T113-S3

User Manual

Revision 1.1

August 30, 2021

Revision History

Revision	Date	Author	Description
1.0	May 18, 2021	AWAXXXX	Initial version
1.1	August 30, 2021	AWA1896	Refresh the specification of section 8 Add the description of CAN

Contents

1 About This Document

1.1 Purpose and Scope

This document describes the features, logical structures, functions, operating modes, and related registers of each module about T113-S3. For details about the interface timings and related parameters, the pins, pin usages, performance parameters, and package dimension, please refer to the T113-S3 Datasheet.

1.2 Intended Audience

The document is intended for:

- Design and maintenance personnel for electronics
- Programmers in writing code or modifying the Allwinner provided code

1.3 Symbol Conventions

1.3.1 Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
WARNING	Indicates potential risk of injury or death exists if the instructions are not obeyed.
CAUTION	Indicates potential risk of equipment damage, data loss, performance degradation, or unexpected results exists if the instructions are not obeyed.
NOTE	Provides additional information to emphasize or supplement important points of the main text.

1.3.2 Table Content Conventions

The table content conventions that may be found in this document are defined as follows.

Symbol	Description
/	The cell is blank.

1.3.3 Reset Value Conventions

In the register definition tables:

If other column value in a bit or multiple bits row is "/", that this bit or these multiple bits are unused.

If the default value of a bit or multiple bits is "UDF", that the default value is undefined.

1.3.4 Register Attributes

The register attributes that may be found in this document are defined as follows.

Symbol	Description
R	Read Only
R/W	Read/Write
R/WAC	Read/Write-Automatic-Clear, clear the bit automatically when the operation of complete. Writing 0 has no effect
R/WC	Read/Write-Clear
R/W0C	Read/Write 0 to Clear. Writing 1 has no effect
R/W1C	Read/Write 1 to Clear. Writing 0 has no effect
R/W1S	Read/Write 1 to Set. Writing 0 has no effect
W	Write Only

1.3.5 Numerical System

The expressions of data capacity, frequency, and data rate are described as follows.

Type	Symbol	Value
Data capacity	1 K	1024
	1 M	1,048,576
	1 G	1,073,741,824
Frequency, data rate	1 k	1000
	1 M	1,000,000
	1 G	1,000,000,000

The expressions of addresses and data are described as follows.

Symbol	Example	Description
0x	0x0200, 0x79	Address or data in hexadecimal
0b	0b010, 0b00 000 111	Data or sequence in binary (register description is excluded.)
X	00X, XX1	In data expression, X indicates 0 or 1. For example, 00X indicates 000 or 001 and XX1 indicates 001, 011, 101 or 111.

Contents

Figures

2 Product Description

2.1 Overview

T113-S3 is an advanced application processor designed for the automotive and industrial control products. It integrates dual-core Cortex™-A7 CPU and single-core HiFi4 DSP to provide the high efficient computing power. T113-S3 supports full format decoding such as H.265, H.264, MPEG-1/2/4, JPEG, VC1, and so on. The independent hardware encoder can encode in JPEG or MJPEG. Integrated multi ADCs/DACs and I2S/PCM/DMIC/OWA audio interfaces can provide the perfect voice interaction solution. T113-S3 comes with extensive connectivity to facilitate product expansion, such as USB, SDIO, EMAC, TWI, UART, SPI, PWM, GPADC, IR TX&RX, and so on.

2.2 Features

2.2.1 CPU Architecture

- Dual-core ARM Cortex™-A7
- 32 KB L1 I-cache + 32 KB L1 D-cache per core, and 256 KB L2 cache

2.2.2 DSP Architecture

- HiFi4
- 32 KB L1 I-cache and 32 KB L1 D-cache
- 64 KB I-ram and 64 KB D-ram

2.2.3 Memory SubSystem

2.2.3.1 Boot ROM (BROM)

- On-chip memory
- Supports system boot from the following devices:
  - SD card
  - eMMC

- SPI NOR Flash
- SPI NAND Flash

- Supports mandatory upgrade process through USB and SD card
- Supports GPIO pin and eFuse module to select the boot media type
- Supports normal booting and secure booting
- Secure BROM loads only certified firmware
- Secure BROM ensures that the Secure Boot is in a trusted environment

2.2.3.2 SDRAM

- Embedded with 128 MB DDR3
- Supports clock frequency up to 800 MHz

2.2.3.3 SMHC

- Three SD/MMC host controller (SMHC) interfaces
- The SMHC0 controls the devices that comply with the protocol Secure Digital Memory (SD mem-version 3.0)
- The SMHC1 controls the device that complies with the protocol Secure Digital I/O (SDIO-version 3.0)
- The SMHC2 controls the device that complies with the protocol Multimedia Card (eMMC-version 5.0)
- Maximum performance:
  - SDR mode 150 MHz@1.8 V IO pad
  - DDR mode 50 MHz@1.8 V IO pad
  - DDR mode 50 MHz@3.3 V IO pad
- Supports 1-bit or 4-bit data width
- Supports block size of 1 to 65535 bytes
- Internal 1024-Bytes RX FIFO and 1024-Bytes TX FIFO
- Supports card insertion and removal interrupt
- Supports hardware CRC generation and error detection
- Supports descriptor-based internal DMA controller

2.2.4 Video Engine

- Video decoding
  - H.265 MP@L4.1 up to 1080p@60fps
  - H.264 BP/MP/HP@L4.2 up to 1080p@60fps
  - H.263 BP up to 1080p@60fps
  - MPEG-4 SP/ASP L5 up to 1080p@60fps
  - MPEG-2 MP/HL up to 1080p@60fps
  - MPEG-1 MP/HL up to 1080p@60fps
  - Xvid up to 1080p@60fps
  - Sorenson Spark up to 1080p@60fps
  - WMV9/VC-1 SP/MP/AP up to 1080p@60fps
  - MJPEG up to 1080p@30fps
- Video encoding
  - JPEG/MJPEG up to 1080p@60fps
  - Supports input picture scaler up/down

2.2.5 Video and Graphics

2.2.5.1 Display Engine (DE)

- Output size up to 2048 x 2048
- Supports two alpha blending channels for main display and one channel for aux display
- Supports four overlay layers in each channel, and has an independent scaler
- Supports potter-duff compatible blending operation
- Supports LBC buffer decoder
- Supports dither output to TCON
- Supports input format Semi-planar YUV422/YUV420/YUV411 and Planar YUV422/YUV420/YUV411, ARGB8888/XRGB8888/RGB888/ARGB4444/ARGB1555/RGB565/palette
- Supports SmartColor2.0 for excellent display experience
  - Adaptive detail/edge enhancement
  - Adaptive color enhancement

- Adaptive contrast enhancement and fresh tone rectify

- Supports write back for aux display

2.2.5.2 De-interlacer (DI)

- Supports YUV420 (Planar/NV12/NV21) and YUV422 (Planar/NV16/NV61) data format
- Supports video resolution from 32 x 32 to 2048 x 1280 pixel
- Supports Inter-field interpolation/motion adaptive de-interlace method
- Performance: module clock 600M for 1080p@60Hz YUV420

2.2.5.3 Graphic 2D (G2D)

- Supports layer size up to 2048 x 2048 pixels
- Supports pre-multiply alpha image data
- Supports color key
- Supports two pipes Porter-Duff alpha blending
- Supports multiple video formats 4:2:0, 4:2:2, 4:1:1 and multiple pixel formats (8/16/24/32 bits graphics layer)
- Supports memory scan order option
- Supports any format convert function
- Supports 1/16× to 32× resize ratio
- Supports 32-phase 8-tap horizontal anti-alias filter and 32-phase 4-tap vertical anti-alias filter
- Supports window clip
- Supports FillRectangle, BitBlit, StretchBlit and MaskBlit
- Supports horizontal and vertical flip, clockwise 0/90/180/270 degree rotate for normal buffer
- Supports horizontal flip, clockwise 0/90/270 degree rotate for LBC buffer

2.2.6 Video Output

2.2.6.1 RGB and LVDS LCD

- Supports RGB interface with DE/SYNC mode, up to 1920 x 1080@60fps
- Supports serial RGB/dummy RGB interface, up to 800 x 480@60fps
- Supports LVDS interface with dual link, up to 1920 x 1080@60fps
- Supports LVDS interface with single link, up to 1366 x 768@60fps
- Supports i8080 interface, up to 800 x 480@60fps
- Supports BT656 interface for NTSC and PAL
- RGB666 and RGB565 with dither function
- Gamma correction with R/G/B channel independence

2.2.6.2 MIPI DSI

- Compliance with MIPI DSI v1.01
- Supports 4-lane MIPI DSI, up to 1920 x 1200@60fps resolution
- Supports non-burst mode with sync pulse/sync event and burst mode
- Supports pixel format: RGB888, RGB666, RGB666 loosely packed and RGB565
- Supports continuous and non-continuous lane clock modes
- Supports bidirectional communication of all generic commands in LP through data lane 0
- Supports low power data transmission
- Supports ULPS and escape modes
- Hardware checksum capabilities

2.2.6.3 CVBS OUT

- 1-channel CVBS output
- Supports NTSC and PAL format
- Plug status auto detecting
- 10 bits DAC output

2.2.7 Video Input

2.2.7.1 Parallel CSI

- Supports 8-bit digital camera interface (RAW8/YUV422/YUV420)
- Supports BT656, BT601 interface (YUV422)
- Supports ITU-R BT.656 time-multiplexed format up to 2*1080p@30fps in DDR sample mode
- Maximum pixel clock of 148.5 MHz
- Supports de-interlacing for interlace video input
- Supports conversion from YUV422 to YUV420, YUV422 to YUV400, YUV420 to YUV400
- Supports horizontal and vertical flip

2.2.7.2 CVBS IN

- 2-channel CVBS input and 1-channel CVBS decoder
- Supports NTSC and PAL format
- Supports YUV422/YUV420 format
- With 1 channel 3D comb filter
- Detection for signal locked and 625 lines
- Programmable brightness, contrast, and saturation
- 10-bit video ADCs

2.2.8 System Peripherals

2.2.8.1 Timer

- The timer module implements the timing and counting functions, which includes timer0, timer1, watchdog, and audio video synchronization (AVS)
- The timer0/timer1 is a 32-bit down counter. The timer0 and timer1 are completely consistent
- The watchdog is used to transmit a reset signal to reset the entire system when an exception occurs in the system

- The AVS is used to synchronize the audio and video. The AVS sub-block includes AVS0 and AVS1, which are completely consistent

2.2.8.2 High Speed Timer (HSTimer)

- The HSTimer module consists of HSTimer0 and HSTimer1. HSTimer0 and HSTimer1 are down counters that implement timing and counting functions. They are completely consistent.
- Configurable 56-bit down timer
- Supports 5 prescale factors
- The clock source is synchronized with AHB0 clock, much more accurate than other timers
- Supports 2 working modes: periodic mode and single counting mode
- Generates an interrupt when the count is decreased to 0

2.2.8.3 GIC

- Supports 16 Software Generated Interrupts (SGIs), 16 Private Peripheral Interrupts (PPIs), and 192 Shared Peripheral Interrupts (SPIs)
- Software-configurable interrupts can be:
  - Enabled or disabled
  - Assigned to one of two groups: Group 0 or Group 1
  - Prioritized
  - Signaled to different processors in multiprocessor implementations
  - Either level-sensitive or edge-triggered
- GIC security extensions
  - Uses Group 0 interrupts as Secure interrupts, and Group 1 interrupts as Non-secure interrupts
  - Uses the FIQ interrupt request to signal Secure interrupts to a connected processor. The GIC-400 always signals Group 1 interrupts using the IRQ interrupt request

2.2.8.4 DMAC

- Up to 16-ch DMA

- Provides 32 peripheral DMA requests for data reading and 32 peripheral DMA requests for data writing
- Flexible data width of 8/16/32/64-bit
- Programmable DMA burst length
- Supports linear and IO address modes
- Supports data transfer types with memory-to-memory, memory-to-peripheral, peripheral-to-memory, peripheral-to-peripheral
- Supports transfer with linked list
- DRQ response includes waiting mode and handshake mode
- DMA channel supports pause function
- Memory devices support non-aligned transform

2.2.8.5 Clock Controller Unit (CCU)

- 8 PLLs
- One on-chip RC oscillator
- Supports one external 24 MHz DCXO and one external 32.768 kHz oscillator
- Supports clock configuration and clock generation for corresponding modules
- Supports software-controlled clock gating and software-controlled reset for corresponding modules

2.2.8.6 Thermal Sensor Controller (THS)

- One thermal sensor located in CPU
- Temperature accuracy: ±3°C from 0°C to +100°C, ±5°C from -25°C to +125°C
- Averaging filter for thermal sensor reading
- Supports over-temperature protection interrupt and over-temperature alarm interrupt

2.2.8.7 LDO Power

- Integrated 2 LDOs (LDOA, LDOB)
- LDOA: 1.8 V power output, LDOB: 1.35 V/1.5 V/1.8 V power output
- LDOA for IO and analog module, LDOB for SDRAM

- Input voltage is 2.4 V to 3.6 V

2.2.8.8 RTC

- Implements time counter and timing wakeup
- Provides a 16-bit counter for counting day, 5-bit counter for counting hour, 6-bit counter for counting minute, 6-bit counter for counting second
- External connect a 32.768 kHz low-frequency oscillator for count clock
- Timer frequency is 1 kHz
- Configurable initial value by software anytime
- Supports timing alarm, and generates interrupt and wakeup the external devices
- 8 general purpose registers for storing power-off information

2.2.8.9 I/O Memory Management Unit (IOMMU)

- Supports virtual address to physical address mapping by hardware implementation
- Supports VE, CSI, DE, G2D, DI parallel address mapping
- Supports VE, CSI, DE, G2D, DI bypass function independently
- Supports VE, CSI, DE, G2D, DI pre-fetch independently
- Supports VE, CSI, DE, G2D, DI interrupt handing mechanism independently
- Supports 2 levels TLB (level1 TLB for special using, and level2 TLB for sharing)
- Supports TLB Fully cleared and Partially disabled
- Supports trigger PTW behavior when TLB miss
- Supports checking the permission

2.2.8.10 Message Box (MSGBOX)

- Supports two CPU to transmit information through channels. Each CPU has a MSGBOX
  - CPU 0: ARM CPUX
  - CPU 1: DSP
- The channel between two CPU has 4 channels, and the FIFO depth of a channel is 8 x 32 bits

2.2.8.11 Spinlock

- Provides hardware synchronization mechanism in multi-core systems
- Supports 32 lock units
- Two kinds of lock status: locked and unlocked
- Lock time of the processor is predictable (less than 200 cycles)

2.2.9 Audio Interfaces

2.2.9.1 Audio Codec

- Two audio digital-to-analog converter (DAC) channels
  - Supports 16-bit and 20-bit sample resolution
  - 8 kHz to 192 kHz DAC sample rate
  - 100 ± 2 dB SNR@A-weight, -85 ± 3 dB THD+N
- One audio output:
  - One stereo headphone output: HPOUTL/R
- Three audio analog-to-digital converter (ADC) channels
  - Supports 16-bit and 20-bit sample resolution
  - 8 kHz to 48 kHz ADC sample rate
  - 95 ± 3dB SNR@A-weight, -80 ± 3dB THD+N
- Three audio inputs:
  - One differential microphone input: MICIN3P/3N, or one single-end microphone input: MICIN3P
  - One stereo LINEIN input: LINEINL/R
  - One stereo FMIN input: FMINL/R
- Stereo headphone driver
  - 95 ± 3 dB SNR@A-weight
  - Output Level 0.55 Vrms@10 kΩ/THD+N -77 ± 3 dB, 0.37 Vrms@16 Ω/THD+N -40 dB
- Supports Dynamic Range Controller adjusting the DAC playback and ADC recording
- One 128x20-bits FIFO for DAC data transmit, one 128x20-bits FIFO for ADC data receive
- Programmable FIFO thresholds

- Supports interrupts and DMA
- Internal HPLDO output for HPVCC
- Internal ALDO output for AVCC

2.2.9.2 I2S/PCM

- Two I2S/PCM external interfaces (I2S1, I2S2) for connecting external power amplifier and MIC ADC
- Compliant with standard Philips Inter-IC sound (I2S) bus specification
  - Left-justified, Right-justified, PCM mode, and Time Division Multiplexing (TDM) format
  - Programmable PCM frame width: 1 BCLK width (short frame) and 2 BCLKs width (long frame)
- Transmit and Receive data FIFOs
  - Programmable FIFO thresholds
  - 128 depth x 32-bit width TXFIFO and 64 depth x 32-bit width RXFIFO
- Supports multiple function clock
  - Clock up to 24.576 MHz Data Output of I2S/PCM in Master mode (Only if the IO PAD and Peripheral I2S/PCM satisfy Timing Parameters)
  - Clock up to 12.288 MHz Data Input of I2S/PCM in Master mode
- Supports TX/RX DMA slave interface
- Supports multiple application scenarios
  - Up to 16 channels (fs = 48 kHz) which has adjustable width from 8-bit to 32-bit
  - Sample rate from 8 kHz to 384 kHz (CHAN = 2)
  - 8-bit u-law and 8-bit A-law companded sample
- Supports master/slave mode

2.2.9.3 DMIC

- Supports maximum 8 digital PDM microphones
- Supports sample rate from 8 kHz to 48 kHz

2.2.9.4 One Wire Audio (OWA)

- One OWA TX and one OWA RX
- Compliance with S/PDIF interface
- IEC-60958 and IEC-61937 transmitter and receiver functionality
  - IEC-60958 supports 16-bit, 20-bit, and 24-bit data formats
  - IEC-61937 uses the IEC-60958 series for the conveying of non-linear PCM bit streams, each sub-frame transmits 16-bit
- TXFIFO and RXFIFO
  - One 128×24bits TXFIFO and one 64×24bits RXFIFO for audio data transfer
  - Programmable FIFO thresholds
- Supports TX/RX DMA slave interface
- Supports multiple function clock
  - Separate clock for OWA TX and OWA RX
  - The clock of TX function includes 24.576 MHz and 22.579 MHz frequency
  - The clock of RX function includes 24.576*8MHz frequency
- Supports hardware parity on TX/RX
  - Hardware parity checking on the receiver
  - Hardware parity generation on the transmitter
- Supports channel status capture on the receiver
- Supports channel sample rate capture on the receiver
- Supports insertion detection for the receiver
- Supports channel status insertion for the transmitter
- Supports interrupts and DMA

2.2.10 Security System

2.2.10.1 Crypto Engine (CE)

- Supports Symmetrical algorithm for encryption and decryption: AES, DES, TDES
  - Supports ECB, CBC, CTS, CTR, CFB, OFB mode for AES

- Supports 128/192/256-bit key for AES
- Supports ECB, CBC, CTR mode for DES/TDES

- Supports Hash algorithm for tamper proofing: MD5, SHA, HMAC
  - Supports SHA1, SHA224, SHA256, SHA384, SHA512 for SHA
  - Supports HMAC-SHA1, HMAC-SHA256 for HMAC
  - Supports multi-package mode for MD5/SHA1/SHA224/SHA256/SHA384/SHA512
- Supports Asymmetrical algorithm for signature verification: RSA
  - RSA supports 512/1024/2048-bit width
- Supports 160-bit hardware PRNG with 175-bit seed
- Supports 256-bit hardware TRNG
- Internal DMA controller for data transfer with memory

2.2.10.2 Security ID (SID)

- Supports 2 Kbits eFuse
- Backup eFuse information by using SID_SRAM
- Burning the key to the SID
- Reading the key use status in the SID
- Loading the key to the CE

2.2.10.3 Secure Memory Control (SMC)

- The SMC is always secure, only secure CPU can access the SMC
- Sets secure area of DRAM
- Sets secure property that Master accesses to DRAM

2.2.10.4 Secure Peripherals Control (SPC)

- The SPC is always secure, only secure CPU can access the SPC
- Sets secure property of peripherals

2.2.11 External Peripherals

2.2.11.1 USB DRD

- One USB 2.0 DRD (USB0), with integrated USB 2.0 analog PHY
- Complies with USB2.0 Specification
- Supports USB Host function
  - Compatible with Enhanced Host Controller Interface (EHCI) Specification, Version 1.0
  - Compatible with Open Host Controller Interface (OHCI) Specification, Version 1.0a
  - Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s), and Low-Speed (LS, 1.5 Mbit/s)
  - Supports only 1 USB Root port shared between EHCI and OHCI
- Supports USB Device function
  - Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s)
  - Supports bi-directional endpoint0 (EP0) for Control transfer
  - Up to 10 user-configurable endpoints (EP1+, EP1-, EP2+, EP2-, EP3+, EP3-, EP4+, EP4-, EP5+, EP5-) for Bulk transfer, Isochronous transfer and Interrupt transfer
  - Up to (8 KB + 64 Bytes) FIFO for all EPs (including EP0)
  - Supports interface to an external Normal DMA controller for every EP
- Supports an internal DMA controller for data transfer with memory
- 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 modes
- Includes automatic ping capabilities
- Soft connect/disconnect function
- Performs all transaction scheduling in hardware
- Power optimization and power management capabilities
- Device and host controller share a 8K SRAM and a physical PHY

2.2.11.2 USB HOST

- One USB 2.0 HOST (USB1), with integrated USB 2.0 analog PHY
- Complies with USB2.0 Specification
- Supports USB2.0 Host function
  - Compatible with Enhanced Host Controller Interface (EHCI) Specification, Version 1.0
  - Compatible with Open Host Controller Interface (OHCI) Specification, Version 1.0a
  - Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s) and Low-Speed (LS, 1.5 Mbit/s) Device
  - Supports only 1 USB Root port shared between EHCI and OHCI
- An internal DMA Controller for data transfer with memory

2.2.11.3 EMAC

- One EMAC interface for connecting external Ethernet PHY
- 10/100/1000 Mbit/s Ethernet port with RGMII and RMII interfaces
- Compliant with IEEE 802.3-2002 standard
- Supports both full-duplex and half-duplex operation
- Provides the management data input/output (MDIO) interface for PHY device configuration and management with configurable clock frequencies
- Programmable frame length to support Standard or Jumbo Ethernet frames with sizes up to 16 KB
- 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
  - Supports linked-list descriptor list structure
  - 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
- 4 KB TXFIFO for transmission packets and 16 KB RXFIFO for reception packets
- Programmable interrupt options for different operational conditions

2.2.11.4 UART

- Up to 6 UART controllers (UART0, UART1, UART2, UART3, UART4, UART5)
- UART0, UART4, UART5: 2-wire; UART1, UART2, UART3: 4-wire
- Compatible with industry-standard 16450/16550 UARTs
- Supports IrDA-compatible slow infrared (SIR) format
- Two separate FIFOs: one is RX FIFO, and the other is TX FIFO
  - Each of them is 64 bytes (For UART0)
  - Each of them is 256 bytes (For UART1, UART2, UART3, UART4, and UART5)
- The working reference clock is from the APB bus clock
  - Speed up to 4 Mbit/s with 64 MHz APB clock
  - Speed up to 1.5 Mbit/s with 24 MHz APB clock
- 5 to 8 data bits for RS-232 characters, or 9 bits RS-485 format
- 1, 1.5 or 2 stop bits
- Programmable parity (even, odd, or no parity)
- Supports TX/RX DMA slave controller interface
- Supports software/hardware flow control
- Supports RX DMA Master interface (Only for UART1)
- Supports auto-flow by using CTS & RTS (Only for UART1/2/3)

2.2.11.5 Two Wire Interface (TWI)

- Up to 4 TWI controllers (TWI0, TWI1, TWI2, TWI3)
- Compliant with I2C bus standard
- Supports standard mode (up to 100 kbit/s) and fast mode (up to 400 kbit/s)
- Supports 7-bit and 10-bit device addressing modes
- Supports master mode or slave mode
- Master mode features:
  - Supports the bus arbitration in the case of multiple master devices
  - Supports clock synchronization and bit and byte waiting

- Supports packet transmission and DMA

- Slave mode features:
  - Interrupt on address detection

- The TWI controller includes one TWI engine and one TWI driver. And the TWI driver supports packet transmission and DMA mode when TWI works in master mode

2.2.11.6 SPI and SPI_DBI

- Up to 2 SPI controllers (SPI0, SPI1)
- The SPI0 only supports SPI mode; The SPI1 supports SPI mode and display bus interface (DBI) mode
- SPI mode:
  - Full-duplex synchronous serial interface
  - Master/slave configurable
  - Mode0 to Mode3 are supported for both transmit and receive operations
  - 8-bit wide by 64-entry FIFO for both transmit and receive data
  - Polarity and phase of the Chip Select (SPI-CS) and SPI Clock (SPI-CLK) are configurable
  - Supports 3-wire/4-wire SPI
  - Supports programmable serial data frame length: 1-bit to 32-bit
  - Supports Standard SPI, Dual-Output/Dual-Input SPI, Dual IO SPI, Quad-Output/Quad-Input SPI
- DBI mode:
  - Supports DBI Type C 3 Line/4 Line Interface Mode
  - Supports 2 Data Lane Interface Mode
  - Supports RGB111/444/565/666/888 video format
  - Maximum resolution of RGB666 240 x 320@30Hz with single data lane
  - Maximum resolution of RGB888 240 x 320@60Hz or 320 x 480@30Hz with dual data lane
  - Supports Tearing effect
  - Supports software flexible control video frame rate

2.2.11.7 CIR Receiver (CIR_RX)

- One CIR_RX interface (IR-RX)
- Full physical layer implementation
- Supports NEC format infra data
- Supports CIR for remote control or wireless keyboard
- 64x8 bits FIFO for data buffer
- Sample clock up to 1 MHz

2.2.11.8 CIR Transmitter (CIR_TX)

- One CIR_TX interface (IR-TX)
- Supports arbitrary wave generator
- Configurable carrier frequency
- Supports handshake mode and waiting mode of DMA
- 128 bytes FIFO for data buffer

2.2.11.9 PWM

- Supports 8 independent PWM channels (PWM0 to PWM7)
  - Supports PWM continuous mode output
  - Supports PWM pulse mode output, and the pulse number is configurable
  - Output frequency range: 0 to 24 MHz or 100 MHz
  - Various duty-cycle: 0% to 100%
  - Minimum resolution: 1/65536
- Supports 4 complementary pairs output
  - PWM01 pair (PWM0 + PWM1), PWM23 pair (PWM2 + PWM3), PWM45 pair (PWM4 + PWM5), PWM67 pair (PWM6 + PWM7)
  - Supports dead-zone generator, and the dead-zone time is configurable
- Supports 4 group of PWM channel output for controlling stepping motors
  - Supports any plural channels to form a group, and output the same duty-cycle pulse

- In group mode, the relative phase of the output waveform for each channel is configurable

- Supports 8 channels capture input
  - Supports rising edge detection and falling edge detection for input waveform pulse
  - Supports pulse-width measurement for input waveform pulse

2.2.11.10 General Purpose ADC (GPADC)

- 1-ch Successive approximation register (SAR) analog-to-digital converter (ADC)
- 12-bit sampling resolution and 8-bit precision
- 64 FIFO depth of data register
- Power reference voltage: AVCC, analog input voltage range: 0 to AVCC
- Maximum sampling frequency up to 1 MHz
- Supports three operation modes: single conversion mode, continuous conversion mode, burst conversion mode

2.2.11.11 Touch Panel ADC (TPADC)

- 12 bit SAR type A/D converter
- Configurable sample frequency up to 1 MHz
- One 32x12 FIFO for storing A/D conversion result
- Supports DMA slave interface
- Supports 4-wire resistive touch panel input detection
  - Supports pen down detection with programmable sensitivity
  - Supports single touch coordinate measurement
  - Supports dual touch detection
  - Supports touch pressure measurement with programmable threshold
  - Supports median and averaging filter for noise reduction
  - Supports X and Y coordinate exchange function
- Supports Aux ADC with up to 4 channels

2.2.11.12 LEDC

- LEDC is used to control the external intelligent control LED lamp
- Configurable LED output high/low level width
- Configurable LED reset time
- LEDC data supports DMA configuration mode and CPU configuration mode
- Maximum 1024 LEDs serial connect
- LED data transfer rate up to 800 kbit/s

2.2.11.13 CAN

- Supports industry-standard AMBA Peripheral Bus (APB) and it is fully compliant with the AMBA Specification, Revision 2.0
- Supports APB 32-bit bus width operation
- Supports the CAN 2.0A and 2.0B protocol specification
- Supports one-shot transmission option
- Supports two configurable filter modes
- Supports listen only mode
- Supports self-test mode

2.2.12 Package

- eLQFP128, 14 mm x 14 mm x 1.4 mm

2.3 Block Diagram

Figure 2-1 shows the system block diagram of the T113-S3.

Figure 2-1 T113-S3 System Block Diagram

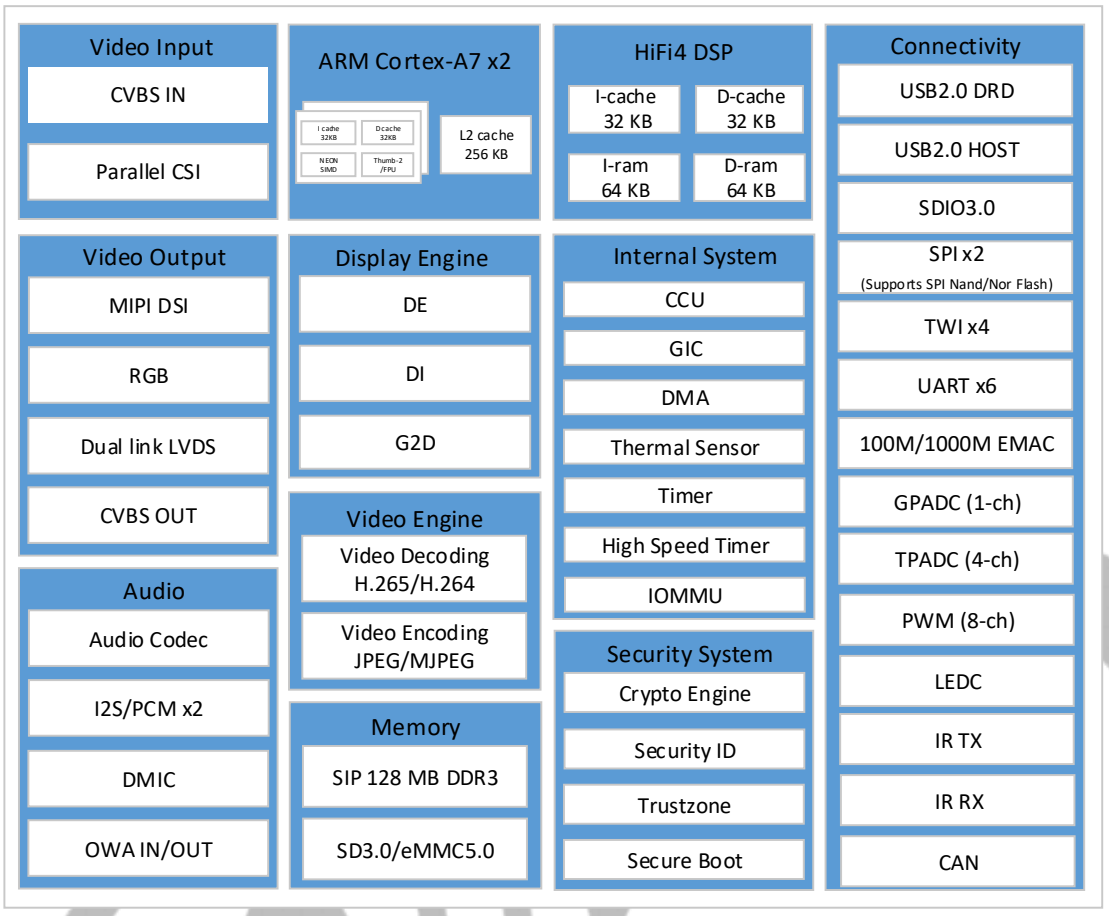


Figure 2-2 to Figure 2-5 show the typical solution diagrams of the T113-S3.

Figure 2-2 Car MP5 Solution of the T113-S3

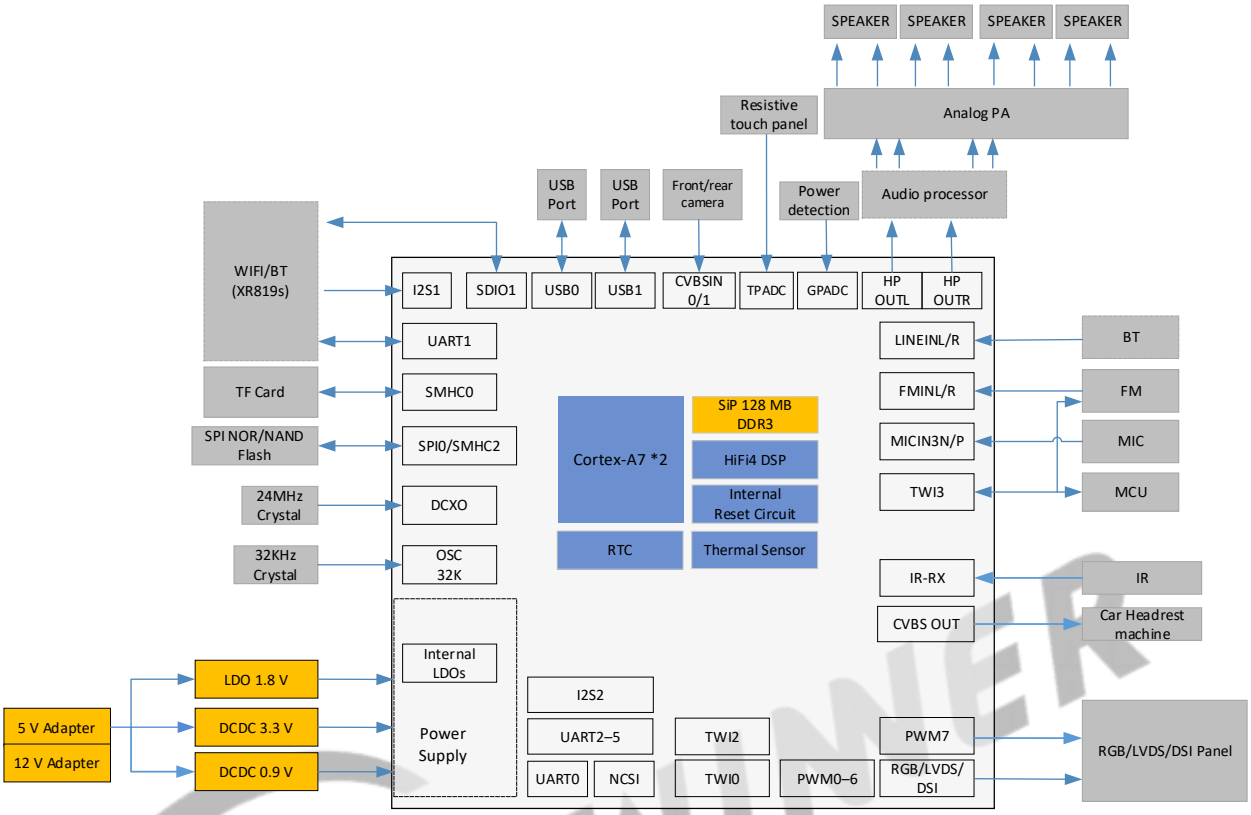


Figure 2-3 Car Instrument Solution of the T113-S3

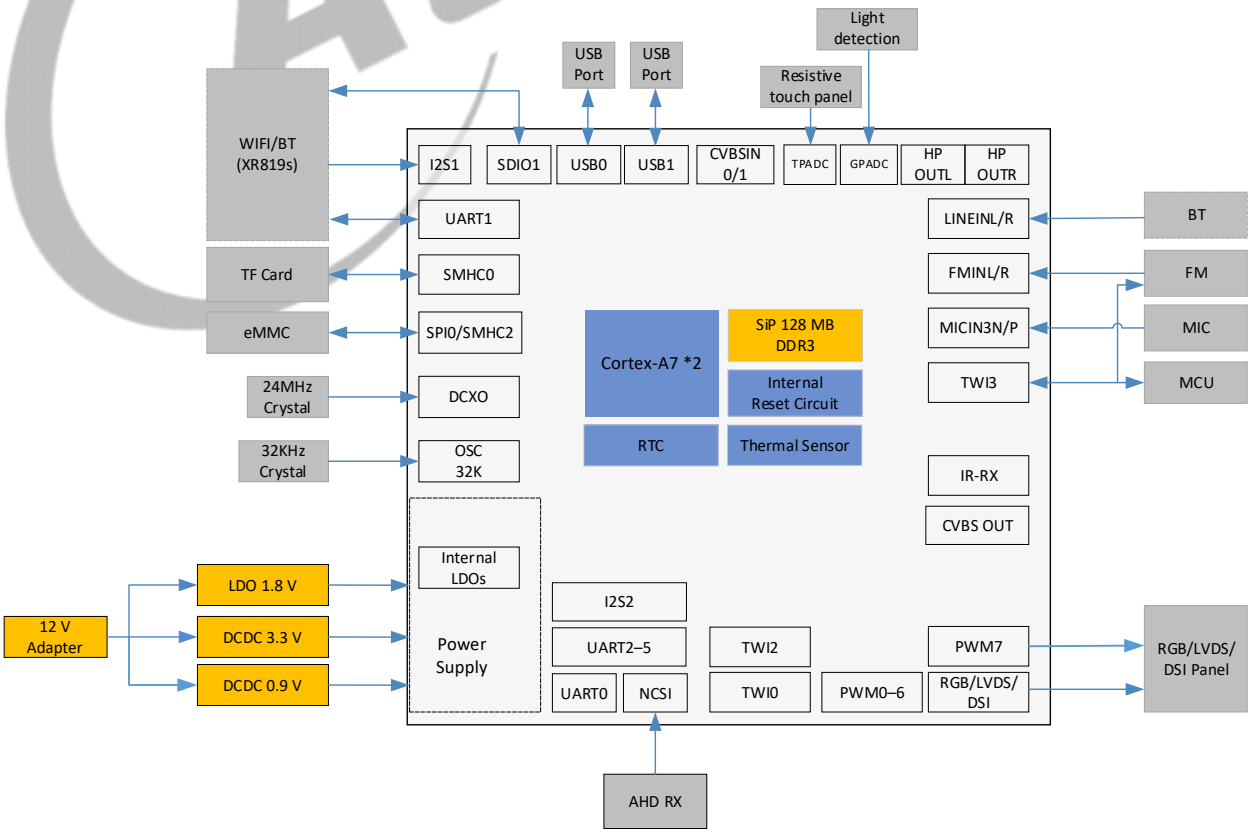


Figure 2-4 Industrial Control HMI Solution of the T113-S3

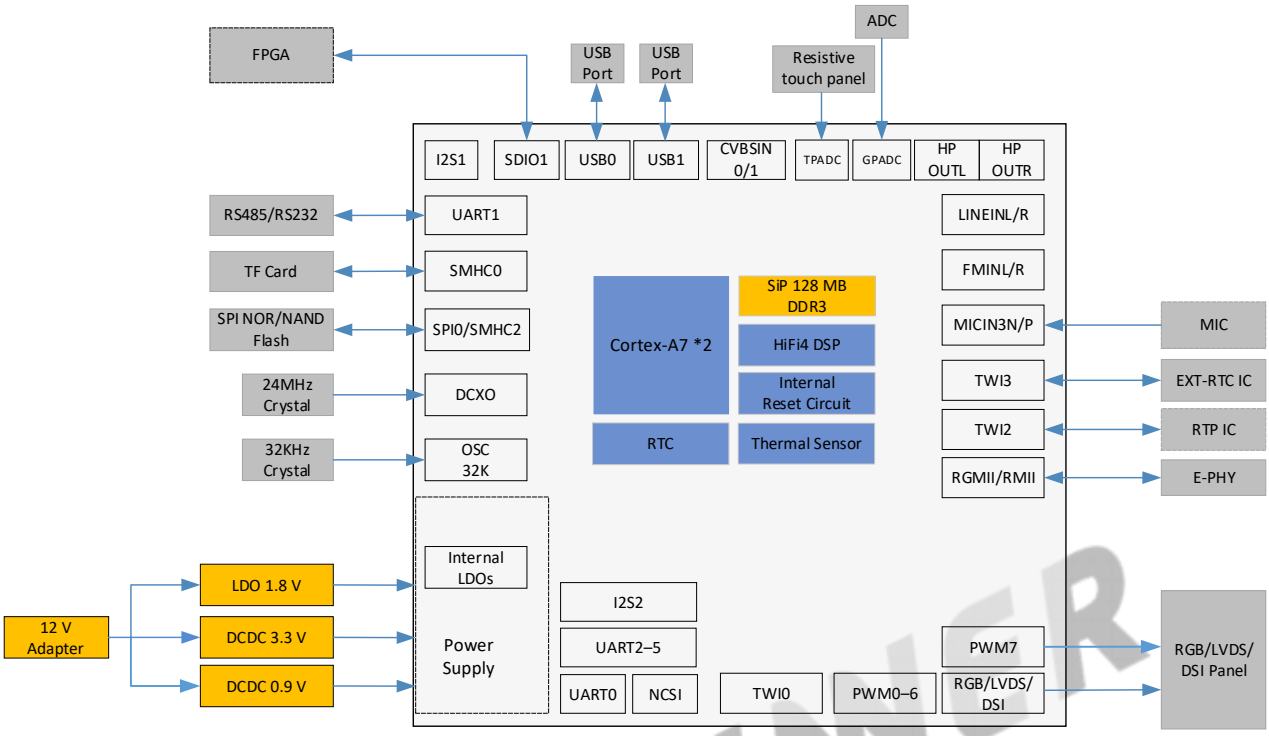
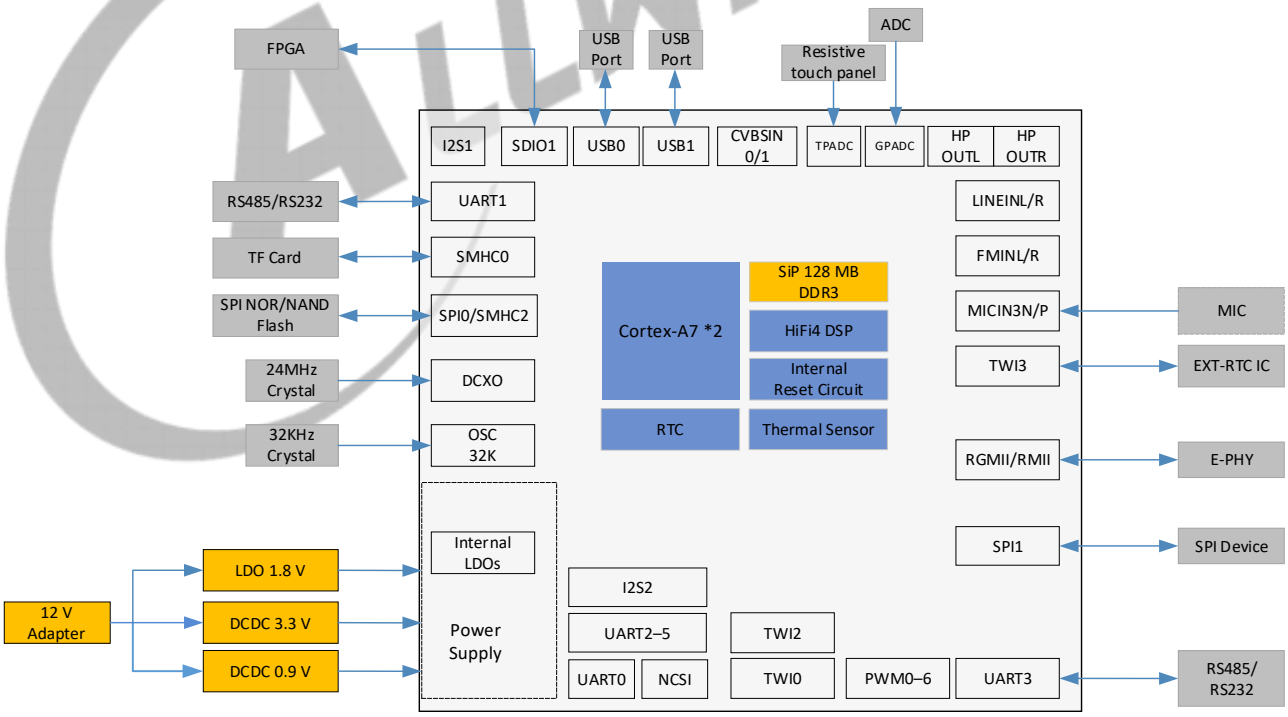


Figure 2-5 Industrial Control PLC Solution of the T113-S3



Contents

Figures

Tables

3 System

3.1 Memory Mapping

Module	Address (It is for Cluster CPU)	Size
BROM & SRAM		
N-BROM	0x0000 0000—0x0000 BFFF	48 KB
S-BROM	0x0000 0000—0x0000 8FFF	36 KB
SRAM A1	0x0002 0000---0x0002 7FFF	32 KB
DSP0 IRAM	0x0002 8000---0x0003 7FFF	64 KB The local sram is switched to system boot.
DSP0 DRAM0	0x0003 8000---0x0003 FFFF	32 KB The local sram is switched to system boot.
DSP0 DRAM1	0x0004 0000---0x0004 7FFF	32 KB The local sram is switched to system boot.
DSP0 IRAM (local sram)	0x0040 0000---0x0040 FFFF	64 KB The local sram is switched to DSP.
DSP0 DRAM0 (local sram)	0x0042 0000---0x0042 7FFF	32 KB The local sram is switched to DSP.
DSP0 DRAM1 (local sram)	0x0044 0000---0x0044 7FFF	32 KB The local sram is switched to DSP.
DSP_SYS Related		
DSP_CFG	0x0170 0000---0x0170 03FF	1 KB
DSP_WDG	0x0170 0400---0x0170 07FF	1 KB
DSP_INTC	0x0170 0800---0x0170 0BFF	1 KB
DSP_TZMA	0x0170 0C00---0x0170 0FFF	1 KB
DSP_MSGBOX	0x0170 1000---0x0170 1FFF	4 KB
VE_SYS Related		
VE	0x01C0 E000---0x01C0 FFFF	8 KB
SP0 (SYS Domain)		
GPIO	0x0200 0000---0x0200 07FF	2 KB
SPC	0x0200 0800---0x0200 0BFF	1 KB

Module	Address (It is for Cluster CPU)	Size
PWM	0x0200 0C00---0x0200 0FFF	1 KB
CCU	0x0200 1000---0x0200 1FFF	4 KB
CIR_TX	0x0200 3000---0x0200 33FF	1 KB
TZMA	0x0200 4000---0x0200 43FF	1 KB
LEDC	0x0200 8000---0x0200 83FF	1 KB
GPADC	0x0200 9000---0x0200 93FF	1 KB
THS	0x0200 9400---0x0200 97FF	1 KB
TPADC	0x0200 9C00---0x0200 9FFF	1 KB
IOMMU	0x0201 0000---0x0201 FFFF	64 KB
Audio Codec	0x0203 0000---0x0203 0FFF	4 KB
DMIC	0x0203 1000---0x0203 13FF	1 KB
I2S1	0x0203 3000---0x0203 3FFF	4 KB
I2S2	0x0203 4000---0x0203 4FFF	4 KB
OWA	0x0203 6000---0x0203 63FF	1 KB
TIMER	0x0205 0000---0x0205 0FFF	4 KB
SP1 (SYS Domain)		
UART0	0x0250 0000---0x0250 03FF	1 KB
UART1	0x0250 0400---0x0250 07FF	1 KB
UART2	0x0250 0800---0x0250 0BFF	1 KB
UART3	0x0250 0C00---0x0250 0FFF	1 KB
UART4	0x0250 1000---0x0250 13FF	1 KB
UART5	0x0250 1400---0x0250 17FF	1 KB
TWI0	0x0250 2000---0x0250 23FF	1 KB
TWI1	0x0250 2400---0x0250 27FF	1 KB
TWI2	0x0250 2800---0x0250 2BFF	1 KB
TWI3	0x0250 2C00---0x0250 2FFF	1 KB
SH0 (SYS Domain)		
SYSCTRL	0x0300 0000---0x0300 0FFF	4 KB
DMAC	0x0300 2000---0x0300 2FFF	4 KB
CPUX_MSGBOX	0x0300 3000---0x0300 3FFF	4 KB
SPINLOCK	0x0300 5000---0x0300 5FFF	4 KB
SID	0x0300 6000---0x0300 6FFF	4 KB
SMC	0x0300 7000---0x0300 7FFF	4 KB
HSTIMER	0x0300 8000---0x0300 8FFF	4 KB

Module	Address (It is for Cluster CPU)	Size
DCU	0x0301 0000---0x0301 FFFF	64 KB
GIC	0x0302 0000---0x0302 FFFF	64 KB
CE_NS	0x0304 0000---0x0304 07FF	2 KB
CE_S	0x0304 0800---0x0304 0FFF	2 KB
CE_KEY_SRAM	0x0304 1000---0x0304 1FFF	4 KB (only CE access)
MSI+MEMC	0x0310 2000---0x0330 1FFF	2 MB
SH2 (SYS Domain)		
SMHC0	0x0402 0000---0x0402 0FFF	4 KB
SMHC1	0x0402 1000---0x0402 1FFF	4 KB
SMHC2	0x0402 2000---0x0402 2FFF	4 KB
SPI0	0x0402 5000---0x0402 5FFF	4 KB
SPI1	0x0402 6000---0x0402 6FFF	4 KB
USB0	0x0410 0000---0x041F FFFF	1 MB
USB1	0x0420 0000---0x042F FFFF	1 MB
EMAC	0x0450 0000---0x0450 FFFF	64 KB
VIDEO_OUT_SYS Related (SYS Domain)		
DE	0x0500 0000---0x053F FFFF	4 MB
DI	0x0540 0000---0x0540 FFFF	64 KB
G2D	0x0541 0000---0x0544 FFFF	256 KB
DSI	0x0545 0000---0x0545 1FFF	8 KB
DISPLAY_TOP	0x0546 0000---0x0546 0FFF	4 KB
TCON_LCD0	0x0546 1000---0x0546 1FFF	4 KB
TCON_TV0	0x0547 0000---0x0547 0FFF	4 KB
TVE_TOP	0x0560 0000---0x0560 3FFF	16 KB
TV Encoder	0x0560 4000---0x0560 7FFF	16 KB
VIDEO_IN_SYS Related (SYS Domain)		
CSI	0x0580 0000---0x05BF FFFF	4 MB
TVD_TOP	0x05C0 0000---0x05C0 0FFF	4 KB
TV Decoder	0x05C0 1000---0x05C0 1FFF	4 KB
APBS0		
CIR_RX	0x0704 0000---0x0704 03FF	1 KB
AHBS		
RTC	0x0709 0000---0x0709 03FF	1 KB
CPUX Related		

Module	Address (It is for Cluster CPU)	Size
CPU_SYS_CFG	0x0810 0000---0x0810 03FF	1 KB
TimeStamp_STA	0x0811 0000---0x0811 0FFF	4 KB
TimeStamp_CTRL	0x0812 0000---0x0812 0FFF	4 KB
IDC	0x0813 0000---0x0813 0FFF	4 KB
C0_CPUX_CFG	0x0901 0000---0x0901 03FF	1 KB
C0_CPUX_MBIST	0x0902 0000---0x0902 0FFF	4 KB
DRAM Space (SYS Domain)		
DRAM SPACE	0x4000 0000---0xBFFF FFFF	2 GB

The following is the system memory mapping seen by the DSP0 host.

