H616 User Manual

High Picture Quality 4K Decoding SoC

Revision 1.0

Dec.28, 2019

DECLARATION

THIS DOCUMENTATION IS THE ORIGINAL WORK AND COPYRIGHTED PROPERTY OF ALLWINNER TECHNOLOGY ("ALLWINNER"). REPRODUCTION IN WHOLE OR IN PART MUST OBTAIN THE WRITTEN APPROVAL OF ALLWINNER AND GIVE CLEAR ACKNOWLEDGEMENT TO THE COPYRIGHT OWNER.

THE PURCHASED PRODUCTS, SERVICES AND FEATURES ARE STIPULATED BY THE CONTRACT MADE BETWEEN ALLWINNER AND THE CUSTOMER. PLEASE READ THE TERMS AND CONDITIONS OF THE CONTRACT AND RELEVANT INSTRUCTIONS CAREFULLY BEFORE USING, AND FOLLOW THE INSTRUCTIONS IN THIS DOCUMENTATION STRICTLY. ALLWINNER ASSUMES NO RESPONSIBILITY FOR THE CONSEQUENCES OF IMPROPER USE(INCLUDING BUT NOT LIMITED TO OVERVOLTAGE, OVERCLOCK, OR EXCESSIVE TEMPERATURE).

THE INFORMATION FURNISHED BY ALLWINNER IS PROVIDED JUST AS A REFERENCE OR TYPICAL APPLICATIONS, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENT DO NOT CONSTITUTE A WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. ALLWINNER RESERVES THE RIGHT TO MAKE CHANGES IN CIRCUIT DESIGN AND/OR SPECIFICATIONS AT ANY TIME WITHOUT NOTICE.

NOR FOR ANY INFRINGEMENTS OF PATENTS OR OTHER RIGHTS OF THE THIRD PARTIES WHICH MAY RESULT FROM ITS USE. NO LICENSE IS GRANTED BY IMPLICATION OR OTHERWISE UNDER ANY PATENT OR PATENT RIGHTS OF ALLWINNER. THIRD PARTY LICENCES MAY BE REQUIRED TO IMPLEMENT THE SOLUTION/PRODUCT. CUSTOMERS SHALL BE SOLELY RESPONSIBLE TO OBTAIN ALL APPROPRIATELY REQUIRED THIRD PARTY LICENCES. ALLWINNER SHALL NOT BE LIABLE FOR ANY LICENCE FEE OR ROYALTY DUE IN RESPECT OF ANY REQUIRED THIRD PARTY LICENCE. ALLWINNER SHALL HAVE NO WARRANTY, INDEMNITY OR OTHER OBLIGATIONS WITH RESPECT TO MATTERS COVERED UNDER ANY REQUIRED THIRD PARTY LICENCE.

Revision History

Revision	Date	Description
1.0	Dec.28, 2019	Initial release version

Contents

Chapter 1 About This Document

Purpose

This document describes the features, logical structures, functions, operating modes, and related registers of each module about H616. For details about the interface timings and related parameters, the pins, pin usages, performance parameters, and package dimension, refer to the ***Allwinner H616 Datasheet.***

Intended Audience

The document is intended for:

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

Symbol Conventions

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

Symbol	Description
WARNING	A warning means that injury or death is possible if the instructions are not obeyed.
CAUTION	A caution means that damage to equipment is possible.
NOTE	Provides additional information to emphasize or supplement important points of the main text.

Notes

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

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.

Numerical System

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

Type	Symbol	Value
Data capacity	1K	1024
	1M	1,048,576
	1G	1,073,741,824
Frequency,data rate	1k	1000
	1M	1,000,000
	1G	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, XX1 indicates 001,011,101 or 111.

Acronyms and Abbreviations

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

ADC	Analog-to-Digital Converter
AE	Automatic Exposure
AEC	Audio Echo Cancellation
AES	Advanced Encryption Standard
AF	Automatic Focus
AGC	Automatic Gain Control
AHB	AMBA High-Speed Bus
ALC	Automatic Level Control
ANR	Active Noise Reduction
APB	Advanced Peripheral Bus
ARM	Advanced RISC Machine
AVS	Audio Video Standard
AWB	Automatic White Balance
BROM	Boot ROM
CIR	Consumer Infrared
CMOS	Complementary Metal-Oxide Semiconductor
CP15	Coprocessor 15
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CVBS	Composite Video Broadcast Signal
DDR	Double Data Rate
DES	Data Encryption Standard
DLL	Delay-Locked Loop
DMA	Direct Memory Access
DRC	Dynamic Range Compression
DVFS	Dynamic Voltage and Frequency Scaling
ECC	Error Correction Code
eFuse	Electrical Fuse, A one-time programmable memory
EHCI	Enhanced Host Controller Interface
eMMC	Embedded Multi-Media Card
ESD	Electrostatic Discharge
FBGA	Fine Pitch Ball Grid Array
FEL	Fireware Exchange Launch
FIFO	First In First Out