Module	Cacheable Property	Address	Size	Address Mapping Description
SRAM A1	Non-Cacheable (Total 512 MB)	0x0002 0000---0x0002 7FFF	32 KB	The start address of the default DSP.
DSP0 IRAM		0x0002 8000---0x0003 7FFF	64 KB	The DSP0 accesses the address through the external bus to operate the internal local SRAM.
DSP0 DRAM0		0x0003 8000---0x0003 FFFF	32 KB	
DSP0 DRAM1		0x0004 0000---0x0004 7FFF	32 KB	
DSP0 IRAM		0x0040 0000---0x0040 FFFF	64 KB	The DSP0 accesses the address through the internal bus to directly access the local SRAM, which is more efficient than the external bus.
DSP0 DRAM0		0x0042 0000---0x0042 7FFF	32 KB	
DSP0 DRAM1		0x0044 0000---0x0044 7FFF	32 KB	
L4_CONN_periph		0x0200 0000---0x09FF FFFF	128 MB	System Peripherals
DSP_SYS_CFG		0x0170 0000---0x0170 0FFF	4 KB	Peripherals
DRAM SPACE		0x1000 0000---0x1FFF FFFF	256 MB	The front 256 MB space of DDR (0x40000000–0x4FFFFFFF) It requires DSP output AHB for address remapping
Cacheable Mapping Area				
SRAM	Cacheable	0x2002 0000---0x2002 7FFF	32 KB	SRAM A1 has two addresses.

Module	Cacheable Property	Address	Size	Address Mapping Description
DSP0 IRAM	(Total 512 MB)	0x2002 8000---0x2003 7FFF	64 KB	The DSP0 accesses the address through the external bus to operate the internal local SRAM.
DSP0 DRAM0		0x2003 8000---0x2003 FFFF	32 KB	
DSP0 DRAM1		0x2004 0000---0x2004 7FFF	32 KB	
DRAM SPACE		0x3000 0000---0x3FFF FFFF	256 MB	The front 256 MB space of DDR (0x40000000–0x47FFFFFF) It requires DSP output AHB for address remapping
DRAM Special Mapping Area				
DRAM SPACE	Non-Cacheable	0x4000 0000---0x7FFF FFFF	1 GB	
DRAM SPACE	Cacheable Configurable	0x8000 0000---0xBFFF FFFF	1 GB	Reserved space
DRAM SPACE	Cacheable	0xC000 0000---0xFFFF FFFF	1 GB	The front 1 GB space of DDR (0x40000000–0x7FFFFFFF)

3.2 CPUX Configuration

3.2.1 Overview

CPUX indicates the ARM CPU subsystem. You can configure the CPUX via the C0_CPUX_CFG and CPU_SUBSYS_CTRL modules.

The C0_CPUX_CFG module is used for configuring Cluster0. Cluster0 is the only cluster which consists of dual-core Cortex™-A7, 32 KB I-cache and 32 KB D-cache per core, and the shared 256 KB L2 cache. The C0_CPUX_CFG module includes the following features:

- CPU reset system: core reset, debug circuit reset, and other functional modules reset
- CPU related control: interface control, CP15 control, and power-on/off control
- CPU status check: idle status, SMP status, interrupt status, and so on
- CPU debug-related register for control and status

The CPU_SUBSYS_CTRL module is used for controlling CPU subsystem resources, such as GIC-400 and JTAG.

3.2.2 Functional Descriptions

3.2.2.1 Signal Description

For details of A7 CPUX signals, refer to the technical reference manual of *DDI0464F_cortex_A7_mpcore_r0p5_trm.pdf*.

3.2.2.2 L2 Idle Mode

Enter the L2 idle mode by configuring the L2 cache of Cluster to enter the wait for interrupt (WFI) mode.

Follow the steps below:

1. Make sure CPU[1:0] of Cluster has entered the WFI mode. You can check it through bit[17:16] of Cluster0 CPU Status Register.
2. Pull high the ACINACTM of Cluster by writing 1 to the bit0 of Cluster 0 Control Register1.
3. Check whether L2 enters idle status by checking whether the STANDBYWFIL2 is high.

To exit the L2 idle mode, set the ACINACTM to low.

3.2.2.3 CPU Reset System

There are three levels of CPUX reset, with the scope size from small to large: core reset, power-on Reset, and H_Reset.

The following table describes all the reset signals in the CPUX reset system.

Table 3-1 Reset Signal Description

Reset Signal	Description
CORE_RST	This is the primary reset signal which resets the corresponding core logic, including NEON, VFP, Debug, ETM, breakpoint, and watchpoint logic. It maps to a warm reset that covers the reset of the processor logic.
PWRON_RST	This power-on reset signal resets all the processor logic, including the Debug, ETM trace unit, breakpoint, watchpoint logic, and performance monitors logic. It maps to a cold reset that covers the reset of the processor logic and the integrated debug functionality. It does not reset debug logic in the debug power domain. PWRON_RST includes CORE_RST, ETM_RST, and DBG_RST.
H_RST	H_RST includes PWRON_RST, L2_RST, MBIST_RST, SOC_DBG_RST, and C0_CPUX_CFG.
AXI2MBUS_RST	This reset signal resets the AXI2MBUS interface logic circuit.
L2_RST	This single, cluster-wide signal resets the L2 memory system and the logic in the SCU.
ETM_RST	This reset signal resets ETM debug logic circuit.
DBG_RST	In the processor power domain, this reset signal resets only the debug, breakpoint, and watchpoint logic. In the debug power domain, it resets the debug logic for each processor.
SOC_DBG_RST	This reset signal resets all the debug logic. It includes DBG_RST.
MBIST_RST	This reset signal resets all resettable registers in the cluster, for entry into and exit from MBIST mode.
CPU_SUBSYS_RST	CPU_SUBSYS_RST includes C0_H_RST, GIC-400, and CPU_SUBSYS_CTRL.

3.2.2.4 CPUX Power Block Diagram

The following table shows the power domain of the modules in the CPU subsystem.

Table 3-2 Power Domain of Modules in the CPU Subsystem

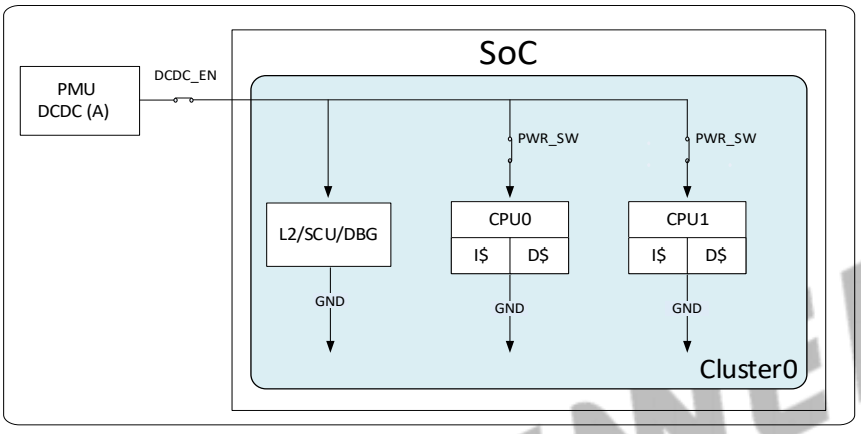
Power Domain	Modules	Description
Cluster0	Cluster0, C0_CPUX_CFG, and C0_MBIST	Cluster0 circuit, C0_CPUX_CFG module, and CPU reset/power/mbist
System	Timestamp, GIC, CPU_SUBSYS_CTRL, and Clock	Provides system source of CPU sub-system

As shown in the table, C0_CPUX_CFG belongs to Cluster0 power domain and CPU_SUBSYS_CTRL belongs to the system power domain.

Because C0_CPUX_CFG and Cluster0 belong to the same power domain, C0_CPUX_CFG holds in the default state when Cluster0 restarts. You need to initial C0_CPUX_CFG via software after C0_H_RST is de-asserted.

The following figure shows the default power domain of Cluster0.

Figure 3-1 Default Power Domain of Cluster0



For Cluster0, the power switch of all CPU cores are on, the PWRON_RST of all CPU cores are de-asserted, the CORE_RST of CPU0 is de-asserted, and the CORE_RST of CPU1 is asserted.

3.2.2.5 CPUX Power Block Diagram

The CPU-related operations, such as opening/closing cores, clustering switches, and querying status, require proper configurations of C0_CPUX_CFG module and the related system control resources like the bus and clock.

3.2.3 Cluster 0 Configuration Register List

Module Name	Base Address
C0_CPUX_CFG	0x09010000

Register Name	Offset	Description
C0_RST_CTRL	0x0000	Cluster 0 Reset Control Register
C0_CTRL_REG0	0x0010	Cluster 0 Control Register0
C0_CTRL_REG1	0x0014	Cluster 0 Control Register1
C0_CTRL_REG2	0x0018	Cluster 0 Control Register2
CACHE_CFG_REG	0x0024	Cache Configuration Register

Register Name	Offset	Description
C0_CPU_STATUS	0x0080	Cluster 0 CPU Status Register
L2_STATUS_REG	0x0084	Cluster 0 L2 Status Register
DBG_REG0	0x00C0	Cluster 0 Debug Control Register0
DBG_REG1	0x00C4	Cluster 0 Debug Control Register1
AXI_MNT_CTRL_REG	0x00D0	AXI Monitor Control Register
AXI_MNT_PRD_REG	0x00D4	AXI Monitor Period Register
AXI_MNT_RLTCY_REG	0x00D8	AXI Monitor Read Total Latency Register
AXI_MNT_WLTCY_REG	0x00DC	AXI Monitor Write Total Latency Register
AXI_MNT_RREQ_REG	0x00E0	AXI Monitor Read Request Times Register
AXI_MNT_WREQ_REG	0x00E4	AXI Monitor Write Request Times Register
AXI_MNT_RBD_REG	0x00E8	AXI Monitor Read Bandwidth Register
AXI_MNT_WBD_REG	0x00EC	AXI Monitor Write Bandwidth Register

3.2.4 Cluster 0 Configuration Register Description

3.2.4.1 0x0000 Cluster 0 Reset Control Register (Default Value: 0x13FF_0101)

Offset: 0x0000			Register Name: C0_RST_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28	R/W	0x1	DDR_RST AXI2MBUS Logic Circuit Reset 0: Assert 1: De-assert
27:26	/	/	/
25	R/W	0x1	MBIST_RST CPUBIST Reset The reset signal is for test. 0: Assert 1: De-assert
24	R/W	0x1	SOC_DBG_RST Cluster SoC Debug Reset 0: Assert 1: De-assert

Offset: 0x0000			Register Name: C0_RST_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
23:20	R/W	0xF	ETM_RST Cluster ETM Reset Assert 0: Assert 1: De-assert
19:16	R/W	0xF	DBG_RST Cluster Debug Reset Assert 0: Assert 1: De-assert
15:9	/	/	/
8	R/W	0x1	L2_RST Cluster L2 Cache Reset 0: Assert 1: De-assert
7:4	/	/	/
3:0	R/W	0x1	CORE_RESET Cluster CPU[1:0] Reset Assert (The bit[3:2] is not used) 0: Assert 1: De-assert

3.2.4.2 0x0010 Cluster 0 Control Register0 (Default Value: 0x8000_0000)

Offset: 0x0010			Register Name: C0_CTRL_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x1	SYSBAR_DISABLE Disable broadcasting barriers to the system bus. 0: Barriers are broadcasted to the system bus. This requires an AMBA4 interconnection. 1: Barriers are broadcasted to the system bus. This is compatible with an AXI3 interconnection.
30	R/W	0x0	BROADCAST_INNER Enable broadcasting inner shareable transactions. 0: Inner shareable transactions are not broadcasted externally. 1: Inner shareable transactions are broadcasted externally.

Offset: 0x0010			Register Name: C0_CTRL_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
29	R/W	0x0	BROADCAST_OUTER Enable broadcasting outer shareable transactions. 0: Outer Shareable transactions are not broadcasted externally. 0: Outer Shareable transactions are broadcasted externally.
28	R/W	0x0	BROADCAST_CACHE_MAINT Enable broadcasting cache maintenance operations to downstream caches. 0: Cache maintenance operations are not broadcasted to downstream caches. 1: Cache maintenance operations are broadcasted to downstream caches.
27:12	/	/	/
11:8	R/W	0x0	CP15S_DISABLE Disable write access to some secure CP15 registers The meaning of each bit: 0: Enable write access to CP15 register 1: Disable write access to CP15 register
7:5	/	/	/
4	R/W	0x0	L2_RST_DISABLE Disable automatic L2 cache invalidation at reset. 0: L2 cache is reset by hardware. 1: L2 cache is not reset by hardware.
3:0	R/W	0x0	L1_RST_DISABLE Disable automatic Cluster CPU[1:0] L1 cache invalidation at reset. (The bit[3:2] is not used) 0: L1 cache is reset by hardware. 1: L1 cache is not reset by hardware.

3.2.4.3 0x0014 Cluster 0 Control Register1 (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: C0_CTRL_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/

Offset: 0x0014			Register Name: C0_CTRL_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x0	CRM_LowFreq_AutoSelect Enable CRM to automatically select low frequency. 0: Disabled 1: Enabled
0	R/W	0x0	ACINACTM Deactivate snoop interface 0: Snoop interface is active 1: Snoop interface is inactive and no longer accepts requests

3.2.4.4 0x0018 Cluster 0 Control Register2 (Default Value: 0x0000_0010)

Offset: 0x0018			Register Name: C0_CTRL_REG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	EVENTI Event input for processor to wake up from WFE state. This bit must remain high for at least one clock cycle to be visible by the cores.
23:20	R/W	0x0	EXM_CLR[1:0] Clear the status of the corresponding interface. (The bit[23:22] is not used) The meaning of each bit: 0: Not to clear 1: Clear
19:0	/	/	/

3.2.4.5 0x0024 Cache Configuration Register (Default Value: 0x0018_001A)

Offset: 0x0024			Register Name: CACHE_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21:19	R/W	0x3	EMA_L2D Indicates the port number of L2 Cache SRAM EMA control port.
18:17	R/W	0x0	EMAW_L2D Indicates the port number of L2 Cache SRAM EMAW control port.

Offset: 0x0024			Register Name: CACHE_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
16	R/W	0x0	EMAS_L2D Indicates the port number of L2 Cache SRAM EMAS control port.
15:12	/	/	/
11:8	R/W	0x0	DVS FARADAY SRAM delay signal
7	R/W	0x0	DVSE FARADAY SRAM delay enable signal
6	R/W	0x0	STOV
5:3	R/W	0x3	EMA Indicates the port number of Cache SRAM EMA control port.
2:1	R/W	0x1	EMAW Indicates the port number of Cache SRAM EMAW control port.
0	R/W	0x0	EMAS Indicates the port number of Cache SRAM EMAS control port.

3.2.4.6 0x0080 Cluster0 CPU Status Register (Default Value: 0x000E_0000)

Offset: 0x0080			Register Name: C0_CPU_STATUS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R	0x0	SMP_AMP Configure the operating mode for CPU[1:0] as Symmetric Multiprocessing mode or Asymmetric Multiprocessing mode. (The bit[27:26] is not used) 0: AMP mode 1: SMP mode
23:20	/	/	/
19:16	R	0xE	STANDBYWFI Indicates if CPU[1:0] of Cluster0 is in WFI standby mode. (The bit[19:18] is not used) 0: The processor is not in WFI standby mode. 1: The processor is in WFI standby mode.
15:12	/	/	/

Offset: 0x0080			Register Name: C0_CPU_STATUS
Bit	**Read/Write**	**Default/Hex**	**Description**
11:8	R	0x0	STANDBYWFE Indicates if CPU[1:0] of Cluster0 is in the WFE standby mode. (The bit[11:10] is not used) 0: The processor is not in WFE standby mode. 1: The processor is in WFE standby mode.
7:1	/	/	/
0	R	0x0	STANDBYWFIL2 Indicates if the L2 memory system of Cluster0 is in WFI standby mode. 0: Cluster0 L2 is not in WFI standby mode. 1: Cluster0 L2 is in WFI standby mode.

3.2.4.7 0x0084 L2 Status Register (Default Value: 0x0000_0000)

Offset: 0x0084			Register Name: L2_STATUS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9	R	0x0	EVENTO Event output. This bit is asserted high for 3 clock cycles when any core in the cluster executes an SEV instruction.
8:0	/	/	/

3.2.4.8 0x00C0 Cluster 0 Debug Control Register0 (Default Value: 0x0000_000F)

Offset: 0x00C0			Register Name: DBG_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11:8	R/W	0x0	DBGRESTART[1:0] Indicates if there are external restart requests for CPU[1:0]. (The bit[11:10] is not used) The meaning of each bit: 0: There are no such requests. 1: There are such requests.
7:4	/	/	/

Offset: 0x00C0			Register Name: DBG_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
3:0	R/W	0xF	C_DBGPWRDUP Indicates if CPU[1:0] is powered-up. (The bit[3:2] is not used) The meaning of each bit: 0: The core is powered down. 1: The core is powered up.

3.2.4.9 0x00C4 Cluster 0 Debug Control Register1 (Default Value: 0x0000_0000)

Offset: 0x00C4			Register Name: DBG_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:12	R/W	0x0	DBGRESTARTED[1:0] (The bit[15:14] is not used) Handshake for DBGRESTART.
11:8	/	/	/
7:4	R	0x0	C_DBGNOPWRDWN Do Not Power Down Request. Indicates if the debugger requests CPU[1:0] not to power down. (The bit[7:6] is not used) The meaning of each bit: 0: There are no such requests. 1: There are such requests.
3:0	R	0x0	C_DBGPWRUPREQ Power Up Request. Indicates if the debugger requests CPU[1:0] to power up. (The bit[3:2] is not used) The meaning of each bit: 0: There are no such requests. 1: There are such requests.

3.2.4.10 0x00D0 CPU AXI Monitor Control Register (Default Value: 0x0000_0000)

Offset: 0x00D0			Register Name: AXI_MNT_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1	R/W	0x0	AXI_MONITOR function clear 0: Not to clear 1: Clear the AXI_MONITOR related registers with the offset address from 0x00D8 to 0x00EC.
0	R/W	0x0	Enable AXI_MONITOR function 0: Disabled 1: Enabled

3.2.4.11 0x00D4 CPU AXI Monitor Period Register (Default Value: 0x0000_0000)

Offset: 0x00D4			Register Name: AXI_MNT_PRD_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	The monitoring period of AXI_MONITOR function (unit: cycles)

3.2.4.12 0x00D8 CPU AXI Monitor Read Latency Register (Default Value: 0x0000_0000)

Offset: 0x00D8			Register Name: AXI_MNT_RLTCY_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	The total latency of AXI_MONITOR function read channel (unit: cycles)

3.2.4.13 0x00DC CPU AXI Monitor Write Latency Register (Default Value: 0x0000_0000)

Offset: 0x00DC			Register Name: AXI_MNT_WLTCY_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	The total latency of AXI_MONITOR function write channel (unit: cycles)

3.2.4.14 0x00E0 CPU AXI Monitor Read Request Register (Default Value: 0x0000_0000)

Offset: 0x00E0			Register Name: AXI_MNT_RREQ_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	The total times of AXI_MONITOR function read channel requests.

3.2.4.15 0x00E4 CPU AXI Monitor Write Request Register (Default Value: 0x0000_0000)

Offset: 0x00E4			Register Name: AXI_MNT_WREQ_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	The total times of AXI_MONITOR function write channel requests.

3.2.4.16 0x00E8 CPU AXI Monitor Read Bandwidth Register (Default Value: 0x0000_0000)

Offset: 0x00E8			Register Name: AXI_MNT_RBD_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	The bandwidth of AXI_MONITOR function read channel (Unit: KB)

3.2.4.17 0x00EC CPU AXI Monitor Write Bandwidth Register (Default Value: 0x0000_0000)

Offset: 0x00EC			Register Name: AXI_MNT_WBD_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	The bandwidth of AXI_MONITOR function write channel (Unit: KB)

3.2.5 CPU Subsystem Control Register List

Module Name	Base Address
CPU_SUBSYS_CTRL	0x08100000

Register Name	Offset	Description
GENER_CTRL_REG0	0x0000	General Control Register0
GIC_JTAG_RST_CTRL	0x000C	GIC and JTAG Reset Control Register
C0_INT_EN	0x0010	Cluster0 Interrupt Enable Control Register

Register Name	Offset	Description
IRQ_FIQ_STATUS	0x0014	IRQ/FIQ Status Register
GENER_CTRL_REG2	0x0018	General Control Register2
DBG_STATE	0x001C	Debug State Register

3.2.6 CPU Subsystem Control Register Description

3.2.6.1 0x0000 General Control Register0 (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: GENER_CTRL_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x0	Cluster 0 Corepll Select Register width state 0: CCU clock 1: Disp PLL clock
1	R/W	0x0	IDC Clock Enable 0: IDC clock is disabled 1: IDC clock is enabled
0	R/W	0x0	GIC_CFGSDISABLE Disable write access to some secure GIC registers. 0: Write access to some secure GIC registers is enabled. 1: Write access to some secure GIC registers is disabled.

3.2.6.2 0x000C GIC and JTAG Reset Control Register (Default Value: 0x0000_0F07)

Offset: 0x000C			Register Name: GIC_JTAG_RST_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/
19:16	R/W	0x0	EXM_CLR[1:0] (The bit[19:18] is not used) Clear the status of interface for debug.
15:12	/	/	/

Offset: 0x000C			Register Name: GIC_JTAG_RST_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
11	R/W	0x1	CS_RST CoreSight reset 0: Assert 1: De-assert
10	R/W	0x1	DAP_RST DAP reset 0: Assert 1: De-assert
9	R/W	0x1	PORTRST DAP reset signal control 0: Assert 1: De-assert
8	R/W	0x1	TRST JTAG RTST signal 0: Assert 1: De-assert
7:2	/	/	/
1	R/W	0x1	IDC_RST Interrupt delay controller reset 0: Assert 1: De-assert
0	R/W	0x1	GIC_RST GIC_reset_cpu_reg 0: Assert 1: De-assert

3.2.6.3 0x0010 Cluster 0 Interrupt Enable Register (Default Value: 0x0000_FFFF)

Offset: 0x0010			Register Name: C0_INT_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/

Offset: 0x0010			Register Name: C0_INT_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
15:0	R/W	0xFFFF	C0_GIC_EN Mask the FIQ_OUT[15:0] and IRQ_OUT[15:0] signals to system domain. The meaning of each bit: 0: Mask the FIQ_OUT and IRQ_OUT signals to system domain. 1: Not to mask the FIQ_OUT and IRQ_OUT signals to system domain.

3.2.6.4 0x0014 GIC IRQ/FIQ Status Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: IRQ_FIQ_STATUS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0000	FIQ_OUT[15:0]
15:0	R/W	0x0000	IRQ_OUT[15:0]

3.2.6.5 0x0018 General Control Register2 (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: GENER_CTRL_REG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	CDBGRSTACK Debug Reset ACK 0: There are no debug reset ACKs. 1: There are debug reset ACKs.
15:1	/	/	/
0	R/W	0x0	C0_TSCLKCHANGE Cluster 0 time stamp change. 0: There are no time stamp changes. 1: There are time stamp changes.

3.2.6.6 0x001C Debug State Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: DBG_STATE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R	0x0	CLU_PWRSW_STA
23:1	/	/	/
0	R	0x0	C0_DBG_STATE Indicates if Cluster0 is in debug mode or normal mode. 0: Normal mode 1: Debug mode

3.3 Clock Controller Unit (CCU)

3.3.1 Overview

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

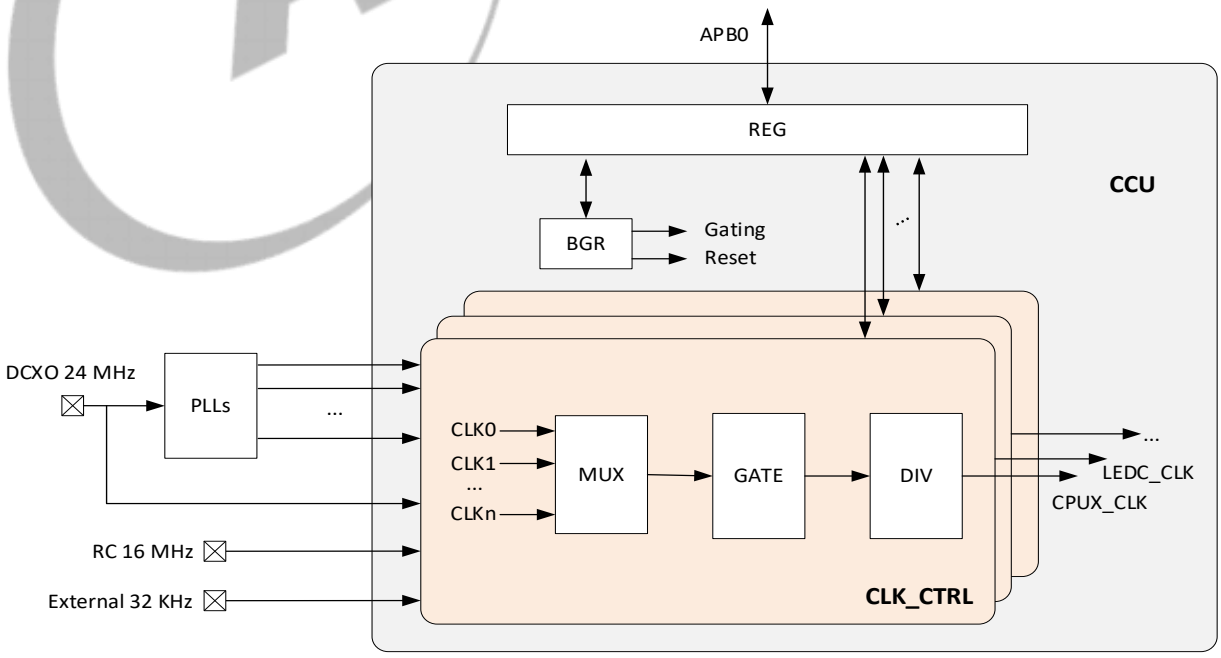
The CCU includes the following features:

- 8 PLLs
- Bus source and divisions
- Clock output control
- Configuring modules clock
- Bus clock gating
- Bus software reset

3.3.2 Block Diagram

The following figure shows the functional block diagram of the CCU.

Figure 3-2 CCU Block Diagram



3.3.3 Functional Description

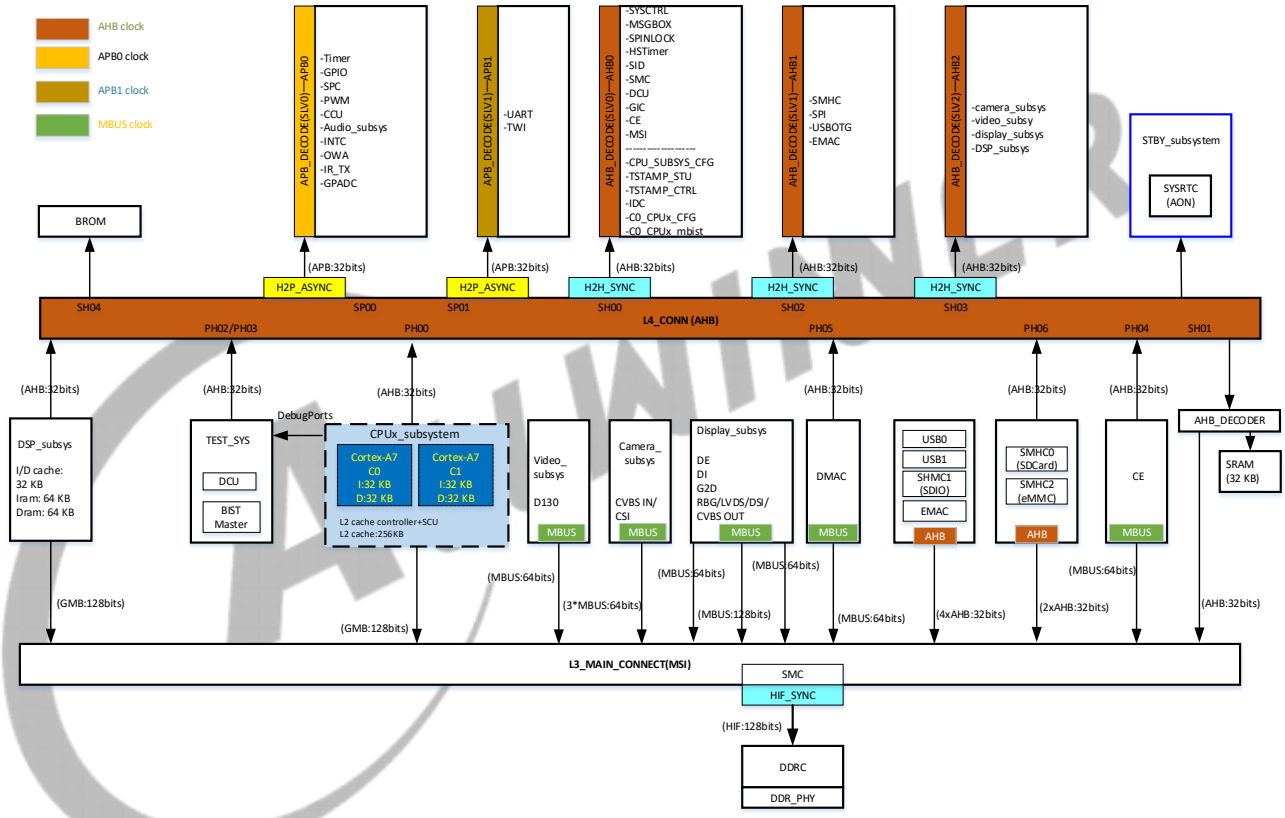
3.3.3.1 System Bus Tree

The system buses include advanced high-performance buses (AHBs), advanced peripheral buses (APBs), and MBUS.

All devices mounted at the bus should use the related bus clocks, and the gating signals for the bus are from the CCU module.

The following figure shows the diagram of the System Bus Tree.

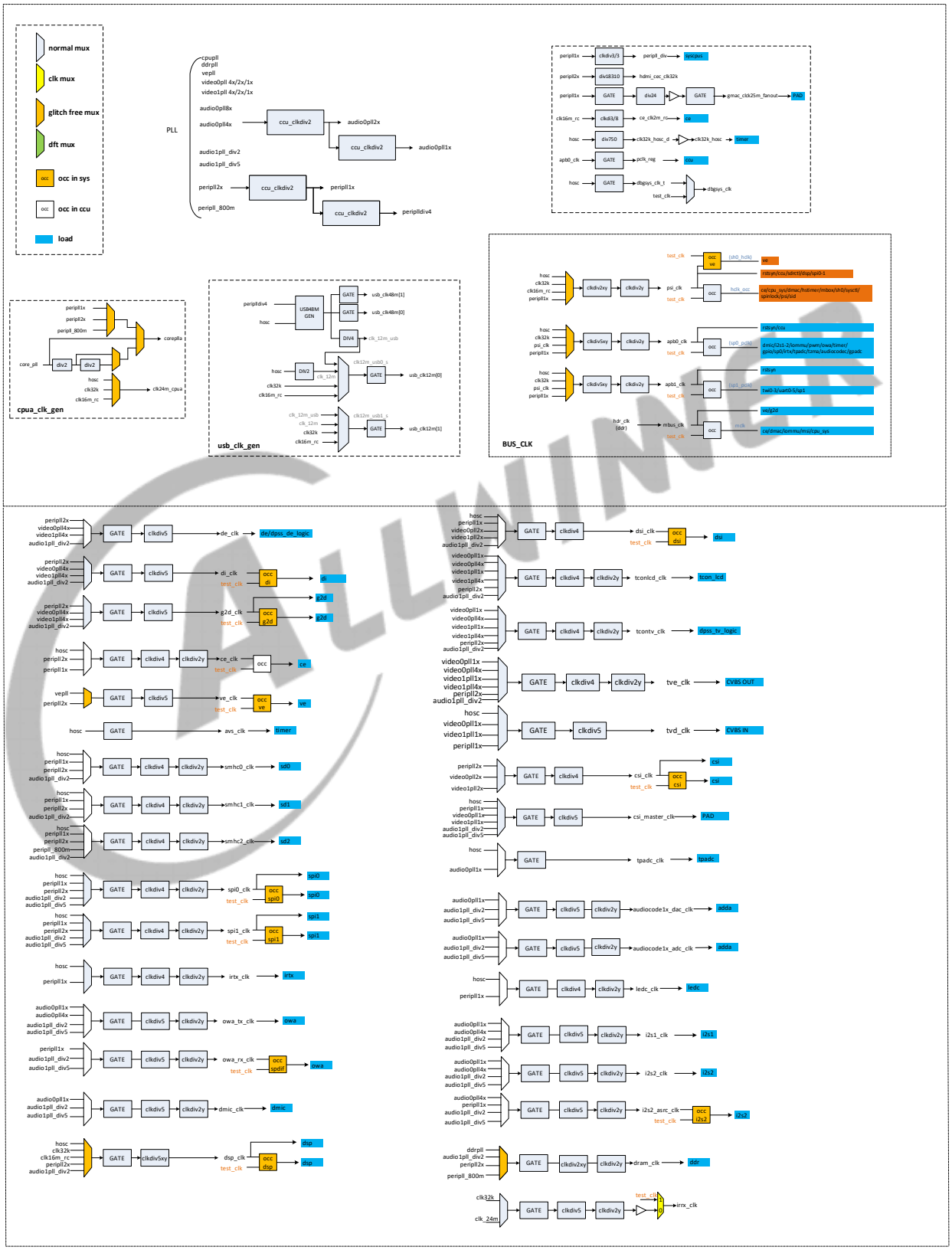
Figure 3-3 System Bus Tree



3.3.3.2 Module Clock Generation

The following figure describes module clock generation.

Figure 3-4 Module Clock Generation



3.3.3.3 PLL Distribution

The following figure shows the block diagram of the PLL distribution.

Figure 3-5 PLL Distribution

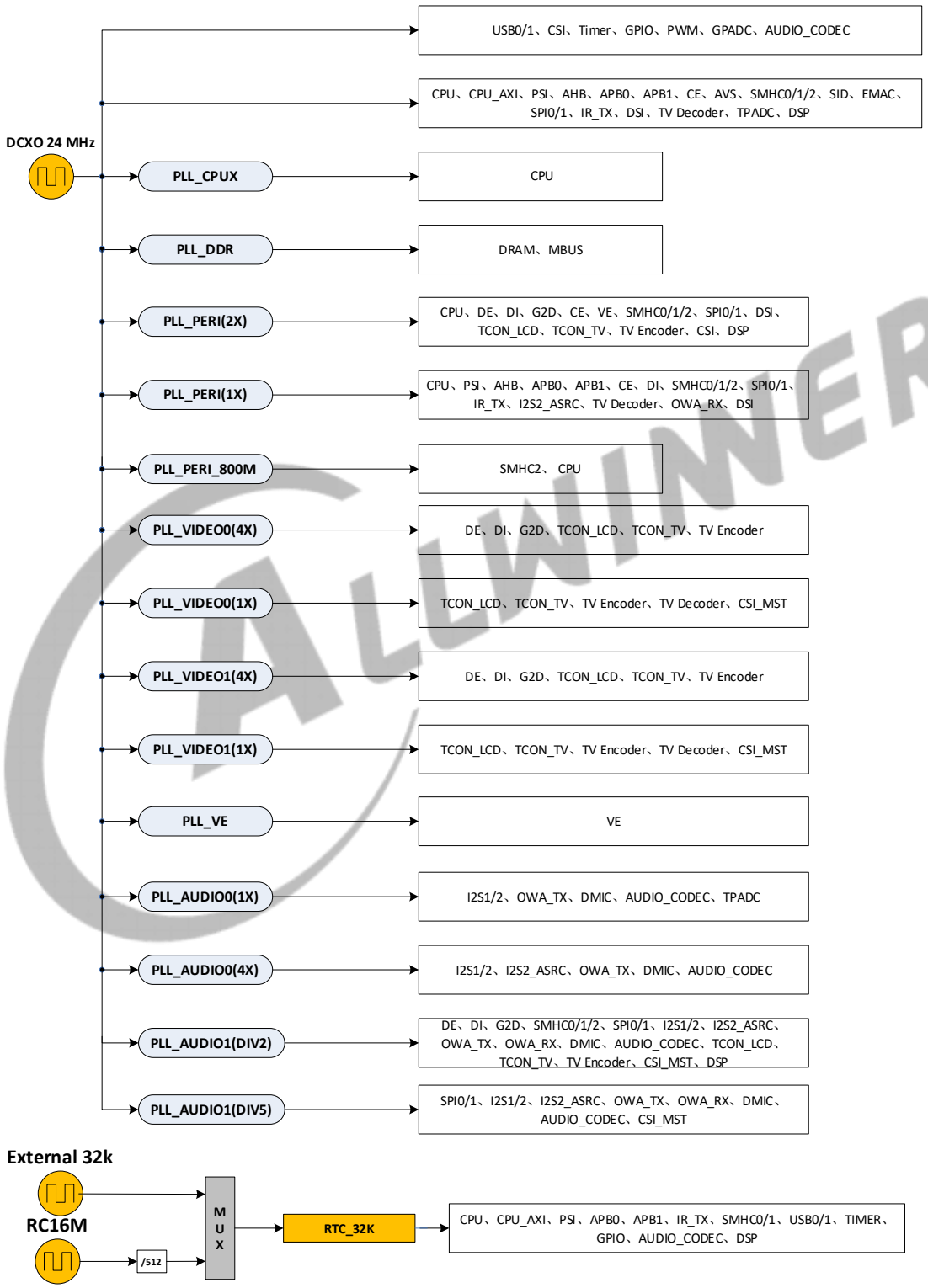


Table 3-3 PLL Typical Application

PLL Type	Application Module	Notes
PLL_CPU	CPU	Support DVFS
PLL_DDR	MBUS, SDRAM	Support spread spectrum No support linear FM
PLL_PERI(2X)	CPU, DE, DI, G2D, CE, VE, SMHC0/1/2, SPI0/1, DSI, TCON_LCD, TCON_TV, TV Encoder, CSI, DSP	No support dynamic FM
PLL_PERI(1X)	CPU, PSI, AHB, APB0, APB1, CE, DI, SMHC0/1/2, SPI0/1, IR_TX, I2S2_ASRC, TV Decoder, OWA_RX, DSI	No support dynamic FM
PLL_PERI(800M)	SMHC2, CPU	No support dynamic FM
PLL_VIDEO0(4X)	DE, DI, G2D, TCON_LCD, TCON_TV, TV Encoder	No support DVFS
PLL_VIDEO0(1X)	TCON_LCD, TCON_TV, TV Encoder, TV Decoder, CSI_MST	No support DVFS
PLL_VIDEO1(4X)	DE, DI, G2D, TCON_LCD, TCON_TV, TV Encoder	No support DVFS
PLL_VIDEO1(1X)	TCON_LCD, TCON_TV, TV Encoder, TV Decoder, CSI_MST	No support DVFS
PLL_VE	VE	No support dynamic FM
PLL_AUDIO0	I2S1/2, OWA_TX, DMIC, AUDIO_CODEC, TPADC	No support DVFS
PLL_AUDIO0(4X)	I2S1/2, I2S2_ASRC, OWA_TX, DMIC, AUDIO_CODEC	No support DVFS
PLL_AUDIO1(DIV2)	DE, DI, G2D, SMHC0/1/2, SPI0/1, I2S1/2, I2S2_ASRC, OWA_TX, OWA_RX, DMIC, AUDIO_CODEC, TCON_LCD, TCON_TV, TV Encoder, CSI_MST, DSP	No support DVFS
PLL_AUDIO1(DIV5)	SPI0/1, I2S1/2, I2S2_ASRC, OWA_TX, OWA_RX, DMIC, AUDIO_CODEC, CSI_MST	No support DVFS

3.3.3.4 PLL Features

The following table shows the PLL features.

Table 3-4 PLL Features

PLL	Stable Operating Frequency	Actual Operating Frequency	Default Frequency	Spread Spectrum	Linear FM	Rate Control	Pk-Pk	Lock Time
PLL_CPU	288 MHz to 3.0 GHz (24*N)	288 MHz to 1.8 GHz	408 MHz	No	Yes	No	< 200 ps	1.5 ms
PLL_AUDIO0	180 MHz to 3.0 GHz (24MHz*N.x/M1/M0/P)	24.576 MHz 22.5792 MHz 24.576*4 MHz 22.5792*4 MHz	24.576 MHz	Yes	No	No	< 200 ps	500 us
PLL_AUDIO1	180 MHz to 3.0 GHz	Fvco/Div (Div: 1–8)	Integer mode: 1/2x: 1.536 GHz 1/5x: 614.4 MHz Decimal mode: 1/2x: 1.1179648 GHz 1/5x: 471.8592 MHz	Yes	No	No	< 200 ps	500 us
PLL_PERI	180 MHz to 3.2 GHz (24*N/M1/M0)	Fvco/Div (Div:1–8)	1/2x: 1.2 GHz 1/3x: 800 MHz	Yes	No	No	< 200 ps	500 us
PLL_Video0(4X)	252 MHz to 3.0 GHz (24*N/M)	192 MHz to 2400 MHz	1x: 297 MHz 2x:594 MHz 4x:1188 MHz	Yes	No	No	< 200 ps	500 us
PLL_Video1(4X)	252 MHz to 3.0 GHz (24*N/M)	192 MHz to 2400 MHz	1x: 297 MHz 2x: 594 MHz 4x: 1188 MHz	Yes	No	No	< 200 ps	500 us
PLL_VE	180 MHz to 3.0 GHz (24*N/M1/M0)	192 MHz to 600 MHz	432 MHz	Yes	No	No	< 200 ps	500 us
PLL_DDR	180 MHz to 3.0 GHz (24*N/M1/M0)	192 MHz to 2.0 GHz	432 MHz	Yes	No	No	200 MHz to 800 MHz (< 200 ps) 800 MHz to 1.3GHz (< 140 ps) 1.3 GHz to 2.0 GHz (< 100 ps)	2 ms

3.3.4 Programming Guidelines

3.3.4.1 Configuring the Frequency of PLL_CPUX

The frequency configuration formula of PLL_CPUX:

PLL_CPUX = 24 MHz*N/P

The parameter N is the frequency-doubling factor of PLL. The next parameter configuration can proceed after the PLL relock. The parameter P is a digital post-frequency-division factor, which can be dynamically switched in real-time, without affecting the normal work of PLL.

NOTE

PLL_CPUX supports dynamic frequency adjustment (modifying the value of N). However, for the system stability, to configure the frequency of the PLL_CPUX from a higher value to a lower one, switch the clock source of the CPU to another clock whose frequency is not higher than the current one first, and configure PLL_CPUX to the target low frequency, and then switch the clock source of the CPU back to PLL_CPUX.

Follow the steps below to adjust the frequency of PLL_CPUX:

Step 1 Before you configure PLL_CPUX, switch the clock source of CPU to PLL_PERI(1X).

Step 2 Modify the parameters N and P of PLL_CPUX.

Step 3 Write the PLL Lock Enable bit (bit[29]) of **PLL_CPU_CTRL_REG** to 0 and then to 1.

Step 4 Wait for the Lock bit (bit[28]) of **PLL_CPU_CTRL_REG** to change to 1.

Step 5 Switch the clock source of the CPU to PLL_CPUX.

3.3.4.2 Configuring the Frequency of PLL_AUDIO

NOTE

PLL_AUDIO includes PLL_AUDIO0 and PLL_AUDIO1.

The frequency configuration formula of PLL_AUDIO:

PLL_AUDIO0 = 24 MHz*N/M0/M1/P/4

PLL_AUDIO1 = 24 MHz*N/M

PLL_AUDIO does not support dynamic adjustment because changing any parameter of N, M0, M1, and P will affect the normal work of PLL, and the PLL will need to be relocked.

Generally, PLL_AUDIO only needs two frequency points: 24.576*4 MHz or 22.5792*4 MHz. For these two frequencies, there are usually special recommended matching factors. To implement the desired frequency point of PLL_AUDIO, you need to use the decimal frequency-division function, so follow the steps below:

Step 1 Configure the N, M0, M1 and P factors.

Step 2 Configure PLL_AUDIOx Control Register[PLL_SDM_ENABLE] to 1.

Step 3 Configure PLL_AUDIOx Pattern0 Control Register to enable the digital spread spectrum.

Step 4 Write PLL_AUDIOx Control Register[Lock Enable] to 0 and then to 1.

Step 5 Wait for PLL_AUDIOx Control Register[Lock] to 1.

NOTE

When the P factor of PLL_AUDIO is an odd number, the clock output is an unequal-duty-cycle signal.

The recommended values for configuration factors of PLL_AUDIO1 are as follows.

Table 3-5 Recommended Values for Configuration Factors of PLL_AUDIO1

Mode	Clock Source (MHz)	Frequency-doubling N	VCO (MHz)	Post Frequency-Division	PLL Output (MHz)	Divisor	Actual Operating Frequency (MHz)	Description
Integer divider	24	128	3072	2	1536	4	384	Provide clock source for peripherals
						8	192	
						16	96	
						……		
				5	614.4	25	24.576	For audio-related modules
Decimal divider	24	98.304	2359.296	2	1179.648	2	589.824	Provides clock source for peripheral devices
						3	393.216	
						6	196.608	
						12	98.304	

Mode	Clock Source (MHz)	Frequency-doubling N	VCO (MHz)	Post Frequency-Division	PLL Output (MHz)	Divisor	Actual Operating Frequency (MHz)	Description
						48	24.576	For audio-related modules
				5	471.8592			

3.3.4.3 Configuring the Frequency of General PLLs

Step 1 Make sure the PLL is enabled. If not, refer to section 3.3.4.4 Enabling the PLL to enable the PLL.

Step 2 Configure the PLL_OUTPUT_ENABLE bit (bit[27]) of the PLL control register as 0 to disable the output gate of the PLL because general PLLs are unavailable in the process of frequency modulation.

Step 3 Configure the N and M factors. (It is not suggested to configure M1 factor).

Step 4 Write the LOCK_ENABLE bit (bit[29]) of the PLL control register to 0 and then to 1.

Step 5 Wait for the LOCK bit (bit[28]) of the PLL control register to 1.

Step 6 Configure PLL_OUTPUT_ENABLE bit (bit[27]) of the PLL control register to 1.

3.3.4.4 Enabling the PLL

Follow the steps below to enable the PLL:

Step 1 Configure the N, M, and P factors of the PLL control register.

Step 2 Write the PLL_ENABLE bit and the LDO_EN bit of the PLL control register to 1, write the PLL_OUTPUT_GATE bit of the PLL control register to 0.

Step 3 Write the LOCK_ENABLE bit of the PLL control register to 1.

Step 4 Wait for the status of the Lock to change to 1.

Step 5 Delay 20 us.

Step 6 Write the PLL_OUTPUT_GATE bit of the PLL control register to 1 and then the PLL will be available.

3.3.4.5 Disabling the PLL

Follow the steps below to disable the PLL:

Step 1 Write the PLL_ENABLE bit (bit[31]) and the LDO_EN bit of the PLL control register to 0.

Step 2 Write the LOCK_ENABLE bit (bit[29]) of the PLL control register to 0.

CAUTION

In the normal use of PLLs, it is unsuggested to enable and disable the PLLs frequently. Turning on and off the PLLs will cause mutual interference between PLLs, which will affect the stability of the system. When the clock is unnecessary, you can write 0 to the PLL_OUTPUT_EN bit of the PLL control register to disable the output gate of the PLL, instead of writing 0 to the Enable bit to disable the PLL.

3.3.4.6 Configuring Bus Clock

The bus clock supports dynamic switching, but the process of switching needs to follow the following two rules.

- From a higher frequency to a lower frequency: switch the clock source first, and then set the frequency division factor;
- From a lower frequency to a higher frequency: configure the frequency division factor first, and then switch the clock source.

3.3.4.7 Configuring Module Clock

For the Bus Gating Reset register of a module, the reset bit is de-asserted first, and then the clock gating bit is enabled to avoid potential problems caused by the asynchronous release of the reset signal.

For all module clocks except the DDR clock, configure the clock source and frequency division factor first, and then release the clock gating (that is, set to 1). For the configuration order of the clock source and frequency division factor, follow the rules below:

- With the increasing of the clock source frequency, configure the frequency division factor before the clock source;
- With the decreasing of the clock source frequency, configure the clock source before the frequency division factor.

3.3.4.8 Implementing Spread Spectrum

The spread spectrum technology is to convert a narrowband signal into a wideband signal. It helps to reduce the effect of electromagnetic interference (EMI) associated with the fundamental frequency of the signal.

For the general PLL frequency, the calculation formula is as follows:

$$f = \frac{N+1+X}{P\cdot(M0+1)\cdot(M1+1)}\cdot 24MHz, 0<X<1$$

Where,

P is the frequency division factor of module or PLL;

M0 is the post-frequency division factor of PLL;

M1 is the pre-frequency division factor of PLL;

N is the frequency doubling factor of PLL;

X is the amplitude coefficient of the spread spectrum.

The parameters N, P, M1, and M0 are for the frequency division.

When M1 = 0, M0 = 0, and P = 1 (no frequency division), the calculation formula of PLL frequency can be simplified as follows:

$$f = (N+1+X)\cdot 24MHz,\ 0<X<1$$

$$[f_1, f_2] = (N+1+[X_1, X_2])\cdot 24MHz$$

$$SDM\_BOT = 2^{17}\cdot X_1$$

$$\text{WAVE\_STEP} = 2^{17}\cdot(X_2 - X_1)/(24\text{MHz}/PREQ)\cdot 2$$

Where, SDM_BOT and WAVE_STEP are bits of the PLL pattern control register, and PREQ is the frequency of the spread spectrum.

NOTE

Different PLLs have different calculate formulas, refer to the CTRL register of the corresponding PLL in section 3.3.6 Register Description.

Configuration Procedure

Follow the steps below to implement the spread spectrum:

Step 1 Configure the control register of the corresponding PLL

a) Calculate the factor N and decimal value X according to the PLL frequency and PLL frequency formula. Refer to the control register of the corresponding PLL (named PLL_xxx_CTRL_REG, where xxx is the module name) in 3.3.6 Register Description for the corresponding PLL frequency formula.

b) Write M0, M1, N, and PLL frequency to the PLL control register.

c) Configure the SDM_Enable bit (bit[24]) of the PLL control register to 1 to enable the spread spectrum function.

Step 2 Configure the pattern control register of the corresponding PLL

a) Calculate the SDM_BOT and WAVE_STEP of the pattern control register according to decimal value X and spread spectrum frequency (the bit[18:17] of the PLL_PAT register)

b) Configure the spread spectrum mode (SPR_FREQ_MODE) to 2 or 3.

c) If the PLL_INPUT_DIV2 of the PLL control register is 1, configure the spread spectrum clock source select bit (SDM_CLK_SEL) of the PLL pattern control register to 1. Otherwise, configure SDM_CLK_SEL to the default value 0.

d) Write SDM_BOT, WAVE_STEP, PREQ, SPR_FREQ_MODE, and SDM_CLK_SEL to the PLL pattern control register, and configure the SIG_DELT_PAT_EN bit (bit[31]) of this register to 1.

Step 3 Delay 20 us

Configuration Example

The following example shows how to configure the spread spectrum frequency as 605.3 MHz to 609.7 MHz.

If M1 = 0, M0 = 0, P = 1, according to the formula $[f_1, f_2] = (N + 1 + [X_1, X_2]) \cdot 24MHz$, you can get:

$$N + 1 + [X_1, X_2] = \frac{[605.3, 609.7]}{24} = \frac{600 + [5.3, 9.7]}{24} = 24 + 1 + [5.3/24, 9.7/24]$$

Obviously,

N = 24, X_1 = 5.3/24, X_2 = 9.7/24

SDM_BOT = 2^{17} * X_1 = 0x7111

WAVE_STEP = 2^{17} * ($X_2 - X_1$) /(24M/PREQ)*2 = 0x3f; PREQ = 31.5 kHz

If M0 = 1, M1=0, P = 1, then total frequency division factor is (M0 + 1) *1 = 2, so the actual output frequency of PLL is 1212.1 MHz to 1219.4 MHz.

Similarly, you can get:

N = 49, X1 = 12.1/24, X2 = 19.4/24

Then calculate the values of SDM_BOT and WAVE_STEP according to the formulas, and follow the steps described in Configuration Procedure.

3.3.5 Register List

Module Name	Base Address
CCU	0x02001000

Register Name	Offset	Description
PLL_CPU_CTRL_REG	0x0000	PLL_CPU Control Register
PLL_DDR_CTRL_REG	0x0010	PLL_DDR Control Register
PLL_PERI_CTRL_REG	0x0020	PLL_PERI Control Register
PLL_VIDEO0_CTRL_REG	0x0040	PLL_VIDEO0 Control Register
PLL_VIDEO1_CTRL_REG	0x0048	PLL_VIDEO1 Control Register
PLL_VE_CTRL_REG	0x0058	PLL_VE Control Register
PLL_AUDIO0_CTRL_REG	0x0078	PLL_AUDIO0 Control Register
PLL_AUDIO1_CTRL_REG	0x0080	PLL_AUDIO1 Control Register
PLL_DDR_PAT0_CTRL_REG	0x0110	PLL_DDR Pattern0 Control Register
PLL_DDR_PAT1_CTRL_REG	0x0114	PLL_DDR Pattern1 Control Register
PLL_PERI_PAT0_CTRL_REG	0x0120	PLL_PERI Pattern0 Control Register
PLL_PERI_PAT1_CTRL_REG	0x0124	PLL_PERI Pattern1 Control Register
PLL_VIDEO0_PAT0_CTRL_REG	0x0140	PLL_VIDEO0 Pattern0 Control Register
PLL_VIDEO0_PAT1_CTRL_REG	0x0144	PLL_VIDEO0 Pattern1 Control Register
PLL_VIDEO1_PAT0_CTRL_REG	0x0148	PLL_VIDEO1 Pattern0 Control Register
PLL_VIDEO1_PAT1_CTRL_REG	0x014C	PLL_VIDEO1 Pattern1 Control Register
PLL_VE_PAT0_CTRL_REG	0x0158	PLL_VE Pattern0 Control Register
PLL_VE_PAT1_CTRL_REG	0x015C	PLL_VE Pattern1 Control Register
PLL_AUDIO0_PAT0_CTRL_REG	0x0178	PLL_AUDIO0 Pattern0 Control Register
PLL_AUDIO0_PAT1_CTRL_REG	0x017C	PLL_AUDIO0 Pattern1 Control Register
PLL_AUDIO1_PAT0_CTRL_REG	0x0180	PLL_AUDIO1 Pattern0 Control Register
PLL_AUDIO1_PAT1_CTRL_REG	0x0184	PLL_AUDIO1 Pattern1 Control Register
PLL_CPU_BIAS_REG	0x0300	PLL_CPU Bias Register
PLL_DDR_BIAS_REG	0x0310	PLL_DDR Bias Register
PLL_PERI_BIAS_REG	0x0320	PLL_PERI Bias Register
PLL_VIDEO0_BIAS_REG	0x0340	PLL_VIDEO0 Bias Register
PLL_VIDEO1_BIAS_REG	0x0348	PLL_VIDEO1 Bias Register
PLL_VE_BIAS_REG	0x0358	PLL_VE Bias Register
PLL_AUDIO0_BIAS_REG	0x0378	PLL_AUDIO0 Bias Register

Register Name	Offset	Description
PLL_AUDIO1_BIAS_REG	0x0380	PLL_AUDIO1 Bias Register
PLL_CPU_TUN_REG	0x0400	PLL_CPU Tuning Register
CPU_AXI_CFG_REG	0x0500	CPU_AXI Configuration Register
CPU_GATING_REG	0x0504	CPU_GATING Configuration Register
PSI_CLK_REG	0x0510	PSI Clock Register
APB0_CLK_REG	0x0520	APB0 Clock Register
APB1_CLK_REG	0x0524	APB1 Clock Register
MBUS_CLK_REG	0x0540	MBUS Clock Register
DE_CLK_REG	0x0600	DE Clock Register
DE_BGR_REG	0x060C	DE Bus Gating Reset Register
DI_CLK_REG	0x0620	DI Clock Register
DI_BGR_REG	0x062C	DI Bus Gating Reset Register
G2D_CLK_REG	0x0630	G2D Clock Register
G2D_BGR_REG	0x063C	G2D Bus Gating Reset Register
CE_CLK_REG	0x0680	CE Clock Register
CE_BGR_REG	0x068C	CE Bus Gating Reset Register
VE_CLK_REG	0x0690	VE Clock Register
VE_BGR_REG	0x069C	VE Bus Gating Reset Register
DMA_BGR_REG	0x070C	DMA Bus Gating Reset Register
MSGBOX_BGR_REG	0x071C	MSGBOX Bus Gating Reset Register
SPINLOCK_BGR_REG	0x072C	SPINLOCK Bus Gating Reset Register
HSTIMER_BGR_REG	0x073C	HSTIMER Bus Gating Reset Register
AVS_CLK_REG	0x0740	AVS Clock Register
DBGSYS_BGR_REG	0x078C	DBGSYS Bus Gating Reset Register
PWM_BGR_REG	0x07AC	PWM Bus Gating Reset Register
IOMMU_BGR_REG	0x07BC	IOMMU Bus Gating Reset Register
DRAM_CLK_REG	0x0800	DRAM Clock Register
MBUS_MAT_CLK_GATING_REG	0x0804	MBUS Master Clock Gating Register
DRAM_BGR_REG	0x080C	DRAM Bus Gating Reset Register
SMHC0_CLK_REG	0x0830	SMHC0 Clock Register
SMHC1_CLK_REG	0x0834	SMHC1 Clock Register
SMHC2_CLK_REG	0x0838	SMHC2 Clock Register
SMHC_BGR_REG	0x084C	SMHC Bus Gating Reset Register
UART_BGR_REG	0x090C	UART Bus Gating Reset Register

Register Name	Offset	Description
TWI_BGR_REG	0x091C	TWI Bus Gating Reset Register
SPI0_CLK_REG	0x0940	SPI0 Clock Register
SPI1_CLK_REG	0x0944	SPI1 Clock Register
SPI_BGR_REG	0x096C	SPI Bus Gating Reset Register
EMAC_25M_CLK_REG	0x0970	EMAC_25M Clock Register
EMAC_BGR_REG	0x097C	EMAC Bus Gating Reset Register
IRTX_CLK_REG	0x09C0	IRTX Clock Register
IRTX_BGR_REG	0x09CC	IRTX Bus Gating Reset Register
GPADC_BGR_REG	0x09EC	GPADC Bus Gating Reset Register
THS_BGR_REG	0x09FC	THS Bus Gating Reset Register
I2S1_CLK_REG	0x0A14	I2S1 Clock Register
I2S2_CLK_REG	0x0A18	I2S2 Clock Register
I2S2_ASRC_CLK_REG	0x0A1C	I2S2_ASRC Clock Register
I2S_BGR_REG	0x0A20	I2S Bus Gating Reset Register
OWA_TX_CLK_REG	0x0A24	OWA_TX Clock Register
OWA_RX_CLK_REG	0x0A28	OWA_RX Clock Register
OWA_BGR_REG	0x0A2C	OWA Bus Gating Reset Register
DMIC_CLK_REG	0x0A40	DMIC Clock Register
DMIC_BGR_REG	0x0A4C	DMIC Bus Gating Reset Register
AUDIO_CODEC_DAC_CLK_REG	0x0A50	AUDIO_CODEC_DAC Clock Register
AUDIO_CODEC_ADC_CLK_REG	0x0A54	AUDIO_CODEC_ADC Clock Register
AUDIO_CODEC_BGR_REG	0x0A5C	AUDIO_CODEC Bus Gating Reset Register
USB0_CLK_REG	0x0A70	USB0 Clock Register
USB1_CLK_REG	0x0A74	USB1 Clock Register
USB_BGR_REG	0x0A8C	USB Bus Gating Reset Register
DPSS_TOP_BGR_REG	0x0ABC	DPSS_TOP Bus Gating Reset Register
DSI_CLK_REG	0x0B24	DSI Clock Register
DSI_BGR_REG	0x0B4C	DSI Bus Gating Reset Register
TCONLCD_CLK_REG	0x0B60	TCONLCD Clock Register
TCONLCD_BGR_REG	0x0B7C	TCONLCD Bus Gating Reset Register
TCONTV_CLK_REG	0x0B80	TCONTV Clock Register
TCONTV_BGR_REG	0x0B9C	TCONTV Bus Gating Reset Register
LVDS_BGR_REG	0x0BAC	LVDS Bus Gating Reset Register
TVE_CLK_REG	0x0BB0	TVE Clock Register

Register Name	Offset	Description
TVE_BGR_REG	0x0BBC	TVE Bus Gating Reset Register
TVD_CLK_REG	0x0BC0	TVD Clock Register
TVD_BGR_REG	0x0BDC	TVD Bus Gating Reset Register
LEDC_CLK_REG	0x0BF0	LEDC Clock Register
LEDC_BGR_REG	0x0BFC	LEDC Bus Gating Reset Register
CSI_CLK_REG	0x0C04	CSI Clock Register
CSI_MASTER_CLK_REG	0x0C08	CSI Master Clock Register
CSI_BGR_REG	0x0C1C	CSI Bus Gating Reset Register
TPADC_CLK_REG	0x0C50	TPADC Clock Register
TPADC_BGR_REG	0x0C5C	TPADC Bus Gating Reset Register
DSP_CLK_REG	0x0C70	DSP Clock Register
DSP_BGR_REG	0x0C7C	DSP Bus Gating Reset Register
PLL_LOCK_DBG_CTRL_REG	0x0F04	PLL Lock Debug Control Register
FRE_DET_CTRL_REG	0x0F08	Frequency Detect Control Register
FRE_UP_LIM_REG	0x0F0C	Frequency Up Limit Register
FRE_DOWN_LIM_REG	0x0F10	Frequency Down Limit Register
CCU_FAN_GATE_REG	0x0F30	CCU FANOUT CLOCK GATE Register
CLK27M_FAN_REG	0x0F34	CLK27M FANOUT Register
PCLK_FAN_REG	0x0F38	PCLK FANOUT Register
CCU_FAN_REG	0x0F3C	CCU FANOUT Register

3.3.6 Register Description

3.3.6.1 0x0000 PLL_CPU Control Register (Default Value: 0x4A00_1000)

Offset: 0x0000			Register Name: PLL_CPU_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_EN PLL Enable 0: Disable 1: Enable The PLL_CPU= InputFreq*N. **The PLL_CPU output frequency must be in the range from 200 MHz to 3 GHz. And the default value of PLL_CPUX is 408 MHz when the crystal oscillator is 24 MHz.**

Offset: 0x0000			Register Name: PLL_CPU_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
30	R/W	0x1	PLL_LDO_EN LDO Enable 0: Disable 1: Enable
29	R/W	0x0	LOCK_ENABLE Lock Enable 0: Disable 1: Enable
28	R	0x0	LOCK PLL Lock Status 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)
27	R/W	0x1	PLL_OUTPUT_GATE PLL Output Gating Enable 0: Disable 1: Enable The bit is used to control the output enable of the PLL.
26:24	R/W	0x2	PLL_LOCK_TIME. PLL Lock Time The bit indicates the step amplitude from one frequency to another.
23:16	/	/	/
15:8	R/W	0x10	PLL_N PLL N N= PLL_N +1 PLL_N is from 0 to 254. In application, PLL_N shall be more than or equal to 12.
7:6	R/W	0x0	PLL_UNLOCK_MDSEL PLL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles
5	R/W	0x0	PLL_LOCK_MDSEL PLL Lock Level 0: 24-26 Clock Cycles 1: 23-27 Clock Cycles

Offset: 0x0000			Register Name: PLL_CPU_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
4:2	/	/	/
1:0	R/W	0x0	PLL_M PLL_M M = PLL_FACTOR_M + 1 PLL_FACTOR_M is from 0 to 3. **Note: The M factor is only for testing.**

3.3.6.2 0x0010 PLL_DDR Control Register (Default Value: 0x4800_2301)

Offset: 0x0010			Register Name: PLL_DDR_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_EN 0: Disable 1: Enable PLL_DDR = InputFreq*N/M1/M0. **The default value of PLL_DDR is 432 MHz when the crystal oscillator is 24 MHz.**
30	R/W	0x1	PLL_LDO_EN LDO ENABLE 0: Disable 1: Enable
29	R/W	0x0	LOCK_ENABLE Lock Enable 0: Disable 1: Enable
28	R	0x0	LOCK PLL Lock 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)
27	R/W	0x1	PLL_OUTPUT_GATE PLL Output Gating Enable 0: Disable 1: Enable The bit is used to control the output enable of the PLL.
26:25	/	/	/

Offset: 0x0010			Register Name: PLL_DDR_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
24	R/W	0x0	PLL_SDM_EN PLL SDM Enable 0: Disable 1: Enable
23:16	/	/	/
15:8	R/W	0x23	PLL_N PLL N N= PLL_N +1 PLL_N is from 0 to 254. In application, PLL_N shall be more than or equal to 12.
7:6	R/W	0x0	PLL_UNLOCK_MDSEL PLL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles
5	R/W	0x0	PLL_LOCK_MDSEL PLL Lock Level 0: 24-26 Clock Cycles 1: 23-27 Clock Cycles
4:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV2 PLL Input Div M1 M1=PLL_INPUT_DIV2 + 1 PLL_INPUT_DIV2 is from 0 to 1.
0	R/W	0x1	PLL_OUTPUT_DIV2 PLL Output Div M0 M0=PLL_OUTPUT_DIV2 + 1 PLL_OUTPUT_DIV2 is from 0 to 1.

3.3.6.3 0x0020 PLL_PERI Control Register (Default Value: 0x4821_6300)

Offset: 0x0020			Register Name: PLL_PERI_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_EN PLL Enable 0: Disable 1: Enable PLL_PERI(2X) = 24 MHz*N/M/P0 PLL_PERI(1X) = 24 MHz*N/M/P0/2 PLL_PERI(800M) = 24 MHz*N/M/P1. When the crystal oscillator is 24 MHz, the default frequency of PLL_PERI(2X) is 1.2 GHz, the default frequency of PLL_PERI(1X) is 600 MHz, and the default frequency of PLL_PERI(800M) is 800 MHz. **The output clock of PLL_PERI(2X) is fixed to 1.2 GHz and not suggested to change the parameter.**
30	R/W	0x1	PLL_LDO_EN LDO Enable 0: Disable 1: Enable
29	R/W	0x0	LOCK_ENABLE Lock Enable 0: Disable 1: Enable
28	R	0x0	LOCK PLL Lock 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)
27	R/W	0x1	PLL_OUTPUT_GATE PLL Output Gating Enable 0: Disable 1: Enable The bit is used to control the output enable of PLL.
26:25	/	/	/

Offset: 0x0020			Register Name: PLL_PERI_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
24	R/W	0x0	PLL_SDM_EN PLL SDM Enable 0: Disable 1: Enable Enable spread spectrum and decimal division.
23	/	/	/
22:20	R/W	0x2	PLL_P1 PLL Output Div P1 P1=PLL_OUTPUT_DIV_P1 + 1 PLL_OUTPUT_DIV_P1 is from 0 to 7.
19	/	/	/
18:16	R/W	0x1	PLL_P0 PLL Output Div P0 P0=PLL_OUTPUT_DIV_P0 + 1 PLL_OUTPUT_DIV_P0 is from 0 to 7.
15:8	R/W	0x63	PLL_N PLL N N= PLL_N +1 PLL_N is from 0 to 254. In application, PLL_N shall be more than or equal to 12.
7:6	R/W	0x0	PLL_UNLOCK_MDSEL PLL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles
5	R/W	0x0	PLL_LOCK_MDSEL PLL Lock Level 0: 24-26 Clock Cycles 1: 23-27 Clock Cycles
4:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV2 PLL Input Div M M1=PLL_INPUT_DIV2 + 1 PLL_INPUT_DIV2 is from 0 to 1.
0	/	/	/

3.3.6.4 0x0040 PLL_VIDEO0 Control Register (Default Value: 0x4800_6203)

Offset: 0x0040			Register Name: PLL_VIDEO0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_EN PLL Enable 0: Disable 1: Enable For application, PLL_VIDEO0(4X)=InputFreq *N/M. PLL_VIDEO0(2X)= InputFreq *N/M/2. PLL_VIDEO0(1X)= InputFreq *N/M/4. When the HOSC is 24 MHz, the default frequency of PLL_VIDEO0(4X) is 1188 MHz.
30	R/W	0x1	PLL_LDO_EN LDO Enable 0: Disable 1: Enable
29	R/W	0x0	LOCK_ENABLE Lock Enable 0: Disable 1: Enable
28	R	0x0	LOCK PLL Lock 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)
27	R/W	0x1	PLL_OUTPUT_GATE PLL Output Gating Enable 0: Disable 1: Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_EN PLL SDM Enable 0: Disable 1: Enable
23:16	/	/	/

Offset: 0x0040			Register Name: PLL_VIDEO0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
15:8	R/W	0x62	PLL_N PLL N N= PLL_N +1 PLL_N is from 0 to 254. In application, PLL_N shall be more than or equal to 12.
7:6	R/W	0x0	PLL_UNLOCK_MDSEL PLL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles
5	R/W	0x0	PLL_LOCK_MDSEL PLL Lock Level 0: 24-26 Clock Cycles 1: 23-27 Clock Cycles
4:2	/	/	/
1	R/W	0x1	PLL_INPUT_DIV2 PLL Input Div M M1=PLL_INPUT_DIV_M + 1 PLL_INPUT_DIV_M is from 0 to 1.
0	R/W	0x1	PLL_OUTPUT_DIV2 PLL Output Div D (The factor is only for testing) M0=PLL_OUTPUT_DIV_D + 1 PLL_OUTPUT_DIV_D is from 0 to 1. For test, PLL_VIDEO0(4X) =24MHz*N/M/D

3.3.6.5 0x0048 PLL_VIDEO1 Control Register (Default Value: 0x4800_6203)

Offset: 0x0048			Register Name: PLL_VIDEO1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_EN PLL Enable 0: Disable 1: Enable For application, PLL_VIDEO1(4X)=InputFreq *N/M. PLL_VIDEO1(2X)= InputFreq *N/M/2. PLL_VIDEO1(1X)= InputFreq *N/M/4. When the HOSC is 24 MHz, the default frequency of PLL_VIDEO1(4X) is 1188 MHz.
30	R/W	0x1	PLL_LDO_EN LDO Enable 0: Disable 1: Enable
29	R/W	0x0	LOCK_ENABLE Lock Enable 0: Disable 1: Enable
28	R	0x0	LOCK PLL Lock 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)
27	R/W	0x1	PLL_OUTPUT_GATE PLL Output Gating Enable 0: Disable 1: Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_EN PLL SDM Enable 0: Disable 1: Enable
23:16	/	/	/

Offset: 0x0048			Register Name: PLL_VIDEO1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
15:8	R/W	0x62	PLL_FACTOR_N PLL FACTOR N N= PLL_FACTOR_N +1 PLL_FACTOR_N is from 0 to 254. In application, PLL_FACTOR_N shall be more than or equal to 12.
7:6	R/W	0x0	PLL_UNLOCK_MDSEL PLL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles
5	R/W	0x0	PLL_LOCK_MDSEL PLL Lock Level 0: 24-26 Clock Cycles 1: 23-27 Clock Cycles
4:2	/	/	/
1	R/W	0x1	PLL_INPUT_DIV2 PLL Input Div M M1=PLL_INPUT_DIV_M + 1 PLL_INPUT_DIV_M is from 0 to 1.
0	R/W	0x1	PLL_OUTPUT_DIV2 PLL Output Div D.(The factor is only for testing) M0=PLL_OUTPUT_DIV_D + 1 PLL_OUTPUT_DIV_D is from 0 to 1. For test, PLL_VIDEO1(4X) =24MHz*N/M/D

3.3.6.6 0x0058 PLL_VE Control Register (Default Value: 0x4800_2301)

Offset: 0x0058			Register Name: PLL_VE_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_EN PLL Enable 0: Disable 1: Enable PLL_VE = InputFreq*N/M1/M0. When the HOSC is 24 MHz, the default frequency of PLL_VE is 432 MHz.
30	R/W	0x1	PLL_LDO_EN LDO Enable 0: Disable 1: Enable
29	R/W	0x0	LOCK_ENABLE Lock Enable 0: Disable 1: Enable
28	R	0x0	LOCK PLL Lock 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)
27	R/W	0x1	PLL_OUTPUT_GATE PLL Output Gating Enable 0: Disable 1: Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_EN PLL SDM Enable 0: Disable 1: Enable
23:16	/	/	/

Offset: 0x0058			Register Name: PLL_VE_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
15:8	R/W	0x23	PLL_N PLL N N= PLL_N +1 PLL_N is from 0 to 254. In application, PLL_N shall be more than or equal to 12.
7:6	R/W	0x0	PLL_UNLOCK_MDSEL PLL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles
5	R/W	0x0	PLL_LOCK_MDSEL PLL Lock Level 0: 24-26 Clock Cycles 1: 23-27 Clock Cycles
4:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV2 PLL Input Div M1 M1=PLL_INPUT_DIV2 + 1 PLL_INPUT_DIV2 is from 0 to 1.
0	R/W	0x1	PLL_OUTPUT_DIV2 PLL Output Div M0 M0=PLL_OUTPUT_DIV2 + 1 PLL_OUTPUT_DIV2 is from 0 to 1.

3.3.6.7 0x0078 PLL_AUDIO0 Control Register (Default Value: 0x4814_5500)

Offset: 0x0078			Register Name: PLL_AUDIO0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_EN PLL Enable 0: Disable 1: Enable PLL_AUDIO0(4X) = 24MHz*N/M1/M0/P PLL_AUDIO0(2X) = (24MHz*N/M1/M0)/P/2 PLL_AUDIO0(1X) = (24MHz*N/M1/M0)/P/4 **Note: 7.5≤N/M0/M1≤125 and 12≤N** The working frequency range of 24MHz*N/M0/M1 is from 180 MHz to 3.0 GHz. The default frequency of PLL_AUDIO0(1X) is 24.5714 MHz.
30	R/W	0x1	PLL_LDO_EN LDO Enable 0: Disable 1: Enable
29	R/W	0x0	LOCK_ENABLE Lock Enable 0: Disable 1: Enable
28	R	0x0	LOCK PLL Lock 0: Unlocked 1: Locked (It indicates that the PLL has been stable.) **Note: The bit is only valid when the bit29 is set to 1.**
27	R/W	0x1	PLL_OUTPUT_GATE 0: Disable 1: Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_EN PLL SDM Enable 0: Disable 1: Enable Enable spread spectrum and decimal division.
23:22	/	/	/

Offset: 0x0078			Register Name: PLL_AUDIO0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
21:16	R/W	0x14	PLL_P PLL Post-div P P= PLL_POST_DIV_P +1 PLL_POST_DIV_P is from 0 to 63.
15:8	R/W	0x55	PLL_N PLL N N= PLL_N +1 PLL_N is from 0 to 254. In application, PLL_N shall be more than or equal to 12.
7:6	R/W	0x0	PLL_UNLOCK_MDSEL PLL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles
5	R/W	0x0	PLL_LOCK_MDSEL PLL Lock Level 0: 24-26 Clock Cycles 1: 23-27 Clock Cycles
4:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV2 PLL Input Div M1 M1=PLL_INPUT_DIV2 + 1 PLL_INPUT_DIV2 is from 0 to 1.
0	R/W	0x0	PLL_OUTPUT_DIV2 PLL Output Div M0 M0=PLL_OUTPUT_DIV2 + 1 PLL_OUTPUT_DIV2 is from 0 to 1.

3.3.6.8 0x0080 PLL_AUDIO1 Control Register (Default Value: 0x4841_7F00)

Offset: 0x0080			Register Name: PLL_AUDIO1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_EN PLL Enable 0: Disable 1: Enable PLL_AUDIO1 = 24MHz*N/M PLL_AUDIO1(DIV2) = 24MHz*N/M/P0 PLL_AUDIO1(DIV5) = 24MHz*N/M/P1 The working frequency range of 24 MHz/M*N is from 180 MHz to 3.5 GHz. The default frequency of PLL_AUDIO1 is 3072 MHz. The default frequency of PLL_AUDIO1(DIV2) is 1536 MHz. The default frequency of PLL_AUDIO1(DIV5) is 614.4 MHz (24.576 MHz*25).
30	R/W	0x1	PLL_LDO_EN LDO Enable 0: Disable 1: Enable
29	R/W	0x0	LOCK_ENABLE Lock Enable 0: Disable 1: Enable
28	R	0x0	LOCK 0: Unlocked 1: Locked (It indicates that the PLL has been stable.) **Note: The bit is only valid when the bit29 is set to 1.**
27	R/W	0x1	PLL_OUTPUT_GATE 0: Disable 1: Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_EN 0: Disable 1: Enable Enable spread spectrum and decimal division.
23	/	/	/

Offset: 0x0080			Register Name: PLL_AUDIO1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
22:20	R/W	0x4	PLL_P1 PLL Output Div P1 P1=PLL_OUTPUT_DIV_P1 + 1 PLL_OUTPUT_DIV_P1 is from 0 to 7.
19	/	/	/
18:16	R/W	0x1	PLL_P0 PLL Output Div P0 P0=PLL_OUTPUT_DIV_P0 + 1 PLL_OUTPUT_DIV_P0 is from 0 to 7.
15:8	R/W	0x7F	PLL_N PLL N N= PLL_N +1 PLL_N is from 0 to 254. In application, PLL_N shall be more than or equal to 12.
7:6	R/W	0x0	PLL_UNLOCK_MDSEL PLL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles
5	R/W	0x0	PLL_LOCK_MDSEL PLL Lock Level 0: 24-26 Clock Cycles 1: 23-27 Clock Cycles
4:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV2 PLL Input Div M M=PLL_INPUT_DIV_M + 1 PLL_INPUT_DIV_M is from 0 to 1.
0	/	/	/

3.3.6.9 0x0110 PLL_DDR Pattern0 Control Register (Default Value: 0x0000_0000)

Offset: 0x0110			Register Name: PLL_DDR_PAT0_CTRL_REG
Bit	**Read/Write**	**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) 11: Triangular(n-bit)
28:20	R/W	0x0	WAVE_STEP Wave Step
19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz **Note: When the PLL_INPUT_DIV2 is 1, the bit needs to be set to 1.**
18:17	R/W	0x0	FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT Wave Bottom

3.3.6.10 0x0114 PLL_DDR Pattern1 Control Register (Default Value: 0x0000_0000)

Offset: 0x0114			Register Name: PLL_DDR_PAT1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	DITHER_EN Dither Enable
23:21	/	/	/

Offset: 0x0114			Register Name: PLL_DDR_PAT1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
20	R/W	0x0	FRAC_EN Fraction Enable
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN Fraction In

3.3.6.11 0x0120 PLL_PERI Pattern0 Control Register (Default Value: 0x0000_0000)

Offset: 0x0120			Register Name: PLL_PERI_PAT0_CTRL_REG
Bit	**Read/Write**	**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) 11: Triangular(n-bit)
28:20	R/W	0x0	WAVE_STEP Wave Step
19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz **Note: When the PLL_INPUT_DIV2 is 1, the bit needs to be set to 1.**
18:17	R/W	0x0	FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT Wave Bottom

3.3.6.12 0x0124 PLL_PERI Pattern1 Control Register (Default Value: 0x0000_0000)

Offset: 0x0124			Register Name: PLL_PERI_PAT1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	DITHER_EN Dither Enable
23:21	/	/	/
20	R/W	0x0	FRAC_EN Fraction Enable
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN Fraction In

3.3.6.13 0x0140 PLL_VIDEO0 Pattern0 Control Register (Default Value: 0x0000_0000)

Offset: 0x0140			Register Name: PLL_VIDEO0_PAT0_CTRL_REG
Bit	**Read/Write**	**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) 11: Triangular(n-bit)
28:20	R/W	0x0	WAVE_STEP Wave Step
19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz **Note: When the PLL_INPUT_DIV2 is 1, the bit needs to be set to 1.**

Offset: 0x0140			Register Name: PLL_VIDEO0_PAT0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
18:17	R/W	0x0	FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT Wave Bottom

3.3.6.14 0x0144 PLL_VIDEO0 Pattern1 Control Register (Default Value: 0x0000_0000)

Offset: 0x0144			Register Name: PLL_VIDEO0_PAT1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	DITHER_EN Dither Enable
23:21	/	/	/
20	R/W	0x0	FRAC_EN Fraction Enable
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN Fraction In

3.3.6.15 0x0148 PLL_VIDEO1 Pattern0 Control Register (Default Value: 0x0000_0000)

Offset: 0x0148			Register Name: PLL_VIDEO1_PAT0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	SIG_DELT_PAT_EN Sigma-Delta Pattern Enable

Offset: 0x0148			Register Name: PLL_VIDEO1_PAT0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
30:29	R/W	0x0	SPR_FREQ_MODE Spread Frequency Mode 00: DC=0 01: DC=1 10: Triangular(1-bit) 11: Triangular(n-bit)
28:20	R/W	0x0	WAVE_STEP Wave Step
19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz **Note: When the PLL_INPUT_DIV2 is 1, the bit needs to be set to 1.**
18:17	R/W	0x0	FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT Wave Bottom

3.3.6.16 0x014C PLL_VIDEO1 Pattern1 Control Register (Default Value: 0x0000_0000)

Offset: 0x014C			Register Name: PLL_VIDEO1_PAT1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	DITHER_EN Dither Enable
23:21	/	/	/
20	R/W	0x0	FRAC_EN Fraction Enable
19:17	/	/	/

Offset: 0x014C			Register Name: PLL_VIDEO1_PAT1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
16:0	R/W	0x0	FRAC_IN Fraction In

3.3.6.17 0x0158 PLL_VE Pattern0 Control Register (Default Value: 0x0000_0000)

Offset: 0x0158			Register Name: PLL_VE_PAT0_CTRL_REG
Bit	**Read/Write**	**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) 11: Triangular(n-bit)
28:20	R/W	0x0	WAVE_STEP Wave Step
19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz **Note: When the PLL_INPUT_DIV2 is 1, the bit needs to be set to 1.**
18:17	R/W	0x0	FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT Wave Bottom

3.3.6.18 0x015C PLL_VE Pattern1 Control Register (Default Value: 0x0000_0000)

Offset: 0x015C			Register Name: PLL_VE_PAT1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	DITHER_EN Dither Enable
23:21	/	/	/
20	R/W	0x0	FRAC_EN Fraction Enable
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN Fraction In

3.3.6.19 0x0178 PLL_AUDIO0 Pattern0 Control Register (Default Value: 0x0000_0000)

Offset: 0x0178			Register Name: PLL_AUDIO0_PAT0_CTRL_REG
Bit	**Read/Write**	**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) 11: Triangular(n-bit)
28:20	R/W	0x0	WAVE_STEP Wave Step
19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz **Note: When the PLL_INPUT_DIV2 is 1, the bit needs to be set to 1.**

Offset: 0x0178			Register Name: PLL_AUDIO0_PAT0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
18:17	R/W	0x0	FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT Wave Bottom

3.3.6.20 0x017C PLL_AUDIO0 Pattern1 Control Register (Default Value: 0x0000_0000)

Offset: 0x017C			Register Name: PLL_AUDIO0_PAT1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	DITHER_EN Dither Enable
23:21	/	/	/
20	R/W	0x0	FRAC_EN Fraction Enable
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN Fraction In

3.3.6.21 0x0180 PLL_AUDIO1 Pattern0 Control Register (Default Value: 0x0000_0000)

Offset: 0x0180			Register Name: PLL_AUDIO1_PAT0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	SIG_DELT_PAT_EN Sigma-Delta Pattern Enable

Offset: 0x0180			Register Name: PLL_AUDIO1_PAT0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
30:29	R/W	0x0	SPR_FREQ_MODE Spread Frequency Mode 00: DC=0 01: DC=1 10: Triangular(1-bit) 11: Triangular(n-bit)
28:20	R/W	0x0	WAVE_STEP Wave Step
19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz **Note: When the PLL_INPUT_DIV2 is 1, the bit needs to be set to 1.**
18:17	R/W	0x0	FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT Wave Bottom

3.3.6.22 0x0184 PLL_AUDIO1 Pattern1 Control Register (Default Value: 0x0000_0000)

Offset: 0x0184			Register Name: PLL_AUDIO1_PAT1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	DITHER_EN Dither Enable
23:21	/	/	/
20	R/W	0x0	FRAC_EN Fraction Enable
19:17	/	/	/

Offset: 0x0184			Register Name: PLL_AUDIO1_PAT1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
16:0	R/W	0x0	FRAC_IN Fraction In

3.3.6.23 0x0300 PLL_CPU Bias Register (Default Value: 0x8010_0000)

Offset: 0x0300			Register Name: PLL_CPU_BIAS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x1	PLL_VCO_RST_IN VCO reset in
30:21	/	/	/
20:16	R/W	0x10	PLL_CP PLL current bias control
15:0	/	/	/

3.3.6.24 0x0310 PLL_DDR Bias Register (Default Value: 0x0003_0000)

Offset: 0x0310			Register Name: PLL_DDR_BIAS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20:16	R/W	0x3	PLL_CP PLL bias control
15:0	/	/	/

3.3.6.25 0x0320 PLL_PERI Bias Register (Default Value: 0x0003_0000)

Offset: 0x0320			Register Name: PLL_PERI_BIAS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20:16	R/W	0x3	PLL_CP PLL bias control
15:0	/	/	/

3.3.6.26 0x0340 PLL_VIDEO0 Bias Register (Default Value: 0x0003_0000)

Offset: 0x0340			Register Name: PLL_VIDEO0_BIAS_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20:16	R/W	0x3	PLL_CP PLL bias control
15:0	/	/	/

3.3.6.27 0x0348 PLL_VIDEO1 Bias Register (Default Value: 0x0003_0000)

Offset: 0x0348			Register Name: PLL_VIDEO1_BIAS_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20:16	R/W	0x3	PLL_CP PLL bias control
15:0	/	/	/

3.3.6.28 0x0358 PLL_VE Bias Register (Default Value: 0x0003_0000)

Offset: 0x0358			Register Name: PLL_VE_BIAS_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20:16	R/W	0x3	PLL_CP PLL bias control
15:0	/	/	/

3.3.6.29 0x0378 PLL_AUDIO0 Bias Register (Default Value: 0x0003_0000)

Offset: 0x0378			Register Name: PLL_AUDIO0_BIAS_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20:16	R/W	0x3	PLL_CP PLL bias control

Offset: 0x0378			Register Name: PLL_AUDIO0_BIAS_REG
Bit	Read/Write	Default/Hex	Description
15:0	/	/	/

3.3.6.30 0x0380 PLL_AUDIO1 Bias Register (Default Value: 0x0003_0000)

Offset: 0x0380			Register Name: PLL_AUDIO1_BIAS_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20:16	R/W	0x3	PLL_CP PLL bias control
15:0	/	/	/

3.3.6.31 0x0400 PLL_CPU Tuning Register (Default Value: 0x4440_4000)

Offset: 0x0400			Register Name: PLL_CPU_TUN_REG
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30:28	R/W	0x4	PLL_VCO VCO range control
27	/	/	/
26:24	R/W	0x4	PLL_VCO_GAIN KVCO gain control
23	/	/	/
22:16	R/W	0x40	PLL_CNT_INT Counter initial control
15	R/W	0x0	PLL_REG_OD PLL-REG-OD0 for verify
14:8	R/W	0x40	PLL_B_IN PLL-B-IN [6:0] for verify
7	R/W	0x0	PLL_REG_OD1 PLL-REG-OD1 for verify
6:0	R	0x0	PLL_B_OUT PLL-B-OUT [6:0] for verify

3.3.6.32 0x0500 CPU_AXI Configuration Register (Default Value: 0x0000_0301)

Offset: 0x0500			Register Name: CPU_AXI_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:27	/	/	/
26:24	R/W	0x0	CPU_CLK_SEL Clock Source Select 000: HOSC 001: CLK32K 010: CLK16M_RC 011: PLL_CPU/P 100: PLL_PERI(1X) 101: PLL_PERI(2X) 110: PLL_PERI(800M) CPU Clock = Clock Source CPU_AXI Clock = Clock Source/M **Note: The clock select is lack of glitch switching and supports dynamic configuration.**
23:18	/	/	/
17:16	R/W	0x0	PLL_CPU_OUT_EXT_DIVP PLL Output External Divider P 00: 1 01: 2 10: 4 11: / When the output clock is less than 288 MHz, the clock frequency can be get by dividing P.
15:10	/	/	/
9:8	R/W	0x3	CPU_DIV2. Factor N (N = FACTOR_N +1) FACTOR_N is from 1 to 3. **Note: The clock division is lack of glitch switching and supports dynamic configuration.**
7:2	/	/	/

Offset: 0x0500			Register Name: CPU_AXI_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
1:0	R/W	0x1	CPU_DIV1. Factor M (M= FACTOR_M +1) FACTOR_M is from 1 to 3. **Note: The clock division is lack of glitch switching and supports dynamic configuration.**

3.3.6.33 0x0504 CPU_GATING Configuration Register (Default Value: 0x8000_0000)

Offset: 0x0504			Register Name: CPU_GATING_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x1	CPU_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON
30:16	/	/	/
15:0	W	0x0	CPU_GATING_FIELD CPU Gating Field If CPU_GATING_FIELD==16'h16AA, the bit31 can be configured.

3.3.6.34 0x0510 PSI Clock Register (Default Value: 0x0000_0000)

Offset: 0x0510			Register Name: PSI_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: CLK32K 010: CLK16M_RC 011: PLL_PERI(1X) PSI_CLK = Clock Source/M/N. **Note: The clock select is lack of glitch switching and supports dynamic configuration.**
23:10	/	/	/

Offset: 0x0510			Register Name: PSI_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8 **Note: The clock division is lack of glitch switching and supports dynamic configuration.**
7:2	/	/	/
1:0	R/W	0x0	FACTOR_M Factor M. M= FACTOR_M +1 FACTOR_M is from 0 to 3. **Note: The clock division is lack of glitch switching and supports dynamic configuration.**

3.3.6.35 0x0520 APB0 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0520			Register Name: APB0_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: HOSC 01: CLK32K 10: PSI_CLK 11: PLL_PERI(1X) APB0_CLK = Clock Source/M/N. **Note: The clock select is lack of glitch switching and supports dynamic configuration.**
23:10	/	/	/

Offset: 0x0520			Register Name: APB0_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
9:8	R/W	0x0	FACTOR_N. Factor N 00: 1 01: 2 10: 4 11: 8 **Note: The clock division is lack of glitch switching and supports dynamic configuration.**
7:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M = FACTOR_M+1 FACTOR_M is from 0 to 31. **Note: The clock division is lack of glitch switching and supports dynamic configuration.**

3.3.6.36 0x0524 APB1 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0524			Register Name: APB1_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: HOSC 01: CLK32K 10: PSI_CLK 11: PLL_PERI(1X) APB1_CLK = Clock Source/M/N. **Note: The clock select is lack of glitch switching and supports dynamic configuration.**
23:10	/	/	/

Offset: 0x0524			Register Name: APB1_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8 **Note: The clock division is lack of glitch switching and supports dynamic configuration.**
7:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M = FACTOR_M+1 FACTOR_M is from 0 to 31. **Note: The clock division is lack of glitch switching and supports dynamic configuration.**

3.3.6.37 0x0540 MBUS Clock Register (Default Value: 0xC000_0000)

Offset: 0x0540			Register Name: MBUS_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R/W	0x1	MBUS_RST MBUS Reset 0: Assert 1: De-assert
29:0	/	/	/

NOTE

The MBUS clock is from the 4 frequency-division of PLL_DDR, and it has the same source with the DRAM clock.

3.3.6.38 0x0600 DE Clock Register (Default Value: 0xC000_0000)

Offset: 0x0600			Register Name: DE_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	DE_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON DE_CLK = Clock Source/M.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: PLL_PERI(2X) 001: PLL_VIDEO0(4X) 010: PLL_VIDEO1(4X) 011: PLL_AUDIO1(DIV2)
23:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.39 0x060C DE Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x060C			Register Name: DE_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	DE_RST DE Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	DE_GATING DE Gating Clock 0: Mask 1: Pass

3.3.6.40 0x0620 DI Clock Register (Default Value: 0x0000_0000)

Offset: 0x0620			Register Name: DI_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	DI_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON DI_CLK = Clock Source/M.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: PLL_PERI(2X) 001: PLL_VIDEO0(4X) 010: PLL_VIDEO1(4X) 011: PLL_AUDIO1(DIV2)
23:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M = FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.41 0x062C DI Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x062C			Register Name: DI_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	DI_RST DI Reset 0: Assert 1: DE-assert
15:1	/	/	/
0	R/W	0x0	DI_GATING DI Gating Clock 0: Mask 1: Pass

3.3.6.42 0x0630 G2D Clock Register (Default Value: 0x0000_0000)

Offset: 0x0630			Register Name: G2D_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	G2D_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON G2D_CLK = Clock Source/M.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: PLL_PERI(2X) 001: PLL_VIDEO0(4X) 010: PLL_VIDEO1(4X) 011: PLL_AUDIO1(DIV2)
23:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.43 0x063C G2D Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x063C			Register Name: G2D_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	G2D_RST G2D Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	G2D_GATING G2D Gating Clock 0: Mask 1: Pass

3.3.6.44 0x0680 CE Clock Register (Default Value: 0x0000_0000)

Offset: 0x0680			Register Name: CE_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CE_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON CE_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: HOSC 01: PLL_PERI(2X) 10: PLL_PERI(1X)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

3.3.6.45 0x068C CE Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x068C			Register Name: CE_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	CE_RST CE Reset 0: Assert 1: De-assert

Offset: 0x068C			Register Name: CE_BGR_REG
Bit	Read/Write	Default/Hex	Description
15:1	/	/	/
0	R/W	0x0	CE_GATING CE Gating Clock 0: Mask 1: Pass

3.3.6.46 0x0690 VE Clock Register (Default Value: 0x0000_0000)

Offset: 0x0690			Register Name: VE_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	VE_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON VE_CLK = Clock Source/M.
30:25	/	/	/
24	R/W	0x0	CLK_SRC_SEL Clock Source Select 0: PLL_VE 1: PLL_PERI(2X)
23:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.47 0x069C VE Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x069C			Register Name: VE_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/

Offset: 0x069C			Register Name: VE_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
16	R/W	0x0	VE_RST VE Reset For VE_PROT 0: Assert 1: DE-assert
15:1	/	/	/
0	R/W	0x0	VE_GATING Gating Clock For VE_PROT 0: Mask 1: Pass

3.3.6.48 0x070C DMA Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x070C			Register Name: DMA_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	DMA_RST DMA Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	DMA_GATING DMA Gating Clock 0: Mask 1: Pass **Note: The working clock is from PSI_CLK.**

3.3.6.49 0x071C MSGBOX Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x071C			Register Name: MSGBOX_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/

Offset: 0x071C			Register Name: MSGBOX_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
17	R/W	0x0	MSGBOX1_RST DSP MSGBOX Reset 0: Assert 1: De-assert
16	R/W	0x0	MSGBOX0_RST CPU MSGBOX Reset 0: Assert 1: De-assert
15:2	/	/	/
1	R/W	0x0	MSGBOX1_GATING Gating Clock for DSP MSGBOX 0: Mask 1: Pass
0	R/W	0x0	MSGBOX0_GATING Gating Clock for CPU MSGBOX 0: Mask 1: Pass

3.3.6.50 0x072C SPINLOCK Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x072C			Register Name: SPINLOCK_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	SPINLOCK_RST SPINLOCK Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	SPINLOCK_GATING Gating Clock For SPINLOCK 0: Mask 1: Pass

3.3.6.51 0x073C HSTIMER Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x073C			Register Name: HSTIMER_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	HSTIMER_RST HSTIMER Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	HSTIMER_GATING Gating Clock for HSTIMER 0: Mask 1: Pass

3.3.6.52 0x0740 AVS Clock Register (Default Value: 0x0000_0000)

Offset: 0x0740			Register Name: AVS_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	AVS_CLK_GATING Gating Clock The AVS_CLK is from HOSC. 0: Clock is OFF 1: Clock is ON
30:0	/	/	/

3.3.6.53 0x078C DBGSYS Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x078C			Register Name: DBGSYS_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	DBGSYS_RST DBGSYS Reset 0: Assert 1: De-assert
15:1	/	/	/

Offset: 0x078C			Register Name: DBGSYS_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	DBGSYS_GATING Gating Clock for DBGSYS The clock of DBGSYS is from HOSC. 0: Mask 1: Pass

3.3.6.54 0x07AC PWM Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x07AC			Register Name: PWM_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	PWM_RST PWM Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	PWM_GATING Gating Clock for PWM 0: Mask 1: Pass

3.3.6.55 0x07BC IOMMU Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x07BC			Register Name: IOMMU_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	IOMMU_GATING Gating Clock for IOMMU 0: Mask 1: Pass

3.3.6.56 0x0800 DRAM Clock Register (Default Value: 0x8000_0000)

Offset: 0x0800			Register Name: DRAM_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x1	DRAM_CLK_GATING Gating Clock DRAM_CLK = Clock Source/M/N. 0: Clock is OFF 1: Clock is ON
30:28	/	/	/
27	R/WAC	0x0	SDRCLK_UPD SDRCLK Configuration 0 Update 0: Invalid 1: Valid Setting 1 will validate the bit. It will be automatically cleared after the bit is valid. Here supports dram req/ack signal. When dram_update is set to 1, dram_clk_sel/dram_div2/dram_clk1 will be updated.
26:24	R/W	0x0	DRAM_CLK_SEL Clock Source Select 00: PLL_DDR 01: PLL_AUDIO1(DIV2) 10: PLL_PERI(2X) 11: PLL_PERI(800M) **Note: The clock select is lack of glitch switching and supports dynamic configuration.**
23:10	/	/	/
9:8	R/W	0x0	DRAM_DIV2 Factor N 00: 1 01: 2 10: 4 11: 8
7:2	/	/	/

Offset: 0x0800			Register Name: DRAM_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
1:0	R/W	0x0	DRAM_DIV1 Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 3. **Note: The clock division is lack of glitch switching and supports dynamic configuration.**

3.3.6.57 0x0804 MBUS Master Clock Gating Register (Default Value: 0x0000_0000)

Offset: 0x0804			Register Name: MBUS_MAT_CLK_GATING_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10	R/W	0x0	G2D_MCLK_EN Gating MBUS Clock For G2D 0: Mask 1: Pass
9	/	/	/
8	R/W	0x0	CSI_MCLK_EN Gating MBUS Clock For CSI 0: Mask 1: Pass
7	R/W	0x0	TVIN_MCLK_EN Gating MBUS Clock For TVIN 0: Mask 1: Pass
6:3	/	/	/
2	R/W	0x0	CE_MCLK_EN Gating MBUS Clock For CE 0: Mask 1: Pass
1	R/W	0x0	VE_MCLK_EN Gating MBUS Clock For VE 0: Mask 1: Pass

Offset: 0x0804			Register Name: MBUS_MAT_CLK_GATING_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	DMA_MCLK_EN Gating MBUS Clock For DMA 0: Mask 1: Pass

3.3.6.58 0x080C DRAM Bus Gating Reset Register (Default Value: 0x0000_0001)

Offset: 0x080C			Register Name: DRAM_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	DRAM_RST DRAM Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x1	DRAM_GATING Gating Clock for DRAM 0: Mask 1: Pass

3.3.6.59 0x0830 SMHC0 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0830			Register Name: SMHC0_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	SMHC0_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON SMHC0_CLK = Clock Source/M/N.
30:27	/	/	/

Offset: 0x0830			Register Name: SMHC0_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: PLL_PERI(1X) 010: PLL_PERI(2X) 011: PLL_AUDIO1(DIV2) Others: Reserved
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M = FACTOR_M +1 FACTOR_M is from 0 to 15.

3.3.6.60 0x0834 SMHC1 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0834			Register Name: SMHC1_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	SMHC1_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:27	/	/	/

Offset: 0x0834			Register Name: SMHC1_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: PLL_PERI(1X) 010: PLL_PERI(2X) 011: PLL_AUDIO1(DIV2) Others: Reserved
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

3.3.6.61 0x0838 SMHC2 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0838			Register Name: SMHC2_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	SMHC2_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:27	/	/	/

Offset: 0x0838			Register Name: SMHC2_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: PLL_PERI(1X) 010: PLL_PERI(2X) 011: PLL_PERI(800M) 100: PLL_AUDIO1(DIV2) Others: Reserved
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

3.3.6.62 0x084C SMHC Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x084C			Register Name: SMHC_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:19	/	/	/
18	R/W	0x0	SMHC2_RST SMHC2 Reset 0: Assert 1: De-assert
17	R/W	0x0	SMHC1_RST SMHC1 Reset 0: Assert 1: De-assert

Offset: 0x084C			Register Name: SMHC_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
16	R/W	0x0	SMHC0_RST SMHC0 Reset 0: Assert 1: De-assert
15:3	/	/	/
2	R/W	0x0	SMHC2_GATING Gating Clock for SMHC2 0: Mask 1: Pass
1	R/W	0x0	SMHC1_GATING Gating Clock for SMHC1 0: Mask 1: Pass
0	R/W	0x0	SMHC0_GATING Gating Clock for SMHC0 0: Mask 1: Pass

3.3.6.63 0x090C UART Bus Gating Reset Register (Default Value: 0x0000_0000)

NOTE

The clock of the UART is from APB1.

Offset: 0x090C			Register Name: UART_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21	R/W	0x0	UART5_RST UART5 Reset 0: Assert 1: De-assert

Offset: 0x090C			Register Name: UART_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
20	R/W	0x0	UART4_RST UART4 Reset 0: Assert 1: De-assert
19	R/W	0x0	UART3_RST UART3 Reset 0: Assert 1: De-assert
18	R/W	0x0	UART2_RST UART2 Reset 0: Assert 1: De-assert
17	R/W	0x0	UART1_RST UART1 Reset 0: Assert 1: De-assert
16	R/W	0x0	UART0_RST UART0 Reset 0: Assert 1: De-assert
15:6	/	/	/
5	R/W	0x0	UART5_GATING Gating Clock for UART5 0: Mask 1: Pass
4	R/W	0x0	UART4_GATING Gating Clock for UART4 0: Mask 1: Pass
3	R/W	0x0	UART3_GATING Gating Clock for UART3 0: Mask 1: Pass

Offset: 0x090C			Register Name: UART_BGR_REG
Bit	Read/Write	Default/Hex	Description
2	R/W	0x0	UART2_GATING Gating Clock for UART2 0: Mask 1: Pass
1	R/W	0x0	UART1_GATING Gating Clock for UART1 0: Mask 1: Pass
0	R/W	0x0	UART0_GATING Gating Clock for UART0 0: Mask 1: Pass

3.3.6.64 0x091C TWI Bus Gating Reset Register (Default Value: 0x0000_0000)

NOTE

The clock of the TWI is from APB1.