GIC	Generic Interrupt Controller
GPIO	General Purpose Input Output
HD	High Definition
HDCP	High-bandwidth Digital Content Protection
I2C	Inter Integrated Circuit
I2S	Inter IC Sound
JEDEC	Joint Electron Device Engineering Council
JPEG	Joint Photographic Experts Group
JTAG	Joint Test Action Group
LCD	Liquid-Crystal Display
LFBGA	Low Profile Fine Pitch Ball Grid Array
LRADC	Low Rate Analog to Digital Converter
LSB	Least Significant Bit
MAC	Media Access Control
MIC	Microphone
MIPI	Mobile Industry Processor Interface
MLC	Multi-Level Cell
MMC	Multimedia Card
MPEG	Motion Pictures Expert Group
MSB	Most Significant Bit
N/A	Not Application
NMI	Non Maskable Interrupt
NTSC	National Television Standards Committee
NVM	Non Volatile Storage Medium
OHCI	Open Host Controller Interface
OSD	On-Screen Display
OTP	One Time Programmable
OWA	One Wire Audio
PAL	Phase Alternating Line
PCM	Pulse Code Modulation
PHY	Physical Layer Controller
PID	Packet Identifier
PLL	Phase-Locked Loop
POR	Power-On Reset
PWM	Pulse Width Modulation
R	Read only/non-Write
RGB	Read Green Blue
RGMII	Reduced Gigabit Media Independent Interface
RMII	Reduced Media Independent Interface
ROM	Read Only Memory
RSA	Rivest-Shamir-Adleman
RTC	Real Time Clock
SAR	Successive Approximation Register
SD	Secure Digital

SDIO	Secure Digital Input Output
SDK	Software Development Kit
SDRAM	Sychronous Dynamic Random Access Memory
SDXC	Secure Digital Extended Capacity
SLC	Single-Level Cell
SoC	System on Chip
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
TDES	Triple Data Encryption Standard
TWI	Two Wire Interface
UART	Universal Asynchronous Receiver Transmitter
UDF	Undefined
USB OTG	Universal Serial Bus On The Go
UTMI	USB2.0 Transceiver Macrocell Interface

Contents

Figures

Chapter 2 Product Description

2.1. Overview

H616 is a high-performance SoC that supports 4K@60fps decoding for over-the-top(OTT) and Internet Protocol television(IPTV) markets. H616 integrates the 4-core 64-bit high performance Cortex™-A53 processor, built-in NEON acceleration engine, powerful CPU processing capabilities to meet a variety of differentiated business requirements. Maintain the best user experience in the industry in terms of stream compatibility, fluency of online video playback, image quality and performance of the whole machine. H616 supports multi formats of video decoder such as H.265, H.264, VP9, AVS2, AVS/AVS+, MPEG-1, MPEG-2, MPEG-4, VC1, VP8, and high-performance H.264 video encoder, which can meet the growing needs of multimedia playback, video communication. H616 also provides rich peripheral interfaces, such as USB2.0, SDIO3.0, 1000Mbps EMAC, TSC, SPI, UART, CIR, etc. H616 adopts the new generation of power consumption technology, and reduces power consumption of 20% than the last generation.