Offset: 0x091C			Register Name: TWI_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:20	/	/	/
19	R/W	0x0	TWI3_RST TWI3 Reset 0: Assert 1: De-assert
18	R/W	0x0	TWI2_RST TWI2 Reset 0: Assert 1: De-assert
17	R/W	0x0	TWI1_RST TWI1 Reset 0: Assert 1: De-assert

Offset: 0x091C			Register Name: TWI_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
16	R/W	0x0	TWI0_RST TWI0 Reset 0: Assert 1: De-assert
15:4	/	/	/
3	R/W	0x0	TWI3_GATING Gating Clock for TWI3 0: Mask 1: Pass
2	R/W	0x0	TWI2_GATING Gating Clock for TWI2 0: Mask 1: Pass
1	R/W	0x0	TWI1_GATING Gating Clock for TWI1 0: Mask 1: Pass
0	R/W	0x0	TWI0_GATING Gating Clock for TWI0 0: Mask 1: Pass

3.3.6.65 0x0940 SPI0 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0940			Register Name: SPI0_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	SPI0_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:27	/	/	/

Offset: 0x0940			Register Name: SPI0_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: PLL_PERI(1X) 010: PLL_PERI(2X) 011: PLL_AUDIO1(DIV2) 100: PLL_AUDIO1(DIV5) Others: Reserved
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

3.3.6.66 0x0944 SPI1 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0944			Register Name: SPI1_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:27	/	/	/

Offset: 0x0944			Register Name: SPI1_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: PLL_PERI(1X) 010: PLL_PERI(2X) 011: PLL_AUDIO1(DIV2) 100: PLL_AUDIO1(DIV5) Others: /
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

3.3.6.67 0x096C SPI Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x096C			Register Name: SPI_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R/W	0x0	SPI1_RST SPI1 Reset 0: Assert 1: De-assert
16	R/W	0x0	SPI0_RST SPI0 Reset 0: Assert 1: De-assert
15:2	/	/	/

Offset: 0x096C			Register Name: SPI_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x0	SPI1_GATING Gating Clock for SPI1 0: Mask 1: Pass
0	R/W	0x0	SPI0_GATING Gating Clock for SPI0 0: Mask 1: Pass

3.3.6.68 0x0970 EMAC_25M Clock Register (Default: 0x0000_0000)

Offset: 0x0970			Register Name: EMAC_25M_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	EMAC_25M_CLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON CLK = PLL_PERI(1X)/24 = 25 MHz
30	R/W	0x0	EMAC_25M_CLK_SRC_GATING Gating the Source Clock of Special Clock It is for low power design. 0: Clock is OFF 1: Clock is ON
29:0	/	/	/

3.3.6.69 0x097C EMAC Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x097C			Register Name: EMAC_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	EMAC_RST EMAC Reset 0: Assert 1: De-assert

Offset: 0x097C			Register Name: EMAC_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
15:1	/	/	/
0	R/W	0x0	EMAC_GATING Gating Clock for EMAC 0: Mask 1: Pass

3.3.6.70 0x09C0 IRTX Clock Register (Default: 0x0000_0000)

Offset: 0x09C0			Register Name: IRTX_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	IRTX_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON IRTX_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 0: HOSC 1: PLL_PERI(1X)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

3.3.6.71 0x09CC IRTX Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x09CC			Register Name: IRTX_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	IRTX_RST IRTX Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	IRTX_GATING Gating Clock for IRTX 0: Mask 1: Pass

3.3.6.72 0x09EC GPADC Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x09EC			Register Name: GPADC_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	GPADC_RST GPADC Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	GPADC_GATING Gating Clock For GPADC 0: Mask 1: Pass

3.3.6.73 0x09FC THS Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x09FC			Register Name: THS_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/

Offset: 0x09FC			Register Name: THS_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
16	R/W	0x0	THS_RST THS Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	THS_GATING Gating Clock For THS 0: Mask 1: Pass

3.3.6.74 0x0A14 I2S/PCM1 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A14			Register Name: I2S/PCM1_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	I2S/PCM1_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON I2S/PCM1_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO0(1X) 01: PLL_AUDIO0(4X) 10: PLL_AUDIO1(DIV2) 11: PLL_AUDIO1(DIV5)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: /1 01: /2 10: /4 11: /8
7:5	/	/	/

Offset: 0x0A14			Register Name: I2S/PCM1_CLK_REG
Bit	Read/Write	Default/Hex	Description
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.75 0x0A18 I2S/PCM2 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A18			Register Name: I2S/PCM2_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	I2S/PCM2_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON I2S/PCM2_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO0(1X) 01: PLL_AUDIO0(4X) 10: PLL_AUDIO1(DIV2) 11: PLL_AUDIO1(DIV5)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: /1 01: /2 10: /4 11: /8
7:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.76 0x0A1C I2S/PCM2_ASRC Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A1C			Register Name: I2S/PCM2_ASRC_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	I2S/PCM2_ASRC_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON I2S/PCM2_ASRC_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO0(4X) 01: PLL_PERI(1X) 10: PLL_AUDIO1(DIV2) 11: PLL_AUDIO1(DIV5)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: /1 01: /2 10: /4 11: /8
7:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M = FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.77 0x0A20 I2S/PCM Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0A20			Register Name: I2S/PCM_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:19	/	/	/

Offset: 0x0A20			Register Name: I2S/PCM_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
18	R/W	0x0	I2S/PCM2_RST I2S/PCM2 Reset 0: Assert 1: De-assert
17	R/W	0x0	I2S/PCM1_RST I2S/PCM1 Reset 0: Assert 1: De-assert
16:3	/	/	/
2	R/W	0x0	I2S/PCM2_GATING Gating Clock for I2S/PCM2 0: Mask 1: Pass
1	R/W	0x0	I2S/PCM1_GATING Gating Clock for I2S/PCM1 0: Mask 1: Pass
0	/	/	/

3.3.6.78 0x0A24 OWA_TX Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A24			Register Name: OWA_TX_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	OWA_TX_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON OWA_TX_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO0(1X) 01: PLL_AUDIO0(4X) 10: PLL_AUDIO1(DIV2) 11: PLL_AUDIO1(DIV5)

Offset: 0x0A24			Register Name: OWA_TX_CLK_REG
Bit	Read/Write	Default/Hex	Description
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: /1 01: /2 10: /4 11: /8
7:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.79 0x0A28 OWA_RX Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A28			Register Name: OWA_RX_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	OWA_RX_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON OWA_RX_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: PLL_PERI(1X) 001: PLL_AUDIO1(DIV2) 010: PLL_AUDIO1(DIV5)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: /1 01: /2 10: /4 11: /8

Offset: 0x0A28			Register Name: OWA_RX_CLK_REG
Bit	Read/Write	Default/Hex	Description
7:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.80 0x0A2C OWA Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0A2C			Register Name: OWA_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	OWA_RST OWA Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	OWA_GATING Gating Clock for OWA 0: Mask 1: Pass

3.3.6.81 0x0A40 DMIC Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A40			Register Name: DMIC_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	DMIC_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON DMIC_CLK = Clock Source/M/N.
30:27	/	/	/

Offset: 0x0A40			Register Name: DMIC_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO0(1X) 01: PLL_AUDIO1(DIV2) 10: PLL_AUDIO1(DIV5)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: /1 01: /2 10: /4 11: /8
7:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.82 0x0A4C DMIC Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0A4C			Register Name: DMIC_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	DMIC_RST DMIC Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	DMIC_GATING Gating Clock for DMIC 0: Mask 1: Pass

3.3.6.83 0x0A50 AUDIO_CODEC_DAC Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A50			Register Name: AUDIO_CODEC_DAC_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	AUDIO_CODEC_DAC_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON AUDIO_CODEC_DAC_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SE Clock Source Select 00: PLL_AUDIO0(1X) 01: PLL_AUDIO1(DIV2) 10: PLL_AUDIO1(DIV5)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: /1 01: /2 10: /4 11: /8
7:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.84 0x0A54 AUDIO_CODEC_ADC Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A54			Register Name: AUDIO_CODEC_ADC_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	AUDIO_CODEC_ADC_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON AUDIO_CODEC_ADC_CLK = Clock Source/M/N.

Offset: 0x0A54			Register Name: AUDIO_CODEC_ADC_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SE Clock Source Select 00: PLL_AUDIO0(1X) 01: PLL_AUDIO1(DIV2) 10: PLL_AUDIO1(DIV5)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: /1 01: /2 10: /4 11: /8
7:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.85 0x0A5C AUDIO_CODEC Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0A5C			Register Name: AUDIO_CODEC_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	AUDIO_CODEC_RST AUDIO_CODEC Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	AUDIO_CODEC_GATING Gating Clock For AUDIO_CODEC 0: Mask 1: Pass

3.3.6.86 0x0A70 USB0 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A70			Register Name: USB0_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	USB0_CLKEN Gating Special Clock For OHCI0 0: Clock is OFF 1: Clock is ON
30	R/W	0x0	USBPHY0_RSTN USB PHY0 Reset 0: Assert 1: De-assert
29:26	/	/	/
25:24	R/W	0x0	USB0_CLK12M_SEL OHCI0 12M Source Select 00: 12M divided from 48 MHz 01: 12M divided from 24 MHz 10: RTC_32K 11: /
23:0	/	/	/

3.3.6.87 0x0A74 USB1 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A74			Register Name: USB1_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	USB1_CLKEN Gating Special Clock For OHCI1 0: Clock is OFF 1: Clock is ON
30	R/W	0x0	USBPHY1_RSTN USB PHY1 Reset 0: Assert 1: De-assert
29:26	/	/	/

Offset: 0x0A74			Register Name: USB1_CLK_REG
Bit	Read/Write	Default/Hex	Description
25:24	R/W	0x0	USB1_CLK12M_SEL OHCI0 12M Source Select 00: 12M divided from 48 MHz 01: 12M divided from 24 MHz 10: RTC_32K 11: /
23:0	/	/	/

3.3.6.88 0x0A8C USB Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0A8C			Register Name: USB_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	USBOTG0_RST USBOTG0 Reset 0: Assert 1: De-assert
23:22	/	/	/
21	R/W	0x0	USBEHCI1_RST USBEHCI1 Reset 0: Assert 1: De-assert
20	R/W	0x0	USBEHCI0_RST USBEHCI0 Reset 0: Assert 1: De-assert
19:18	/	/	/
17	R/W	0x0	USBOHCI1_RST USBOHCI1 Reset 0: Assert 1: De-assert
16	R/W	0x0	USBOHCI0_RST USBOHCI0 Reset 0: Assert 1: De-assert

Offset: 0x0A8C			Register Name: USB_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
15:9	/	/	/
8	R/W	0x0	USBOTG0_GATING Gating Clock For USBOTG0 0: Mask 1: Pass
7:6	/	/	/
5	R/W	0x0	USBEHCI1_GATING Gating Clock For USBEHCI1 0: Mask 1: Pass
4	R/W	0x0	USBEHCI0_GATING Gating Clock For USBEHCI0 0: Mask 1: Pass
3:2	/	/	/
1	R/W	0x0	USBOHCI1_GATING Gating Clock For USBOHCI1 0: Mask 1: Pass
0	R/W	0x0	USBOHCI0_GATING Gating Clock For USBOHCI0 0: Mask 1: Pass

3.3.6.89 0x0ABC DPSS_TOP Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0ABC			Register Name: DPSS_TOP_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	DPSS_TOP_RST DPSS_TOP Reset 0: Assert 1: De-assert
15:1	/	/	/

Offset: 0x0ABC			Register Name: DPSS_TOP_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	DPSS_TOP_GATING Gating Clock For DPSS_TOP 0: Mask 1: Pass

3.3.6.90 0x0B24 DSI Clock Register (Default Value: 0x0000_0000)

Offset: 0x0B24			Register Name: DSI_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	DSI_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON DSI_CLK = Clock Source/M.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: PLL_PERI(1X) 010: PLL_VIDEO0(2X) 011: PLL_VIDEO1(2X) 100: PLL_AUDIO1(DIV2)
23:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

3.3.6.91 0x0B4C DSI Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0B4C			Register Name: DSI_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/

Offset: 0x0B4C			Register Name: DSI_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
16	R/W	0x0	DSI_RST DSI Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	DSI_GATING Gating Clock For DSI 0: Mask 1: Pass

3.3.6.92 0x0B60 TCONLCD Clock Register (Default Value: 0x0000_0000)

Offset: 0x0B60			Register Name: TCONLCD_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	TCONLCD_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON TCONLCD_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: PLL_VIDEO0(1X) 001: PLL_VIDEO0(4X) 010: PLL_VIDEO1(1X) 011: PLL_VIDEO1(4X) 100: PLL_PERI(2X) 101: PLL_AUDIO1(DIV2)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8

Offset: 0x0B60			Register Name: TCONLCD_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1. FACTOR_M is from 0 to 15.

3.3.6.93 0x0B7C TCONLCD Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0B7C			Register Name: TCONLCD_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	TCONLCD_RST TCON LCD Reset 0: Assert 1: DE-assert
15:1	/	/	/
0	R/W	0x0	TCONLCD_GATING Gating Clock For TCON LCD 0: Mask 1: Pass

3.3.6.94 0x0B80 TCONTV Clock Register (Default Value: 0x0000_0000)

Offset: 0x0B80			Register Name: TCONTV_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	TCONTV_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON TCONTV_CLK = Clock Source/M/N.
30:27	/	/	/

Offset: 0x0B80			Register Name: TCONTV_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: PLL_VIDEO0(1X) 001: PLL_VIDEO0(4X) 010: PLL_VIDEO1(1X) 011: PLL_VIDEO1(4X) 100: PLL_PERI(2X) 101: PLL_AUDIO1(DIV2)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

3.3.6.95 0x0B9C TCONTV Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0B9C			Register Name: TCONTV_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	TCONTV_RST TCONTV Reset 0: Assert 1: DE-assert
15:1	/	/	/
0	R/W	0x0	TCONTV_GATING. Gating Clock For TCONTV 0: Mask 1: Pass

3.3.6.96 0x0BAC LVDS Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0BAC			Register Name: LVDS_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	LVDS0_RST LVDS0 Reset 0: Assert 1: De-assert
15:0	/	/	/

3.3.6.97 0x0BB0 TVE Clock Register (Default Value: 0x0000_0000)

Offset: 0x0BB0			Register Name: TVE_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	TVE_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON TVE_CLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: PLL_VIDEO0(1X) 001: PLL_VIDEO0(4X) 010: PLL_VIDEO1(1X) 011: PLL_VIDEO1(4X) 100: PLL_PERI(2X) 101: PLL_AUDIO1(DIV2)
23:10	/	/	/

Offset: 0x0BB0			Register Name: TVE_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

3.3.6.98 0x0BBC TVE Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0BBC			Register Name: TVE_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R/W	0x0	TVE_RST TVE Reset 0: Assert 1: DE-assert
16	R/W	0x0	TVE_TOP_RST TVE_TOP Reset 0: Assert 1: DE-assert
15:2	/	/	/
1	R/W	0x0	TVE_GATING Gating Clock For TVE 0: Mask 1: Pass
0	R/W	0x0	TVE_TOP_GATING Gating Clock For TVE_TOP 0: Mask 1: Pass

3.3.6.99 0x0BC0 TVD Clock Register (Default Value: 0x0000_0000)

Offset: 0x0BC0			Register Name: TVD_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	TVD_CLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON TVD_CLK = Clock Source/M.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: PLL_VIDEO0(1X) 010: PLL_VIDEO1(1X) 011: PLL_PERI(1X)
23:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.100 0x0BDC TVD Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0BDC			Register Name: TVD_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R/W	0x0	TVD_RST TVD Reset 0: Assert 1: De-assert
16	R/W	0x0	TVD_TOP_RST TVD_TOP Reset 0: Assert 1: De-assert
15:2	/	/	/

Offset: 0x0BDC			Register Name: TVD_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x0	TVD_GATING Gating Clock For TVD 0: Mask 1: Pass
0	R/W	0x0	TVD_TOP_GATING Gating Clock For TVD_TOP 0: Mask 1: Pass

3.3.6.101 0x0BF0 LEDC Clock Register (Default Value: 0x0000_0000)

Offset: 0x0BF0			Register Name: LEDC_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	LDEC_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON LEDC_CLK = Clock Source/M/N.
30:25	/	/	/
24	R/W	0x0	CLK_SRC_SEL Clock Source Select 0: HOSC 1: PLL_PERI(1X)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

3.3.6.102 0x0BFC LEDC Bus Gating Reset Register (Default: 0x0000_0000)

Offset: 0x0BFC			Register Name: LEDC_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	LEDC_RST LEDC Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	LEDC_GATING Gating Clock For LEDC 0: Mask 1: Pass

3.3.6.103 0x0C04 CSI Clock Register (Default Value: 0x0000_0000)

Offset: 0x0C04			Register Name: CSI_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CSI_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON CSI_CLK = Clock Source/M.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: PLL_PERI(2X) 001: PLL_VIDEO0(2X) 010: PLL_VIDEO1(2X)
23:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 15.

3.3.6.104 0x0C08 CSI Master Clock Register (Default Value: 0x0000_0000)

Offset: 0x0C08			Register Name: CSI_MASTER_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CSI_MASTER_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON CSI_MASTER_CLK = Clock Source/M.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: PLL_PERI(1X) 010: PLL_VIDEO0(1X) 011: PLL_VIDEO1(1X) 100: PLL_AUDIO1(DIV2) 101: PLL_AUDIO1(DIV5)
23:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.105 0x0C1C CSI Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0C1C			Register Name: CSI_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	CSI_RST CSI Reset 0: Assert 1: DE-assert
15:1	/	/	/

Offset: 0x0C1C			Register Name: CSI_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	CSI_GATING Gating Clock For CSI 0: Mask 1: Pass

3.3.6.106 0x0C50 TPADC Clock Register (Default Value: 0x0000_0000)

Offset: 0x0C50			Register Name: TPADC_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	TPADC_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON TPADC_CLK = Clock Source.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: PLL_AUDIO0(1X)
23:0	/	/	/

3.3.6.107 0x0C5C TPADC Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0C5C			Register Name: TPADC_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	TPADC_RST TPADC Reset 0: Assert 1: De-assert
15:1	/	/	/

Offset: 0x0C5C			Register Name: TPADC_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	TPADC_GATING Gating Clock For TPADC 0: Mask 1: Pass

3.3.6.108 0x0C70 DSP Clock Register (Default Value: 0x0000_0000)

Offset: 0x0C70			Register Name: DSP_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	DSP_CLK_GATING Gating Clock 0: Clock is OFF 1: Clock is ON DSP_CLK = Clock Source/M.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: HOSC 001: CLK32K 010: CLK16M_RC 011: PLL_PERI(2X) 100: PLL_AUDIO1(DIV2)
23:5	/	/	/
4:0	R/W	0x0	FACTOR_M Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.109 0x0C7C DSP Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0C7C			Register Name: DSP_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:19	/	/	/

Offset: 0x0C7C			Register Name: DSP_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
18	R/W	0x0	DSP_DBG_RST DSP_DBG Reset 0: Assert 1: De-assert
17	R/W	0x0	DSP_CFG_RST DSP_CFG Reset 0: Assert 1: De-assert
16	R/W	0x0	DSP_RST DSP Reset 0: Assert 1: De-assert
15:2	/	/	/
1	R/W	0x0	DSP_CFG_GATING Gating Clock For DSP_CFG 0: Mask 1: Pass
0	/	/	/

3.3.6.110 0x0F04 PLL Lock Debug Control Register (Default Value: 0x0000_0000)

Offset: 0x0F04			Register Name: PLL_LOCK_DBG_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_LOCK_FLAG_EN Debug Enable 0: Disable 1: Enable
30:23	/	/	/

Offset: 0x0F04			Register Name: PLL_LOCK_DBG_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
22:20	R/W	0x0	PLL_LOCK_FLAG_SEL Debug Select 000: PLL_CPUX 001: PLL_DDR 010: PLL_PERI(2X) 011: PLL_VIDEO0(4X) 100: PLL_VIDEO1(4X) 101: PLL_VE 110: PLL_AUDIO0 111: PLL_AUDIO1 Others: /
19:0	/	/	/

3.3.6.111 0x0F08 Frequence Detect Control Register (Default Value: 0x0000_0020)

Offset: 0x0F08			Register Name: FRE_DET_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W0C	0x0	ERROR_FLAG Error Flag 0: Write 0 to clear 1: Error
30:9	/	/	/
8:4	R/W	0x2	DET_TIME Detect Time Time=1/32k*(2^RegValue) RegValue is from 0 to 16.
3:2	/	/	/
1	R/W	0x0	FRE_DET_IRQ_EN Frequence Detect IRQ Enable 0: Disable 1: Enable
0	R/W	0x0	FRE_DET_FUN_EN Frequence Detect Function Enable 0: Disable 1: Enable

3.3.6.112 0x0F0C Frequence Up Limit Register (Default Value: 0x0000_0000)

Offset: 0x0F0C			Register Name: FRE_UP_LIM_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	FRE_UP_LIM Frequence Up Limit **The value of the register must be an integral multiple of 32. The unit is kHz.**

3.3.6.113 0x0F10 Frequence Down Limit Register (Default Value: 0x0000_0000)

Offset: 0x0F10			Register Name: FRE_DOWN_LIM_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	FRE_DOWN_LIM Frequence Down Limit **The value of the register must be an integral multiple of 32. The unit is kHz.**

3.3.6.114 0x0F30 CCU FANOUT Clock Gate Register (Default Value: 0x0000_0000)

Offset: 0x0F30			Register Name: CCU_FAN_GATE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4	R/W	0x0	CLK32K_EN Gating for CLK32K 0: Clock is OFF 1: Clock is ON
3	R/W	0x0	CLK25M_EN Gating for CLK25M 0: Clock is OFF 1: Clock is ON

Offset: 0x0F30			Register Name: CCU_FAN_GATE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x0	CLK16M_EN Gating for CLK16M 0: Clock is OFF 1: Clock is ON
1	R/W	0x0	CLK12M_EN Gating for CLK12M 0: Clock is OFF 1: Clock is ON
0	R/W	0x0	CLK24M_EN Gating for CLK24M 0: Clock is OFF 1: Clock is ON

3.3.6.115 0x0F34 CLK27M FANOUT Register (Default Value: 0x0000_0000)

Offset: 0x0F34			Register Name: CLK27M_FAN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CLK27M_EN Gating for CLK27M 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:26	/	/	/
25:24	R/W	0x0	CLK27M_SCR_SEL Clock Source Select 000: PLL_VIDEO0(1X) 001: PLL_VIDEO1(1X) 010: / 011: /
23:10	/	/	/
9:8	R/W	0x0	CLK27M_DIV1 Factor N N= FACTOR_N +1. FACTOR_N is from 0 to 3.
7:5	/	/	/

Offset: 0x0F34			Register Name: CLK27M_FAN_REG
Bit	Read/Write	Default/Hex	Description
4:0	R/W	0x0	CLK27M_DIV0 Factor M M= FACTOR_M +1. FACTOR_M is from 0 to 31.

3.3.6.116 0x0F38 PCLK FANOUT Register (Default Value: 0x0000_0000)

Offset: 0x0F38			Register Name: PCLK_FAN_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	PCLK_DIV_EN Gating for PCLK 0: Clock is OFF 1: Clock is ON PCLK = APB0_CLK/M.
30:5	/	/	/
4:0	R/W	0x0	PCLK_DIV Factor M M= FACTOR_M +1 FACTOR_M is from 0 to 31.

3.3.6.117 0x0F3C CCU FANOUT Register (Default Value: 0x0000_0000)

Offset: 0x0F3C			Register Name: CCU_FAN_REG
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26:24	R/W	0x0	Reserved
23	R/W	0x0	CLK_FANOUT2_EN Gating for CLK_FANOUT2 0: Clock is OFF 1: Clock is ON
22	R/W	0x0	CLK_FANOUT1_EN Gating for CLK_FANOUT1 0: Clock is OFF 1: Clock is ON

Offset: 0x0F3C			Register Name: CCU_FAN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
21	R/W	0x0	CLK_FANOUT0_EN Gating for CLK_FANOUT0 0: Clock is OFF 1: Clock is ON
20:18	/	/	/
17:9	R/W	0x0	Reserved
8:6	R/W	0x0	CLK_FANOUT2_SEL 000:CLK32K (From PLL_PERI(2X)) 001:CLK12M (From DCXO/2) 010:CLK16M (From PLL_PERI(2X)) 011:CLK24M (From DCXO) 100:CLK25M (From PLL_PERI(1X)) 101:CLK27M 110:PCLK CLK_FANOUT2 can be selected to output from the above seven sources.
5:3	R/W	0x0	CLK_FANOUT1_SEL 000:CLK32K (From PLL_PERI(2X)) 001:CLK12M (From DCXO/2) 010:CLK16M (From PLL_PERI(2X)) 011:CLK24M (From DCXO) 100:CLK25M (From PLL_PERI(1X)) 101:CLK27M 110:PCLK CLK_FANOUT1 can be selected to output from the above seven sources.
2:0	R/W	0x0	CLK_FANOUT0_SEL 000:CLK32K (From PLL_PERI(2X)) 001:CLK12M (From DCXO) 010:CLK16M (From PLL_PERI(2X)) 011:CLK24M (From DCXO) 100:CLK25M (From PLL_PERI(1X)) 101:CLK27M 110:PCLK CLK_FANOUT0 can be selected to output from the above seven sources.

3.4 BROM System

3.4.1 Overview

The system has several ways to boot. It has an integrated on-chip Boot ROM (BROM) that is considered the primary program-loader. On the startup process, the T113-S3 starts to fetch the first instruction from address 0x0, where is the BROM located at.

The BROM system is divided into two parts: the firmware exchange launch (FEL) module and the Medium Boot module. FEL is responsible for writing the external data to the local NVM, and Medium Boot is responsible for loading an effective and legitimate BOOT0 from NVM and running.

The BROM system includes the following features:

- Supports CPU0 boot process
- Supports mandatory upgrade process through USB and SD card
- Supports GPIO pin and eFuse to select the boot media type
- Supports normal booting and secure booting
- Secure BROM loads only certified firmware
- Secure BROM ensures that the Secure Boot is in a trusted environment

3.4.2 Functional Description

3.4.2.1 Selecting the Boot Medium

The BROM system supports the following boot media:

- SD card
- eMMC
- SPI NOR FLASH
- SPI NAND FLASH

There are two ways to select the boot medium: GPIO Pin Select and eFuse Select. On startup, the BROM will read the state of BOOT_MODE, and decide whether the GPIO or eFuse to select the type of boot medium based on the state of BOOT_MODE. The BOOT_MODE is the bit of the SID module (register: 0x03006210).

The following table shows BOOT_MODE Setting.

Table 3-6 BOOT_MODE Setting

BOOT_MODE	Boot_Select Type
0	GPIO boot select, indicates that the boot medium is decided by the value of the GPIO pin.
1	eFuse boot select, indicates that the boot medium is decided by the value of the eFuse type.

GPIO Boot Select

If the state of the BOOT_MODE is 0, the boot medium is decided by the value of the GPIO pin. The following table shows the boot medium priority. The boot medium priority describes the possibility that each medium to be selected as the boot medium. The BROM reads the boot0 of the medium with the highest priority first. If the medium does not exist or has any problems, the BROM will try the next medium. Otherwise, the medium will be selected as the boot medium.

Table 3-7 GPIO Boot Select Configuration

Pin_Boot_Select[1:0]	Boot Medium Priority
00	SPI NOR->SPI NAND
01	SMHC0->SPI NOR->other media
10	SMHC0->SPI NAND->other media
11	SMHC0->EMMC2_BOOT->EMMC2_USR->other media

NOTE

The status of the GPIO boot select pin can be read by the bit[12:11] of the system configuration module (register: 0x03000024).

eFuse Boot Select

If the state of the BOOT_MODE is 1, the boot medium is decided by the value of eFuse_Boot_Select_Cfg. The eFuse_Boot_Select_Cfg is divided into 4 groups and each group is 3-bit. The following table shows the groups of eFUSE_Boot_Select.

NOTE

The status of the efuse boot select pin can be read by the bit[27:16] of the SID module (register: 0x03006210).

Table 3-8 Groups of eFuse_Boot_Select

eFuse_Boot_Select_Cfg[11:0]	Group
eFuse_Boot_Select_Cfg[2:0]	eFuse_Boot_Select_1
eFuse_Boot_Select_Cfg[5:3]	eFuse_Boot_Select_2
eFuse_Boot_Select_Cfg[8:6]	eFuse_Boot_Select_3
eFuse_Boot_Select_Cfg[11:9]	eFuse_Boot_Select_4

The four groups take effect with the following priority:

eFuse_Boot_Select_1 -> eFuse_Boot_Select_2 -> eFuse_Boot_Select_3 -> eFuse_Boot_Select_4

For example, eFuse_Boot_Select_2 will not take effect unless eFuse_Boot_Select_1 is set as 0x111, eFuse_Boot_Select_3 will not take effect unless eFuse_Boot_Select_2 is set as 0x111, and so on.

The following table shows the boot medium priority for the different values of eFuse_Boot_Select_n, where n = [4:1]. The eFuse_Boot_Select_1 to eFuse_Boot_Select_3 are the same setting. But for eFuse_Boot_Select_4, if its value is 0x111, the BROM will select the boot medium in the Try mode. The BROM in the Try mode follows the order below to select the boot medium:

SMHC0 -> SPI NOR -> SPI NAND -> SMHC2

Table 3-9 eFuse Boot Select Setting

eFuse_Boot_Select_n	Boot Medium Priority
000	Select the boot medium in Try mode.
001	Reserved
010	SHMC2
011	SPI NOR
100	SPI NAND
101	Reserved
110	Reserved

eFuse_Boot_Select_n	Boot Medium Priority
111	When n is 1 to 3: The boot medium is decided by the value of **eFuse_Boot_Select_(n + 1)**. When n is 4: Select the boot medium in Try mode.

3.4.2.2 Selecting the Boot Mode

For SoCs that have implemented and enabled the ARM TrustZone technology, there are two boot modes: Normal BROM Mode and Secure BROM Mode. Secure BROM Mode is designed to protect against the potential threat that attackers modify the code or data areas in programmable memory.

On startup, the BROM will select the boot mode according to the value of the Secure Enable bit. If the value of Secure Enable bit is 0, the system will boot in Normal BROM Mode. Otherwise, it will boot in Secure BROM Mode.

NOTE

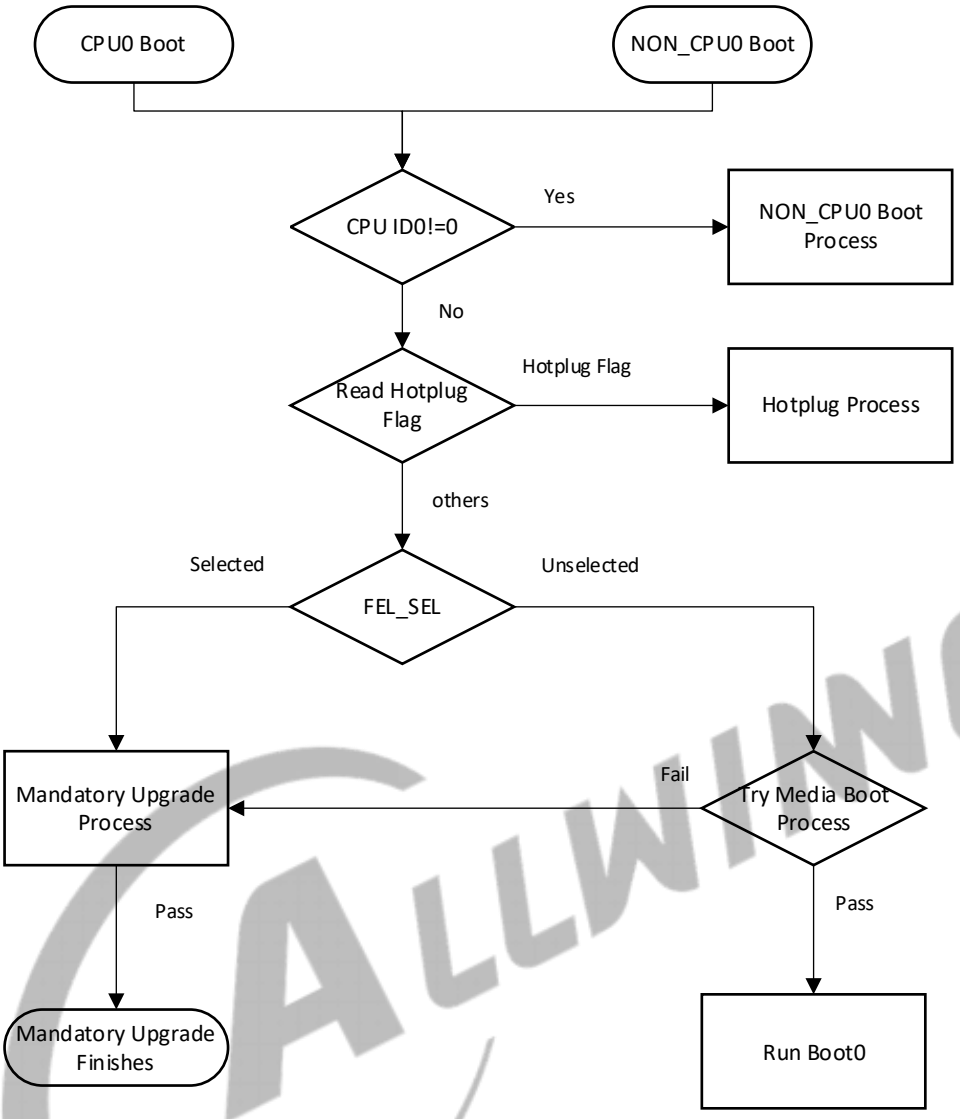
The ARM TrustZone feature is the minimal security functionality an ARM-TrustZone-technology-based system must implement for its Trusted Boot. You can enable the ARM TrustZone feature by configuring the Secure Enable Bit as enabled.

Normal BROM Mode

In Normal BROM Mode, the system boot starts from CPU0, and then the BROM will read the state of the FEL pin. If the FEL pin is high, the system will jump to the fast boot process. If it is low, the system will jump to the mandatory upgrade process.

The following figure shows the boot process in Normal BROM Mode.

Figure 3-6 Boot Process in Normal BROM Mode



Secure BROM Mode

The Secure BROM Mode has the following features:

- Supports X509 certificate

  The certificate is used to check whether the Security Boot software is modified or replaced. Before running the Security Boot software, the system checks the integrity of the certificate make sure the software has not been modified or replaced.

- Supports cryptographic algorithms

  - SHA-256

  - RSA-2048

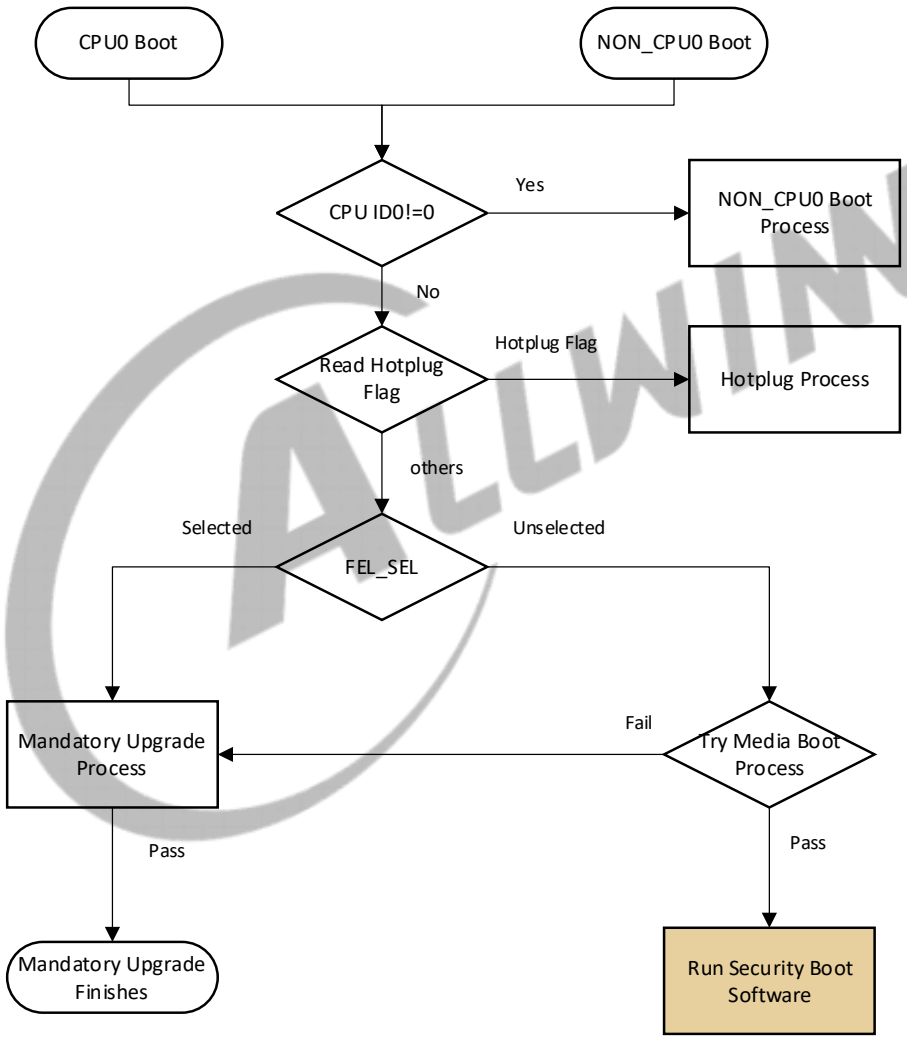
The system uses the Crypto Engine (CE) hardware module to accelerate the speed of encryption and decryption. The use of standard cryptography ensures the reliability of the firmware images. The reliable firmware image ensures that the system security state can be as expected.

- Supports OTP/eFuse

The process of selecting the boot medium in Secure BROM Mode is the same as that in Normal BROM Mode.

In Secure BROM Mode, after the boot medium is selected, the system additionally runs the Security Boot software to authenticate the Sboot bin file.

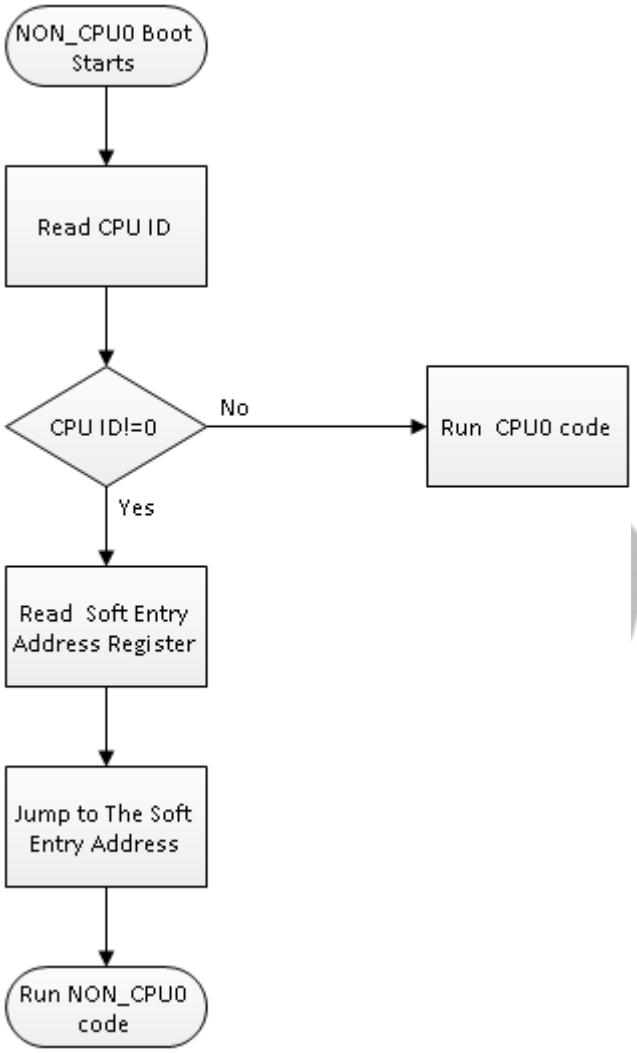
Figure 3-7 Boot Process in Secure BROM Mode



3.4.2.3 NON_CPU0 Boot Process

If CPU ID is greater than 0, the system boot from NON_CPU0, the BROM will read the Soft Entry Address Register, then jump the Soft Entry Address, and run NON_CPU0 boot code. The following figure shows the NON_CPU0 Boot Process.

Figure 3-8 NON_CPU0 Boot Process Diagram



NOTE

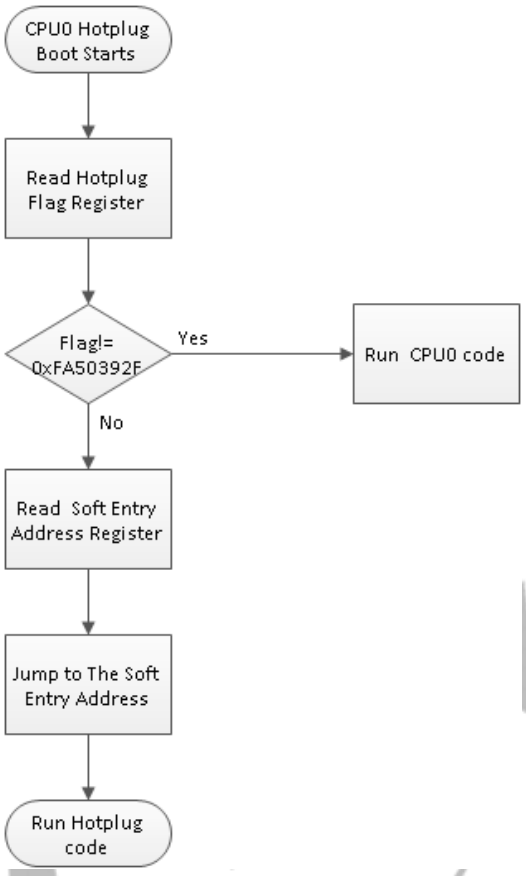
The Soft Entry Address Register of CPU0 is 0x070005C4.

The Soft Entry Address Register of CPU1 is 0x070005C8.

3.4.2.4 CPU0 Hotplug Process

The Hotplug Flag determines whether the system will do Hotplug boot, if the CPU Hotplug Flag value is equal to 0xFA50392F, then read the Soft Entry Register and the system will jump to the Soft Entry Address. The following figure shows the CPU0 Hotplug Process.

Figure 3-9 CPU0 Hotplug Process Diagram



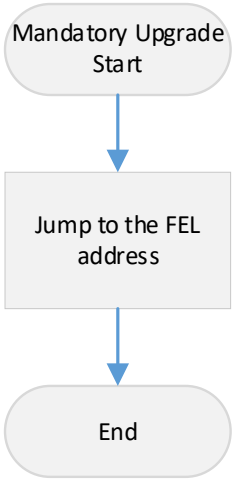
NOTE

- The Hotplug Flag Register is 0x070005C0.
- The Soft Entry Address Register is 0x070005C4.

3.4.2.5 Mandatory Upgrade Process

If the FEL pin is detected to pull low, the system will jump to the mandatory upgrade process. The following figure shows the mandatory upgrade process.

Figure 3-10 Mandatory Upgrade Process



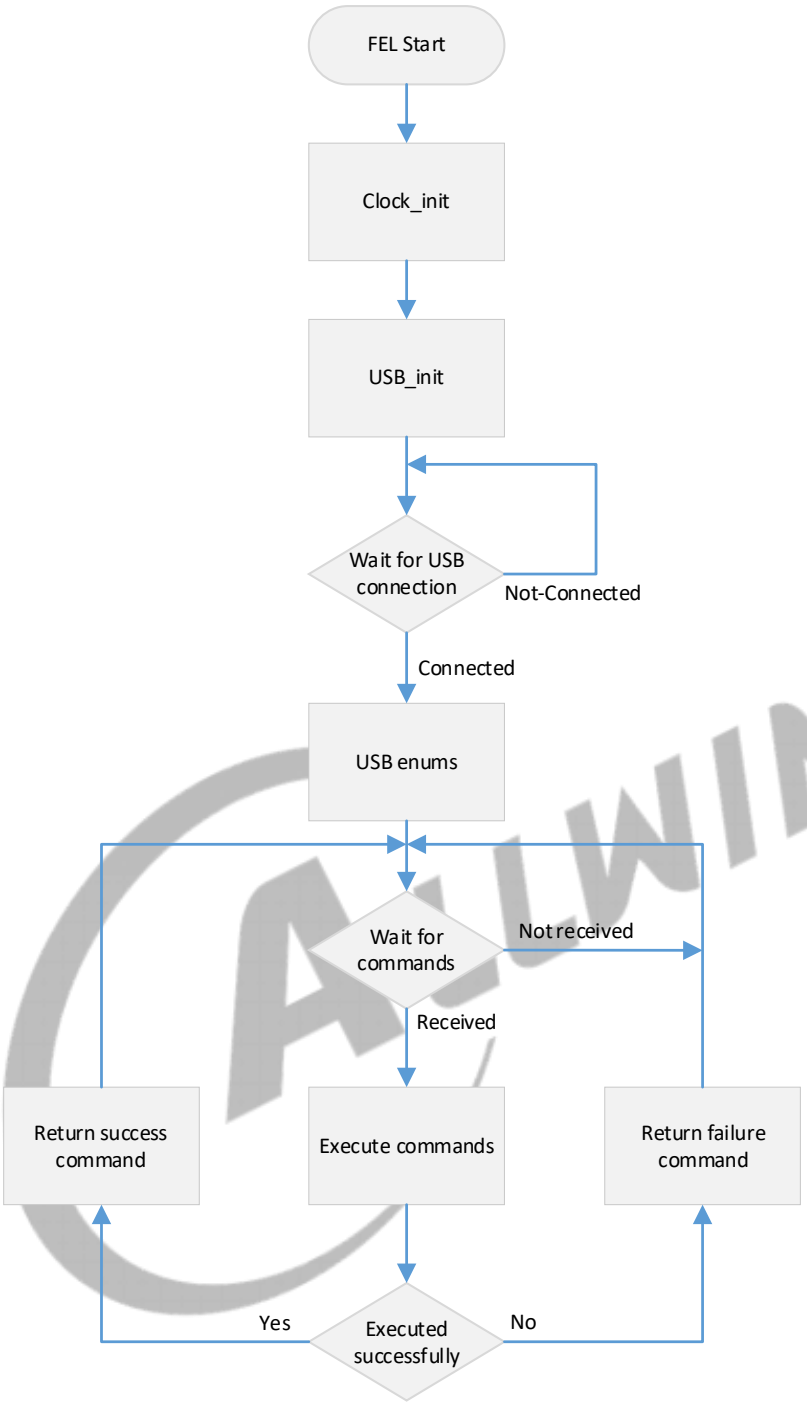
NOTE

- The FEL address of the Normal BROM is 0x20.
- The FEL address of the Secure BROM is 0x64.
- The status of the FEL pin is the bit[8] of the system configuration module (register: 0x03000024).

3.4.2.6 FEL Process

When the system chooses to enter the Mandatory Upgrade Process, the system will jump to the FEL process. The following figure shows the FEL upgrade process.

Figure 3-11 USB FEL Process



3.4.2.7 Fast Boot Process

If the value of the Fast Boot register (0x07090120) in RTC module is not zero, the system will enter the Fast Boot Process. The following table shows the boot medium priority for different values of the Fast Boot register.

Table 3-10 Fast Boot Select Setting

Reg_bit[31:28]	Boot Medium Priority
1	Try process
2	SMHC0->EMMC2_USER->EMMC2_BOOT->Other media
3	SMHC0->SPI NOR->Other media
4	SMHC0->SPI NAND->Other media
5	EMMC2_BOOT->EMMC2_USER->Other media
6	SMHC0->Try process
7	SMHC0->Try process

NOTE

- The bit[28:0] of Fast Boot register is used to record the media information.
- The try process is SMHC0->SPI NOR->SPI NAND->EMMC2_USER->EMMC2_BOOT.

3.5 System Configuration

3.5.1 Overview

The system configuration module is used to configure parameters for system domain, such as SRAM, CPU, PLL, BROM, and so on.

3.5.2 Register List

Module Name	Base Address
SYS_CFG	0x03000000

Register Name	Offset	Description
DSP_BOOT_RAMMAP_REG	0x0008	DSP Boot SRAM Remap Control Register
VER_REG	0x0024	Version Register
EMAC_EPHY_CLK_REG0	0x0030	EMAC-EPHY Clock Register 0
SYS_LDO_CTRL_REG	0x0150	System LDO Control Register
RESCAL_CTRL_REG	0x0160	Resistor Calibration Control Register
RES240_CTRL_REG	0x0168	240ohms Resistor Manual Control Register
RESCAL_STATUS_REG	0x016C	Resistor Calibration Status Register

3.5.3 Register Description

3.5.3.1 0x0008 DSP Boot SRAM Remap Register (Default Value: 0x0000_0001)

Offset: 0x0008			Register Name: DSP_BOOT_RAMMAP_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x1	DSP BOOT SRAM REMAP ENABLE 0: DSP 128K Local SRAM Remap for DSP_SYS 1: DSP 128K Local SRAM Remap for System Boot After system boots up, this bit must be set to 0 before using DSP.

3.5.3.2 0x0024 Version Register (Default Value: UDF)

Offset: 0x0024			Register Name: VER_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
12:11	R	UDF	BOOT_SEL_PAD_STA The value of this bit decides the priority order for each medium type to be selected as the boot media.
10:9	/	/	/
8	R	UDF	FEL_SEL_PAD_STA Fel_Select_Pin_Status 0: Run_FEL 1: Try Media Boot
7:0	/	/	/

3.5.3.3 0x0030 EMAC-EPHY Clock Register 0 (Default Value: 0x0005_8000)

Offset: 0x0030			Register Name: EMAC_EPHY_CLK_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	BPS_EFUSE
27	R/W	0x0	XMII_SEL 0: Internal SMI and MII 1: External SMI and MII
26:25	R/W	0x0	EPHY_MODE Operation Mode Selection 00: Normal Mode 01: Simulation Mode 10: AFE Test Mode 11: Reserved
24:20	R/W	0x0	PHY_ADDR PHY Address
19	R/W	0x0	Reserved
18	R/W	0x1	CLK_SEL 0: 25 MHz 1: 24 MHz

Offset: 0x0030			Register Name: EMAC_EPHY_CLK_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
17	R/W	0x0	LED_POL 0: High active 1: Low active
16	R/W	0x1	SHUTDOWN 0: Power up 1: Shut down
15	R/W	0x1	PHY_SELECT 0: External PHY 1: Internal PHY
14	/	/	/
13	R/W	0x0	RMII_EN 0: Disable RMII Module 1: Enable RMII Module This bit is prior to bit[2]. When this bit is asserted, the MII and RGMII interfaces will be both disabled.
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: Disabled 1: Enabled
3	R/W	0x0	ETXIE Enable EMAC Transmit Clock Invertor 0: Disabled 1: Enabled
2	R/W	0x0	EPIT EMAC PHY Interface Type 0: MII 1: RGMII

Offset: 0x0030			Register Name: EMAC_EPHY_CLK_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
1:0	R/W	0x0	ETCS EMAC Transmit Clock Source 00: Transmit clock source for MII 01: External transmit clock source for GMII and RGMII 10: Internal transmit clock source for GMII and RGMII 11: Reserved

3.5.3.4 0x0150 System LDO Control Register (Default Value: 0x0000_0E0F)

Offset: 0x0150			Register Name: SYS_LDO_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0x0	SPARE Reserved spare register
23:22	/	/	/
21	R/W	0x0	reserved
20	R/W	0x0	reserved
19:18	/	/	/
17	R/W	0x0	reserved
16	R/W	0x0	reserved
15:8	R/W	0xE	LDOB_TRIM LDOB Trimming. Adjust LDOB output, only the low 6-bit is used. 000000:1.167 000001:1.18 000010:1.193 000011:1.207 000100:1.22 000101:1.233 000110:1.247 000111:1.260 001000:1.273 001001:1.287 001010:1.3 001011:1.313 001100:1.327

Offset: 0x0150			Register Name: SYS_LDO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
			001101:1.340 001110:1.353 (default) 001111:1.367 010000:1.38 010001:1.393 010010:1.407 010011:1.42 010100:1.433 010101:1.447 010110:1.46 010111:1.473 011000:1.487 011001:1.5 011010:1.513 011011:1.527 011100:1.54 011101:1.553 011110:1.567 011111:1.58 100000:1.593 100001:1.607 100010:1.627 100011:1.64 100100:1.653 100101:1.667 100110:1.680 100111:1.693 101000:1.707 101001:1.720 101010:1.733 101011:1.747 101100:1.76 101101:1.773 101110:1.787 101111:1.8 110000:1.813 110001:1.827

Offset: 0x0150			Register Name: SYS_LDO_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			110010:1.84 110011:1.853 110100:1.867 110101:1.88 110110:1.893 110111:1.907 111000:1.92 111001:1.933 111010:1.947 111011:1.96 111100:1.973 111101:1.987 111110:2 111111:2.013
7:0	R/W	0xF	LDOA_TIM. LDOA Trimming Adjust LDOA output, only the low 5-bit is used. 00000:1.593 00001:1.607 00010:1.627 00011:1.64 00100:1.653 00101:1.667 00110:1.680 00111:1.693 01000:1.707 01001:1.720 01010:1.733 01011:1.747 01100:1.76 01101:1.773 01110:1.787 01111:1.8 (default) 10000:1.813 10001:1.827 10010:1.84 10011:1.853

Offset: 0x0150			Register Name: SYS_LDO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
			10100:1.867 10101:1.88 10110:1.893 10111:1.907 11000:1.92 11001:1.933 11010:1.947 11011:1.96 11100:1.973 11101:1.987 11110:2 11111:2.013

3.5.3.5 0x0160 Resistor Calibration Control Register (Default Value: 0x0033_0003)

Offset: 0x0160			Register Name: RESCAL_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	Reserved
23:22	/	/	/
21:16	R/W	0x33	Reserved
15:9	/	/	/
8	R/W	0x0	DDR_RES240_Trimming_SEL 240ohms Resistor Trimming Source Select 0: Trimming value from RESCAL 1: Trimming value from RES240_TRIM
7:3	/	/	/
2	R/W	0x0	RESCAL_MODE RESCAL Calibration Mode Select 0: Auto Calibration 1: Reserved
1	R/W	0x1	CAL_ANA_EN. Calibration Circuits Analog Enable 0: Disable 1: Enable

Offset: 0x0160			Register Name: RESCAL_CTRL_REG
Bit	Read/Write	Default/Hex	Description
0	R/W	0x1	CAL_EN Auto Calibration Enable 0: Disable 1: Enable

3.5.3.6 0x0168 240ohms Resistor Manual Control Register (Default Value: 0x0000_0033)

Offset: 0x0168			Register Name: RES240_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:6	/	/	/
5:0	R/W	0x33	DDR_RES240_TRIM 240ohms Resistor trimming bit

3.5.3.7 0x016C Resistor Calibration Status Register (Default Value: 0x0000_0000)

Offset: 0x016C			Register Name: RESCAL_STATUS_REG
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8	RO	0x0	COUT Calibration Circuits Analog Compare Output
7:6	/	/	/
5:0	RO	0x0	RES_CAL_DO RESCAL Calibration Results Output

3.6 Timer

3.6.1 Overview

The timer module implements the timing and counting functions. The timer module includes timer0, timer1, watchdog and audio video synchronization (AVS).

The timer0 and timer1 are completely consistent. The main features for timer0 and timer1 are as follows:

- Alternative count clock: LOSC or OSC24M. The LOSC can be either the internal or external low-frequency clock, and the external one has more accuracy.
- Supports 8 prescale factors
- Programmable 32-bit down timer
- Supports two timing modes: periodic mode and single counting mode
- Generates an interrupt when the count is decreased to 0

The watchdog is used to transmit a reset signal to reset the entire system when an exception occurs in the system. The main features for the watchdog are as follows:

- Single clock source: OSC24M/750
- Supports 12 initial values
- Supports the generation of timeout interrupts
- Supports the generation of reset signals
- Supports Watchdog Restart

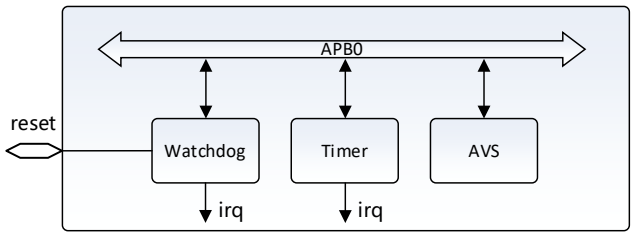
The AVS is used to synchronize the audio and video. The AVS module includes AVS0 and AVS1, which are completely consistent. The main features for the AVS are as follows:

- Single clock source: OSC24M
- Programmable 33-bit up timer
- Supports updating the initial value anytime
- 12-bit frequency divider factor
- Supports Pause/Start function

3.6.2 Block Diagram

The following figure shows the functional block diagram of the timer module.

Figure 3-12 Timer Block Diagram



The watchdog, timer (including timer0 and timer1), and AVS are all mounted at the APB0 bus. The system configures the parameters of these configure registers via APB0 bus.

The timer and watchdog are both down counters and support generating interrupts after the counting value reaches 0.

For watchdog, the system is responsible for configuring the interval value. If the system fails to restart the watchdog regularly because of some exceptional situations, such as the bus hang, the watchdog will send out a Watchdog Reset External signal to reset the system. And the signal will be transmitted to the Reset pad to reset the PMIC.

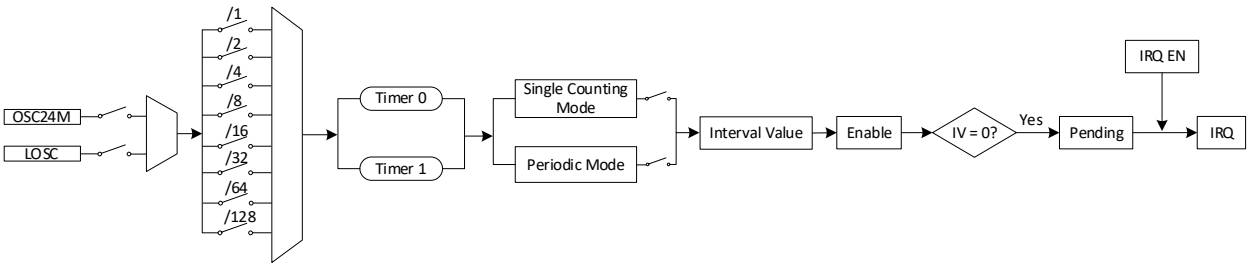
3.6.3 Functional Description

3.6.3.1 Timer

The timer (including timer0 and timer1) is a 32-bit down counter. The counter value is decremented by 1 on each rising edge of the timer clock.

The following figure shows the block diagram for the timer.

Figure 3-13 Block Diagram for the Timer



The clock source for the timer can be either OSC24M or LOSC. For LOSC, it can be either the internal or external low-frequency clock. The external one has more accuracy.

Each timer has a prescale that divides the working clock frequency by 1, 2, 4, 8, 16, 32, 64, or 128. And each timer can generate independent interrupts.

Timing Modes

The timer has two timing modes: the single counting mode and periodic mode. You can configure the timing mode via the bit[7] of **TMRn_CTRL_REG** (n = 0 or 1). The value 0 is for the period mode and value 1 is for the single counting mode.

- Single Counting Mode

  In the single counting mode, the timer starts counting from the interval value and generates an interrupt after the counter decreases to 0, and then stops counting. It starts to count again only when a new interval value is loaded.

- Periodic Mode

  In the periodic mode, the timer restarts another round of counting after generating the interrupt. It reloads data from the **TMRn_INTV_VALUE_REG** and then continues to count.

Formula for Calculating the Timer Time

The following formula decribes the relationship among timer parameters.

$$T_{timer} = \frac{TMRn\_INTV\_VALUE\_REG - TMRn\_CUR\_VALUE\_REG}{TMRn\_CLK\_SRC} \times TMRn\_CLK\_PRES$$

Where,

The parameter n is either 0 or 1;

T_{timer} is the remaining time of the timer;

TMRn_INTV_VALUE_REG is the interval value of the timer;

TMRn_CUR_VALUE_REG is the current value of the timer;

TMRn_CLK_SRC is the frequency of the timer clock source;

TMRn_CLK_PRES is the prescale ratio of the timer clock.

Initializing the Timer

Follow the steps below to initialize the timer:

1. Configure the timer parameters clock source, prescale factor, and timing mode by writing **TMRn_CTRL_REG.** There is no sequence requirement of configuring the parameters.

2. Write the interval value.

   a) Write TMRn_INTV_VALUE_REG to configure the interval value for the timer.

   b) Write bit[1] of TMRn_CTRL_REG to load the interval value to the timer. The value of the bit will be cleared automatically after loading the interval value.

3. Write bit[0] of TMRn_CTRL_REG to start the timer. To get the current value of the timer, read TMRn_CUR_VALUE_REG.

Processing the Interrupt

Follow the steps below to process the interrupt:

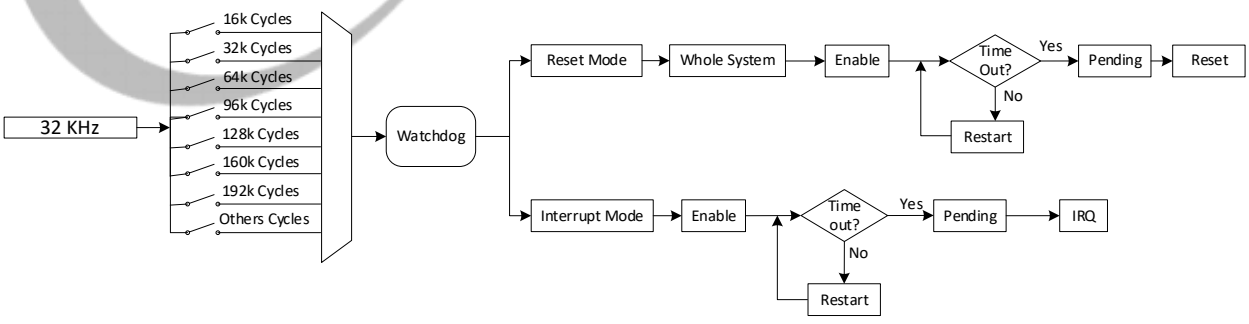
1. Enable interrupts for the timer: write the enable bit of the corresponding interrupt in TMR_IRQ_EN_REG for the timer. The timer will generate an interrupt everytime the count value reaches 0.
2. After entering the interrupt process, write the pending bit of the corresponding interrupt in TMR_IRQ_STA_REG to clear the interrupt pending, and execute the process of waiting for the interrupt.
3. Resume the interrupt and continue to execute the interrupted process.

3.6.3.2 Watchdog

The watchdog is a 32-bit down counter. The counter value is decremented by 1 on each rising edge of the count clock.

The following figure shows the block diagram for the watchdog.

Figure 3-14 Block Diagram for the Watchdog



The clock source of the watchdog is OSC24M/750. There are 12 configurable initial count values.

Operating Modes

The watchdog has two operating modes: the interrupt mode and reset mode.

- In the interrupt mode, when the counter value reaches 0 and WDOG_IRQ_EN_REG is enabled, the watchdog generates an interrupt.
- In the reset mode, when the counter value reaches 0, the watchdog generates a reset signal to reset the entire system.

You can configure the operating mode for the watchdog via the bit[1:0] of the WDOG_CFG_REG. The value 0x2 is for the interrupt mode and the value 0x1 is for the reset mode.

Both the interrupt mode and reset mode support Watchdog Restart. You can make the watchdog to count from the initial value at any time by configuring the WDOG_CTRL_REG: write 0xA57 to bit[12:1], then write 1 to bit[0].

Initializing the Watchdog

Follow the steps below to initialize the watchdog:

1. Write the bit[1:0] of **WDOG_CFG_REG** to configure the watchdog operating mode so that the watchdog can generate interrupts or output reset signals.
2. Write the bit[7:4] of **WDOG_MODE_REG** to configure the initial count value.
3. Write the bit[0] of **WDOG_MODE_REG** to enable the watchdog.

Processing the Interrupt

In the interrupt mode, the watchdog is used as a counter. It generates an interrupt everytime the count value reaches 0.

Follow the steps below to process the interrupt:

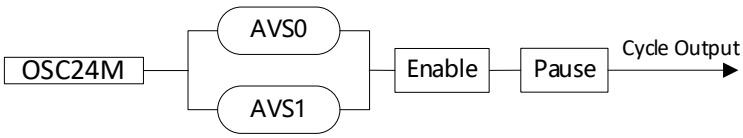
1. Write the enable bit of **WDOG_IRQ_EN_REG** to enable the interrupt.
2. After entering the interrupt process, write the pending bit of **WDOG_IRQ_STA_REG** to clear the interrupt pending and execute the process of waiting for the interrupt.
3. Resume the interrupt and continue to execute the interrupted process.

3.6.3.3 AVS

The AVS is a 33-bit up counter. The counter value is increased by 1 on each rising edge of the count clock. There is a clock gate in CCU module to control the output of the AVS counter. To operate the AVS, open the clock gate first.

The following figure shows the block diagram for the AVS.

Figure 3-15 Block Diagram for the AVS



The clock source of the AVS is OSC24M. There is a 12-bit division factor for each AVS, N0 for AVS0 and N1 for AVS1. When the timer increases from 0 to N1 or N2, the AVS counter adds 1. When the counter reaches 33-bit upper limit, the AVS will start to count from the initial value again.

The AVS supports changing the initial value and division factor at anytime. And the AVS supports restarting from the initial value or pausing at anytime.

Starting or Pausing the AVS

Follow the steps below:

1. Write **AVS_CNT_DIV_REG** to configure the division factor.
2. Write **AVS_CNTn_REG** (n = 0 or 1) to configure the initial value.
3. Write **AVS_CNT_CTL_REG** to enable the AVS. You can pause the AVS at any time.

3.6.4 Programming Guidelines

3.6.4.1 Configuring the Timer

The following example shows how to make a one-millisecond delay with the clock source selected as OSC24M, the operating mode sets as single counting mode, and the pre-scale sets as 2.

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

```
writel(readl(TMR_0_CTRL)|(1<<0), TMR_0_CTRL);      //Enable Timer0
```

3.6.4.2 Resetting the Watchdog

The following example shows how to make the watchdog to generate a reset signal to the whole system after 1 second. The clock source for the watchdog is OSC24M/750.

```
writel(0x1, WDOG_CONFIG);                          //Set the operating mode as the reset mode.
writel(0x10, WDOG_MODE);                           //Set the interval value as 1 s.
writel(readl(WDOG_MODE)|(1<<0), WDOG_MODE);        //Enable the Watchdog.
```

3.6.4.3 Restarting the Watchdog

The following example shows how to restart the watchdog. In this example, the clock source is OSC24M/750, the interval value is 1 second, and the watchdog operating mode is the reset mode.

If the execution time of "other codes" is shorter than 1 second, the watchdog will restart from the interval value before it count to zero and generates the reset signal. Otherwise, the watchdog will reset the whole system before the code "writel(readl(WDOG_CTRL)|(0xA57<<1)|(1<<0), WDOG_CTRL)" is executed.

```
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); //Write 0xA57 at Key Field and Restart Watchdog
```

3.6.5 Register List

Module Name	Base Address
Timer	0x02050000

Register Name	Offset	Description
TMR_IRQ_EN_REG	0x0000	Timer IRQ Enable Register
TMR_IRQ_STA_REG	0x0004	Timer Status Register

Register Name	Offset	Description
TMR0_CTRL_REG	0x0010	Timer0 Control Register
TMR0_INTV_VALUE_REG	0x0014	Timer0 Interval Value Register
TMR0_CUR_VALUE_REG	0x0018	Timer0 Current Value Register
TMR1_CTRL_REG	0x0020	Timer1 Control Register
TMR1_INTV_VALUE_REG	0x0024	Timer1 Interval Value Register
TMR1_CUR_VALUE_REG	0x0028	Timer1 Current Value Register
WDOG_IRQ_EN_REG	0x00A0	Watchdog IRQ Enable Register
WDOG_IRQ_STA_REG	0x00A4	Watchdog Status Register
WDOG_SOFT_RST_REG	0x00A8	Watchdog Software Reset Register
WDOG_CTRL_REG	0x00B0	Watchdog Control Register
WDOG_CFG_REG	0x00B4	Watchdog Configuration Register
WDOG_MODE_REG	0x00B8	Watchdog Mode Register
WDOG_OUTPUT_CFG_REG	0x00BC	Watchdog Output Configuration Register
AVS_CNT_CTL_REG	0x00C0	AVS Control Register
AVS_CNT0_REG	0x00C4	AVS Counter 0 Register
AVS_CNT1_REG	0x00C8	AVS Counter 1 Register
AVS_CNT_DIV_REG	0x00CC	AVS Divisor Register

3.6.6 Register Description

3.6.6.1 0x0000 Timer IRQ Enable Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: TMR_IRQ_EN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1	R/W	0x0	TMR1_IRQ_EN Timer1 Interrupt Enable 0: Disabled 1: Enabled
0	R/W	0x0	TMR0_IRQ_EN Timer0 Interrupt Enable 0: Disabled 1: Enabled

3.6.6.2 0x0004 Timer IRQ Status Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: TMR_IRQ_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1	R/W1C	0x0	TMR1_IRQ_PEND The IRQ pending bit for Timer1 0: No effect 1: Pending, indicates that the interval value of the timer 1 is reached. Write 1 to clear the pending status.
0	R/W1C	0x0	TMR0_IRQ_PEND The IRQ pending bit for Timer0 0: No effect 1: Pending, indicates that the interval value of the timer 0 is reached. Write 1 to clear the pending status.

3.6.6.3 0x0010 Timer0 Control Register (Default Value: 0x0000_0004)

Offset: 0x0010			Register Name: TMR0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	TMR0_MODE Select the timing mode for timer0 0: Periodic mode. When the interval value of the timer 0 is reached, the timer will restart another round of counting automatically. 1: Single counting mode. When the interval value of the timer 0 is reached, the timer will stop counting.

Offset: 0x0010			Register Name: TMR0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
6:4	R/W	0x0	TMR0_CLK_PRES Select the pre-scale of timer0 clock source 000: /1 001: /2 010: /4 011: /8 100: /16 101: /32 110: /64 111: /128
3:2	R/W	0x1	TMR0_CLK_SRC Select the clock source for timer0 00: LOSC 01: OSC24M 10: / 11: /
1	R/W	0x0	TMR0_RELOAD Timer0 Reload 0: No effect 1: Reload the Interval value for timer0 After the bit is set, it can not be written again before it is cleared automatically.
0	R/W	0x0	TMR0_EN Timer0 Enable 0: Stop/Pause 1: Start By setting the bit to 1, the timer will be started. It reloads the interval value register and then counts from the interval value to 0. By setting the bit to 0 before the timer counts to 0, the timer will be paused. The bit will be locked to 0 for at least 2 cycles. Within the 2 cycles, you cannot set the bit to 1 to restart the timer. The timer supports updating the interval value in the pause state. To start to down-count from the updated interval value, set both the reload bit and enable bit to 1. Additionally, in the single counting mode, after the count value reaches 0, the system will automatically change the bit to 0 to stop the timer.

3.6.6.4 0x0014 Timer0 Interval Value Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: TMR0_INTV_VALUE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	TMR0_INTV_VALUE Timer0 Interval Value

NOTE

Take the system clock and timer clock source into consideration when setting the interval value.

3.6.6.5 0x0018 Timer0 Current Value Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: TMR0_CUR_VALUE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	TMR0_CUR_VALUE Timer0 Current Value Timer0 current value is a 32-bit down-counter (from interval value to 0).

3.6.6.6 0x0020 Timer1 Control Register (Default Value: 0x0000_0004)

Offset: 0x0020			Register Name: TMR1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	TMR1_MODE Select the timing mode for timer1 0: Periodic mode. When the interval value of the timer 1 is reached, the timer will restart another round of counting automatically. 1: Single counting mode. When the interval value of the timer 1 is reached, the timer will stop counting.

Offset: 0x0020			Register Name: TMR1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
6:4	R/W	0x0	TMR1_CLK_PRES Select the pre-scale of timer1 clock source 000: /1 001: /2 010: /4 011: /8 100: /16 101: /32 110: /64 111: /128
3:2	R/W	0x1	TMR1_CLK_SRC Select the pre-scale of timer1 clock source 00: LOSC 01: OSC24M 10: / 11: /
1	R/W	0x0	TMR1_RELOAD Timer1 Reload 0: No effect 1: Reload the interval value for timer1 After the bit is set, it can not be written again before it is cleared automatically.
0	R/W	0x0	TMR1_EN Timer1 Enable 0: Stop/Pause 1: Start By setting the bit to 1, the timer will be started. It reloads the interval value register and then counts from the interval value to 0. By setting the bit to 0 before the timer counts to 0, the timer will be paused. The bit will be locked to 0 for at least 2 cycles. Within the 2 cycles, you cannot set the bit to 1 to restart the timer. The timer supports updating the interval value in the pause state. To start to down-count from the updated interval value, set both the reload bit and enable bit to 1. Additionally, in the single counting mode, after the count value reaches 0, the system will automatically change the bit to 0 to stop the timer.

3.6.6.7 0x0024 Timer1 Interval Value Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: TMR1_INTV_VALUE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	TMR1_INTV_VALUE Timer1 Interval Value

NOTE

Take the system clock and timer clock source into consideration when setting the interval value.

3.6.6.8 0x0028 Timer1 Current Value Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: TMR1_CUR_VALUE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	TMR1_CUR_VALUE Timer1 Current Value Timer1 current value is a 32-bit down-counter (from interval value to 0).

3.6.6.9 0x00A0 Watchdog IRQ Enable Register (Default Value: 0x0000_0000)

Offset: 0x00A0			Register Name: WDOG_IRQ_EN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	WDOG_IRQ_EN Watchdog Interrupt Enable 0: Disabled 1: Enabled

3.6.6.10 0x00A4 Watchdog Status Register (Default Value: 0x0000_0000)

Offset: 0x00A4			Register Name: WDOG_IRQ_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W1C	0x0	WDOG_IRQ_PEND The IRQ pending bit for the watchdog Write 1 to clear the pending status. 0: No effect 1: Pending, indicates that the interval value of the watchdog is reached.

3.6.6.11 0x00A8 Watchdog Software Reset Register (Default Value: 0x0000_0000)

Offset: 0x00A8			Register Name: WDOG_SOFT_RST_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	W	0x0	KEY_FIELD Key Field To change the value of bit[0], this field should be filled with 0x16AA.
15:1	/	/	/
0	R/W1C	0x0	Soft Reset Enable 0: De-assert 1: Reset the system **Note: To use the bit to reset the system, the watchdog first needs to be disabled.**

3.6.6.12 0x00B0 Watchdog Control Register (Default Value: 0x0000_0000)

Offset: 0x00B0			Register Name: WDOG_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
12:1	W	0x0	WDOG_KEY_FIELD Watchdog Key Field It should be written to 0xA57. Writing any other value in this field aborts the write operation.

Offset: 0x00B0			Register Name: WDOG_CTRL_REG
Bit	Read/Write	Default/Hex	Description
0	R/W1S	0x0	WDOG_RESTART Watchdog Restart 0: No effect 1: Restart the watchdog

3.6.6.13 0x00B4 Watchdog Configuration Register (Default Value: 0x0000_0001)

Offset: 0x00B4			Register Name: WDOG_CFG_REG
Bit	Read/Write	Default/Hex	Description
31:16	W	0x0	KEY_FIELD Key Field To change the value of bit[15:0], this field should be filled with 0x16AA.
15:9	/	/	/
8	R/W	0x0	WDOG_CLK_SRC Select the clock source for the watchdog. 0: HOSC_32K, that is, OSC24M/750. It is a 32 KHz clock divided from the OSC24M. 1: LOSC_32K. A clock provided by the LOSC.
7:2	/	/	/
1:0	R/W	0x1	WDOG_MODE Configure the operating mode for the watchdog 00: / 01: To whole system 10: Only interrupt mode 11: /

3.6.6.14 0x00B8 Watchdog Mode Register (Default Value: 0x0000_0000)

Offset: 0x00B8			Register Name: WDOG_MODE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	W	0x0	KEY_FIELD Key Field To change the value of bit[15:0], this field should be filled with 0x16AA.
15:8	/	/	/
7:4	R/W	0x0	WDOG_INTV_VALUE Watchdog Interval Value 0000: 16000 cycles (0.5 s) 0001: 32000 cycles (1 s) 0010: 64000 cycles (2 s) 0011: 96000 cycles (3 s) 0100: 128000 cycles (4 s) 0101: 160000 cycles (5 s) 0110: 192000 cycles (6 s) 0111: 256000 cycles (8 s) 1000: 320000 cycles (10 s) 1001: 384000 cycles (12 s) 1010: 448000 cycles (14 s) 1011: 512000 cycles (16 s) Others: Reserved **Note: The corresponding clock cycles for the interval value (IV) depends on the frequency of the clock: Cycles = F_{CLK} * IV.** For example, to get a interval value of 0.5 second, if the clock source is HOSC_32K (whose frequency is 32 KHz), the cycle number is 16,000; if the clock source is LOSC_32K (whose frequency is 32.768 kHz), the cycle number is 16,384.
3:1	/	/	/
0	R/W	0x0	WDOG_EN Watchdog Enable 0: No effect 1: Enable the Watchdog

3.6.6.15 0x00BC Watchdog Output Configuration Register (Default Value: 0x0000_001F)

Offset: 0x00BC			Register Name: WDOG_OUTPUT_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11:0	R/W	0x1F	WDOG OUTPUT CONFIG Configure the valid time for the watchdog reset signal. T = 1/32ms*(N + 1) The default value is 1 ms.

3.6.6.16 0x00C0 AVS Counter Control Register (Default Value: 0x0000_0000)

Offset: 0x00C0			Register Name: AVS_CNT_CTL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9	R/W	0x0	AVS_CNT1_PS Audio/Video Sync Counter 1 Pause Control 0: Do not pause. 1: Pause the AVS counter1.
8	R/W	0x0	AVS_CNT0_PS Audio/Video Sync Counter 0 Pause Control 0: Do not pause. 1: Pause the AVS counter0.
7:2	/	/	/
1	R/W	0x0	AVS_CNT1_EN Audio/Video Sync Counter 1 Enable/Disable The clock source is OSC24M. 0: Disabled 1: Enabled
0	R/W	0x0	AVS_CNT0_EN Audio/Video Sync Counter 0 Enable/Disable The clock source is OSC24M. 0: Disabled 1: Enabled

3.6.6.17 0x00C4 AVS Counter 0 Register (Default Value: 0x0000_0000)

Offset: 0x00C4			Register Name: AVS_CNT0_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	AVS_CNT0 The higher 32 bits of AVS counter0. AVS counter0 is a 33-bit up counter. The initial value consists of two parts: this register forms the bit[32:1] of AVS counter0 and the bit[0] is zero. You can set the initial value of the AVS counter0 by software. The initial value can be updated at anytime. You can also pasuse the counter by setting AVS_CNT0_PS to ”1”. The counter value will not increase when it is paused.

3.6.6.18 0x00C8 AVS Counter 1 Register (Default Value: 0x0000_0000)

Offset: 0x00C8			Register Name: AVS_CNT1_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	AVS_CNT1 The higher 32 bits of AVS counter1. AVS counter1 is a 33-bit up counter. The initial value consists of two parts: this register forms the bit[32:1] of AVS counter0 and the bit[0] is zero. You can set the initial value of the AVS counter1 by software. The initial value can be updated at anytime. You can also pasuse the counter by setting AVS_CNT1_PS to ”1”. The counter value will not increase when it is paused.

3.6.6.19 0x00CC AVS Counter Divisor Register (Default Value: 0x05DB_05DB)

Offset: 0x00CC			Register Name: AVS_CNT_DIV_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/

Offset: 0x00CC			Register Name: AVS_CNT_DIV_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
27:16	R/W	0x5DB	AVS_CNT1_D N1, the divisor factor for AVS1. The clock for AVS1 is 24 MHz/N1. N1 = Bit[27:16] + 1. The valid value for N1 is from 1 to 0x7ff. There is an internal 12-bit counter maintained by the engine of the 33-bit AVS1. The 12-bit counter is used for counting the cycle number of the clock OSC24M. When the value of the 12-bit counter reaches N1, the internal 33-bit counter register will increase 1 and the 12-bit counter will reset to zero and restart again. You can change the value of N1 via the software at any time.
15:12	/	/	/
11:0	R/W	0x5DB	AVS_CNT0_D N0, the divisor factor for AVS0. The clock for AVS0 is 24MHz/N0. N0 = Bit[11:0] + 1. The valid value for N0 is from 1 to 0x7ff. There is an internal 12-bit counter maintained by the engine of the 33-bit AVS0. The 12-bit counter is used for counting the cycle number of the clock OSC24M. When the value of the 12-bit counter reaches N0, the internal 33-bit counter register will increase 1 and the 12-bit counter will reset to zero and restart again. You can change the value of N0 via the software at any time.

3.7 High Speed Timer

3.7.1 Overview

The high speed timer (HSTimer) module consists of HSTimer0 and HSTimer1. HSTimer0 and HSTimer1 are down counters that implement timing and counting functions. They are completely consistent. Compared with the timer module, the HSTimer module provides more precise timing and counting.

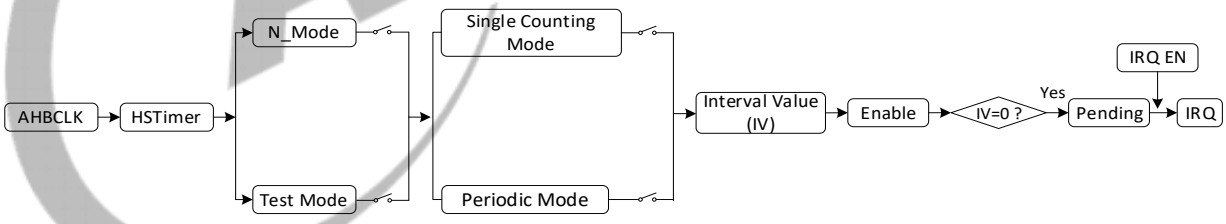
The HSTimer has the following features:

- Single clock source: AHB0
- Supports 5 prescale factors
- Configurable 56-bit down timer
- Supports 2 timing modes: periodic mode and one-shot mode
- Supports the test mode
- Generates an interrupt when the count is decreased to 0

3.7.2 Block Diagram

The following figure shows the block diagram of the HSTimer.

Figure 3-16 HSTimer Block Diagram



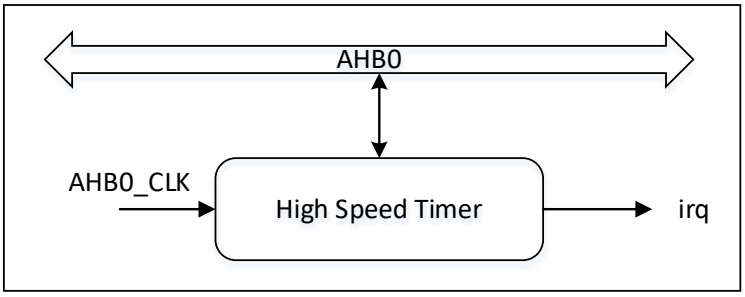
3.7.3 Functional Description

The HSTimers are 56-bit down counters. The counter value is decremented by 1 on each rising edge of the count clock. Each HSTimer has a prescaler that divides the working clock frequency of each working timer by 1, 2, 4, 8, or 16.

3.7.3.1 Typical Application

The following figure shows a typical application of HSTimer module.

Figure 3-17 Typical Application for HSTimer



The HSTimer module is mounted at AHB0, and can control the registers via AHB0. AHB0 is the clock source of the HSTimer. When the count value reaches zero, the HSTimer generates an interrupt.

3.7.3.2 Count Modes

The HSTimer has two count modes: one-shot mode and periodic mode. You can configure the timing mode via the bit[7] of **HS_TMRn_CTRL_REG** (n = 0 or 1). The value 0 is for the period mode and value 1 is for the one-shot mode.

- One-shot Mode

  When the count value of the HSTimer reaches 0, the HSTimer stops counting. The HSTimer starts to count again only when a new value is loaded.

- Periodic Mode

  The HSTimer counts continuously. When the count value of the HSTimer reaches 0, the HSTimer reloads an initial value from **HS_TMRn_INTV_LO_REG** and **HS_TMRn_INTV_HI_REG** and then continues to count.

3.7.3.3 Operating Modes

The HSTimer has two operating modes: the normal mode and test mode. You can configure the operating mode via the bit[31] of **HS_TMRn_CTRL_REG**. The value 0 is for the normal mode and value 1 is for the test mode.

- Normal Mode

  In the normal mode, the HSTimer is used as a 56-bit down counter, which can finish one-shot counting and periodic counting. The interval value for the HSTimer consists of two parts: **HS_TMRn_INTV_LO_REG** forms the bit[31:0] and **HS_TMRn_INTV_HI_REG** forms the bit[55:32]. To read or write the interval value, **HS_TMRn_INTV_LO_REG** should be done before **HS_TMRn_INTV_HI_REG**.

- Test Mode

In the test mode, the HSTimer is used as a 24-bit down counter. **HS_TMRn_INTV_LO_REG** must be set to 0x1, and **HS_TMRn_INTV_HI_REG** acts as the initial value for the HSTimer.

3.7.3.4 HSTimer Formula

The following formula describes the relationship among HSTimer parameters in the normal mode.

$$T_{HSTimer} = \frac{(HS\_TMRn\_INTV\_HI\_REG<<32 + HS\_TMRn\_INTV\_LO\_REG) - (HS\_TMRn\_CURNT\_HI\_REG<<32 + HS\_TMRn\_CURNT\_LO\_REG)}{AHB0CLK} \times HS\_TMRn\_CLK$$

Where,

The parameter n is either 0 or 1;

$T_{HSTimer}$ is the remaining time of the timer;

HS_TMRn_INTV_HI_REG is bit[55:32] of the HSTimer interval value;

HS_TMRn_INTV_LO_REG is bit[31:0] of the HSTimer interval value;

HS_TMRn_CURNT_HI_REG is bit[55:32] of the HSTimer current value;

HS_TMRn_CURNT_LO_REG is bit[31:0] of the HSTimer current value;

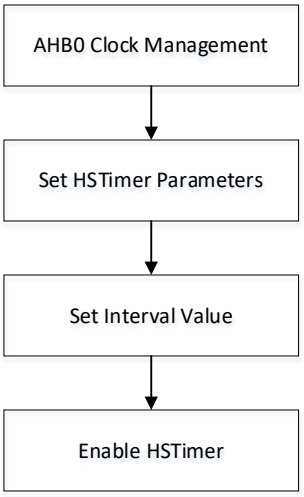
AHB0CLK is the frequency of AHB0 Clock (the HSTimer clock source);

HS_TMRn_CLK is the prescale ratio of the HSTimer clock.

3.7.3.5 Initializing the HSTimer

The following figure shows the process of HSTimer initialization.

Figure 3-18 HSTimer Initialization Process



1. AHB0 clock management: Open the clock gate of AHB0 and de-assert the soft reset of AHB0 in CCU.
2. Configure the corresponding parameters of the HSTimer: clock source, prescaler factor, operating mode, and timing mode. There is no sequence requirement when writing the above parameters to **HS_TMRn_CTRL_REG**.
3. Write the initial value: Write the lower 32 bits of the initial value to **HS_TMRn_INTV_LO_REG** first, and then the higher 24 bits to **HS_TMRn_INTV_HI_REG**. Normally, write the bit[1] of **HS_TMRn_CTRL_REG** to load the initial value. If the HSTimer is in the timing stop stage, write 1 to the bit[1] and bit[0] of **HS_TMRn_CTRL_REG** at the same time to reload the initial value.
4. Enable HSTimer: Write the bit[0] of **HS_TMRn_CTRL_REG** to enable HSTimer to count.
5. Reading **HS_TMRn_CURNT_LO_REG** and **HS_TMRn_CURNT_HI_REG** can get current counting value.

3.7.3.6 Processing the HSTimer Interrupt

Follow the steps below to process the HSTimer interrupt:

1. Enable interrupt: Write the corresponding interrupt enable bit of **HS_TMR_IRQ_EN_REG**, when the counting time of HSTimer reaches, the corresponding interrupt generates.
2. After entering the interrupt process, write **HS_TMR_IRQ_STAS_REG** to clear the interrupt pending.
3. Resume the interrupt and continue to execute the interrupted process.

3.7.4 Programming Guidelines

The following example shows how to make a 1 us delay with HSTimer0. The frequency of the AHB0 clock is 100 MHz, the operating mode is the normal mode, the timing mode is single counting mode, and the pre-scale is 2.

```
writel(0x32, HS_TMR0_INTV_LO);                  //Set bit[31:0] of the interval value as 0x32.
writel(0x0, HS_TMR0_INTV_HI);                   //Set bit[55:32] of the interval value as 0x0.
writel(0x90, HS_TMR0_CTRL);
//Set the operating mode as Normal Mode, the pre-scale as 2, and the timing mode as Single Counting Mode.
writel(readl(HS_TMR0_CTRL)|(1<<1), HS_TMR0_CTRL);   //Set the reload bit.
writel(readl(HS_TMR0_CTRL)|(1<<0), HS_TMR0_CTRL);   //Enable HSTimer0.
while(!(readl(HS_TMR_IRQ_STAS)&1));             //Wait for HSTimer0 to generate pending.
writel(1, HS_TMR_IRQ_STAS);                     //Clear HSTimer0 pending.
```

3.7.5 Register List

Module Name	Base Address
HSTimer	0x03008000

Register Name	Offset	Description
HS_TMR_IRQ_EN_REG	0x0000	HS Timer IRQ Enable Register
HS_TMR_IRQ_STAS_REG	0x0004	HS Timer Status Register
HS_TMR0_CTRL_REG	0x0020	HS Timer0 Control Register
HS_TMR0_INTV_LO_REG	0x0024	HS Timer0 Interval Value Low Register
HS_TMR0_INTV_HI_REG	0x0028	HS Timer0 Interval Value High Register
HS_TMR0_CURNT_LO_REG	0x002C	HS Timer0 Current Value Low Register
HS_TMR0_CURNT_HI_REG	0x0030	HS Timer0 Current Value High Register
HS_TMR1_CTRL_REG	0x0040	HS Timer1 Control Register
HS_TMR1_INTV_LO_REG	0x0044	HS Timer1 Interval Value Low Register
HS_TMR1_INTV_HI_REG	0x0048	HS Timer1 Interval Value High Register
HS_TMR1_CURNT_LO_REG	0x004C	HS Timer1 Current Value Low Register
HS_TMR1_CURNT_HI_REG	0x0050	HS Timer1 Current Value High Register

3.7.6 Register Description

3.7.6.1 0x0000 HS Timer IRQ Enable Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: HS_TMR_IRQ_EN_REG
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W	0x0	HS_TMR1_INT_EN HSTimer1 Interrupt Enable 0: Disabled 1: Enabled
0	R/W	0x0	HS_TMR0_INT_EN HSTimer0 Interrupt Enable 0: Disabled 1: Enabled

3.7.6.2 0x0004 HS Timer IRQ Status Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: HS_TMR_IRQ_STAS_REG
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W1C	0x0	HS_TMR1_IRQ_PEND HSTimer1 IRQ Pending The IRQ pending bit for HSTimer1. Write 1 to clear the pending status. 0: No effect 1: Pending, indicates that the initial value of the HSTimer is reached.
0	R/W1C	0x0	HS_TMR0_IRQ_PEND HSTimer0 IRQ Pending The IRQ pending bit for HSTimer0. Write 1 to clear the pending status. 0: No effect 1: Pending, indicates that the initial value of the HSTimer is reached.

3.7.6.3 0x0020 HS Timer0 Control Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: HS_TMR0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	HS_TMR0_TEST Select the operating mode for HSTimer0 0: Normal mode 1: Test mode In the test mode, the **HS_TMR0_INTV_LO_REG** must be set to 0x1, and **HS_TMR0_INTV_HI_REG** acts as the initial value for HSTimer0.
30:8	/	/	/
7	R/W	0x0	HS_TMR0_MODE Select the timing mode for HSTimer0 0: Periodic mode. When the count value is decreased to 0, the timer will restart another round of counting automatically. 1: One-shot mode. When the count value is decreased to 0, the timer will stop counting.
6:4	R/W	0x0	HS_TMR0_CLK Select the pre-scale for the HSTimer0 clock sources 000: /1 001: /2 010: /4 011: /8 100: /16 101: / 110: / 111: /
3:2	/	/	/
1	R/W1S	0x0	HS_TMR0_RELOAD HSTimer0 Reload 0: No effect 1: Reload the interval value of the HSTimer0

Offset: 0x0020			Register Name: HS_TMR0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	HS_TMR0_EN HSTimer0 Enable 0: Stop/Pause 1: Start By setting the bit to 1, the timer will be started. It reloads the interval value register and then counts from the interval value to 0. By setting the bit to 0 before the timer counts to 0, the timer will be paused. The bit will be locked to 0 for at least 2 cycles. Within the 2 cycles, you cannot set the bit to 1 to restart the timer. The timer supports updating the interval value in the pause state. To start from the updated interval value, set both the reload bit and enable bit to 1. Additionally, in the one-shot mode, after the count value reaches 0, the system will automatically change the bit to 0 to stop the timer.

3.7.6.4 0x0024 HS Timer0 Interval Value Lo Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: HS_TMR0_INTV_LO_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	HS_TMR0_INTV_VALUE_LO Bit[31:0] of the HSTimer0 interval value.

3.7.6.5 0x0028 HS Timer0 Interval Value Hi Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: HS_TMR0_INTV_HI_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:0	R/W	0x0	HS_TMR0_INTV_VALUE_HI Bit[55:32] of the HSTimer0 interval value.

NOTE

HSTimer0 is a 56-bit counter. The interval value consists of two parts: HS_TMR0_INTV_VALUE_LO acts as the bit[31:0] and HS_TMR0_INTV_VALUE_HI acts as the bit[55:32]. To read or write the interval value, HS_TMR0_INTV_LO_REG should be done before HS_TMR0_INTV_HI_REG.

3.7.6.6 0x002C HS Timer0 Current Value Lo Register (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: HS_TMR0_CURNT_LO_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	HS_TMR0_CUR_VALUE_LO Bit[31:0] of the HSTimer0 current value.

3.7.6.7 0x0030 HS Timer0 Current Value Hi Register (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: HS_TMR0_CURNT_HI_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:0	R/W	0x0	HS_TMR0_CUR_VALUE_HI Bit[55:32] of the HSTimer0 current value.

NOTE

HSTimer0 is a 56-bit counter. The current value consists of two parts: HS_TMR0_CUR_VALUE_LO acts as the bit[31:0] and HS_TMR0_CUR_VALUE_HI acts as the bit[55:32]. To read or write the current value, HS_TMR0_CUR_VALUE_LO should be done before HS_TMR0_CUR_VALUE_HI.

3.7.6.8 0x0040 HS Timer1 Control Register (Default Value: 0x0000_0000)

Offset: 0x0040			Register Name: HS_TMR1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	HS_TMR1_TEST Select the operating mode for HSTimer1. 0: Normal mode 1: Test mode In the test mode, the **HS_TMR1_INTV_LO_REG** must be set to 0x1, and **HS_TMR1_INTV_HI_REG** acts as the interval value for HSTimer1.
30:8	/	/	/
7	R/W	0x0	HS_TMR1_MODE Select the timing mode for HSTimer1. 0: Periodic mode. When the count value is decreased to 0, the timer will restart another round of counting automatically. 1: One-shot mode. When the count value is decreased to 0, the timer will stop counting.
6:4	R/W	0x0	HS_TMR1_CLK Select the pre-scale of the HSTimer1 clock sources. 000: /1 001: /2 010: /4 011: /8 100: /16 101: / 110: / 111: /
3:2	/	/	/
1	R/W1S	0x0	HS_TMR1_RELOAD HSTimer1 Reload 0: No effect 1: Reload the HSTimer1 interval value.

Offset: 0x0040			Register Name: HS_TMR1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	HS_TMR1_EN HSTimer1 Enable 0: Stop/Pause 1: Start By setting the bit to 1, the timer will be started. It reloads the interval value register and then counts from the interval value to 0. By setting the bit to 0 before the timer counts to 0, the timer will be paused. The bit will be locked to 0 for at least 2 cycles. Within the 2 cycles, you cannot set the bit to 1 to restart the timer. The timer supports updating the interval value in the pause state. To start from the updated interval value, set both the reload bit and enable bit to 1. Additionally, in the one-shot mode, after the count value reaches 0, the system will automatically change the bit to 0 to stop the timer.

3.7.6.9 0x0044 HS Timer1 Interval Value Lo Register (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: HS_TMR1_INTV_LO_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	HS_TMR1_INTV_VALUE_LO Bit[31:0] of the HSTimer1 interval value

3.7.6.10 0x0048 HS Timer1 Interval Value Hi Register (Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: HS_TMR1_INTV_HI_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:0	R/W	0x0	HS_TMR1_INTV_VALUE_HI Bit[55:32] of the HSTimer1 interval value

NOTE

HSTimer1 is a 56-bit counter. The interval value consists of two parts: HS_TMR1_INTV_VALUE_LO acts as the bit[31:0] and HS_TMR1_INTV_VALUE_HI acts as the bit[55:32]. To read or write the interval value, HS_TMR1_INTV_LO_REG should be done before HS_TMR1_INTV_HI_REG.

3.7.6.11 0x004C HS Timer1 Current Value Lo Register (Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: HS_TMR1_CURNT_LO_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	HS_TMR1_CUR_VALUE_LO Bit[31:0] of the HSTimer1 current value

3.7.6.12 0x0050 HS Timer1 Current Value Hi Register (Default Value: 0x0000_0000)

Offset: 0x0050			Register Name: HS_TMR1_CURNT_HI_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:0	R/W	0x0	HS_TMR1_CUR_VALUE_HI Bit[55:32] of the HSTimer1 current value

NOTE

HSTimer1 is a 56-bit counter. The current value consists of two parts: HS_TMR1_CUR_VALUE_LO acts as the bit[31:0] and HS_TMR1_CUR_VALUE_HI acts as the bit[55:32]. To read or write the current value, HS_TMR1_CUR_VALUE_LO should be done before HS_TMR1_CUR_VALUE_HI.

3.8 Generic Interrupt Controller (GIC)

3.8.1 Overview

The GIC-400 is a high-performance, area-optimized interrupt controller with an Advanced Microcontroller Bus Architecture (AMBA) Advanced eXtensible Interface (AXI) interface. It detects, manages, and distributes interrupts in System on Chip (SoC) configurations. You can configure the GIC-400 to provide the optimum features, performance, and gate count required for your intended application.

For details about GIC, please refer to the *GIC PL400 technical reference* manual and *ARM GIC Architecture Specification V2.0*.

3.8.2 Functional Description

The following table describes the details of interrupt sources.

CAUTION

The CPU interrupt seen by DSP is subtracted 32 from the following interrupt number. For example, the interrupt number of MSGBOX is 32, then the interrupt number seen by DSP is 0.