2.2. Features

2.2.1. CPU Architecture

- Quad-core ARM Cortex™-A53 processor
- Power-efficient ARM v8 architecture
- 64 and 32bit execution states for scalable high performance
- Trustzone technology supported
- Supports NEON Advanced SIMD instruction for acceleration of media and signal processing functions
- Large Physical Address Extensions(LPAE)
- VFPv4 Floating Point Unit

2.2.2. GPU Architecture

- G31
- Supports OpenGL ES 1.0/2.0/3.2, Vulkan 1.1, OpenCL 2.0

2.2.3. Memory Subsystem

2.2.3.1. Boot ROM

- On-chip memory
- Supports system boot from the following devices:
  - SD/eMMC(SMHC0, SMHC2)
  - Nand Flash
  - SPI Nor Flash
  - SPI Nand Flash
- Supports secure boot and normal boot
- Supports mandatory upgrade process through SMHC0 and USB
- Secure brom supports load only certified firmware
- Secure brom ensures that the secure boot is a trusted environment

2.2.3.2. SDRAM

- 32-bit DDR4/DDR3/DDR3L/LPDDR3/LPDDR4 interface
- Memory capacity up to 4 GB

2.2.3.3. Nand Flash

- Compliant with ONFI 2.0 and Toggle 2.0
- Up to 80-bit ECC per 1024 bytes
- Supports 1K/2K/4K/8K/16K/32K bytes page size
- Up to 8-bit data bus width
- Supports 2 chip selects, and 2 ready_busy signals
- Supports SLC/MLC/TLC flash and EF-NAND
- Supports SDR/Toggle DDR/ONFI DDR NAND interface

2.2.3.4. SMHC

- Three SD/MMC host controller(SMHC) interfaces
- SMHC0 controls the devices that comply with the Secure Digital (SD3.0)
  - 4-bit bus width
  - SDR mode 50 MHz@3.3V IO pad
  - SDR mode 150 MHz@1.8V IO pad
  - DDR mode 50 MHz@3.3V IO pad
- SMHC1 controls the devices that comply with the Secure Digital Input/Output (SDIO3.0)
  - 4-bit bus width
  - SDR mode 50 MHz@3.3V IO pad
  - SDR mode 150 MHz@1.8V IO pad

  - DDR mode 50 MHz@3.3V IO pad
- SMHC2 controls the devices that comply with the Multimedia Card (eMMC 5.1)
  - 8-bit bus width
  - SDR mode 50 MHz@3.3V IO pad
  - SDR mode 150 MHz@1.8V IO pad
  - DDR mode 50 MHz@3.3V IO pad
  - DDR mode 100 MHz@1.8V IO pad
- Supports hardware CRC generation and error detection
- Supports block size of 1 to 65535 bytes

2.2.4. Video Engine

2.2.4.1. Video Decoding

- Supports video decoding up to 4K@60fps
- Supports multi-formats:
  - H.265 Main10@L5.1 up to 4K@60fps, or 6K@30fps
  - VP9 Profile 2 up to 4K@60fps
  - AVS2 JiZhun 10bit Profile up to 4K@60fps
  - H.264 BP/MP/HP@L4.2 up to 4K@30fps
  - 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
  - VP8 up to 1080p@60fps
  - AVS/AVS+ JiZhun Profile up to 1080p@60fps
  - WMV9/VC1 SP/MP/AP up to 1080p@60fps
  - JPEG HFIF file format up to 45MPPS

2.2.4.2. Video Encoding

- H.264 BP/MP/HP
- H.264 supports I/P frame, and only supports single reference frame
- MJPEG/JPEG baseline
- Maximum 16-megapixel(4096 x 4096) resolution for H.264 encoding
- H.264 encoding capability: 4K@25fps
- JPEG snapshot performance of 1080p@60fps independently
- Supports the constant bit rate(CBR)/variable bit rate(VBR) bit rate control mode,ranging from 256 kbit/s to 100 Mbit/s
- Encoding of eight regions of interest(ROIs)

2.2.5. Video and Graphics

2.2.5.1. Display Engine (DE)

- Output size up to 4096 x 2048
- Six configurable alpha blending channels
- Four overlay layers in each channel, and has an independent scaler
- Potter-duff compatible blending operation
- Supports AFBC buffer
- Supports keystone correction
- Input format: semi-planar YUV422/YUV420/YUV411/P010/P210 and planar YUV422/YUV420/YUV411, ARGB8888/XRGB8888/RGB888/ARGB4444/ARGB1555/RGB565
- Frame Packing/Top-and-Bottom/Side-by-Side Full/Side-by-Side Half 3D format data
- Supports 10-bit processing path for HDR video
- Supports SDR/HDR10/Hybrid-log gamma EOTF and color space conversion
- Supports SmartColor™ 3.3 for excellent display experience
  - Adaptive detail/edge enhancement
  - Adaptive color enhancement and fresh tone protection
  - Adaptive contrast enhancement
  - Adaptive de-noising for compression noise or mosquito noise with YUV420/YUV422 input
- Supports write back only for high efficient dual display and miracast
- Supports output format YUV444/YUV422/YUV420/RGB444 for 10/8bit
- Supports Register Configuration Queue for register update function

2.2.5.2. De-interlacer (DI)

- Supports off-line processing mode only
- Supports 8-bit NV12/NV21/YV12 and planar YUV422/planar YUV422 UV-combined input data format
- Supports 8-bit NV12/NV21/YV12 and planar YUV422/planar YUV422 UV-combined output data format for DIT, and YV12/planar YUV422 output data format for TNR
- Supports video resolution from 32 x 32 to 2048 x 1280 pixel
- Supports weave/pixel-motion-adaptive de-interlace method
- Supports temporal noise reduction function
- Supports film mode detection with video-on-film detection
- Performance: module clock 150MHz for 1080p@60Hz

2.2.5.3. Graphic 2D (G2D)

- Supports layer size up to 2048 x 2048 pixels
- Supports input/output formats: YUV422(semi-planar and planar format)/YUV420(semi-planar and planar format)/P010/P210/P410/Y8/ARGB8888/XRGB8888/RGB888/ARGB4444/ARGB1555/ARGB2101010 and RGB565
- Supports horizontal and vertical flip, clockwise 0/90/180/270 degree rotate

2.2.6. System Peripherals

2.2.6.1. Timer

- The timer module implements the timing and counting functions, including Timer0, Timer1, Watchdog and AVS0, AVS1
- Timer0 and Timer1 for system scheduler counting
  - Configurable 8 prescale factor
  - Programmable 32-bit down timer
  - Supports two working modes: continue mode and single count mode
  - Generates an interrupt when the count is decreased to 0
- 1 Watchdog for transmitting a reset signal to reset the entire system after an exception occurs in the system
  - Supports 12 initial values to configure
  - Generation of timeout interrupts
  - Generation of reset signal
  - Watchdog restart the timing
- 2 AVS counters (AVS0 and AVS1) for synchronizing video and audio in the player
  - Programmable 33-bit up timer
  - Initial value can be updated anytime
  - 12-bit frequency divider factor
  - Pause/Start function

2.2.6.2. High Speed Timer

- One high speed timer with 56-bit counter
- Configurable 5 prescale factor
- Clock source is synchronized with AHB1 clock, much more accurate than other timers
- Supports 2 working modes: continuous mode and single mode
- Generates an interrupt when the count is decreased to 0

2.2.6.3. RTC

- 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
- Supports one solution without low-frequency crystal, a precise 32.768 kHz counter clock can be generated by using HOSC to calibrate the internal RC clock
- Configurable initial value by software anytime
- Periodically alarm to wakeup the external devices
- 16 general purpose registers for storing power-off information

2.2.6.4. GIC

- Supports 16 Software Generated Interrupts(SGIs), 16 Private Peripheral Interrupts(PPIs) and 160 Shared Peripheral Interrupts(SPIs)
- Enabling, disabling, and generating processor interrupts from hardware interrupt
- Interrupt masking and prioritization

2.2.6.5. DMA

- Up to 16-channel DMA
- Interrupt generated for each DMA channel
- Flexible data width of 8/16/32/64-bit
- 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 wait mode and handshake mode
- DMA channel supports pause function

2.2.6.6. CCU

- 12 PLLs
- One on-chip RC oscillator and one external 24 MHz DCXO
- Supports clock configuration and clock generated for corresponding modules
- Supports software-controlled clock gating and software-controlled reset for corresponding modules

2.2.6.7. Thermal Sensor Controller

- Temperature accuracy: ±3°C from 0°C to +100°C, ±5°C from -25°C to +125°C
- Supports over-temperature protection interrupt and over-temperature alarm interrupt
- Averaging filter for thermal sensor reading
- Four thermal sensors: sensor0 located in the GPU, sensor1 located in the VE, sensor2 located in the CPU and sensor3 located in the DDR

2.2.6.8. CPU Configuration

- Capable of CPU reset, including core reset, debug circuit reset, etc
- Capable of other CPU-related control, including interface control and CP15 control
- Capable of checking CPU status, including idle status, SMP status, and interrupt status, etc
- Including CPU debug control and status register

2.2.6.9. IOMMU

- Supports virtual address to physical address mapping by hardware implementation
- Supports DE, DI, VE_R, VE, G2D parallel address mapping
- Supports DE, DI, VE_R, VE, G2D bypass function independently
- Supports DE, DI, VE_R, VE, G2D prefetch independently
- Supports DE, DI, VE_R, VE, G2D 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