Table 3-11 Interrupt Sources

Interrupt Number	Interrupt Source	Interrupt Vector	Description
0	SGI 0	0x0000	SGI 0 interrupt
1	SGI 1	0x0004	SGI 1 interrupt
2	SGI 2	0x0008	SGI 2 interrupt
3	SGI 3	0x000C	SGI 3 interrupt
4	SGI 4	0x0010	SGI 4 interrupt
5	SGI 5	0x0014	SGI 5 interrupt
6	SGI 6	0x0018	SGI 6 interrupt
7	SGI 7	0x001C	SGI 7 interrupt
8	SGI 8	0x0020	SGI 8 interrupt
9	SGI 9	0x0024	SGI 9 interrupt
10	SGI 10	0x0028	SGI 10 interrupt
11	SGI 11	0x002C	SGI 11 interrupt

Interrupt Number	Interrupt Source	Interrupt Vector	Description
12	SGI 12	0x0030	SGI 12 interrupt
13	SGI 13	0x0034	SGI 13 interrupt
14	SGI 14	0x0038	SGI 14 interrupt
15	SGI 15	0x003C	SGI 15 interrupt
16	PPI 0	0x0040	PPI 0 interrupt
17	PPI 1	0x0044	PPI 1 interrupt
18	PPI 2	0x0048	PPI 2 interrupt
19	PPI 3	0x004C	PPI 3 interrupt
20	PPI 4	0x0050	PPI 4 interrupt
21	PPI 5	0x0054	PPI 5 interrupt
22	PPI 6	0x0058	PPI 6 interrupt
23	PPI 7	0x005C	PPI 7 interrupt
24	PPI 8	0x0060	PPI 8 interrupt
25	PPI 9	0x0064	PPI 9 interrupt
26	PPI 10	0x0068	PPI 10 interrupt
27	PPI 11	0x006C	PPI 11 interrupt
28	PPI 12	0x0070	PPI 12 interrupt
29	PPI 13	0x0074	PPI 13 interrupt
30	PPI 14	0x0078	PPI 14 interrupt
31	PPI 15	0x007C	PPI 15 interrupt
32	CPUX_MSGBOX_R	0x0080	CPUX MSGBOX READ IRQ
33		0x0084	
34	UART0	0x0088	
35	UART1	0x008C	
36	UART2	0x0090	
37	UART3	0x0094	
38	UART4	0x0098	
39	UART5	0x009C	
40		0x00A0	
41	TWI0	0x00A4	
42	TWI1	0x00A8	
43	TWI2	0x00AC	
44	TWI3	0x00B0	
45		0x00B4	

Interrupt Number	Interrupt Source	Interrupt Vector	Description
46		0x00B8	
47	SPI0	0x00BC	
48	SPI1	0x00C0	
49		0x00C4	
50	PWM	0x00C8	
51	CIR_TX	0x00CC	
52	LEDC	0x00D0	
53		0x00D4	
54		0x00D8	
55	OWA	0x00DC	
56	DMIC	0x00E0	
57	AUDIO_CODEC	0x00E4	
58		0x00E8	
59	I2S/PCM1	0x00EC	
60	I2S/PCM2	0x00F0	
61	USB0_DEVICE	0x00F4	
62	USB0_EHCI	0x00F8	
63	USB0_OHCI	0x00FC	
64		0x0100	
65	USB1_EHCI	0x0104	
66	USB1_OHCI	0x0108	
67		0x010C	
68		0x0110	
69		0x0114	
70		0x0118	
71		0x011C	
72	SMHC0	0x0120	
73	SMHC1	0x0124	
74	SMHC2	0x0128	
75	MSI	0x012C	
76	SMC	0x0130	
77		0x0134	
78	EMAC	0x0138	
79	TZMA_ERR	0x013C	SRAM_A1 TZMA IRQ

Interrupt Number	Interrupt Source	Interrupt Vector	Description
80	CCU_FERR	0x0140	
81	AHB_HREADY_TIME_OUT	0x0144	SYS_CTRL ahb_hready time out
82	DMAC_NS	0x0148	
83	DMAC_S	0x014C	
84	CE_NS	0x0150	
85	CE_S	0x0154	
86	SPINLOCK	0x0158	
87	HSTIMER0	0x015C	
88	HSTIMER1	0x0160	
89	GPADC	0x0164	
90	THS	0x0168	Thermal Sensor IRQ
91	TIMER0	0x016C	
92	TIMER1	0x0170	
93		0x0174	
94	TPADC	0x0178	
95	WATCHDOG	0x017C	
96	IOMMU	0x0180	
97		0x0184	
98	VE	0x0188	
99		0x018C	
100		0x0190	
101	GPIOB_NS	0x0194	
102	GPIOB_S	0x0198	
103	GPIOC_NS	0x019C	
104	GPIOC_S	0x01A0	
105	GPIOD_NS	0x01A4	
106	GPIOD_S	0x01A8	
107	GPIOE_NS	0x01AC	
108	GPIOE_S	0x01B0	
109	GPIOF_NS	0x01B4	
110	GPIOF_S	0x01B8	
111	GPIOG_NS	0x01BC	
112	GPIOG_S	0x01C0	

Interrupt Number	Interrupt Source	Interrupt Vector	Description
113		0x01C4	
114		0x01C8	
115		0x01CC	
116		0x01D0	
117	CPUX_MSGBOX_DSP_W	0x01D4	CPUX MSGBOX WRITE IRQ for DSP
118		0x01D8	
119	DE	0x01DC	
120	DI	0x01E0	
121	G2D	0x01E4	
122	LCD	0x01E8	
123	TV	0x01EC	
124	DSI	0x01F0	
125		0x01F4	
126	TVE	0x01F8	CVBS OUT interrupt
127	CSI_DMA0	0x01FC	
128	CSI_DMA1	0x0200	
129		0x0204	
130		0x0208	
131		0x020C	
132	CSI_PARSER0	0x0210	
133		0x0214	
134		0x0218	
135		0x021C	
136		0x0220	
137		0x0224	
138	CSI_TOP_PKT	0x0228	
139	TVD	0x022C	CVBS IN interrupt
140		0x0230	
141		0x0234	
142		0x0238	
143		0x023C	
144		0x0240	
145		0x0244	

Interrupt Number	Interrupt Source	Interrupt Vector	Description
146		0x0248	
147		0x024C	
148		0x0250	
149		0x0254	
150		0x0258	
151		0x025C	
152	DSP_DEE	0x0260	DSP_SYS[0]: DoubleExceptionError
153	DSP_PFE	0x0264	DSP_SYS[1]: PFatalError
154	DSP_WDG	0x0268	
155	DSP_MBOX_CPUX_W	0x026C	
156		0x0270	
157	DSP_TZMA	0x0274	
158		0x0278	
159		0x027C	
160		0x0280	
161		0x0284	
162		0x0288	
163		0x028C	
164		0x0290	
165		0x0294	
166		0x0298	
167		0x029C	
CPUS Related			
168	NMI	0x02A0	
169	PPU	0x02A4	
170	TWD	0x02A8	
171		0x02AC	
172	TIMER0	0x02B0	
173	TIMER1	0x02B4	
174	TIMER2	0x02B8	
175	TIMER3	0x02BC	
176	ALARM0	0x02C0	
177		0x02C4	

Interrupt Number	Interrupt Source	Interrupt Vector	Description
178		0x02C8	
179		0x02CC	
180		0x02D0	
181		0x02D4	
182		0x02D8	
183	IRRX	0x02DC	CPUS_IRRX_IRQ[9:0] OR to 1-bit in syscpus top
184		0x02E0	
185		0x02E4	
186	AHBS_HREADY_TIME_OUT	0x02EC	CPUS AHB READY TIME OUT IRQ
187		0x02F0	
188		0x02F4	
189		0x02F8	
190		0x02FC	
191		0x0300	
CPUX Related			
192	C0_CTI0	0x0304	C0_CTI0 interrupt
193	C0_CTI1	0x0308	C0_CTI1 interrupt
194		0x030C	
195		0x0310	
196	C0_COMMTX0	0x0314	C0_COMMTX0 interrupt
197	C0_COMMTX1	0x0318	C0_COMMTX1 interrupt
198		0x031C	
199		0x0320	
200	C0_COMMRX0	0x0324	C0_COMMRX0 interrupt
201	C0_COMMRX1	0x0328	C0_COMMRX1 interrupt
202		0x032C	
203		0x0330	
204	C0_PMU0	0x0334	C0_PMU0 interrupt
205	C0_PMU1	0x0338	C0_PMU1 interrupt
206		0x033C	
207		0x0340	
208	C0_AXI_ERROR	0x0344	C0_AXI_ERROR interrupt

Interrupt Number	Interrupt Source	Interrupt Vector	Description
209		0x0348	
210	AXI_WR_IRQ	0x034C	
211	AXI_RD_IRQ	0x0350	
212	DBGPWRUPREQ_out[0]	0x0354	
213	DBGPWRUPREQ_out[1]	0x0358	
214		0x035C	
215		0x0360	
216		0x0364	
217		0x0368	
218		0x036C	
219		0x0370	
220		0x0374	
221		0x0378	
222		0x037C	
223		0x0380	

3.9 Direct Memory Access Controller (DMAC)

3.9.1 Overview

The direct memory access (DMA) is a method of transferring data between peripherals and memories (including the SRAM and DRAM) without using the CPU. It is an efficient way to offload data transfer duties from the CPU. Without DMA, the CPU has to control all the data transfers. While with DMA, the DMAC directly transfers data between a peripheral and a memory, between peripherals, or between memories.

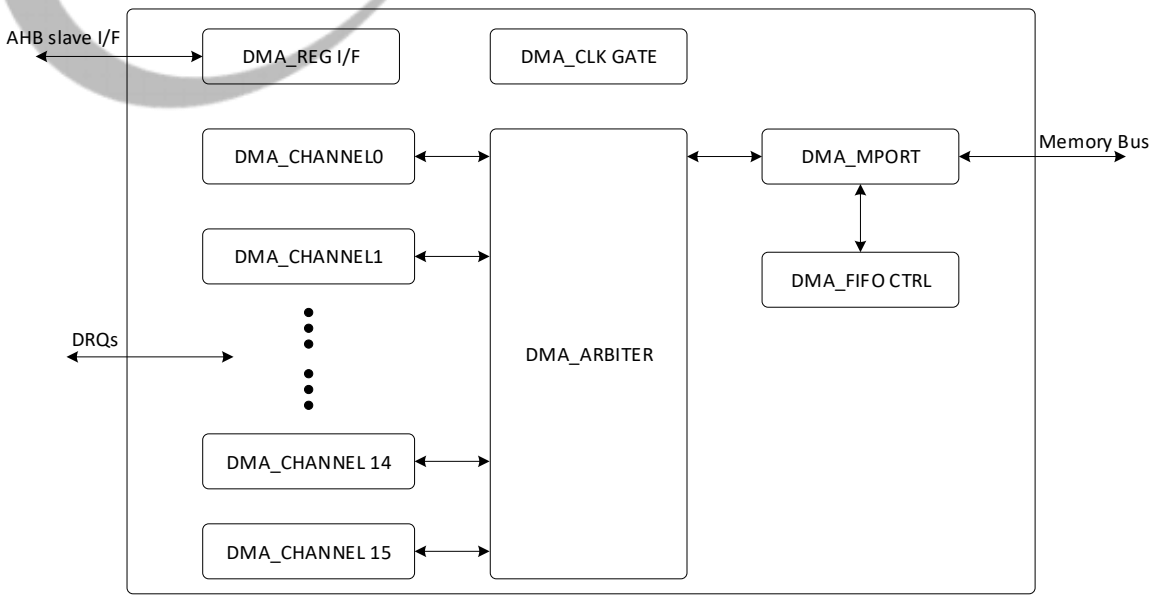
The DMAC has the following features:

- Up to 16 DMA channels
- Provides 32 peripheral DMA requests for data reading and 32 peripheral DMA requests for data writing
- Supports transferring data with a linked list
- Supports programmable 8-bit, 16-bit, 32-bit, and 64-bit data width
- Supports programmable DMA burst length
- DRQ response includes the waiting mode and handshake mode
- DMA channel supports pause function
- Memory devices support non-aligned transform

3.9.2 Block Diagram

The following figure shows a block diagram of DMAC.

Figure 3-19 DMAC Block Diagram



DMAC contains the following sub-blocks:

Table 3-12 DMAC Sub-blocks

Sub-block	Description
DMA_ARBITER	Arbitrates the DMA read/write requests from all channels, and converts the requests to the read/write requests of ports.
DMA_CHANNELs	DMA transfer engine. Each channel is independent. When the DMA requests from multiple peripherals are valid simultaneously, the channel with the highest priority starts data transfer first. The system uses the polling mechanism to decide the priorities of DMA channels. When DMA_ARBITER is idle, channel 0 has the highest priority, whereas channel 15 has the lowest priority. When DMA_ARBITER is busy processing the request from channel n, channel (n+1) has the highest priority. For n = 15, the channel (n + 1) should be channel 0.
DRQs	DMA requests. Peripherals use the DMA request signals to request a data transfer.
DMA_MPORT	Receives the read/write requests from DMA_ARBITER, and converts the requests to the corresponding MBUS access requests. It is mainly used for accessing the DRAM.
DMA_HPORT	The port for accessing the AHB Master. It is mainly used for accessing the SRAM and IO devices.
DMA_FIFO CTRL	Internal FIFO cell control module.
DMA_REG Interface	DMA_REG is the common register module that is mainly used to resolve AHB demands.
DMA_CLKGATE	The control module for hardware auto clock gating.

The DMAC integrates 16 independent DMA channels and each channel has an independent FIFO controller. When the DMA channel starts, the DMAC gets a DMA descriptor from the DMA_DESC_ADDR_REG and uses it as the configuration information for the data transfer of the current DMA package. Then the DMAC can transfer data between the specified devices. After transferring a DMA package, the DMAC judges if the current channel transfer is finished via the linked address in the descriptor. If the linked address shows all the packages are transferred, the DMAC will end the chain transmission and close the channel.

3.9.3 Functional Description

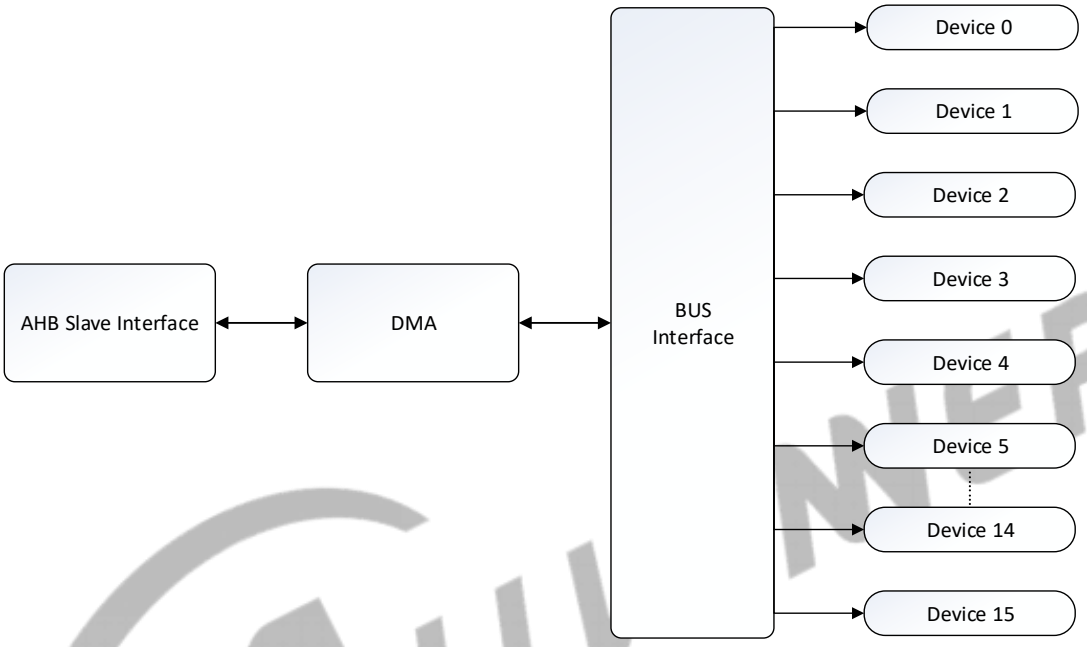
3.9.3.1 Clock

The DMAC is on MBUS. The clock of MBUS influences the transfer efficiency of the DMAC.

3.9.3.2 Typical Application

The following figure shows a typical application of the DMAC.

Figure 3-20 DMAC Typical Application Diagram



3.9.3.3 DRQ Port of Peripherals

The following table shows the source DRQ types and destination DRQ types of different ports.

Table 3-13 DMA DRQ Type

Source DRQ Type		Destination DRQ Type	
port0	SRAM	port0	SRAM
port1	DRAM	port1	DRAM
port2	OWA-RX	port2	OWA-TX
port3		port3	
port4	I2S/PCM1-RX	port4	I2S/PCM1-TX
port5	I2S/PCM2-RX	port5	I2S/PCM2-TX
port6		port6	
port7	Audio_Codec	port7	Audio_Codec
port8	DMIC	port8	

Source DRQ Type		Destination DRQ Type	
port9		port9	
port10		port10	
port11		port11	
port12	GPADC	port12	
port13	TPADC	port13	IR-TX
port14	UART0-RX	port14	UART0-TX
port15	UART1-RX	port15	UART1-TX
port16	UART2-RX	port16	UART2-TX
port17	UART3-RX	port17	UART3-TX
port18	UART4-RX	port18	UART4-TX
port19	UART5-RX	port19	UART5-TX
port20		port20	
port21		port21	
port22	SPI0-RX	port22	SPI0-TX
port23	SPI1-RX	port23	SPI1-TX
port24		port24	
port25		port25	
port26		port26	
port27		port27	
port28		port28	
port29		port29	
Port30	USB0_EP1	Port30	USB0_EP1
Port31	USB0_EP2	Port31	USB0_EP2
Port32	USB0_EP3	Port32	USB0_EP3
Port33	USB0_EP4	Port33	USB0_EP4
Port34	USB0_EP5	Port34	USB0_EP5
Port35		Port35	
Port36		Port36	
Port37		Port37	
Port38		Port38	
Port39		Port39	
Port40		Port40	
Port41		Port41	
Port42		Port42	LEDC

Source DRQ Type		Destination DRQ Type	
Port43	TWI0	Port43	TWI0
Port44	TWI1	Port44	TWI1
Port45	TWI2	Port45	TWI2
Port46	TWI3	Port46	TWI3
Port47		Port47	
Port48		Port48	
Port49		Port49	
Port50		Port50	
Port51		Port51	
Port52		Port52	
Port53		Port53	

3.9.3.4 DMA Descriptor

The DMAC descriptor is the configuration information of DMA transfer that decides the DMA working mode. Each descriptor includes 6 words: Configuration, Source Address, Destination Address, Byte Counter, Parameter, and Link. The following figure shows the structure of the DMA descriptor.

Figure 3-21 DMA Descriptor

- **Configuration: Configure the following information by DMA_CFG_REG.**
  - DRQ type: DRQ type of the source and destination devices.
  - Address counting mode: For both the source and destination devices, there are two address counting modes: the IO mode and linear mode. The IO mode is for IO devices whose address is fixed during the data transfer and the linear mode is for the memory whose address is increasing during the data transfer.

- Transferred block length: The amount of data that non-memory peripherals can transfer in a valid DRQ. The block length supports 1 bit, 4 bits, 8 bits, and 16 bits.
- Transferred data width: The data width of operating the non-memory peripherals. The data width supports 8 bits, 16 bits, 32 bits, and 64 bits.

NOTE

The configuration supports BMODE mode. The BMODE is used in the following scenario: the source is an IO device, and the destination is a memory device. Setting the BMODE mode can limit the amount of block data transferred in DMA block transmission to the amount of data transferred when the DRQ threshold of the source IO device is 1. For example,

- **Source Address: Configure the address of the source device.**
- **Destination Address: Configure the address of the destination device.**

DMA reads data from the source address and then writes data to the destination address.

Both the DMA source and destination addresses have 34 bits. In the descriptor, because there are only 32 bits in the **Source/Destination Address** field, another 2 bits are stored in the **Parameter** field.

The following table shows the details of the related fields in the descriptor.

Table 3-14 Source/Destination Address Distribution

Descriptor Group	Bit	Description
Source Address	31:0	DMA transfers the lower 32 bits of the 34-bit source address
Destination Address	31:0	DMA transfers the lower 32 bits of the 34-bit destination address
Parameter	31:20	Reserved
	19:18	DMA transfers the higher 2 bits of the 34-bit destination address
	17:16	DMA transfers the high 2 bits of the 34-bit source address
	15:8	Reserved
	7:0	Wait Clock Cycles Set the waiting time in DRQ mode
Link	31:2	The address of the next group descriptor, the lower 30 bits of the word address
	1:0	The address of the next group descriptor, the higher 2 bits of the word address

From the above table, you can get:

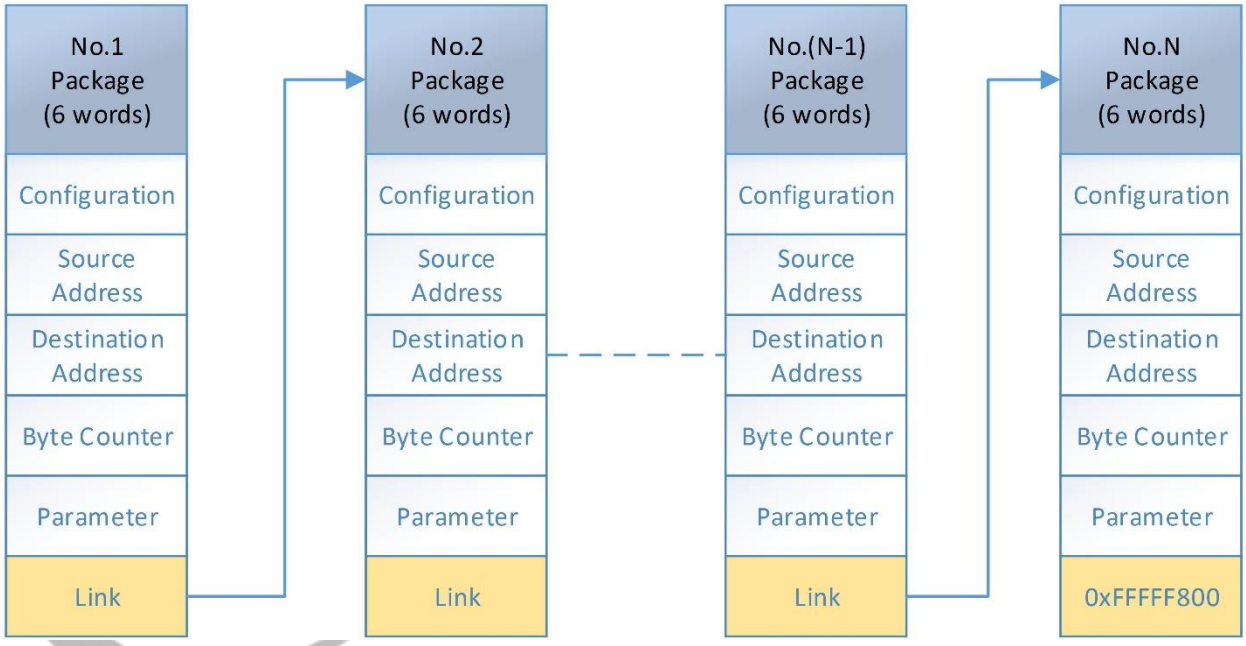
Real DMA source address (in byte mode) = {Parameter [17:16], Source Address [31:0]};

Real DMA destination address (in byte mode) = {Parameter [19:18], Destination Address [31:0]};

Real link address (in byte mode) = {Link[1:0], Link[31:2], 2'b00}.

- **Byte counter: Configure the data amount of a package. The maximum value is (2^25-1) bytes. If the data amount of the package reaches the maximum value, even if DRQ is valid, the DMA will stop the current transfer.**
- **Parameter: Configure the interval between the data block. The parameter is valid for non-memory peripherals. When DMA detects that the DRQ is high, the DMA transfers the data block and ignores the status changes of the DRQ until the data transfer finishes. After that, the DMA waits for certain clock cycles (WAIT_CYC) and executes the next DRQ detection.**
- **Link: If the value of the link is 0xFFFFF800, the current package is at the end of the linked list. The DMAC will stop the data transfer after transferring the package; otherwise, the value of the link is considered as the descriptor address of the next package.**

Figure 3-22 DMA Chain Transfer



3.9.3.5 Interrupts

There are three kinds of DMA interrupts: the half package interrupt, package end interrupt, and queue end interrupt.

Half package interrupt: When enabled, the DMAC sends out a half package interrupt after transferring half of a package.

Package end interrupt: When enabled, the DMAC sends out a package end interrupt after transferring a complete package.

Queue end interrupt: When enabled, the DMAC sends out a queue end interrupt after transferring a complete queue.

Notice that when CPU does not respond to the interrupts timely, or two DMA interrupts are generated very closely, the later interrupt may override the former one. That is, from the perspective of the CPU, the DMAC has only a system interrupt source.

NOTE

The DMAC has 16 channels and 2 groups of interrupts. The channel [7:0] corresponds to one group of interrupt, the channel [15:8] corresponds to anther group of interrupt. The DSP is fixed to use the interrupt of the channel [15:8].

3.9.3.6 Clock Gating

The DMA_CLK_GATE module is a hardware module for controlling the clock gating automatically. It provides clock sources for sub-modules in DMAC and the module local circuits.

The DMA_CLK_GATE module consists of two parts: the channel clock gate and the common clock gate.

Channel clock gate: Controls the DMA clock of the DMA channels. When the system accesses the register of the current DMA channel and the DMA channel is enabled, the channel clock gate automatically opens the DMA clock. With a 16-HCLK-cycle delay after the system finishes accessing the register or the DMA data transfer is completed, the channel clock gate automatically closes the DMA clock. Also, the clock for the related circuits, such as for the channel control and FIFO control modules, will be closed.

Common clock gate: Controls the clocks of the DMA common circuits. The common circuits include the common circuit of the FIFO control module, MPORT module, and MBUS. When all the DMA channels are enabled, the common clock gate automatically closes the clocks for the above circuits.

The DMA clock gating can support all the functions stated above or not by software.

3.9.3.7 Transfer Mode

The peripherals initiate data transfer by transmitting DMA request signals to the DMAC. After receiving the request signal, the DMAC converts it to the internal DRQ signal and controls the DMA data transfer.

The DMAC supports two data transfer modes: the waiting mode and handshake mode.

The principle of waiting mode

- When the DMAC detects a valid external request signal, the DMAC starts to operate the peripheral device. The internal DRQ always holds high before the transferred data amount reaches the transferred block length.
- When the transferred data amount reaches the transferred block length, the internal DRQ pulls low automatically.
- The internal DRQ holds low for certain clock cycles (WAIT_CYC), and then the DMAC restarts to detect the external requests. If the external request signal is valid, then the next transfer starts.

The principle of handshake mode

- When the DMAC detects a valid external request signal, the DMAC starts to operate the peripheral device. The internal DRQ always holds high before the transferred data amount reaches the transferred block length.
- When the transferred data amount reaches the transferred block length, the internal DRQ will be pulled down automatically. For the last data transfer of the block, the DMAC sends a DMA Last signal with the DMA commands to the peripheral device. The DMA Last signal will be packed as part of the DMA commands and transmitted on the bus. It is used to inform the peripheral device that it is the end of the data transfer for the current DRQ.
- When the peripheral device receives the DMA Last signal, it can judge that the data transfer for the current DRQ is finished. To continue the data transfer, it sends a DMA Active signal to the DMAC.

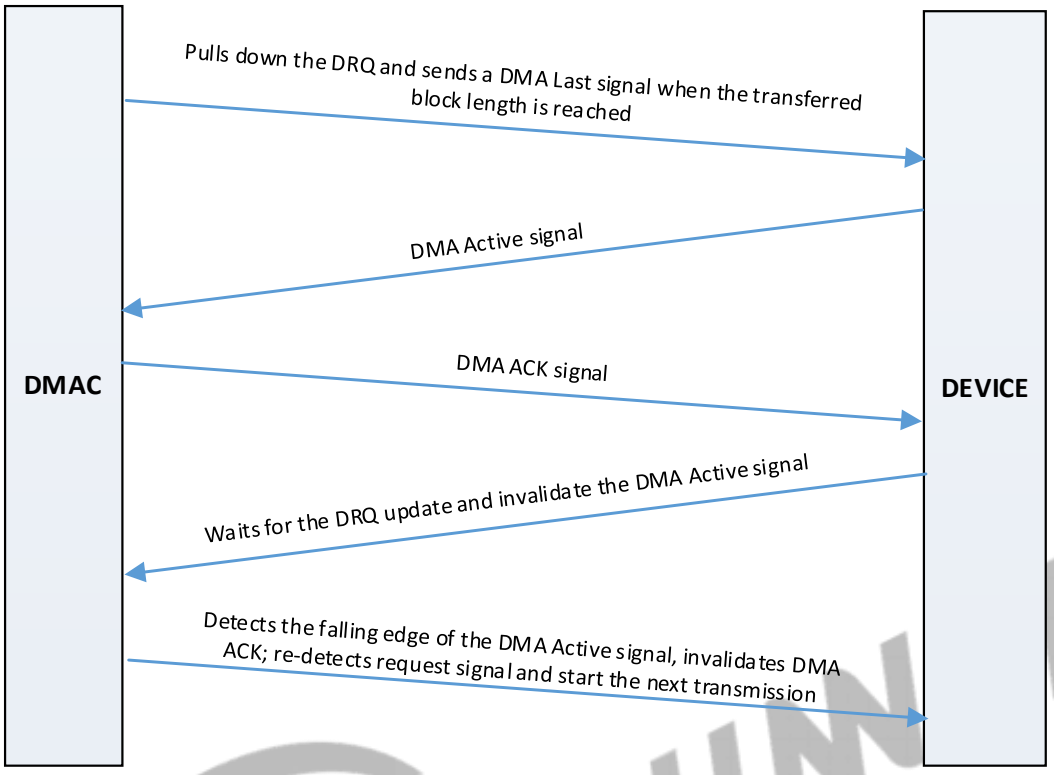
NOTE

One DMA Active signal will be converted to one DRQ signal in the DMA module. To generate multiple DRQs, the peripheral device needs to send out multiple DMA Active signals via the bus protocol.

- When the DMAC received the DMA Active signal, it sends back a DMA ACK signal to the peripheral device.
- When the peripheral device receives the DMA ACK signal, it waits for all the operations on the local device completed, and both the FIFO and DRQ status refreshed. Then it invalidates the DMA Active signal.
- When the DMAC detects the falling edge of the DMA Active signal, it invalidates the corresponding DMA ACK signal, and restarts to detect the external request signals. If a valid request signal is detected, the next data transfer starts.

The following figure shows the workflow of the handshake mode.

Figure 3-23 Workflow of the DMAC Handshake Mode



3.9.3.8 Address Auto-Alignment

For the non-IO devices whose start address is not 32-byte-aligned, the DMAC will adjust the address to 32-byte-aligned through the burst transfer within 32 bytes. Adjusting address to 32-byte-aligned improves the DRAM access efficiency.

The following example shows how the DMAC adjusts the address: when the peripheral device of a DMA channel is a non-IO device whose start address is 0x86 (not 32-byte-aligned), the DMAC firstly uses a 26-byte burst transfer to align the address to 0xA0 (32-byte-aligned), and then transfers data by 64-byte burst (the maximum transfer amount that MBUS allows).

The IO devices do not support address alignment, so the bit width of IO devices must match the address offset; otherwise, the DMAC will ignore the inconsistency and directly transmit data of the corresponding bit width to the address.

The address of the DMA descriptor does not support the address auto-alignment. Make sure the address is word-aligned; otherwise the DMAC cannot identify the descriptor.

3.9.3.9 DMAC Clock Control

- The DMAC clock is synchronous with the AHB0 clock. Make sure that the DMAC gating bit of AHB0 clock is enabled before accessing the DMAC register.
- The reset input signal of the DMAC is asynchronous with AHB0 and is low valid by default. Make sure that the reset signal of the DMAC is de-asserted before accessing the DMA register.
- To avoid the indefinite state within registers, de-assert the reset signal first, and then open the gating bit of AHB0.
- The DMAC supports Clock Auto Gating function to reduce power consumption, the system will automatically disable the DMAC clock in the DMAC idle state. Clock Auto Gating is enabled by default.

3.9.4 Programming Guidelines

3.9.4.1 Using DMAC Transfer Process

The DMAC transfer process is as follows.

Step 1 Request DMA channel, and check if the DMA channel is idle by checking if it is enabled. A disabled channel indicates it is idle, while an enabled channel indicates it is busy.

Step 2 Write the descriptor with 6 words into the memory. The descriptor must be word-aligned. For more details, refer to section 3.9.3.4 "DMA Descriptor".

Step 3 Write the start address of the descriptor to **DMA_DESC_ADDR_REGN**.

Step 4 Enable the DMA channel, and write the corresponding channel to **DMAC_EN_REGN**.

Step 5 The DMA obtains the descriptor information.

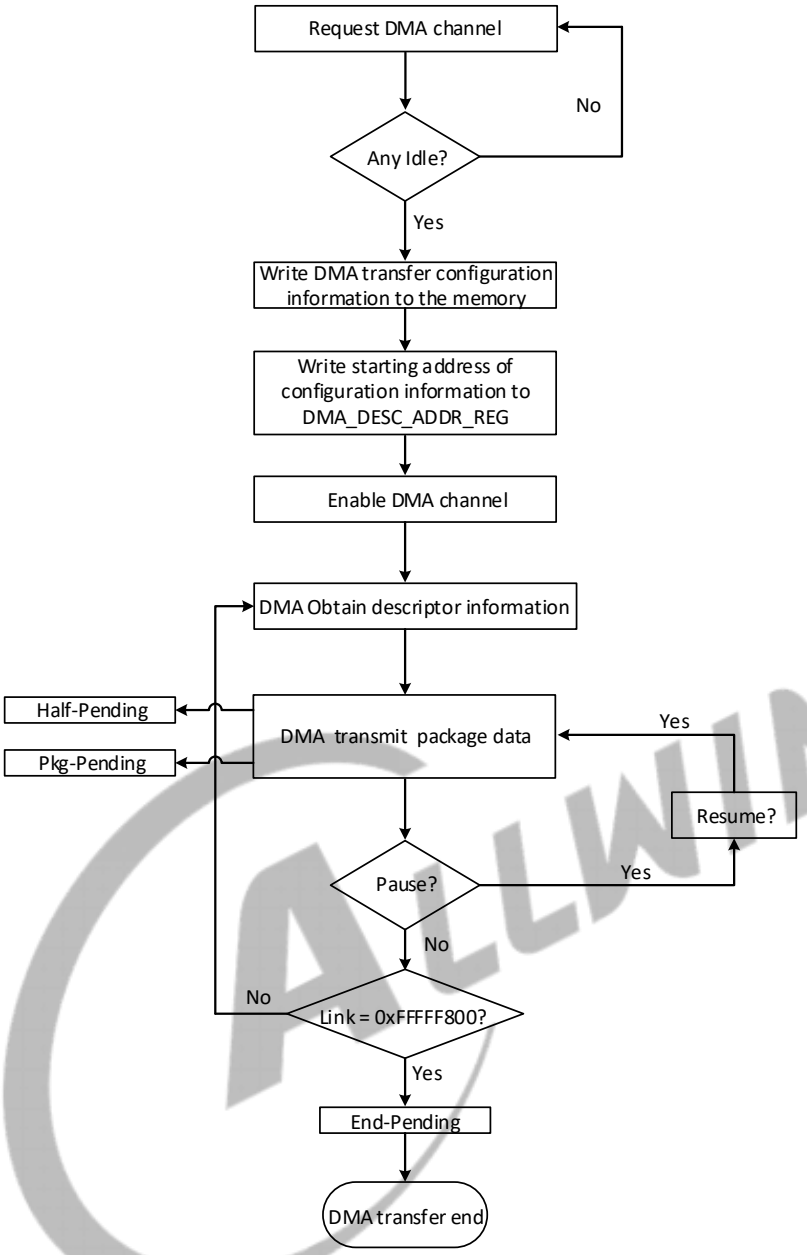
Step 6 Start to transmit a package. When half of the package is completed, the DMA sends a Half Package Transfer Interrupt; when a total package is completed, the DMA sends a Package End Transfer Interrupt. These interrupt status can be read by **DMAC_IRQ_PEND_REG0**.

Step 7 Set **DMAC_PAU_REGN** to pause or resume the data transfer.

Step 8 After completing a total package transfer, the DMA decides to start the next package transfer or end the transfer by the link of the descriptor. If the link is 0xFFFFF800, the transfer ends; otherwise, the next package starts to transmit. When the transfer ends, the DMA sends a Queue End Transfer Interrupt.

Step 9 Disable the DMA channel.

Figure 3-24 DMAC Transfer Process



3.9.4.2 Processing DMAC Interrupt

Follow the steps below to process the DMAC interrupt:

Step 1 Enable interrupt: write the corresponding interrupt enable bit of **DMAC_IRQ_EN_REG0**. The system generates an interrupt when the corresponding condition is satisfied.

Step 2 After entering the interrupt process, write C:\Users\zengjing\Downloads_Hlk52357013_-_Hlk52357103to clear the interrupt pending and execute the process of waiting for the interrupt.

Step 3 Resume the interrupt and continue to execute the interrupted process.

3.9.4.3 Configuring DMAC

To configure the DMAC, follow the guidelines below:

- Make sure the transfer bit width of IO devices is consistent with the offset of the start address.
- The MBUS protocol does not support the read operation of non-integer words. For the devices whose bit width is not word-aligned, after receiving the read command, they should resolve the read command according to their FIFO bit width instead of the command bit width, and ignore the redundant data caused by the inconsistency of the bit width.
- When the DMA transfer is paused, this is equivalent to invalid DRQ. Because there is a certain time delay between DMA transfer commands, the DMAC will not stop data transfer until the DMAC finishes processing the current command and the commands in Arbiter (at most 32 bytes data).

DMAC application example:

```
writel(0x00000000, mem_address + 0x00);   //Set configurations. The mem_address must be word-aligned.
writel(0x00001000, mem_address + 0x04);    // Set the start address for the source device.
writel(0x20000000, mem_address + 0x08);   //Set the start address for the destination device.
writel(0x00000020, mem_address + 0x0C);   // Set the data package size.
writel(0x00000000, mem_address + 0x10);   //Set the parameters.
writel(0xFFFFF800, mem_address + 0x14);   //Set the start address for the next descriptor.
writel(mem_address, 0x01C02000+ 0x100 + 0x08); //Set the start address for the DMA channel0 descriptor.
do{
If(mem_address == readl(0x01C02000 + 0x100 + 0x08));
break;
}while(1);              //Make sure that the writing operation is valid.
writel(0x000000001, 0x01C02000 + 0x100 + 0x00);      // Enable DMA channel0 transfer.
```

The DMAC supports increasing data package in transfer, pay attention to the following points:

- The 0xFFFFF800 value of **DMAC_FDESC_ADDR_REGN** indicates that the DMA channel has got back the descriptor of the last package. The DMA channel will automatically stop the data transfer after transferring the current package.
- To add a package during the data transfer, check if the DMA channel has got back the descriptor of the last package. If yes, you cannot add any package in the current queue. Request another DMA channel with

a new DRQ to transfer the package. Otherwise, you can add the package by modifying the DMAC_FDESC_ADDR_REGN of the last package from 0xFFFFF800 to the start address of the to-be-added package.

To ensure that the modification is valid, read the value of DMAC_FDESC_ADDR_REGN after the modification. The value 0xFFFFF800 indicates the modification fails and the other values indicate you have successfully added packages to the queue.

Another problem is, the system needs some time to process the modification, during which the DMA channel may get back the descriptor of the last package. You can read DMAC_CUR_SRC_REGN and DMAC_CUR_DEST_REGN and check if the increasing memory address accords with the information of the added package. If yes, the package is added successfully; otherwise, the modification failed.

To ensure a higher rate of success, it is suggested that you add the package before the half package interrupt of the penultimate package.

3.9.5 Register List

Module Name	Base Address
DMAC	0x03002000

Register Name	Offset	Description
DMAC_IRQ_EN_REG0	0x0000	DMAC IRQ Enable Register 0
DMAC_IRQ_EN_REG1	0x0004	DMAC IRQ Enable Register 1
DMAC_IRQ_PEND_REG0	0x0010	DMAC IRQ Pending Register 0
DMAC_IRQ_PEND_REG1	0x0014	DMAC IRQ Pending Register 1
DMAC_AUTO_GATE_REG	0x0028	DMAC Auto Gating Register
DMAC_STA_REG	0x0030	DMAC Status Register
DMAC_EN_REGN	0x0100 + N*0x0040	DMAC Channel Enable Register N (N = 0 to 15)
DMAC_PAU_REGN	0x0104 + N*0x0040	DMAC Channel Pause Register N (N = 0 to 15)
DMAC_DESC_ADDR_REGN	0x0108 + N*0x0040	DMAC Channel Start Address Register N (N = 0 to 15)
DMAC_CFG_REGN	0x010C + N*0x0040	DMAC Channel Configuration Register N (N = 0 to 15)
DMAC_CUR_SRC_REGN	0x0110 + N*0x0040	DMAC Channel Current Source Register N (N = 0 to 15)
DMAC_CUR_DEST_REGN	0x0114 + N*0x0040	DMAC Channel Current Destination Register N (N = 0 to 15)
DMAC_BCNT_LEFT_REGN	0x0118 + N*0x0040	DMAC Channel Byte Counter Left Register N (N = 0 to 15)
DMAC_PARA_REGN	0x011C + N*0x0040	DMAC Channel Parameter Register N (N = 0 to 15)

Register Name	Offset	Description
DMAC_MODE_REGN	0x0128 + N*0x0040	DMAC Mode Register N (N = 0 to 15)
DMAC_FDESC_ADDR_REGN	0x012C + N*0x0040	DMAC Former Descriptor Address Register N (N = 0 to 15)
DMAC_PKG_NUM_REGN	0x0130 + N*0x0040	DMAC Package Number Register N (N = 0 to 15)

3.9.6 Register Description

3.9.6.1 0x0000 DMAC IRQ Enable Register0 (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: DMAC_IRQ_EN_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R/W	0x0	DMA7_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 7 0: Disabled 1: Enabled
29	R/W	0x0	DMA7_PKG_IRQ_EN Enable the package end interrupt of DMA channel 7 0: Disabled 1: Enabled
28	R/W	0x0	DMA7_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 7 0: Disabled 1: Enabled
27	/	/	/
26	R/W	0x0	DMA6_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 6 0: Disabled 1: Enabled
25	R/W	0x0	DMA6_PKG_IRQ_EN Enable the package end interrupt of DMA channel 6 0: Disabled 1: Enabled

Offset: 0x0000			Register Name: DMAC_IRQ_EN_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
24	R/W	0x0	DMA6_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 6 0: Disabled 1: Enabled
23	/	/	/
22	R/W	0x0	DMA5_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 5 0: Disabled 1: Enabled
21	R/W	0x0	DMA5_PKG_IRQ_EN Enable the package end interrupt of DMA channel 5 0: Disabled 1: Enabled
20	R/W	0x0	DMA5_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 5 0: Disabled 1: Enabled
19	/	/	/
18	R/W	0x0	DMA4_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 4 0: Disabled 1: Enabled
17	R/W	0x0	DMA4_PKG_IRQ_EN Enable the package end interrupt of DMA channel 4 0: Disabled 1: Enabled
16	R/W	0x0	DMA4_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 4 0: Disabled 1: Enabled
15	/	/	/
14	R/W	0x0	DMA3_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 3 0: Disabled 1: Enabled

Offset: 0x0000			Register Name: DMAC_IRQ_EN_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
13	R/W	0x0	DMA3_PKG_IRQ_EN Enable the package end interrupt of DMA channel 3 0: Disabled 1: Enabled
12	R/W	0x0	DMA3_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 3 0: Disabled 1: Enabled
11	/	/	/
10	R/W	0x0	DMA2_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 2 0: Disabled 1: Enabled
9	R/W	0x0	DMA2_PKG_IRQ_EN Enable the package end interrupt of DMA channel 2 0: Disabled 1: Enabled
8	R/W	0x0	DMA2_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 2 0: Disabled 1: Enabled
7	/	/	/
6	R/W	0x0	DMA1_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 1 0: Disabled 1: Enabled
5	R/W	0x0	DMA1_PKG_IRQ_EN Enable the package end interrupt of DMA channel 1 0: Disabled 1: Enabled
4	R/W	0x0	DMA1_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 1 0: Disabled 1: Enabled
3	/	/	/

Offset: 0x0000			Register Name: DMAC_IRQ_EN_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x0	DMA0_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 0 0: Disabled 1: Enabled
1	R/W	0x0	DMA0_PKG_IRQ_EN Enable the package end interrupt of DMA channel 0 0: Disabled 1: Enabled
0	R/W	0x0	DMA0_HLAF_IRQ_EN Enable the half package interrupt of DMA channel 0 0: Disabled 1: Enabled

3.9.6.2 0x0004 DMAC IRQ Enable Register1 (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: DMAC_IRQ_EN_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R/W	0x0	DMA15_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 15 0: Disabled 1: Enabled
29	R/W	0x0	DMA15_PKG_IRQ_EN Enable the package end interrupt of DMA channel 15 0: Disabled 1: Enabled
28	R/W	0x0	DMA15_HALF_IRQ_EN Enable the half package interrupt of DMA channel 15 0: Disabled 1: Enabled
27	/	/	/
26	R/W	0x0	DMA14_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 14 0: Disabled 1: Enabled

Offset: 0x0004			Register Name: DMAC_IRQ_EN_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
25	R/W	0x0	DMA14_PKG_IRQ_EN Enable the package end interrupt of DMA channel 14 0: Disabled 1: Enabled
24	R/W	0x0	DMA14_HALF_IRQ_EN Enable the half package interrupt of DMA channel 14 0: Disabled 1: Enabled
23	/	/	/
22	R/W	0x0	DMA13_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 13 0: Disabled 1: Enabled
21	R/W	0x0	DMA13_PKG_IRQ_EN Enable the package end interrupt of DMA channel 13 0: Disabled 1: Enabled
20	R/W	0x0	DMA13_HALF_IRQ_EN Enable the half package interrupt of DMA channel 13 0: Disabled 1: Enabled
19	/	/	/
18	R/W	0x0	DMA12_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 12 0: Disabled 1: Enabled
17	R/W	0x0	DMA12_PKG_IRQ_EN Enable the package end interrupt of DMA channel 12 0: Disabled 1: Enabled
16	R/W	0x0	DMA12_HALF_IRQ_EN Enable the half package interrupt of DMA channel 12 0: Disabled 1: Enabled
15	/	/	/

Offset: 0x0004			Register Name: DMAC_IRQ_EN_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
14	R/W	0x0	DMA11_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 11 0: Disabled 1: Enabled
13	R/W	0x0	DMA11_PKG_IRQ_EN Enable the package end interrupt of DMA channel 11 0: Disabled 1: Enabled
12	R/W	0x0	DMA11_HALF_IRQ_EN Enable the half package interrupt of DMA channel 11 0: Disabled 1: Enabled
11	/	/	/
10	R/W	0x0	DMA10_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 10 0: Disabled 1: Enabled
9	R/W	0x0	DMA10_PKG_IRQ_EN Enable the package end interrupt of DMA channel 10 0: Disabled 1: Enabled
8	R/W	0x0	DMA10_HALF_IRQ_EN Enable the half package interrupt of DMA channel 10 0: Disabled 1: Enabled
7	/	/	/
6	R/W	0x0	DMA9_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 9 0: Disabled 1: Enabled
5	R/W	0x0	DMA9_PKG_IRQ_EN Enable the package end interrupt of DMA channel 9 0: Disabled 1: Enabled

Offset: 0x0004			Register Name: DMAC_IRQ_EN_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
4	R/W	0x0	DMA9_HALF_IRQ_EN Enable the half package interrupt of DMA channel 9 0: Disabled 1: Enabled
3	/	/	/
2	R/W	0x0	DMA8_QUEUE_IRQ_EN Enable the queue end interrupt of DMA channel 8 0: Disabled 1: Enabled
1	R/W	0x0	DMA8_PKG_IRQ_EN Enable the package end interrupt of DMA channel 8 0: Disabled 1: Enabled
0	R/W	0x0	DMA8_HALF_IRQ_EN Enable the half package interrupt of DMA channel 8 0: Disabled 1: Enabled

3.9.6.3 0x0010 DMAC IRQ Pending Status Register 0 (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R/W1C	0x0	DMA7_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 7. Write 1 to clear the pending status. 0: No effect 1: Pending
29	R/W1C	0x0	DMA7_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 7. Write 1 to clear the pending status. 0: No effect 1: Pending

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
28	R/W1C	0x0	DMA7_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 7. Write 1 to clear the pending status. 0: No effect 1: Pending
27	/	/	/
26	R/W1C	0x0	DMA6_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 6. Write 1 to clear the pending status. 0: No effect 1: Pending
25	R/W1C	0x0	DMA6_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 6. Write 1 to clear the pending status. 0: No effect 1: Pending
24	R/W1C	0x0	DMA6_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 6. Write 1 to clear the pending status. 0: No effect 1: Pending
23	/	/	/
22	R/W1C	0x0	DMA5_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 5. Write 1 to clear the pending status. 0: No effect 1: Pending
21	R/W1C	0x0	DMA5_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 5. Write 1 to clear the pending status. 0: No effect 1: Pending
20	R/W1C	0x0	DMA5_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 5. Write 1 to clear the pending status. 0: No effect 1: Pending

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
19	/	/	/
18	R/W1C	0x0	DMA4_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 4. Write 1 to clear the pending status. 0: No effect 1: Pending
17	R/W1C	0x0	DMA4_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 4. Write 1 to clear the pending status. 0: No effect 1: Pending
16	R/W1C	0x0	DMA4_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 4. Write 1 to clear the pending status. 0: No effect 1: Pending
15	/	/	/
14	R/W1C	0x0	DMA3_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 3. Write 1 to clear the pending status. 0: No effect 1: Pending
13	R/W1C	0x0	DMA3_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 3. Write 1 to clear the pending status. 0: No effect 1: Pending
12	R/W1C	0x0	DMA3_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 3. Write 1 to clear the pending status. 0: No effect 1: Pending
11	/	/	/

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
10	R/W1C	0x0	DMA2_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 2. Write 1 to clear the pending status. 0: No effect 1: Pending
9	R/W1C	0x0	DMA2_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 2. Write 1 to clear the pending status. 0: No effect 1: Pending
8	R/W1C	0x0	DMA2_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 2. Write 1 to clear the pending status. 0: No effect 1: Pending
7	/	/	/
6	R/W1C	0x0	DMA1_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 1. Write 1 to clear the pending status. 0: No effect 1: Pending
5	R/W1C	0x0	DMA1_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 1. Write 1 to clear the pending status. 0: No effect 1: Pending
4	R/W1C	0x0	DMA1_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 1. Write 1 to clear the pending status. 0: No effect 1: Pending
3	/	/	/
2	R/W1C	0x0	DMA0_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 0. Write 1 to clear the pending status. 0: No effect 1: Pending

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W1C	0x0	DMA0_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 0. Write 1 to clear the pending status. 0: No effect 1: Pending
0	R/W1C	0x0	DMA0_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 0. Write 1 to clear the pending status. 0: No effect 1: Pending

3.9.6.4 0x0014 DMAC IRQ Pending Status Register 1 (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R/W1C	0x0	DMA15_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 15. Write 1 to clear the pending status. 0: No effect 1: Pending
29	R/W1C	0x0	DMA15_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 15. Write 1 to clear the pending status. 0: No effect 1: Pending
28	R/W1C	0x0	DMA15_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 15. Write 1 to clear the pending status. 0: No effect 1: Pending
27	/	/	/

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
26	R/W1C	0x0	DMA14_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 14. Write 1 to clear the pending status. 0: No effect 1: Pending
25	R/W1C	0x0	DMA14_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 14. Write 1 to clear the pending status. 0: No effect 1: Pending
24	R/W1C	0x0	DMA14_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 14. Write 1 to clear the pending status. 0: No effect 1: Pending
23	/	/	/
22	R/W1C	0x0	DMA13_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 13. Write 1 to clear the pending status. 0: No effect 1: Pending
21	R/W1C	0x0	DMA13_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 13. Write 1 to clear the pending status. 0: No effect 1: Pending.
20	R/W1C	0x0	DMA13_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 13. Write 1 to clear the pending status. 0: No effect 1: Pending
19	/	/	/
18	R/W1C	0x0	DMA12_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 12. Write 1 to clear the pending status. 0: No effect 1: Pending

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
17	R/W1C	0x0	DMA12_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 12. Write 1 to clear the pending status. 0: No effect 1: Pending
16	R/W1C	0x0	DMA12_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 12. Write 1 to clear the pending status. 0: No effect 1: Pending
15	/	/	/
14	R/W1C	0x0	DMA11_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 11. Write 1 to clear the pending status. 0: No effect 1: Pending
13	R/W1C	0x0	DMA11_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 11. Write 1 to clear the pending status. 0: No effect 1: Pending
12	R/W1C	0x0	DMA11_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 11. Write 1 to clear the pending status. 0: No effect 1: Pending
11	/	/	/
10	R/W1C	0x0	DMA10_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 10. Write 1 to clear the pending status. 0: No effect 1: Pending
9	R/W1C	0x0	DMA10_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 10. Write 1 to clear the pending status. 0: No effect 1: Pending

Offset: 0x0010			Register Name: DMAC_IRQ_PEND_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
8	R/W1C	0x0	DMA10_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 10. Write 1 to clear the pending status. 0: No effect 1: Pending
7	/	/	/
6	R/W1C	0x0	DMA9_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 9. Write 1 to clear the pending status. 0: No effect 1: Pending
5	R/W1C	0x0	DMA9_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 9. Write 1 to clear the pending status. 0: No effect 1: Pending
4	R/W1C	0x0	DMA9_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 9. Write 1 to clear the pending status. 0: No effect 1: Pending
3	/	/	/
2	R/W1C	0x0	DMA8_QUEUE_IRQ_PEND The IRQ pending bit for the queue end interrupt of the DMA channel 8. Write 1 to clear the pending status. 0: No effect 1: Pending
1	R/W1C	0x0	DMA8_PKG_IRQ_PEND The IRQ pending bit for the package end interrupt of the DMA channel 8. Write 1 to clear the pending status. 0: No effect 1: Pending
0	R/W1C	0x0	DMA8_HALF_IRQ_PEND The IRQ pending bit for the half package interrupt of the DMA channel 8. Write 1 to clear the pending status. 0: No effect 1: Pending

3.9.6.5 0x0028 DMAC Auto Gating Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: DMAC_AUTO_GATE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x0	DMA_MCLK_CIRCUIT Auto gating bit of DMA MCLK interface circuit 0: Auto gating enabled 1: Auto gating disabled
1	R/W	0x0	DMA_COMMON_CIRCUIT Auto gating bit of DMA common circuit 0: Auto gating enabled 1: Auto gating disabled
0	R/W	0x0	DMA_CHAN_CIRCUIT Auto gating bit of DMA channel circuit 0: Auto gating enabled 1: Auto gating disabled

NOTE

When initializing the DMA Controller, the bit[2] should be set up.

3.9.6.6 0x0030 DMAC Status Register (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: DMAC_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x0	MBUS FIFO Status 0: Empty 1: Not Empty
30:16	/	/	/

Offset: 0x0030			Register Name: DMAC_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
15:0	R	0x0	DMA_STATUS DMA Channel[15:0] Status The meaning of each bit: 0: Idle 1: Busy

3.9.6.7 0x0100 + N*0x0040 DMAC Channel Enable Register N (Default Value: 0x0000_0000)

Offset: 0x0100 + N*0x0040 (N = 0 to 15)			Register Name: DMAC_EN_REGN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	DMA_EN DMA Channel Enable 0: Disabled 1: Enabled

3.9.6.8 0x0104 + N*0x0040 DMAC Channel Pause Register N (Default Value: 0x0000_0000)

Offset: 0x0104 + N*0x0040 (N = 0 to 15)			Register Name: DMAC_PAU_REGN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	DMA_PAUSE Pause the DMA Channel Transfer Data 0: Resume Transferring 1: Pause Transferring

3.9.6.9 0x0108 + N*0x0040 DMAC Channel Descriptor Address Register N (Default Value: 0x0000_0000)

Offset: 0x0108 + N*0x0040 (N = 0 to 15)			Register Name: DMAC_DESC_ADDR_REGN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	R/W	0x0	DMA_DESC_ADDR Lower 30 bits of DMA channel descriptor address The descriptor address must be word-aligned.
1:0	R/W	0x0	DMA_DESC_HIGH_ADDR Higher 2 bits of DMA channel descriptor high address The real address is as follows. DMA Channel Descriptor Address = {bit[1:0], bit[31:2], 2'b00}

3.9.6.10 0x010C + N*0x0040 DMAC Channel Configuration Register N (Default Value: 0x0000_0000)

Offset: 0x010C + N*0x0040 (N = 0 to 15)			Register Name: DMAC_CFG_REGN
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R	0x0	BMODE_SEL 0: Normal Mode 1: BMODE
29:27	/	/	/
26:25	R	0x0	DMA_DEST_DATA_WIDTH DMA Destination Data Width 00: 8-bit 01: 16-bit 10: 32-bit 11: 64-bit
24	R	0x0	DMA_ADDR_MODE DMA Destination Address Mode 0: Linear Mode 1: IO Mode

Offset: 0x010C + N*0x0040 (N = 0 to 15)			Register Name: DMAC_CFG_REGN
Bit	**Read/Write**	**Default/Hex**	**Description**
23:22	R	0x0	DMA_DEST_BLOCK_SIZE DMA Destination Block Size 00: 1 01: 4 10: 8 11: 16
21:16	R	0x0	DMA_DEST_DRQ_TYPE DMA Destination DRQ Type The details in DRQ Type and Port Corresponding Relation.
15:11	/	/	/
10:9	R	0x0	DMA_SRC_DATA_WIDTH DMA Source Data Width 00: 8-bit 01: 16-bit 10: 32-bit 11: 64-bit
8	R	0x0	DMA_SRC_ADDR_MODE DMA Source Address Mode 0: Linear Mode 1: IO Mode
7:6	R	0x0	DMA_SRC_BLOCK_SIZE DMA Source Block Size 00: 1 01: 4 10: 8 11: 16
5:0	R	0x0	DMA_SRC_DRQ_TYPE DMA Source DRQ Type The details in DRQ Type and Port Corresponding Relation.

3.9.6.11 0x0110 + N*0x0040 DMAC Channel Current Source Address Register N (Default Value: 0x0000_0000)

Offset: 0x0110 + N*0x0040 (N = 0 to 15)			Register Name: DMAC_CUR_SRC_REGN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DMA_CUR_SRC DMA Channel Current Source Address.

3.9.6.12 0x0114 + N*0x0040 DMAC Channel Current Destination Address Register N (Default Value: 0x0000_0000)

Offset: 0x0114 + N*0x0040 (N = 0 to 15)			Register Name: DMAC_CUR_DEST_REGN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DMA_CUR_DEST DMA Channel Current Destination Address.

3.9.6.13 0x0118 + N*0x0040 DMAC Channel Byte Counter Left Register N (Default Value: 0x0000_0000)

Offset: 0x0118 + N*0x0040 (N = 0 to 15)			Register Name: DMAC_BCNT_LEFT_REGN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24:0	R	0x0	DMA_BCNT_LEFT DMA Channel Byte Counter Left.

3.9.6.14 0x011C + N*0x0040 DMAC Channel Parameter Register N (Default Value: 0x0000_0000)

Offset: 0x011C + N*0x0040 (N = 0 to 15)			Register Name: DMAC_PARA_REGN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R	0x0	WAIT_CYC Wait Clock Cycles

3.9.6.15 0x0128 + N*0x0040 DMAC Mode Register N (Default Value: 0x0000_0000)

Offset: 0x0128 + N*0x0040 (N = 0 to 15)			Register Name: DMAC_MODE_REGN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3	R/W	0x0	DMA_DST_MODE Destination Communication Mode Select 0: Waiting mode 1: Handshake mode
2	R/W	0x0	DMA_SRC_MODE Source Communication Mode Select 0: Waiting mode 1: Handshake mode
1:0	/	/	/

3.9.6.16 0x012C + N*0x0040 DMAC Former Descriptor Address Register N (Default Value: 0x0000_0000)

Offset: 0x012C + N*0x0040 (N = 0 to 15)			Register Name: DMAC_FDESC_ADDR_REGN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DMA_FDESC_ADDR This register is used to store the former value of DMA Channel Descriptor Address Register.

3.9.6.17 0x0130 + N*0x0040 DMAC Package Number Register N (Default Value: 0x0000_0000)

Offset: 0x0130 + N*0x0040 (N = 0 to 15)			Register Name: DMAC_PKG_NUM_REGN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DMA_PKG_NUM This register records the number of packages which has been completed in one transmission.

3.10 Thermal Sensor Controller (THS)

3.10.1 Overview

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

The thermal sensor controller (THS) embeds one thermal sensor located in the CPU. When the temperature reaches a certain thermal threshold, the thermal sensor can generate interrupts to the software to lower the temperature via the dynamic voltage and frequency scaling (DVFS) technology.

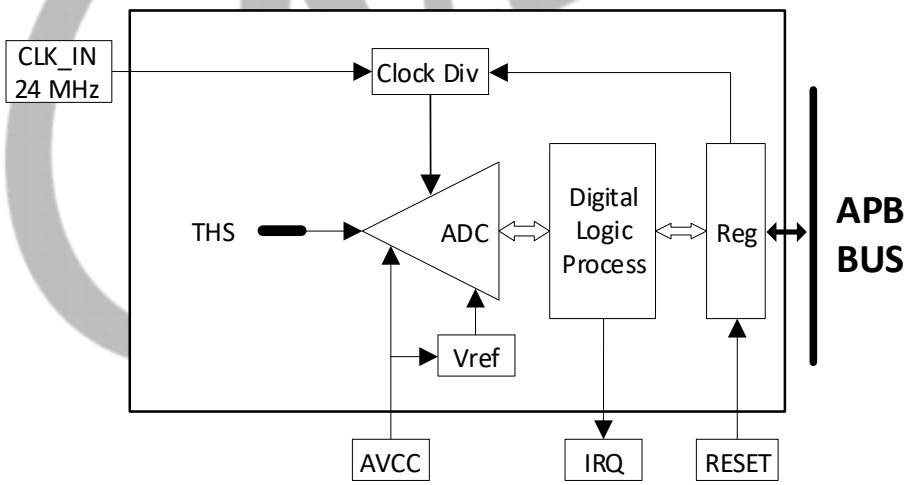
The THS has the following features:

- Temperature accuracy: ±3°C from 0°C to +100°C, ±5°C from -25°C to +125°C
- Averaging filter for thermal sensor reading
- Supports over-temperature protection interrupt and over-temperature alarm interrupt

3.10.2 Block Diagram

The following figure shows a block diagram of the THS.

Figure 3-25 THS Block Diagram



3.10.3 Functional Description

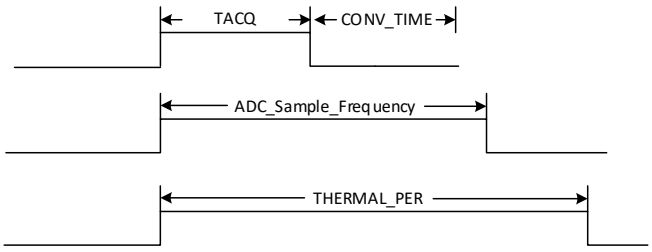
3.10.3.1 Clock Source

The THS gets one clock source: OSC24M. For details about clock configurations, refer to section 3.3 “CCU”.

3.10.3.2 Timing Requirements

The following figure shows the timing requirements for the THS.

Figure 3-26 Thermal Sensor Timing Requirement



CLK_IN = 24 MHz

CONV_TIME (Conversion Time) = 1/24 MHz x 14 Cycles = 0.583 us

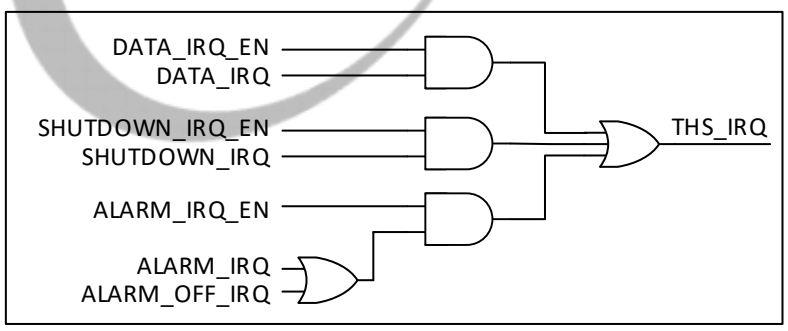
TACQ > 1/24 MHz x 24 Cycles

THERMAL_PER > ADC_Sample_Frequency > TACQ + CONV_TIME

3.10.3.3 Interrupts

The THS has four interrupt sources: DATA_IRQ, SHUTDOWN_IRQ, ALARM_IRQ, and ALARM_OFF_IRQ. The following figure shows thermal sensor interrupt sources.

Figure 3-27 Thermal Sensor Controller Interrupt Source



DATA_IRQ: The interrupt is generated when the measured sensor_data is updated.

SHUTDOWN_IRQ: The interrupt is generated when the temperature is higher than the shutdown threshold.

ALARM_IRQ: The interrupt is generated when the temperature is higher than the Alarm_Threshold.

ALARM_OFF_IRQ: The interrupt is generated when the temperature drops to lower than the Alarm_Off_Thershold. It is triggered at the fall edge.

3.10.3.4 THS Temperature Conversion Formula

T = (sensor_data - 2800)/(-14.85)

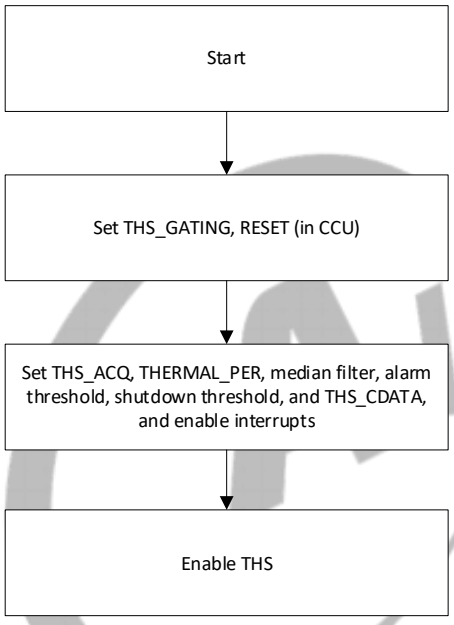
Unit of T: Celsius degree (°C).

The sensor_data is read from the sensor data register.

3.10.4 Programming Guidelines

The initial process of the THS is as follows.

Figure 3-28 THS Initial Process



In the final test (FT) stage, the THS is calibrated through the ambient temperature, and the calibration value is written in the SID module. The following table shows the THS information in the SID.

Table 3-15 THS Information in the SID

Base Address: 0x14	Register Name: THS
Bit	**Description**
27:16	The calibration value of the T-sensor.

Before enabling THS, read eFuse value and write the value to **THS_CDATA**.

Query Mode

Step 1 Write 0x1 to the bit[16] of **THS_BGR_REG** to dessert the reset.

Step 2 Write 0x1 to the bit[0] of **THS_BGR_REG** to open the THS clock.

Step 3 Write 0x2F to the bit[15:0] of **THS_CTRL** to set the ADC acquire time.

Step 4 Write 0x1DF to the bit[31:16] of **THS_CTRL** to set the ADC sample frequency divider.

Step 5 Write 0x3A to the bit[31:12] of **THS_PER** to set the THS work period.

Step 6 Write 0x1 to the bit[2] of **THS_FILTER** to enable the temperature convert filter.

Step 7 Write 0x1 to the bit[1:0] of **THS_FILTER** to select the filter type.

Step 8 Read THS eFuse value from SID, then write the eFuse value to **THS_CDATA** to calibrate THS.

Step 9 Write 0x1 to the bit[0] of THS0_EN to enable THS.

Step 10 Read the bit[0] of **THS_DATA_INTS**. If it is 1, the temperature conversion is complete.

Step 11 Read the bit[11:0] of **THS_DATA**, and calculate the THS temperature based on section 3.10.3.4 "THS Temperature Conversion Formula".

Interrupt Mode

Step 1 Write 0x1 to the bit16 of **THS_BGR_REG** to dessert the reset.

Step 2 Write 0x1 to the bit0 of **THS_BGR_REG** to open the THS clock.

Step 3 Write 0x2F to the bit[15:0] of **THS_CTRL** to set the ADC acquire time.

Step 4 Write 0x1DF to the bit[31:16] of **THS_CTRL** to set the ADC sample frequency divider.

Step 5 Write 0x3A to the bit[31:12] of **THS_PER** to set the THS work period.

Step 6 Write 0x1 to the bit2 of **THS_FILTER** to enable the temperature convert filter.

Step 7 Write 0x1 to the bit[1:0] of **THS_FILTER** to select the filter type.

Step 8 Read THS eFuse value from SID, and then write the eFuse value to **THS_CDATA** to calibrate THS.

Step 9 Write 0x1 to the bit[0] of **THS_DATA_INTC** to enable the interrupt of THS.

Step 10 Set interrupt based on GIC module.

Step 11 Put the interrupt handler address into the interrupt vector table.

Step 12 Write 0x1 to the bit[0] of THS0_EN to enable THS.

Step 13 Read the bit[0] of **THS_DATA_INTS**. If it is 1, the temperature conversion is complete.

Step 14 Read the bit[11:0] of **THS_DATA**, and calculate the THS temperature based on section 3.10.3.4 "THS Temperature Conversion Formula".

3.10.5 Register List

Module Name	Base Address
THS	0x02009400

Register Name	Offset	Description
THS_CTRL	0x0000	THS Control Register
THS_EN	0x0004	THS Enable Register
THS_PER	0x0008	THS Period Control Register
THS_DATA_INTC	0x0010	THS Data Interrupt Control Register
THS_SHUT_INTC	0x0014	THS Shut Interrupt Control Register
THS_ALARM_INTC	0x0018	THS Alarm Interrupt Control Register
THS_DATA_INTS	0x0020	THS Data Interrupt Status Register
THS_SHUT_INTS	0x0024	THS Shut Interrupt Status Register
THS_ALARMO_INTS	0x0028	THS Alarm off Interrupt Status Register
THS_ALARM_INTS	0x002C	THS Alarm Interrupt Status Register
THS_FILTER	0x0030	THS Median Filter Control Register
THS_ALARM_CTRL	0x0040	THS Alarm Threshold Control Register
THS_SHUTDOWN_CTRL	0x0080	THS Shutdown Threshold Control Register
THS_CDATA	0x00A0	THS Calibration Data
THS_DATA	0x00C0	THS Data Register

3.10.6 Register Description

3.10.6.1 0x0000 THS Control Register (Default Value: 0x01DF_002F)

Offset: 0x0000			Register Name: THS_CTRL
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x1DF	TACQ ADC acquire time CLK_IN/(n + 1) The default value is 2 us.
15:0	R/W	0x2F	Reserved

3.10.6.2 0x0004 THS Enable Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: THS_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	THS_EN Enable temperature measurement sensor 0: Disable 1: Enable

3.10.6.3 0x0008 THS Period Control Register (Default Value: 0x0003_A000)

Offset: 0x0008			Register Name: THS_PER
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	R/W	0x3A	THERMAL_PER Temperature measurement period 4096*(n + 1)/CLK_IN The default value is 10 ms.
11:0	/	/	/

3.10.6.4 0x0010 THS Data Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: THS_DATA_INTC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	THS_DATA_IRQ_EN Enable the interrupt of sensor_data update If enabled, when the measured sensor_data is updated, it will generate an interrupt. 0: Disabled 1: Enabled

3.10.6.5 0x0014 THS Shut Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: THS_SHUT_INTC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	SHUT_INT_EN Enable the shutdown interrupt for the sensor 0: Disabled 1: Enabled

3.10.6.6 0x0018 THS Alarm Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: THS_ALARM_INTC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	ALARM_INT_EN Enable the alarm interrupt for the sensor 0: Disabled 1: Enabled

3.10.6.7 0x0020 THS Data Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: THS_DATA_INTS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W1C	0x0	THS_DATA_IRQ_STS Indicates the pending status of the sensor's data interrupt. Write 1 to clear the pending status. 0: No effect 1: Pending

3.10.6.8 0x0024 THS Shut Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: THS_SHUT_INTS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W1C	0x0	SHUT_INT_STS Indicates the pending status of the sensor's shutdown interrupt. Write 1 to clear the pending status. 0: No effect 1: Pending

3.10.6.9 0x0028 THS Alarm Off Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: THS_ALARM_INTS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W1C	0x0	ALARM_OFF_STS Alarm interrupt off pending for sensor Write 1 to clear the pending status. 0: No effect 1: Pending

3.10.6.10 0x002C THS Alarm Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: THS_ALARM_INTS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W1C	0x0	ALARM_INT_STS Alarm interrupt pending for sensor Write 1 to clear the pending status. 0: No effect 1: Pending

3.10.6.11 0x0030 Median Filter Control Register (Default Value: 0x0000_0001)

Offset: 0x0030			Register Name: THS_FILTER
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x0	FILTER_EN Filter enable 0: Disabled 1: Enabled
1:0	R/W	0x1	FILTER_TYPE Averaging filter type 00: 2 01: 4 10: 8 11: 16

3.10.6.12 0x0040 THS Alarm Threshold Control Register (Default Value: 0x05A0_0684)

Offset: 0x0040			Register Name: THS_ALARM_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0x5A0	ALARM_T_HOT Thermal sensor alarm threshold for hot temperature
15:12	/	/	/
11:0	R/W	0x684	ALARM_T_HYST Thermal sensor alarm threshold for hysteresis temperature

3.10.6.13 0x0080 THS Shutdown Threshold Control Register (Default Value: 0x0000_04E9)

Offset: 0x0080			Register Name: THS_SHUTDOWN_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
11:0	R/W	0x4E9	SHUT_T_HOT Thermal sensor shutdown threshold for hot temperature

3.10.6.14 0x00A0 THS Calibration Data Register (Default Value: 0x0000_0800)

Offset: 0x00A0			Register Name: THS_CDATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11:0	R/W	0x800	THS_CDATA Thermal sensor calibration data

3.10.6.15 0x00C0 THS Data Register (Default Value: 0x0000_0000)

Offset: 0x00C0			Register Name: THS_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11:0	R	0x0	THS_DATA Temperature measurement data of sensor

3.11 IOMMU

3.11.1 Overview

The I/O Memory management unit (IOMMU) is designed for the specific memory requirements of products. It maps the virtual address (sent by peripheral access memory) to the physical address. The IOMMU allows multiple ways to manage the location of physical address, and it can use physical address which has potentially conflict mapping for different processes to allocate memory space, and also allow application of non-continuous address mapping to continuous virtual address space.

Features:

- Supports virtual address to physical address mapping by hardware implementation
- Supports VE, CSI, DE, G2D, DI parallel address mapping
- Supports VE, CSI, DE, G2D, DI bypass function independently
- Supports VE, CSI, DE, G2D, DI pre-fetch independently
- Supports VE, CSI, DE, G2D, DI interrupt handing mechanism independently
- Supports 2 levels TLB (level1 TLB for special using, and level2 TLB for sharing)
- Supports TLB Fully cleared and Partially disabled
- Supports trigger PTW behavior when TLB miss
- Supports checking the permission

3.11.2 Block Diagram

The internal module of IOMMU mainly has the following parts.

Micro TLB: level1 TLB, 64 words. Each peripheral corresponds to a TLB, which caching the level2 page table for the peripheral.

Macro TLB: level2 TLB, 4K words. Each peripheral shares a level2 TLB for caching the level2 page table.

Prefetch Logic: Each Micro TLB corresponds to a Prefetch Logic. By monitoring each master device to predict the bus access, the secondary page table corresponding to the address to be accessed can be read from memory and stored in the secondary TLB to improve hit ratio.

PTW Logic: Page Table Walk, mainly contains PTW Cache and PTW. The PTW Cache is used to store the level1 page table; when the virtual address VA missed in the level1 and level2 TLB, it will trigger the PTW. PTW Cache can store 512 level1 page tables, that is, 512 words.

PMU: Performance Monitoring Unit, which is used to count hit efficiency and latency.

APB Interface: IOMMU register instantiation module. CPU reads and writes the IOMMU register by APB bus.

The following figure shows the internal block diagram of IOMMU.

Figure 3-29 IOMMU Block Diagram

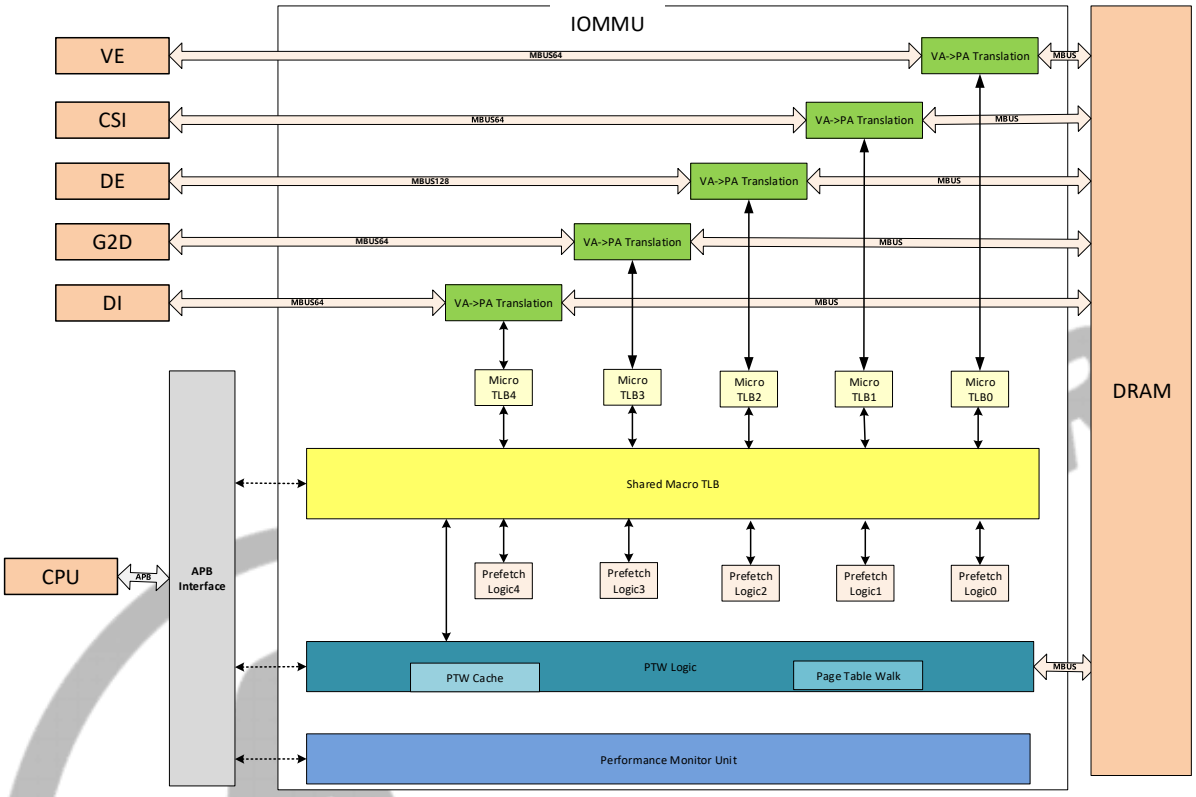


Table 3-16 Correspondence Relation between Master and Module

Master Number	Module
Master0	VE
Master1	CSI
Master2	DE
Master3	G2D
Master4	DI

3.11.3 Functional Description

3.11.3.1 Initialization

- Release the IOMMU reset signal by writing 1 to **IOMMU_RESET_REG[31]**;

- Write the base address of the first TLB to **IOMMU_TTB_REG**;
- Set **IOMMU_INT_ENABLE_REG**;
- Enable the IOMMU by configuring **IOMMU_ENABLE_REG** in the final.

3.11.3.2 Address Changing

In the process of address mapping, The peripheral virtual address VA[31:12] are retrieved in the Level1 TLB, when TLB hits, the mapping finished, or they are retrieved in the Level2 TLB in the same way. If TLB hits, it will write the hit mapping to the Level1 TLB, and hits in Level1 TLB. If Level1 and Level2 TLB are retrieved fail, it will trigger the PTW. After opening peripheral bypass function by setting IOMMU Bypass Register, IOMMU will not map the address for peripheral typed the address, and it will output the virtual address as physical address. The typical scenarios are as follows.

Micro TLB hit

Step 1 The master device sends a transfer command to send the address to the corresponding Micro TLB, and searches the Level2 page table corresponding to virtual address;

Step 2 If Micro TLB hits, it will return a corresponding physical addresses and the Level2 page table of permission Index;

Step 3 Address transform module converts the virtual address into physical address, and checks the permissions at the same time. If pass, the transfer is completed.

Micro TLB miss, Macro TLB hit

Step 1 The master device sends a transfer command to send the address to the corresponding Micro TLB, and searches the Level2 page table corresponding to virtual address;

Step 2 If Micro TLB misses, then continue to search Macro TLB;

Step 3 If Macro TLB hits, it will return the Level2 page table to Micro TLB;

Step 4 Micro TLB receives the page table and puts it to Micro TLB (if this Micro TLB is full, there will happen the replace activity), at the same time, the page table entry is sent to address translation module;

Step 5 Address transform module converts the virtual address into physical address, and checks the permissions at the same time. If pass, the transfer is completed.

Micro TLB miss, Macro TLB miss, PTW Cache hit

Step 1 The master device sends a transfer command to send the address to the corresponding Micro TLB, and searches the Level2 page table corresponding to virtual address;

Step 2 If Micro TLB misses, then continue to search Macro TLB;

Step 3 If Macro TLB misses, then it will send the request to the PTW to return the corresponding page table;

Step 4 PTW first accesses PTW Cache, confirms that the required Level1 page table exists in the PTW Cache, sends the page table to PTW logic;

Step 5 PTW logic returns the corresponding Level2 page table from memory page table according to Level1 page table, checks the effectiveness, and sends to Macro TLB;

Step 6 Macro TLB stores the Level2 page table (there may happen the replace activity), and will return the Level2 page table to Micro TLB;

Step 7 Micro TLB receives the page table entries and puts it to the Micro TLB (if this Micro TLB is full, there will happen the replace activity), and sends page table entries to address translation module;

Step 8 Address transform module converts the virtual address into physical address, and checks the permissions at the same time. If pass, the transfer is completed.

Micro TLB miss, Macro TLB miss, PTW Cache miss

Step 1 The master device sends a transfer command to send the address to the corresponding Micro TLB, and searches the Level2 page table corresponding to virtual address;

Step 2 If Micro TLB misses, then continue to search Macro TLB;

Step 3 If Macro TLB misses, there will send the request to the PTW to return the corresponding page table;

Step 4 PTW accesses PTW Cache, the Level1 page table is unnecessary;

Step 5 PTW accesses memory to get the corresponding Level1 page table and stores it to the PTW Cache (there may happen the replace activity);

Step 6 PTW logic returns the corresponding Level2 page table from memory page table according to Level1 page table, checks the effectiveness, and sends to Macro TLB;

Step 7 Macro TLB stores the Level2 page table (there may happen the replace activity), and returns the Level 2 page table to Micro TLB;

Step 8 Micro TLB receives the page table entries and puts it to the Micro TLB (if this Micro TLB is full, there will happen the replace activity), and sends page table entries to address translation module;

Step 9 Address transform module converts the virtual address into physical address, and checks the permissions at the same time. If pass, the transfer is completed.

Permission error

Step 1 Permission checking always performs in the address conversion;

Step 2 Once the permission checking makes mistake, the new access of the master suspends, but continues before this access;

Step 3 Set the error status register;

Step 4 Trigger interrupt.

Invalid Level1 page table

Step 1 Invalid Level1 page table is checked when PTW logic reads the new level page table from memory;

Step 2 The PTW reads sequentially two page table entries from the memory (64-bit data, a complete cache line), and stores in the PTW cache;

Step 3 If the current page table is detected invalid, then the error flag is set, and the interrupt is triggered, the cache line need to be invalidated.

NOTE

- Invalid page table has two situations: the reading target page table from the memory is invalid; and the page table stored in PTW Cache with target page table is found to be invalid after using;
- If a page table is invalid, then the total cache line (that is two page tables) need to be invalidated.

Invalid Level2 page table

Step 1 Invalid Level2 page table checks when Macro TLB reads the new level page table from memory;

Step 2 The Macro TLB reads sequentially two page table entries from the memory (64-bit data, a complete cache line), and stores in the Macro TLB;

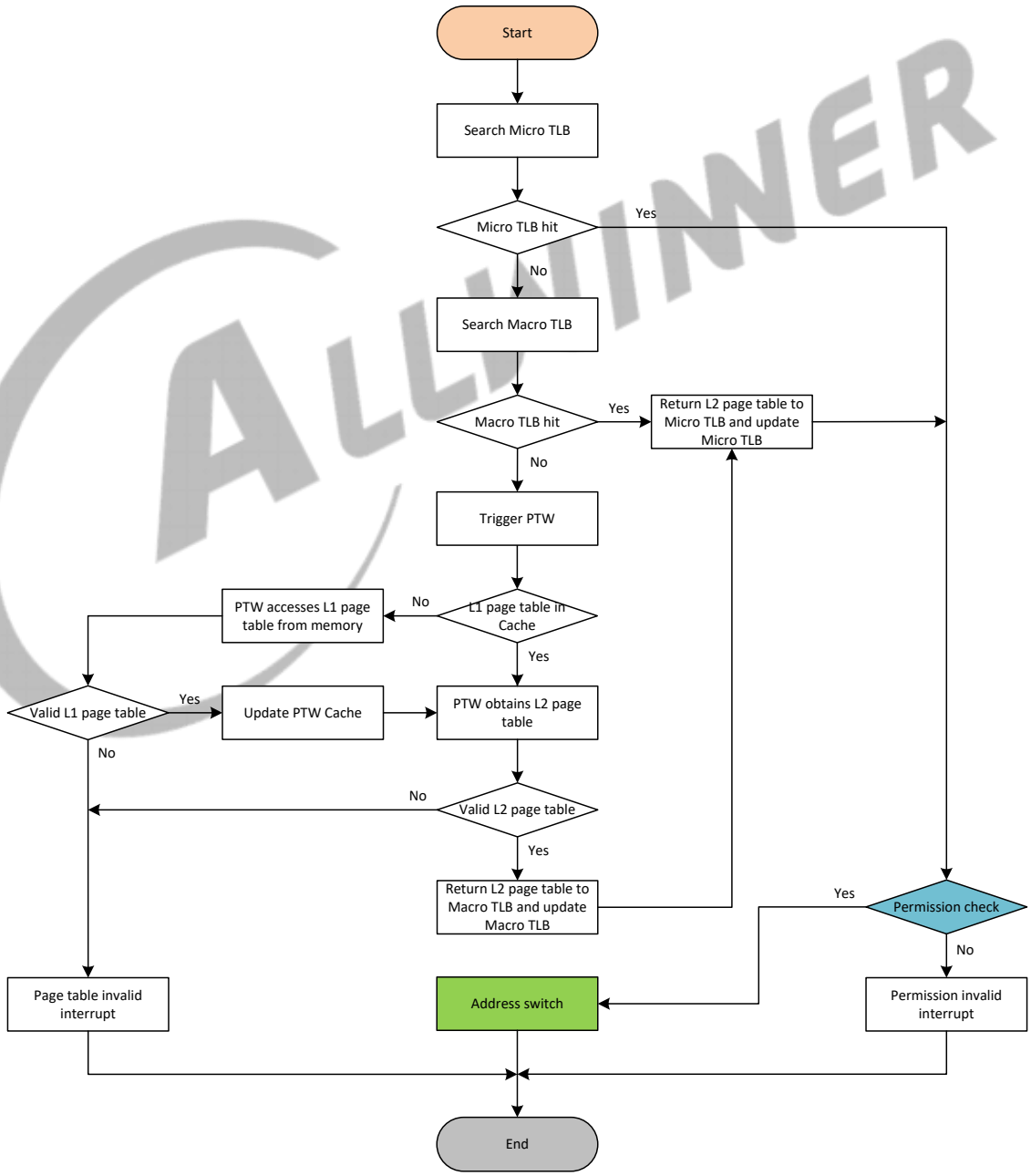
Step 3 If the current page table is detected invalid, then the error flag is set, and the interrupt is triggered, the cache line need to be invalidated.

NOTE

- Invalid page table has two situations: the reading target page table from the memory is invalid; and the page table stored in Macro TLB with target page table is found to be invalid after using.
- If a page table is invalid, then the total cache line (that is two page tables) need to be invalidated.

The following figure shows the internal address switch process.

Figure 3-30 Internal Switch Process



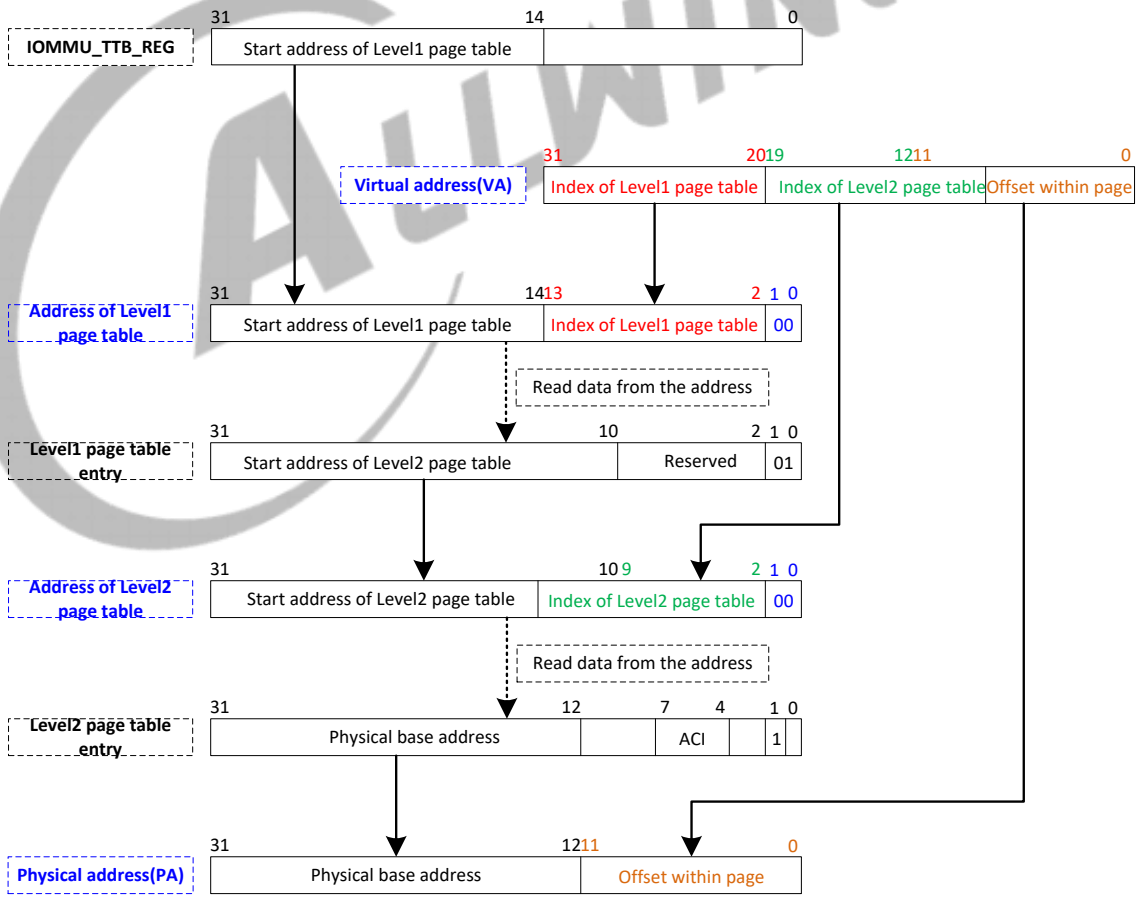
3.11.3.3 VA-PA Mapping

IOMMU page table is defined as Level2 mapping, the first level is 1M address space mapping, the second level is 4K address space. This version does not support 1K, 16K and other page table size. IOMMU supports a page table only, its meaning is:

- All peripherals connected to IOMMU use the same virtual address space;
- The virtual address space of the peripherals can overlap;
- Different virtual addresses can map to the same physical address space;

Base address of the page table is defined by software, and it needs 16 KB address alignment; Page table of the Level2 table item needs 1 KB address alignment. The following figure shows a complete VA-PA address translation process.

Figure 3-31 VA-PA Switch Process



3.11.3.4 Clear and Invalidate TLB

When the content of multi page tables refreshes or the address of page table changes, all VA-PA mapping which has been cached in TLB will no longer be valid, then you need configure **IOMMU TLB Flush Enable Register** to clear the TLB or PTW Cache. First suspend the access to TLB or Cache, then configure the corresponding Flush bit of **IOMMU TLB Flush Enable Register**, after the operation takes effect, the related peripherals can continue to send new access memory operations.

When some page table is invalid or incorrect mapping, you can set the TLB Invalidation relevant register to invalidate some TLB VA-PA mapping pairs. There are two modes to invalidate the TLB operation.

(1) Mode0 (Old mode)

Step 1 Set **IOMMU TLB Invalidation Mode Select Register** to 0 to select mode0;

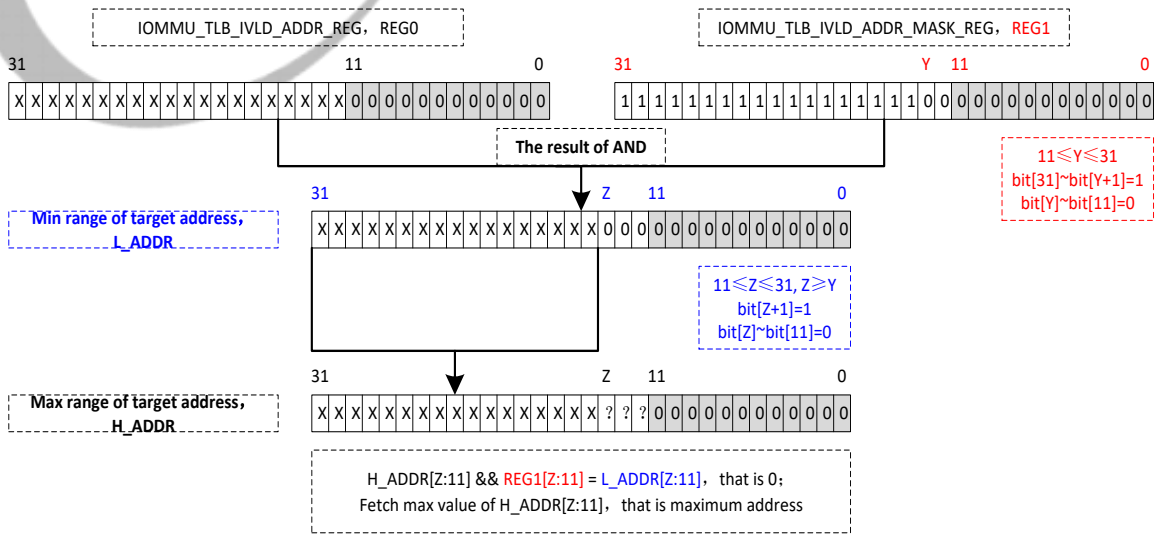
Step 2 Write target address to **IOMMU TLB Invalidation Address Register**;

Step 3 Set configuration values to **IOMMU TLB Invalidation Address Mask Register**, the requirements are as follows:

- The value of **IOMMU TLB Invalidation Address Mask Register** cannot be less than the **IOMMU TLB Invalidation Address Register**.
- The higher bit of **IOMMU TLB Invalidation Address Mask Register** must be continuous 1, the lower bit must be continuous 0. For example, 0xfffff000, 0xffffe000, 0xfffc000, 0xffff8000, 0xffff0000 belongs to the legal value; and 0xffffd000, 0xffffb000, 0xfffa000, 0xffff9000, 0xffff7000 belongs to the illegal values.

Step 4 Configure **IOMMU TLB Invalidation Enable Register** to enable the invalid operation. Among the way to determine the invalid address is to get maximum valid bit and determine target address range by target address AND mask address. The following figure shows the process.

Figure 3-32 Invalid TLB Address Range



For example:

a) When the value of **IOMMU TLB Invalidation Address Mask Register** is 0xFFFFF000 by default, the result of AND is target address, that is, only target address is invalid.

b) When the value of **IOMMU TLB Invalidation Address Mask Register** is 0xFFFF0000, the value of **IOMMU TLB Invalidation Address Register** is 0xEEEE1000, then the range of target address is from 0xEEEE0000 to 0xEEEEF000.

c) When the value of **IOMMU TLB Invalidation Address Mask Register** is 0xFFFFC000, the value of **IOMMU TLB Invalidation Address Register** is 0xEEEE8000, then the range of target address is from 0xEEEE8000 to 0xEEEEB000.

d) When the value of **IOMMU TLB Invalidation Address Mask Register** is 0xFFFF8000, the value of **IOMMU TLB Invalidation Address Register** is 0xEEEEC000, then the range of target address is from 0xEEEE8000 to 0xEEEEF000.

e) When the value of **IOMMU TLB Invalidation Address Mask Register** is 0xFFFFC000, the value of **IOMMU TLB Invalidation Address Register** is 0xEEEE0000, then the range of target address is from 0xEEEE0000 to 0xEEEE3000.

(2) Mode1 (New mode)

Step 1 Set **IOMMU TLB Invalidation Mode Select Register** to 1 to select mode1;

Step 2 Set the starting address of invalid TLB by **IOMMU TLB Invalidation Start Address Register,** and set the ending address of invalid TLB by **IOMMU TLB Invalidation Start Address Register;**

Step 3 Configure **IOMMU TLB Invalidation Enable Register** to enable invalid operation, then the related TLB operation is invalidated.

3.11.3.5 Clear and Invalidate PTW Cache

There are two modes to invalidate the PTW cache operation.

(1) Mode0 (Old mode)

Step 1 Set **IOMMU PC Invalidation Mode Select Register** to 0 to select mode0;

Step 2 Set the address register that needs to be invalidated to **IOMMU PC Invalidation Address Register** (the addresses need to be aligned with 1 MB);

Step 3 Configure **IOMMU PC Invalidation Enable Register** to enable the invalid operation. That is, the PTW cache operation of a cacheline is invalidated.

(2) Mode1 (New mode)

Step 1 Set **IOMMU PC Invalidation Mode Select Register** to 1 to select mode1;

Step 2 Set the starting address of invalid TLB by **IOMMU PTW Invalidation Start Address Register,** and set the ending address of invalid TLB by **IOMMU PC Invalidation Start Address Register;**

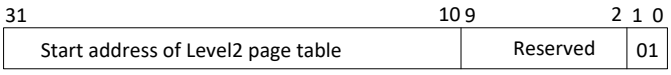
Step 3 Configure **IOMMU PC Invalidation Enable Register** to enable invalid operation, then to invalid the related PWM cache operation is completed.

3.11.3.6 Page Table Format

Level1 Page Table

The format of Level1 page table is as follows.

Figure 3-33 Level1 Page Table Format



Bit[31:10]: Base address of Level2 page table;

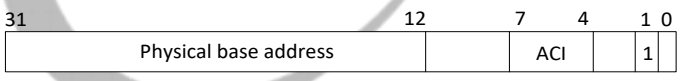
Bit[9:2]: Reserved;

Bit[1:0]: 01 is valid page table; others are fault;

Level2 Page Table

The format of Level2 page table is as follows.

Figure 3-34 Level1 Page Table Format



Bit[31:12]: Physical address of 4K address;

Bit[11:8]: Reserved;

Bit[7:4]: ACI, permission control index; correspond to permission control bit of **IOMMU Domain Authority Control Register**;

Bit[3:2]: Reserved;

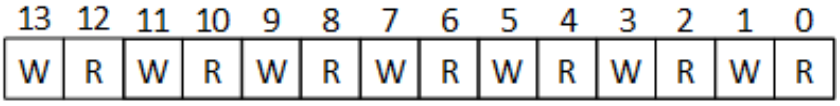
Bit[1]: 1 is valid page table; 0 is fault;

Bit[0]: Reserved

Permission Index

The read/write access control of series register such as **IOMMU Domain Authority Control Register** is as follows.

Figure 3-35 Read/Write Permission Control



Bit[1:0]/Bit[17:16]: Master0 read/write permission control;

Bit[3:2]/Bit[19:18]: Master1 read/write permission control;

Bit[5:4]/Bit[21:20]: Master2 read/write permission control;

Bit[7:6]/Bit[23:22]: Master3 read/write permission control;

Bit[9:8]/Bit[25:24]: Master4 read/write permission control;

Bit[11:10]/Bit[27:26]: Master5 read/write permission control;

Bit[13:12]/Bit[29:28]: Master6 read/write permissiom control.

The value of **IOMMU Domain Authority Control Register** is read-only by default. Other registers can configure through system requirement. In address switch process, the corresponding relation between ACI and Domain is as follows.

Table 3-17 Relation between ACI and Domain

ACI	Domain	Register
0	Domain 0	IOMMU Domain Authority Control Register 0
1	Domain 1	IOMMU Domain Authority Control Register 0
2	Domain 2	IOMMU Domain Authority Control Register 1
3	Domain 3	IOMMU Domain Authority Control Register 1
4	Domain 4	IOMMU Domain Authority Control Register 2
5	Domain 5	IOMMU Domain Authority Control Register 2
6	Domain 6	IOMMU Domain Authority Control Register 3
7	Domain 7	IOMMU Domain Authority Control Register 3
8	Domain 8	IOMMU Domain Authority Control Register 4
9	Domain 9	IOMMU Domain Authority Control Register 4
10	Domain 10	IOMMU Domain Authority Control Register 5
11	Domain 11	IOMMU Domain Authority Control Register 5

ACI	Domain	Register
12	Domain 12	IOMMU Domain Authority Control Register 6
13	Domain 13	IOMMU Domain Authority Control Register 6
14	Domain 14	IOMMU Domain Authority Control Register 7
15	Domain 15	IOMMU Domain Authority Control Register 7

After enabled **IOMMU Domain Authority Overwrite Register**, the read/write control permission can override all **IOMMU Domain Authority Control Register**.

3.11.4 Programming Guidelines

3.11.4.1 Resetting IOMMU

Before the IOMMU module software reset operation, make sure IOMMU is never opened, or all bus operations are completed, or DRAM and peripherals already open the corresponding switch, to shield the influence of IOMMU reset.

3.11.4.2 Enabling IOMMU

Before opening the IOMMU address mapping function, the **Translation Table Base Register** should be correctly configured, or all the masters are in the bypass state, or all the masters do not send the bus command.

3.11.4.3 Configuring TTB

Operating the register must close the address mapping function of IOMMU, that is, the IOMMU_ENABLE_REG[0] is 0; or the bypass function of all masters is set to 1; or the state of the TX bus command is none.

3.11.4.4 Clearing TTB

In the Flush operation, all TLB/Cache access will be suspended; but the operation entered the TLB will continue to complete before the Flush starts.

3.11.4.5 Reading/Writing VA Data

For target virtual address, read and write the corresponding physical address data to make sure whether IOMMU module address mapping function is normal. First, make sure to read or write, and then configure the target virtual address or write data, then start to read or write function, check whether the results are as expected after the operation is finished.

3.11.4.6 Using PMU Statistics Function

When PMU function is used for the first time, set **IOMMU PMU Enable Register** to enable statistics function; when reading the relevant Register, clear the enable bit of **IOMMU PMU Enable Register**; when PMU function is used next time, first **IOMMU PMU Clear Register** is set, after counter is cleared, set the enable bit of **IOMMU PMU Enable Register**.