2.2.7. Video Output

2.2.7.1. TCON_TV

- Supports 10-bit pixel depth YUV422/YUV420, HV format output up to 4K@60Hz
- Supports 8-bit pixel depth YUV444, HV format output up to 4K@60Hz

2.2.7.2. TVE

- Supports 1-ch TV CVBS output
- Supports NTSC and PAL mode
- Plug status auto detecting

2.2.7.3. HDMI

- Compatible with HDCP 2.2 and HDCP 1.4
- Supports DDC and SCDC
- Integrated CEC hardware engine
- Video support:
  - 2D Video: 4K/1080P/1080I/720P/576P/480P/576I/480I, up to 4K@60fps
  - 3D Video: 4K/1080P/720P/576P/480P, up to 4K@30fps
  - Supports RGB/YUV444/YUV422/YUV420 output
  - Color depth: 8/10-bit
  - HDR10: compliant with CTA-861.3 and SMPTE ST 2048
- Audio support:
  - Uncompressed audio formats: IEC60985 L-PCM audio samples, up to 192 kHz
  - Compressed audio formats: IEC61937 compressed audio, up to 1536 kHz

2.2.8. Audio Subsystem

2.2.8.1. Audio Codec

- Two audio digital-to-analog(DAC) channels
  - Supports 16-bit and 20-bit sample resolution
  - 8 kHz to 192 kHz DAC sample rate
  - 95±2dB SNR@A-weight, -80±3dB THD+N, output Level more than 0.55Vrms
- One audio output:
  - One differential LINEOUTP/N or single-ended LINEOUTL/R output
- One low-noise analog microphone bias output
- Supports Dynamic Range Controller adjusting the DAC playback and ADC capture
- One 128x24-bits FIFO for DAC data transmit, one 128x24-bits FIFO for ADC data receive
- Programmable FIFO thresholds
- DMA and Interrupt support

2.2.8.2. Audio HUB

- One Audio HUB
- Supports 2 Digital Audio MIXER(DAM)
- Supports 3 I2S/PCM interfaces for connecting external devices, and 1 I2S/PCM for connecting internal HDMI
- Supports Left-justified, Right-justified, Standard I2S mode, PCM mode, and TDM mode
- I2S mode supports 8 channels, and 32-bit/192 kbit sample rate
- I2S and TDM modes support maximum 16 channels, and 32-bit/96 kbit sample rate

2.2.8.3. DMIC

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

2.2.8.4. OWA

- One OWA TX
- IEC-60958 transmitter functionality
- Supports channel status insertion for the transmitter
- Hardware parity generation on the transmitter
- One 128×24bits TXFIFO for audio data transfer
- Programmable FIFO thresholds
- Interrupt and DMA support
- Supports 16-bit, 20-bit and 24-bit data formats

2.2.9. Security Engine

2.2.9.1. Crypto Engine (CE)

- Supports Symmetrical algorithm: AES, DES, TDES, XTS
  - ECB, CBC, CTS, CTR, CFB, OFB, CBC-MAC mode for AES
  - 128/192/256-bit key for AES
  - 256-bit, 512-bit key for XTS
  - ECB, CBC, CTR, CBC-MAC mode for DES/TDES
- Supports Hash algorithm: MD5, SHA, HMAC
  - SHA1, SHA224, SHA256, SHA384, SHA512 for SHA
  - HMAC-SHA1, HMAC-SHA256 for HMAC
  - MD5, SHA, HMAC are padded using hardware
- Supports Asymmetrical algorithm: RSA, ECC
  - RSA supports 512/1024/2048/4096-bit width
  - ECC supports 160/224/256/384/521-bit width
- Supports 160-bit hardware PRNG
- Supports 256-bit hardware TRNG
- Internal embedded DMA to do data transfer
- Supports secure and non-secure interfaces respectively
- Supports task chain mode for each request. Task or task chain are executed at request order
- 8 scatter group(sg) are supported for both input and output data
- DMA has multiple channels, each corresponding to one suit of algorithm

2.2.9.2. Security ID

- Supports one EFUSE for chip ID and security application
- EFUSE has secure zone and non-secure zone

2.2.9.3. Secure Memory Control (SMC)

- The SMC is always secure, only secure CPU can access the SMC
- Set secure area of DRAM
- Set secure property that Master accesses to DRAM
- Set DRM area
- Set whether DRM master can access to DRM area or not

2.2.9.4. Secure Peripherals Control (SPC)

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

2.2.10. External Peripherals

2.2.10.1. USB

- One USB 2.0 OTG (USB0), with integrated USB 2.0 analog PHY
  - Compatible with USB2.0 Specification
  - Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s) and Low-Speed (LS, 1.5 Mbit/s) in host mode
  - Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s) in device mode
  - Compatible with Enhanced Host Controller Interface (EHCI) Specification, Version 1.0, and the Open Host Controller Interface (OHCI) Specification, Version 1.0a for host mode
  - Up to 8 User-Configurable Endpoints (EPs) for Bulk, Isochronous and Interrupt bi-directional transfers
  - Supports (4 KB + 64 bytes) FIFO for all EPs (including EP0)
  - Supports point-to-point and point-to-multipoint transfer in both host and peripheral mode
- Three USB 2.0 HOST (USB1, USB2, USB3), with integrated USB 2.0 analog PHY
  - Compatible with Enhanced Host Controller Interface (EHCI) Specification, Version 1.0, and the Open Host Controller Interface (OHCI) Specification, Version 1.0a.
  - Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s) and Low-Speed (LS,1.5 Mbit/s) device
  - Only USB2 supports USB standby

2.2.10.2. EMAC

- Two EMAC interfaces
  - EMAC0: 10/100/1000 Mbps Ethernet port with RGMII and RMII interfaces, for connecting the external EPHY
  - EMAC1: 10/100 Mbps Ethernet port with RMII interface, and it embedded with 100M EPHY
  - EMAC1 has no external pins
  - EMAC0 and EMAC1 can use at the same time
- Compliant with IEEE 802.3-2002 standard
- Supports both full-duplex and half-duplex operation
- Supports MDIO
- 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

2.2.10.3. UART

- Up to 6 UART controllers (UART0, UART1, UART2, UART3, UART4, UART5)

- UART0, UART5: 2-wire; UART1, UART2, UART3, UART4: 4-wire
- 2-wire UART can be used for printing; 4-wire UART can be used for flow control
- Compatible with industry-standard 16550 UARTs
- Capable of speed up to 4 Mbit/s
- Supports 5 to 8 data bits and 1/1.5/2 stop bits
- Supports even, odd or no parity
- Supports DMA controller interface
- Supports software/hardware flow control
- Supports IrDA 1.0 SIR
- Supports RS-485/9-bit mode

2.2.10.4. SPI

- Up to 2 SPI controllers (SPI0,SPI1)
- Full-duplex synchronous serial interface
- Master/slave configurable
- Mode0~3 are supported for both transmit and receive operations
- Two 64 bytes FIFO for both transmit and receive data
- Polarity and phase of the Chip Select (SPI-CS) and SPI Clock (SPI-CLK) are configurable
- Interrupt or DMA support
- 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

2.2.10.5. Two Wire Interface (TWI)

- Up to 6 TWI controllers (TWI0, TWI1, TWI2, TWI3, TWI4, S_TWI0)
- Software-programmable for slave or master
- Supports repeated START signal
- Multi-master system supported
- Allows 10-bit addressing transactions
- Performs arbitration and clock synchronization
- Own address and General Call address detection
- Interrupt on address detection
- Supports Standard mode(up to 100 kbit/s) and Fast mode(up to 400 kbit/s)
- Allows operation from a wide range of input clock frequency
- TWI driver supports packet transmission and DMA when TWI works in Master mode

2.2.10.6. CIR Receiver

- Full physical layer implementation
- Supports NEC format infrared data

- Supports CIR for remote control or wireless keyboard
- 64x8 bits FIFO for data buffer
- Sample clock up to 1 MHz

2.2.10.7. PWM

- 4 PWM channels(PWM1, PWM2, PWM3, PWM4)
- PWM23 pair consists of PWM2 and PWM3
- PWM23 pair supports deadzone function
- PWM1/PWM4 has the single channel characteristics of PWM module, and has no pair function
- Supports pulse, cycle and complementary pair output
- Supports capture input
- Build-in the programmable dead-time generator, controllable dead-time
- Three kinds of output waveform: continuous waveform, pulse waveform and complementary pair
- Output frequency range: 0~24 MHz/100 MHz
- Various duty-cycle: 0%~