Given a Level2 page table administers continuous 4 KB address, if Micro TLB misses in continuous virtual address, there may need to return a Level2 page table to hit from Macro TLB; but the hit number is not recorded in the Macro TLB hit and Micro TLB hit related register. So the true hit rate calculation is as follows:

Hit Rate = N1/M1 + (1-N1/M1)*N2/M2

N1: Micro TLB hit number
M1: Micro TLB access number
N2: Macro TLB hit number
M2: Macro TLB access number

3.11.5 Register List

Module Name	Base Address
IOMMU	0x02010000

Register Name	Offset	Description
IOMMU_RESET_REG	0x0010	IOMMU Reset Register
IOMMU_ENABLE_REG	0x0020	IOMMU Enable Register
IOMMU_BYPASS_REG	0x0030	IOMMU Bypass Register
IOMMU_AUTO_GATING_REG	0x0040	IOMMU Auto Gating Register
IOMMU_WBUF_CTRL_REG	0x0044	IOMMU Write Buffer Control Register
IOMMU_OOO_CTRL_REG	0x0048	IOMMU Out of Order Control Register
IOMMU_4KB_BDY_PRT_CTRL_REG	0x004C	IOMMU 4KB Boundary Protect Control Register

Register Name	Offset	Description
IOMMU_TTB_REG	0x0050	IOMMU Translation Table Base Register
IOMMU_TLB_ENABLE_REG	0x0060	IOMMU TLB Enable Register
IOMMU_TLB_PREFETCH_REG	0x0070	IOMMU TLB Prefetch Register
IOMMU_TLB_FLUSH_ENABLE_REG	0x0080	IOMMU TLB Flush Enable Register
IOMMU_TLB_IVLD_MODE_SEL_REG	0x0084	IOMMU TLB Invalidation Mode Select Register
IOMMU_TLB_IVLD_STA_ADDR_REG	0x0088	IOMMU TLB Invalidation Start Address Register
IOMMU_TLB_IVLD_END_ADDR_REG	0x008C	IOMMU TLB Invalidation End Address Register
IOMMU_TLB_IVLD_ADDR_REG	0x0090	IOMMU TLB Invalidation Address Register
IOMMU_TLB_IVLD_ADDR_MASK_REG	0x0094	IOMMU TLB Invalidation Address Mask Register
IOMMU_TLB_IVLD_ENABLE_REG	0x0098	IOMMU TLB Invalidation Enable Register
IOMMU_PC_IVLD_MODE_SEL_REG	0x009C	IOMMU PC Invalidation Mode Select Register
IOMMU_PC_IVLD_ADDR_REG	0x00A0	IOMMU PC Invalidation Address Register
IOMMU_PC_IVLD_STA_ADDR_REG	0x00A4	IOMMU PC Invalidation Start Address Register
IOMMU_PC_IVLD_ENABLE_REG	0x00A8	IOMMU PC Invalidation Enable Register
IOMMU_PC_IVLD_END_ADDR_REG	0x00AC	IOMMU PC Invalidation End Address Register
IOMMU_DM_AUT_CTRL0_REG	0x00B0	IOMMU Domain Authority Control 0 Register
IOMMU_DM_AUT_CTRL1_REG	0x00B4	IOMMU Domain Authority Control 1 Register
IOMMU_DM_AUT_CTRL2_REG	0x00B8	IOMMU Domain Authority Control 2 Register
IOMMU_DM_AUT_CTRL3_REG	0x00BC	IOMMU Domain Authority Control 3 Register
IOMMU_DM_AUT_CTRL4_REG	0x00C0	IOMMU Domain Authority Control 4 Register
IOMMU_DM_AUT_CTRL5_REG	0x00C4	IOMMU Domain Authority Control 5 Register
IOMMU_DM_AUT_CTRL6_REG	0x00C8	IOMMU Domain Authority Control 6 Register
IOMMU_DM_AUT_CTRL7_REG	0x00CC	IOMMU Domain Authority Control 7 Register
IOMMU_DM_AUT_OVWT_REG	0x00D0	IOMMU Domain Authority Overwrite Register
IOMMU_INT_ENABLE_REG	0x0100	IOMMU Interrupt Enable Register
IOMMU_INT_CLR_REG	0x0104	IOMMU Interrupt Clear Register
IOMMU_INT_STA_REG	0x0108	IOMMU Interrupt Status Register
IOMMU_INT_ERR_ADDR0_REG	0x0110	IOMMU Interrupt Error Address 0 Register
IOMMU_INT_ERR_ADDR1_REG	0x0114	IOMMU Interrupt Error Address 1 Register
IOMMU_INT_ERR_ADDR2_REG	0x0118	IOMMU Interrupt Error Address 2 Register
IOMMU_INT_ERR_ADDR3_REG	0x011C	IOMMU Interrupt Error Address 3 Register
IOMMU_INT_ERR_ADDR4_REG	0x0120	IOMMU Interrupt Error Address 4 Register
IOMMU_INT_ERR_ADDR5_REG	0x0124	IOMMU Interrupt Error Address 5 Register
IOMMU_INT_ERR_ADDR6_REG	0x0128	IOMMU Interrupt Error Address 6 Register

Register Name	Offset	Description
IOMMU_INT_ERR_ADDR7_REG	0x0130	IOMMU Interrupt Error Address 7 Register
IOMMU_INT_ERR_ADDR8_REG	0x0134	IOMMU Interrupt Error Address 8 Register
IOMMU_INT_ERR_DATA0_REG	0x0150	IOMMU Interrupt Error Data 0 Register
IOMMU_INT_ERR_DATA1_REG	0x0154	IOMMU Interrupt Error Data 1 Register
IOMMU_INT_ERR_DATA2_REG	0x0158	IOMMU Interrupt Error Data 2 Register
IOMMU_INT_ERR_DATA3_REG	0x015C	IOMMU Interrupt Error Data 3 Register
IOMMU_INT_ERR_DATA4_REG	0x0160	IOMMU Interrupt Error Data 4 Register
IOMMU_INT_ERR_DATA5_REG	0x0164	IOMMU Interrupt Error Data 5 Register
IOMMU_INT_ERR_DATA6_REG	0x0168	IOMMU Interrupt Error Data 6 Register
IOMMU_INT_ERR_DATA7_REG	0x0170	IOMMU Interrupt Error Data 7 Register
IOMMU_INT_ERR_DATA8_REG	0x0174	IOMMU Interrupt Error Data 8 Register
IOMMU_L1PG_INT_REG	0x0180	IOMMU L1 Page Table Interrupt Register
IOMMU_L2PG_INT_REG	0x0184	IOMMU L2 Page Table Interrupt Register
IOMMU_VA_REG	0x0190	IOMMU Virtual Address Register
IOMMU_VA_DATA_REG	0x0194	IOMMU Virtual Address Data Register
IOMMU_VA_CONFIG_REG	0x0198	IOMMU Virtual Address Configuration Register
IOMMU_PMU_ENABLE_REG	0x0200	IOMMU PMU Enable Register
IOMMU_PMU_CLR_REG	0x0210	IOMMU PMU Clear Register
IOMMU_PMU_ACCESS_LOW0_REG	0x0230	IOMMU PMU Access Low 0 Register
IOMMU_PMU_ACCESS_HIGH0_REG	0x0234	IOMMU PMU Access High 0 Register
IOMMU_PMU_HIT_LOW0_REG	0x0238	IOMMU PMU Hit Low 0 Register
IOMMU_PMU_HIT_HIGH0_REG	0x023C	IOMMU PMU Hit High 0 Register
IOMMU_PMU_ACCESS_LOW1_REG	0x0240	IOMMU PMU Access Low 1 Register
IOMMU_PMU_ACCESS_HIGH1_REG	0x0244	IOMMU PMU Access High 1 Register
IOMMU_PMU_HIT_LOW1_REG	0x0248	IOMMU PMU Hit Low 1 Register
IOMMU_PMU_HIT_HIGH1_REG	0x024C	IOMMU PMU Hit High 1 Register
IOMMU_PMU_ACCESS_LOW2_REG	0x0250	IOMMU PMU Access Low 2 Register
IOMMU_PMU_ACCESS_HIGH2_REG	0x0254	IOMMU PMU Access High 2 Register
IOMMU_PMU_HIT_LOW2_REG	0x0258	IOMMU PMU Hit Low 2 Register
IOMMU_PMU_HIT_HIGH2_REG	0x025C	IOMMU PMU Hit High 2 Register
IOMMU_PMU_ACCESS_LOW3_REG	0x0260	IOMMU PMU Access Low 3 Register
IOMMU_PMU_ACCESS_HIGH3_REG	0x0264	IOMMU PMU Access High 3 Register
IOMMU_PMU_HIT_LOW3_REG	0x0268	IOMMU PMU Hit Low 3 Register
IOMMU_PMU_HIT_HIGH3_REG	0x026C	IOMMU PMU Hit High 3 Register

Register Name	Offset	Description
IOMMU_PMU_ACCESS_LOW4_REG	0x0270	IOMMU PMU Access Low 4 Register
IOMMU_PMU_ACCESS_HIGH4_REG	0x0274	IOMMU PMU Access High 4 Register
IOMMU_PMU_HIT_LOW4_REG	0x0278	IOMMU PMU Hit Low 4 Register
IOMMU_PMU_HIT_HIGH4_REG	0x027C	IOMMU PMU Hit High 4 Register
IOMMU_PMU_ACCESS_LOW5_REG	0x0280	IOMMU PMU Access Low 5 Register
IOMMU_PMU_ACCESS_HIGH5_REG	0x0284	IOMMU PMU Access High 5 Register
IOMMU_PMU_HIT_LOW5_REG	0x0288	IOMMU PMU Hit Low 5 Register
IOMMU_PMU_HIT_HIGH5_REG	0x028C	IOMMU PMU Hit High 5 Register
IOMMU_PMU_ACCESS_LOW6_REG	0x0290	IOMMU PMU Access Low 6 Register
IOMMU_PMU_ACCESS_HIGH6_REG	0x0294	IOMMU PMU Access High 6 Register
IOMMU_PMU_HIT_LOW6_REG	0x0298	IOMMU PMU Hit Low 6 Register
IOMMU_PMU_HIT_HIGH6_REG	0x029C	IOMMU PMU Hit High 6 Register
IOMMU_PMU_ACCESS_LOW7_REG	0x02D0	IOMMU PMU Access Low 7 Register
IOMMU_PMU_ACCESS_HIGH7_REG	0x02D4	IOMMU PMU Access High 7 Register
IOMMU_PMU_HIT_LOW7_REG	0x02D8	IOMMU PMU Hit Low 7 Register
IOMMU_PMU_HIT_HIGH7_REG	0x02DC	IOMMU PMU Hit High 7 Register
IOMMU_PMU_ACCESS_LOW8_REG	0x02E0	IOMMU PMU Access Low 8 Register
IOMMU_PMU_ACCESS_HIGH8_REG	0x02E4	IOMMU PMU Access High 8 Register
IOMMU_PMU_HIT_LOW8_REG	0x02E8	IOMMU PMU Hit Low 8 Register
IOMMU_PMU_HIT_HIGH8_REG	0x02EC	IOMMU PMU Hit High 8 Register
IOMMU_PMU_TL_LOW0_REG	0x0300	IOMMU Total Latency Low 0 Register
IOMMU_PMU_TL_HIGH0_REG	0x0304	IOMMU Total Latency High 0 Register
IOMMU_PMU_ML0_REG	0x0308	IOMMU Max Latency 0 Register
IOMMU_PMU_TL_LOW1_REG	0x0310	IOMMU Total Latency Low 1 Register
IOMMU_PMU_TL_HIGH1_REG	0x0314	IOMMU Total Latency High 1 Register
IOMMU_PMU_ML1_REG	0x0318	IOMMU Max Latency 1 Register
IOMMU_PMU_TL_LOW2_REG	0x0320	IOMMU Total Latency Low 2 Register
IOMMU_PMU_TL_HIGH2_REG	0x0324	IOMMU Total Latency High 2 Register
IOMMU_PMU_ML2_REG	0x0328	IOMMU Max Latency 2 Register
IOMMU_PMU_TL_LOW3_REG	0x0330	IOMMU Total Latency Low 3 Register
IOMMU_PMU_TL_HIGH3_REG	0x0334	IOMMU Total Latency High 3 Register
IOMMU_PMU_ML3_REG	0x0338	IOMMU Max Latency 3 Register
IOMMU_PMU_TL_LOW4_REG	0x0340	IOMMU Total Latency Low 4 Register
IOMMU_PMU_TL_HIGH4_REG	0x0344	IOMMU Total Latency High 4 Register

Register Name	Offset	Description
IOMMU_PMU_ML4_REG	0x0348	IOMMU Max Latency 4 Register
IOMMU_PMU_TL_LOW5_REG	0x0350	IOMMU Total Latency Low 5 Register
IOMMU_PMU_TL_HIGH5_REG	0x0354	IOMMU Total Latency High 5 Register
IOMMU_PMU_ML5_REG	0x0358	IOMMU Max Latency 5 Register
IOMMU_PMU_TL_LOW6_REG	0x0360	IOMMU Total Latency Low 6 Register
IOMMU_PMU_TL_HIGH6_REG	0x0364	IOMMU Total Latency High 6 Register
IOMMU_PMU_ML6_REG	0x0368	IOMMU Max Latency 6 Register

3.11.6 Register Description

3.11.6.1 0x0010 IOMMU Reset Register (Default Value: 0x8003_007F)

Offset: 0x0010			Register Name: IOMMU_RESET_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x1	IOMMU_RESET IOMMU Software Reset Switch 0: Set reset signal 1: Release reset signal Before IOMMU software reset operation, ensure IOMMU never be opened; or all bus operations are completed; or DRAM and the peripherals have opened the corresponding switch, for shielding the effects of IOMMU reset.
30:18	/	/	/
17	R/W	0x1	PC_RST PTW Cache Reset PTW Cache address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When PTW Cache occurs abnormal, the bit is used to reset PTW Cache individually.
16	R/W	0x1	MTLB_RST Macrotlb Reset Macro TLB address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When PTW Cache occurs abnormal, the bit is used to reset PTW Cache individually.

Offset: 0x0010			Register Name: IOMMU_RESET_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
15:7	/	/	/
6	R/W	0x1	M6_RST Master6 Reset Master6 address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When Master6 occurs abnormal, the bit is used to reset PTW Cache individually. **Note: This bit is not used.**
5	R/W	0x1	M5_RST Master5 Reset Master5 address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When Master5 occurs abnormal, the bit is used to reset PTW Cache individually.
4	R/W	0x1	M4_RST Master4 Reset Master4 address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When Master4 occurs abnormal, the bit is used to reset PTW Cache individually.
3	R/W	0x1	M3_RST Master3 Reset Master3 address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When Master3 occurs abnormal, the bit is used to reset PTW Cache individually.
2	R/W	0x1	M2_RST Master2 Reset Master2 address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When Master2 occurs abnormal, the bit is used to reset PTW Cache individually.

Offset: 0x0010			Register Name: IOMMU_RESET_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x1	M1_RST Master1 Reset Master1 address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When Master1 occurs abnormal, the bit is used to reset PTW Cache individually.
0	R/W	0x1	M0_RST Master0 Reset Master0 address convert lane software reset switch. 0: Set reset signal 1: Release reset signal When Master0 occurs abnormal, the bit is used to reset PTW Cache individually.

3.11.6.2 0x0020 IOMMU Enable Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: IOMMU_ENABLE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	ENABLE IOMMU module enable switch 0: Disable IOMMU 1: Enable IOMMU Before IOMMU address mapping function opens, configure the Translation Table Base register; or ensure all masters are in bypass status or no the status of sending bus demand(such as reset)

3.11.6.3 0x0030 IOMMU Bypass Register (Default Value: 0x0000_007F)

Offset: 0x0030			Register Name: IOMMU_BYPASS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6	R/W	0x1	M6_BP Master6 bypass switch

Offset: 0x0030			Register Name: IOMMU_BYPASS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			After bypass function is opened, IOMMU can not map the address of Master6 sending, and directly output the virtual address to MBUS as physical address. 0: Disable bypass function 1: Enable bypass function **Note: The bit is not used.**
5	R/W	0x1	M5_BP Master5 bypass switch After bypass function is opened, IOMMU can not map the address of Master5 sending, and directly output the virtual address to MBUS as physical address. 0: Disable bypass function 1: Enable bypass function
4	R/W	0x1	M4_BP Master4 bypass switch After bypass function is opened, IOMMU can not map the address of Master4 sending, and directly output the virtual address to MBUS as physical address. 0: Disable bypass function 1: Enable bypass function
3	R/W	0x1	M3_BP Master3 bypass switch After bypass function is opened, IOMMU can not map the address of Master3 sending, and directly output the virtual address to MBUS as physical address. 0: Disable bypass function 1: Enable bypass function
2	R/W	0x1	M2_BP Master2 bypass switch After bypass function is opened, IOMMU can not map the address of Master2 sending, and directly output the virtual address to MBUS as physical address. 0: Disable bypass function 1: Enable bypass function
1	R/W	0x1	M1_BP Master1 bypass switch

Offset: 0x0030			Register Name: IOMMU_BYPASS_REG
Bit	Read/Write	Default/Hex	Description
			After bypass function is opened, IOMMU can not map the address of Master1 sending, and directly output the virtual address to MBUS as physical address. 0: Disable bypass function 1: Enable bypass function
0	R/W	0x1	M0_BP Master0 bypass switch After bypass function is opened, IOMMU can not map the address of Master0 sending, and directly output the virtual address to MBUS as physical address. 0: Disable bypass function 1: Enable bypass function

NOTE

- Operating the register belongs to non-accurate timing sequence control function. That is, before the function is valid, master operation will complete address mapping function, and any subsequent operation will not perform address mapping.
- It is suggested that master is in reset state or in no any bus operation before operating the register.

3.11.6.4 0x0040 IOMMU Auto Gating Register (Default Value: 0x0000_0001)

Offset: 0x0040			Register Name: IOMMU_AUTO_GATING_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x1	IOMMU_AUTO_GATING IOMMU circuit auto gating control The purpose is to decrease power consumption of the module. 0: Disable auto gating function 1: Enable auto gating function

3.11.6.5 0x0044 IOMMU Write Buffer Control Register (Default Value: 0x0000_007F)

Offset: 0x0044			Register Name: IOMMU_WBUF_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6	R/W	0x1	Reserved
5	R/W	0x1	Reserved
4	R/W	0x1	Reserved
3	R/W	0x1	Reserved
2	R/W	0x1	Reserved
1	R/W	0x1	Reserved
0	R/W	0x1	Reserved

3.11.6.6 0x0048 IOMMU Out Of Order Control Register (Default Value: 0x0000_007F)

Offset: 0x0048			Register Name: IOMMU_OOO_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6	R/W	0x1	M6_OOO_CTRL Master6 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order **Note: This bit is not used.**
5	R/W	0x1	M5_OOO_CTRL Master5 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order
4	R/W	0x1	M4_OOO_CTRL Master4 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order
3	R/W	0x1	M3_OOO_CTRL Master3 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order
2	R/W	0x1	M2_OOO_CTRL Master2 out-of-order control bit

Offset: 0x0048			Register Name: IOMMU_OOO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
			0: Disable out-of-order 1: Enable out-of-order
1	R/W	0x1	M1_OOO_CTRL Master1 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order
0	R/W	0x1	M0_OOO_CTRL Master0 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order

3.11.6.7 0x004C IOMMU 4KB Boundary Protect Control Register (Default Value: 0x0000_007F)

NOTE

When the virtual address sent by master is over the 4 KB boundary, 4 KB protection unit will split it into two serial access.

Offset: 0x004C			Register Name: IOMMU_4KB_BDY_PRT_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6	R/W	0x1	M6_4KB_BDY_PRT_CTRL Master6 4 KB boundary protect control bit 0: Disable 4 KB boundary protect 1: Enable 4 KB boundary protect **Note: This bit is not used.**
5	R/W	0x1	M5_4KB_BDY_PRT_CTRL Master4 4 KB boundary protect control bit 0: Disable 4 KB boundary protect 1: Enable 4 KB boundary protect
4	R/W	0x1	M4_4KB_BDY_PRT_CTRL Master4 4 KB boundary protect control bit 0: Disable 4 KB boundary protect

Offset: 0x004C			Register Name: IOMMU_4KB_BDY_PRT_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			1: Enable 4 KB boundary protect
3	R/W	0x1	M3_4KB_BDY_PRT_CTRL Master3 4 KB boundary protect control bit 0: Disable 4 KB boundary protect 1: Enable 4 KB boundary protect
2	R/W	0x1	M2_4KB_BDY_PRT_CTRL Master2 4 KB boundary protect control bit 0: Disable 4 KB boundary protect 1: Enable 4 KB boundary protect
1	R/W	0x1	M1_4KB_BDY_PRT_CTRL Master1 4 KB boundary protect control bit 0: Disable 4 KB boundary protect 1: Enable 4 KB boundary protect
0	R/W	0x1	M0_4KB_BDY_PRT_CTRL Master0 4 KB boundary protect control bit 0: Disable 4 KB boundary protect 1: Enable 4 KB boundary protect

3.11.6.8 0x0050 IOMMU Translation Table Base Register (Default Value: 0x0000_0000)

Offset: 0x0050			Register Name: IOMMU_TTB_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:14	R/W	0x0	TTB Translation Table Base Level1 page table starting address, aligned to 16 KB. When operating the register, IOMMU address mapping function must be closed, namely IOMMU_ENABLE_REG is 0; Or Bypass function of all main equipment is set to 1, or no the state of transfer bus commands (such as setting).
13:0	/	/	/

3.11.6.9 0x0060 IOMMU TLB Enable Register (Default Value: 0x0003_007F)

Offset: 0x0060			Register Name: IOMMU_TLB_ENABLE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R/W	0x1	PTW_CACHE_ENABLE PTW Cache enable bit 0: Disable 1: Enable
16	R/W	0x1	MACRO_TLB_ENABLE Macro TLB enable bit 0: Disable 1: Enable
15:7	/	/	/
6	R/W	0x1	MICRO_TLB6_ENABLE Micro TLB6 enable bit 0: Disable 1: Enable
5	R/W	0x1	MICRO_TLB5_ENABLE Micro TLB5 enable bit 0: Disable 1: Enable
4	R/W	0x1	MICRO_TLB4_ENABLE Micro TLB4 enable bit 0: Disable 1: Enable
3	R/W	0x1	MICRO_TLB3_ENABLE Micro TLB3 enable bit 0: Disable 1: Enable
2	R/W	0x1	MICRO_TLB2_ENABLE Micro TLB2 enable bit 0: Disable 1: Enable
1	R/W	0x1	MICRO_TLB1_ENABLE Micro TLB1 enable bit 0: Disable 1: Enable

Offset: 0x0060			Register Name: IOMMU_TLB_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
0	R/W	0x1	MICRO_TLB0_ENABLE Micro TLB0 enable bit 0: Disable 1: Enable

3.11.6.10 0x0070 IOMMU TLB Prefetch Register (Default Value: 0x0000_0000)

Offset: 0x0070			Register Name: IOMMU_TLB_PREFETCH_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R/W	0x1	PF_VL_PT_TO_PC Prefetch Value Pagetable to PTW Cache 0: Disable 1: Enable If the function is enabled, the prefetch function will not update the invalid Level1 page table to PTW cache.
16	R/W	0x1	PF_VL_PT_TO_MT Prefetch Value Pagetable to Macro TLB 0: Disable 1: Enable If the function is enabled, the prefetch function will not update the invalid Level2 page table to Macro TLB.
15:7	/	/	/
6	R/W	0x0	MI_TLB6_PF Micro TLB6 prefetch enable 0: Disable 1: Enable
5	R/W	0x0	MI_TLB5_PF Micro TLB5 prefetch enable 0: Disable 1: Enable
4	R/W	0x0	MI_TLB4_PF Micro TLB4 prefetch enable 0: Disable 1: Enable
3	R/W	0x0	MI_TLB3_PF

Offset: 0x0070			Register Name: IOMMU_TLB_PREFETCH_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			Micro TLB3 prefetch enable 0: Disable 1: Enable **Note: If G2D accesses DDR, it is suggested that disable the prefetch function.**
2	R/W	0x0	MI_TLB2_PF Micro TLB2 prefetch enable 0: Disable 1: Enable
1	R/W	0x0	MI_TLB1_PF Micro TLB1 prefetch enable 0: Disable 1: Enable
0	R/W	0x0	MI_TLB0_PF Micro TLB0 prefetch enable 0: Disable 1: Enable

3.11.6.11 0x0080 IOMMU TLB Flush Enable Register (Default Value: 0x0000_0000)

When performing flush operations, all TLB/Cache access will be paused.

Before flush starts, the operation that has entered TLB continues to complete.

Offset: 0x0080			Register Name: IOMMU_TLB_FLUSH_ENABLE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R/WAC	0x0	PC_FS PTW Cache Flush Clear PTW Cache 0: No clear operation or clear operation is completed 1: Enable clear operation After the Flush operation is completed, the bit can clear automatically.
16	R/WAC	0x0	MA_TLB_FS Macro TLB Flush Clear Macro TLB 0: No clear operation or clear operation is completed

Offset: 0x0080			Register Name: IOMMU_TLB_FLUSH_ENABLE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			1: Enable clear operation After the Flush operation is completed, the bit can clear automatically.
15:7	/	/	/
6	R/WAC	0x0	MI_TLB6_FS Micro TLB6 Flush Clear Micro TLB6 0: No clear operation or clear operation is completed 1: Enable clear operation After the Flush operation is completed, the bit can clear automatically.
5	R/WAC	0x0	MI_TLB5_FS Micro TLB5 Flush Clear Micro TLB5 0: No clear operation or clear operation is completed 1: Enable clear operation After the Flush operation is completed, the bit can clear automatically.
4	R/WAC	0x0	MI_TLB4_FS Micro TLB4 Flush Clear Micro TLB4 0: No clear operation or clear operation is completed 1: Enable clear operation After the Flush operation is completed, the bit can clear automatically.
3	R/WAC	0x0	MI_TLB3_FS Micro TLB3 Flush Clear Micro TLB3 0: No clear operation or clear operation is completed 1: Enable clear operation After the Flush operation is completed, the bit can clear automatically.
2	R/WAC	0x0	MI_TLB2_FS Micro TLB2 Flush Clear Micro TLB2 0: No clear operation or clear operation is completed 1: Enable clear operation

Offset: 0x0080			Register Name: IOMMU_TLB_FLUSH_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
			After the Flush operation is completed, the bit can clear automatically.
1	R/WAC	0x0	MI_TLB1_FS Micro TLB1 Flush Clear Micro TLB1 0: No clear operation or clear operation is completed 1: Enable clear operation After the Flush operation is completed, the bit can clear automatically.
0	R/WAC	0x0	MI_TLB0_FS Micro TLB0 Flush Clear Micro TLB0 0: No clear operation or clear operation is completed 1: Enable clear operation After the Flush operation is completed, the bit can clear automatically.

3.11.6.12 0x0084 IOMMU TLB Invalidation Mode Select Register (Default Value: 0x0000_0000)

Offset: 0x0084			Register Name: IOMMU_TLB_IVLD_MODE_SEL_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	TLB_IVLD_MODE_SEL 0: Invalidate TLB by using the Mask mode 1: Invalidate TLB by using the Start and End mode

3.11.6.13 0x0088 IOMMU TLB Invalidation Start Address Register (Default Value: 0x0000_0000)

Offset: 0x0088			Register Name: IOMMU_TLB_IVLD_STA_ADDR_REG
Bit	Read/Write	Default/Hex	Description
31:12	R/W	0x0	TLB_IVLD_STA_ADDR TLB invalid start address, 4 KB aligned.
11:0	/	/	/

3.11.6.14 0x008C IOMMU TLB Invalidation End Address Register (Default Value: 0x0000_0000)

Offset: 0x008C			Register Name: IOMMU_TLB_IVLD_END_ADDR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	R/W	0x0	TLB_IVLD_END_ADDR TLB invalid end address, 4 KB aligned.
11:0	/	/	/

3.11.6.15 0x0090 IOMMU TLB Invalidation Address Register (Default Value: 0x0000_0000)

Offset: 0x0090			Register Name: IOMMU_TLB_IVLD_ADDR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	R/W	0x0	TLB_IVLD_ADDR TLB invalid address, 4 KB aligned
11:0	/	/	/

Operation:

1) Set the virtual address that needs to be operated in **IOMMU_TLB_IVLD_ADDR_REG**.
2) Set the mask of virtual address that needs to be operated in **IOMMU_TLB_IVLD_ADDR_MASK_REG**.
3) Write '1' to **IOMMU_TLB_IVLD_ENABLE_REG[0].**
4) Read **IOMMU_TLB_IVLD_ENABLE_REG[0]**, when it is '0', it indicates that invalidation behavior is finished.

NOTE

- When performing invalidation operation, TLB/Cache operation has not affected.
- After or before invalidation operation starts, there is no absolute relationship between the same address switch operation and invalidation operation.

3.11.6.16 0x0094 IOMMU TLB Invalidation Address Mask Register (Default Value: 0x0000_0000)

Offset: 0x0094			Register Name: IOMMU_TLB_IVLD_ADDR_MASK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	R/W	0x0	TLB_IVLD_ADDR_MASK TLB invalid address mask register, 4 KB aligned
11:0	/	/	/

3.11.6.17 0x0098 IOMMU TLB Invalidation Enable Register (Default Value: 0x0000_0000)

Offset: 0x0098			Register Name: IOMMU_TLB_IVLD_ENABLE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/WAC	0x0	TLB_IVLD_ENABLE Enable TLB invalidation operation 0: No operation or operation is completed 1: Enable invalidation operation After invalidation operation is completed, the bit can clear automatically. When operating invalidation operation, TLB/Cache operation has not affected. After or before invalidation operation starts, there is no absolute relationship between the same address switch operation and invalidation operation.

3.11.6.18 0x009C IOMMU PC Invalidation Mode Select Register (Default Value: 0x0000_0000)

Offset: 0x009C			Register Name: IOMMU_PC_IVLD_MODE_SEL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	PC_IVLD_MODE_SEL PTW Cache Invalid Mode Select 0: Invalidate PTW by using the Mask mode 1: Invalidate PTW by using the Start and End mode

3.11.6.19 0x00A0 IOMMU PC Invalidation Address Register (Default Value: 0x0000_0000)

Offset: 0x00A0			Register Name: IOMMU_PC_IVLD_ADDR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	R/W	0x0	PC_IVLD_ADDR PTW Cache invalid address, 1 MB aligned.
19:0	/	/	/

3.11.6.20 0x00A4 IOMMU PC Invalidation Start Address Register (Default Value: 0x0000_0000)

Offset: 0x00A4			Register Name: IOMMU_PC_IVLD_STA_ADDR_REG
Bit	Read/Write	Default/Hex	Description
31:20	R/W	0x0	PC_IVLD_SA PTW Cache invalid start address, 1 MB aligned.
19:0	/	/	/

3.11.6.21 0x00A8 IOMMU PC Invalidation Enable Register (Default Value: 0x0000_0000)

Offset: 0x00A8			Register Name: IOMMU_PC_IVLD_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/WAC	0x0	PC_IVLD_ENABLE Enable PTW Cache invalidation operation 0: No operation or operation is completed 1: Enable invalidation operation After invalidation operation is completed, the bit can clear automatically. After or before invalidation operation starts, there is no absolute relationship between the same address switch operation and invalidation operation.

3.11.6.22 0x00AC IOMMU PC Invalidation End Address Register (Default Value: 0x0000_0000)

Offset: 0x00AC			Register Name: IOMMU_PC_IVLD_END_ADDR_REG
Bit	Read/Write	Default/Hex	Description
31:20	R/W	0x0	PC_IVLD_EA PTW Cache invalid end address, 1 MB aligned.
19:0	/	/	/

3.11.6.23 0x00B0 IOMMU Domain Authority Control 0 Register (Default Value: 0x0000_0000)

Software can set 15 different permission control types in IOMMU_DM_AUT_CTRL_REG0–7. A default access control type is DOMAIN0. The read/write operation of DOMAIN1–15 is unlimited by default.

Software needs to set the index of the permission control domain corresponding to the page table item in the bit[7:4] of the Level2 page table, the default value is 0 (use domian0), that is, the read/write operation is not controlled.

Setting REG_ARD_OVWT can mask the Domain control defined by IOMMU_DM_AUT_CTRL_REG0–7. All Level2 page table type are covered by the type of REG_ARD_OVWT. The read/write operation is permitted by default.

Offset: 0x00B0			Register Name: IOMMU_DM_AUT_CTRL0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29	R/W	0x0	DM1_M6_WT_AUT_CTRL Domain1 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited **Note: The bit is not used.**
28	R/W	0x0	DM1_M6_RD_AUT_CTRL Domain1 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited **Note: The bit is not used.**
27	R/W	0x0	DM1_M5_WT_AUT_CTRL Domain1 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
26	R/W	0x0	DM1_M5_RD_AUT_CTRL Domain1 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited
25	R/W	0x0	DM1_M4_WT_AUT_CTRL Domain1 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
24	R/W	0x0	DM1_M4_RD_AUT_CTRL Domain1 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
23	R/W	0x0	DM1_M3_WT_AUT_CTRL Domain1 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
22	R/W	0x0	DM1_M3_RD_AUT_CTRL

Offset: 0x00B0			Register Name: IOMMU_DM_AUT_CTRL0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			Domain1 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
21	R/W	0x0	DM1_M2_WT_AUT_CTRL Domain1 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
20	R/W	0x0	DM1_M2_RD_AUT_CTRL Domain1 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
19	R/W	0x0	DM1_M1_WT_AUT_CTRL Domain1 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited
18	R/W	0x0	DM1_M1_RD_AUT_CTRL Domain1 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
17	R/W	0x0	DM1_M0_WT_AUT_CTRL Domain1 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
16	R/W	0x0	DM1_M0_RD_AUT_CTRL Domain1 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited
15:14	/	/	/
13	R	0x0	DM0_M6_WT_AUT_CTRL Domain0 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited **Note: The bit is not used.**
12	R	0x0	DM0_M6_RD_AUT_CTRL Domain0 read permission control for master6 0: The read-operation is permitted

Offset: 0x00B0			Register Name: IOMMU_DM_AUT_CTRL0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			1: The read-operation is prohibited **Note: The bit is not used.**
11	R	0x0	DM0_M5_WT_AUT_CTRL Domain0 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
10	R	0x0	DM0_M5_RD_AUT_CTRL Domain0 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited
9	R	0x0	DM0_M4_WT_AUT_CTRL Domain0 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
8	R	0x0	DM0_M4_RD_AUT_CTRL Domain0 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
7	R	0x0	DM0_M3_WT_AUT_CTRL Domain0 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
6	R	0x0	DM0_M3_RD_AUT_CTRL Domain0 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
5	R	0x0	DM0_M2_WT_AUT_CTRL Domain0 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
4	R	0x0	DM0_M2_RD_AUT_CTRL Domain0 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
3	R	0x0	DM0_M1_WT_AUT_CTRL Domain0 write permission control for master1

Offset: 0x00B0			Register Name: IOMMU_DM_AUT_CTRL0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			0: The write-operation is permitted 1: The write-operation is prohibited
2	R	0x0	DM0_M1_RD_AUT_CTRL Domain0 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
1	R	0x0	DM0_M0_WT_AUT_CTRL Domain0 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
0	R	0x0	DM0_M0_RD_AUT_CTRL Domain0 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited

3.11.6.24 0x00B4 IOMMU Domain Authority Control 1 Register (Default Value: 0x0000_0000)

Offset: 0x00B4			Register Name: IOMMU_DM_AUT_CTRL1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29	R/W	0x0	DM3_M6_WT_AUT_CTRL Domain3 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited **Note: The bit is not used.**
28	R/W	0x0	DM3_M6_RD_AUT_CTRL Domain3 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited **Note: The bit is not used.**
27	R/W	0x0	DM3_M5_WT_AUT_CTRL Domain3 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
26	R/W	0x0	DM3_M5_RD_AUT_CTRL Domain3 read permission control for master5

Offset: 0x00B4			Register Name: IOMMU_DM_AUT_CTRL1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			0: The read-operation is permitted 1: The read-operation is prohibited
25	R/W	0x0	DM3_M4_WT_AUT_CTRL Domain3 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
24	R/W	0x0	DM3_M4_RD_AUT_CTRL Domain3 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
23	R/W	0x0	DM3_M3_WT_AUT_CTRL Domain3 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
22	R/W	0x0	DM3_M3_RD_AUT_CTRL Domain3 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
21	R/W	0x0	DM3_M2_WT_AUT_CTRL Domain3 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
20	R/W	0x0	DM3_M2_RD_AUT_CTRL Domain3 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
19	R/W	0x0	DM3_M1_WT_AUT_CTRL Domain3 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited
18	R/W	0x0	DM3_M1_RD_AUT_CTRL Domain3 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
17	R/W	0x0	DM3_M0_WT_AUT_CTRL Domain3 write permission control for master0

Offset: 0x00B4			Register Name: IOMMU_DM_AUT_CTRL1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			0: The write-operation is permitted 1: The write-operation is prohibited
16	R/W	0x0	DM3_M0_RD_AUT_CTRL Domain3 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited
15:14	/	/	/
13	R/W	0x0	DM2_M6_WT_AUT_CTRL Domain2 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited **Note: The bit is not used.**
12	R/W	0x0	DM2_M6_RD_AUT_CTRL Domain2 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited **Note: The bit is not used.**
11	R/W	0x0	DM2_M5_WT_AUT_CTRL Domain2 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
10	R/W	0x0	DM2_M5_RD_AUT_CTRL Domain2 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited
9	R/W	0x0	DM2_M4_WT_AUT_CTRL Domain2 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
8	R/W	0x0	DM2_M4_RD_AUT_CTRL Domain2 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
7	R/W	0x0	DM2_M3_WT_AUT_CTRL Domain2 write permission control for master3 0: The write-operation is permitted

Offset: 0x00B4			Register Name: IOMMU_DM_AUT_CTRL1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			1: The write-operation is prohibited
6	R/W	0x0	DM2_M3_RD_AUT_CTRL Domain2 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
5	R/W	0x0	DM2_M2_WT_AUT_CTRL Domain2 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
4	R/W	0x0	DM2_M2_RD_AUT_CTRL Domain2 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
3	R/W	0x0	DM2_M1_WT_AUT_CTRL Domain2 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited
2	R/W	0x0	DM2_M1_RD_AUT_CTRL Domain2 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
1	R/W	0x0	DM2_M0_WT_AUT_CTRL Domain2 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
0	R/W	0x0	DM2_M0_RD_AUT_CTRL Domain2 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited

3.11.6.25 0x00B8 IOMMU Domain Authority Control 2 Register (Default Value: 0x0000_0000)

Offset: 0x00B8			Register Name: IOMMU_DM_AUT_CTRL2_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29	R/W	0x0	DM5_M6_WT_AUT_CTRL

Offset: 0x00B8			Register Name: IOMMU_DM_AUT_CTRL2_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			Domain5 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited **Note: The bit is not used.**
28	R/W	0x0	DM5_M6_RD_AUT_CTRL Domain5 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited **Note: The bit is not used.**
27	R/W	0x0	DM5_M5_WT_AUT_CTRL Domain5 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
26	R/W	0x0	DM5_M5_RD_AUT_CTRL Domain5 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited
25	R/W	0x0	DM5_M4_WT_AUT_CTRL Domain5 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
24	R/W	0x0	DM5_M4_RD_AUT_CTRL Domain5 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
23	R/W	0x0	DM5_M3_WT_AUT_CTRL Domain5 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
22	R/W	0x0	DM5_M3_RD_AUT_CTRL Domain5 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
21	R/W	0x0	DM5_M2_WT_AUT_CTRL Domain5 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited

Offset: 0x00B8			Register Name: IOMMU_DM_AUT_CTRL2_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
20	R/W	0x0	DM5_M2_RD_AUT_CTRL Domain5 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
19	R/W	0x0	DM5_M1_WT_AUT_CTRL Domain5 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited
18	R/W	0x0	DM5_M1_RD_AUT_CTRL Domain5 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
17	R/W	0x0	DM5_M0_WT_AUT_CTRL Domain5 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
16	R/W	0x0	DM5_M0_RD_AUT_CTRL Domain5 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited
15:14	/	/	/
13	R/W	0x0	DM4_M6_WT_AUT_CTRL Domain4 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited **Note: The bit is not used.**
12	R/W	0x0	DM4_M6_RD_AUT_CTRL Domain4 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited **Note: The bit is not used.**
11	R/W	0x0	DM4_M5_WT_AUT_CTRL Domain4 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
10	R/W	0x0	DM4_M5_RD_AUT_CTRL

Offset: 0x00B8			Register Name: IOMMU_DM_AUT_CTRL2_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			Domain4 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited
9	R/W	0x0	DM4_M4_WT_AUT_CTRL Domain4 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
8	R/W	0x0	DM4_M4_RD_AUT_CTRL Domain4 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
7	R/W	0x0	DM4_M3_WT_AUT_CTRL Domain4 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
6	R/W	0x0	DM4_M3_RD_AUT_CTRL Domain4 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
5	R/W	0x0	DM4_M2_WT_AUT_CTRL Domain4 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
4	R/W	0x0	DM4_M2_RD_AUT_CTRL Domain4 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
3	R/W	0x0	DM4_M1_WT_AUT_CTRL Domain4 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited
2	R/W	0x0	DM4_M1_RD_AUT_CTRL Domain4 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
1	R/W	0x0	DM4_M0_WT_AUT_CTRL

Offset: 0x00B8			Register Name: IOMMU_DM_AUT_CTRL2_REG
Bit	Read/Write	Default/Hex	Description
			Domain4 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
0	R/W	0x0	DM4_M0_RD_AUT_CTRL Domain4 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited

3.11.6.26 0x00BC IOMMU Domain Authority Control 3 Register (Default Value: 0x0000_0000)

Offset: 0x00BC			Register Name: IOMMU_DM_AUT_CTRL3_REG
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29	R/W	0x0	DM7_M6_WT_AUT_CTRL Domain7 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited **Note: The bit is not used.**
28	R/W	0x0	DM7_M6_RD_AUT_CTRL Domain7 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited **Note: The bit is not used.**
27	R/W	0x0	DM7_M5_WT_AUT_CTRL Domain7 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
26	R/W	0x0	DM7_M5_RD_AUT_CTRL Domain7 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited
25	R/W	0x0	DM7_M4_WT_AUT_CTRL Domain7 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
24	R/W	0x0	DM7_M4_RD_AUT_CTRL

Offset: 0x00BC			Register Name: IOMMU_DM_AUT_CTRL3_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			Domain7 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
23	R/W	0x0	DM7_M3_WT_AUT_CTRL Domain7 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
22	R/W	0x0	DM7_M3_RD_AUT_CTRL Domain7 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
21	R/W	0x0	DM7_M2_WT_AUT_CTRL Domain7 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
20	R/W	0x0	DM7_M2_RD_AUT_CTRL Domain7 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
19	R/W	0x0	DM7_M1_WT_AUT_CTRL Domain7 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited
18	R/W	0x0	DM7_M1_RD_AUT_CTRL Domain7 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
17	R/W	0x0	DM7_M0_WT_AUT_CTRL Domain7 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
16	R/W	0x0	DM7_M0_RD_AUT_CTRL Domain7 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited
15:14	/	/	/

Offset: 0x00BC			Register Name: IOMMU_DM_AUT_CTRL3_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
13	R/W	0x0	DM6_M6_WT_AUT_CTRL Domain6 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited **Note: The bit is not used.**
12	R/W	0x0	DM6_M6_RD_AUT_CTRL Domain6 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited **Note: The bit is not used.**
11	R/W	0x0	DM6_M5_WT_AUT_CTRL Domain6 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
10	R/W	0x0	DM6_M5_RD_AUT_CTRL Domain6 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited
9	R/W	0x0	DM6_M4_WT_AUT_CTRL Domain6 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
8	R/W	0x0	DM6_M4_RD_AUT_CTRL Domain6 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
7	R/W	0x0	DM6_M3_WT_AUT_CTRL Domain6 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
6	R/W	0x0	DM6_M3_RD_AUT_CTRL Domain6 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
5	R/W	0x0	DM6_M2_WT_AUT_CTRL Domain6 write permission control for master2 0: The write-operation is permitted

Offset: 0x00BC			Register Name: IOMMU_DM_AUT_CTRL3_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			1: The write-operation is prohibited
4	R/W	0x0	DM6_M2_RD_AUT_CTRL Domain6 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
3	R/W	0x0	DM6_M1_WT_AUT_CTRL Domain6 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited
2	R/W	0x0	DM6_M1_RD_AUT_CTRL Domain6 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
1	R/W	0x0	DM6_M0_WT_AUT_CTRL Domain6 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
0	R/W	0x0	DM6_M0_RD_AUT_CTRL Domain6 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited

3.11.6.27 0x00C0 IOMMU Domain Authority Control 4 Register (Default Value: 0x0000_0000)

Offset: 0x00C0			Register Name: IOMMU_DM_AUT_CTRL4_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29	R/W	0x0	DM9_M6_WT_AUT_CTRL Domain9 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited **Note: The bit is not used.**
28	R/W	0x0	DM9_M6_RD_AUT_CTRL Domain9 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited

Offset: 0x00C0			Register Name: IOMMU_DM_AUT_CTRL4_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			Note: The bit is not used.
27	R/W	0x0	DM9_M5_WT_AUT_CTRL Domain9 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
26	R/W	0x0	DM9_M5_RD_AUT_CTRL Domain9 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited
25	R/W	0x0	DM9_M4_WT_AUT_CTRL Domain9 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
24	R/W	0x0	DM9_M4_RD_AUT_CTRL Domain9 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
23	R/W	0x0	DM9_M3_WT_AUT_CTRL Domain9 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
22	R/W	0x0	DM9_M3_RD_AUT_CTRL Domain9 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
21	R/W	0x0	DM9_M2_WT_AUT_CTRL Domain9 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
20	R/W	0x0	DM9_M2_RD_AUT_CTRL Domain9 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
19	R/W	0x0	DM9_M1_WT_AUT_CTRL Domain9 write permission control for master1 0: The write-operation is permitted

Offset: 0x00C0			Register Name: IOMMU_DM_AUT_CTRL4_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			1: The write-operation is prohibited
18	R/W	0x0	DM9_M1_RD_AUT_CTRL Domain9 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
17	R/W	0x0	DM9_M0_WT_AUT_CTRL Domain9 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
16	R/W	0x0	DM9_M0_RD_AUT_CTRL Domain9 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited
15:14	/	/	/
13	R/W	0x0	DM8_M6_WT_AUT_CTRL Domain8 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited **Note: The bit is not used.**
12	R/W	0x0	DM8_M6_RD_AUT_CTRL Domain8 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited **Note: The bit is not used.**
11	R/W	0x0	DM8_M5_WT_AUT_CTRL Domain8 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
10	R/W	0x0	DM8_M5_RD_AUT_CTRL Domain8 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited
9	R/W	0x0	DM8_M4_WT_AUT_CTRL Domain8 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited

Offset: 0x00C0			Register Name: IOMMU_DM_AUT_CTRL4_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
8	R/W	0x0	DM8_M4_RD_AUT_CTRL Domain8 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
7	R/W	0x0	DM8_M3_WT_AUT_CTRL Domain8 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
6	R/W	0x0	DM8_M3_RD_AUT_CTRL Domain8 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
5	R/W	0x0	DM8_M2_WT_AUT_CTRL Domain8 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
4	R/W	0x0	DM8_M2_RD_AUT_CTRL Domain8 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
3	R/W	0x0	DM8_M1_WT_AUT_CTRL Domain8 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited
2	R/W	0x0	DM8_M1_RD_AUT_CTRL Domain8 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
1	R/W	0x0	DM8_M0_WT_AUT_CTRL Domain8 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
0	R/W	0x0	DM8_M0_RD_AUT_CTRL Domain8 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited

3.11.6.28 0x00C4 IOMMU Domain Authority Control 5 Register (Default Value: 0x0000_0000)

Offset: 0x00C4			Register Name: IOMMU_DM_AUT_CTRL5_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29	R/W	0x0	DM11_M6_WT_AUT_CTRL Domain11 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited **Note: The bit is not used.**
28	R/W	0x0	DM11_M6_RD_AUT_CTRL Domain11 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited **Note: The bit is not used.**
27	R/W	0x0	DM11_M5_WT_AUT_CTRL Domain11 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
26	R/W	0x0	DM11_M5_RD_AUT_CTRL Domain11 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited
25	R/W	0x0	DM11_M4_WT_AUT_CTRL Domain11 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
24	R/W	0x0	DM11_M4_RD_AUT_CTRL Domain11 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
23	R/W	0x0	DM11_M3_WT_AUT_CTRL Domain11 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
22	R/W	0x0	DM11_M3_RD_AUT_CTRL Domain11 read permission control for master3

Offset: 0x00C4			Register Name: IOMMU_DM_AUT_CTRL5_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			0: The read-operation is permitted 1: The read-operation is prohibited
21	R/W	0x0	DM11_M2_WT_AUT_CTRL Domain11 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
20	R/W	0x0	DM11_M2_RD_AUT_CTRL Domain11 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
19	R/W	0x0	DM11_M1_WT_AUT_CTRL Domain11 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited
18	R/W	0x0	DM11_M1_RD_AUT_CTRL Domain11 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
17	R/W	0x0	DM11_M0_WT_AUT_CTRL Domain11 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
16	R/W	0x0	DM11_M0_RD_AUT_CTRL Domain11 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited
15:14	/	/	/
13	R/W	0x0	DM10_M6_WT_AUT_CTRL Domain10 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited **Note: The bit is not used.**
12	R/W	0x0	DM10_M6_RD_AUT_CTRL Domain10 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited

Offset: 0x00C4			Register Name: IOMMU_DM_AUT_CTRL5_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			Note: The bit is not used.
11	R/W	0x0	DM10_M5_WT_AUT_CTRL Domain10 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
10	R/W	0x0	DM10_M5_RD_AUT_CTRL Domain10 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited
9	R/W	0x0	DM10_M4_WT_AUT_CTRL Domain10 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
8	R/W	0x0	DM10_M4_RD_AUT_CTRL Domain10 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
7	R/W	0x0	DM10_M3_WT_AUT_CTRL Domain10 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
6	R/W	0x0	DM10_M3_RD_AUT_CTRL Domain10 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
5	R/W	0x0	DM10_M2_WT_AUT_CTRL Domain10 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
4	R/W	0x0	DM10_M2_RD_AUT_CTRL Domain10 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
3	R/W	0x0	DM10_M1_WT_AUT_CTRL Domain10 write permission control for master1 0: The write-operation is permitted

Offset: 0x00C4			Register Name: IOMMU_DM_AUT_CTRL5_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			1: The write-operation is prohibited
2	R/W	0x0	DM10_M1_RD_AUT_CTRL Domain10 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
1	R/W	0x0	DM10_M0_WT_AUT_CTRL Domain10 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
0	R/W	0x0	DM10_M0_RD_AUT_CTRL Domain10 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited

3.11.6.29 0x00C8 IOMMU Domain Authority Control 6 Register (Default Value: 0x0000_0000)

Offset: 0x00C8			Register Name: IOMMU_DM_AUT_CTRL6_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29	R/W	0x0	DM13_M6_WT_AUT_CTRL Domain13 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited
28	R/W	0x0	DM13_M6_RD_AUT_CTRL Domain13 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited
27	R/W	0x0	DM13_M5_WT_AUT_CTRL Domain13 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
26	R/W	0x0	DM13_M5_RD_AUT_CTRL Domain13 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited
25	R/W	0x0	DM13_M4_WT_AUT_CTRL

Offset: 0x00C8			Register Name: IOMMU_DM_AUT_CTRL6_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			Domain13 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
24	R/W	0x0	DM13_M4_RD_AUT_CTRL Domain13 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
23	R/W	0x0	DM13_M3_WT_AUT_CTRL Domain13 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
22	R/W	0x0	DM13_M3_RD_AUT_CTRL Domain13 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
21	R/W	0x0	DM13_M2_WT_AUT_CTRL Domain13 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
20	R/W	0x0	DM13_M2_RD_AUT_CTRL Domain13 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
19	R/W	0x0	DM13_M1_WT_AUT_CTRL Domain13 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited
18	R/W	0x0	DM13_M1_RD_AUT_CTRL Domain13 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
17	R/W	0x0	DM13_M0_WT_AUT_CTRL Domain13 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
16	R/W	0x0	DM13_M0_RD_AUT_CTRL

Offset: 0x00C8			Register Name: IOMMU_DM_AUT_CTRL6_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			Domain13 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited
15:14	/	/	/
13	R/W	0x0	DM12_M6_WT_AUT_CTRL Domain12 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited
12	R/W	0x0	DM12_M6_RD_AUT_CTRL Domain12 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited
11	R/W	0x0	DM12_M5_WT_AUT_CTRL Domain12 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
10	R/W	0x0	DM12_M5_RD_AUT_CTRL Domain12 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited
9	R/W	0x0	DM12_M4_WT_AUT_CTRL Domain12 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
8	R/W	0x0	DM12_M4_RD_AUT_CTRL Domain12 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
7	R/W	0x0	DM12_M3_WT_AUT_CTRL Domain12 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
6	R/W	0x0	DM12_M3_RD_AUT_CTRL Domain12 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited

Offset: 0x00C8			Register Name: IOMMU_DM_AUT_CTRL6_REG
Bit	Read/Write	Default/Hex	Description
5	R/W	0x0	DM12_M2_WT_AUT_CTRL Domain12 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
4	R/W	0x0	DM12_M2_RD_AUT_CTRL Domain12 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited
3	R/W	0x0	DM12_M1_WT_AUT_CTRL Domain12 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited
2	R/W	0x0	DM12_M1_RD_AUT_CTRL Domain12 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
1	R/W	0x0	DM12_M0_WT_AUT_CTRL Domain12 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
0	R/W	0x0	DM12_M0_RD_AUT_CTRL Domain12 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited

3.11.6.30 0x00CC IOMMU Domain Authority Control 7 Register (Default Value: 0x0000_0000)

Offset: 0x00CC			Register Name: IOMMU_DM_AUT_CTRL7_REG
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29	R/W	0x0	DM15_M6_WT_AUT_CTRL Domain15 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited **Note: The bit is not used.**
28	R/W	0x0	DM15_M6_RD_AUT_CTRL

Offset: 0x00CC			Register Name: IOMMU_DM_AUT_CTRL7_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			Domain15 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited **Note: The bit is not used.**
27	R/W	0x0	DM15_M5_WT_AUT_CTRL Domain15 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
26	R/W	0x0	DM15_M5_RD_AUT_CTRL Domain15 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited
25	R/W	0x0	DM15_M4_WT_AUT_CTRL Domain15 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
24	R/W	0x0	DM15_M4_RD_AUT_CTRL Domain15 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
23	R/W	0x0	DM15_M3_WT_AUT_CTRL Domain15 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
22	R/W	0x0	DM15_M3_RD_AUT_CTRL Domain15 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
21	R/W	0x0	DM15_M2_WT_AUT_CTRL Domain15 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
20	R/W	0x0	DM15_M2_RD_AUT_CTRL Domain15 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited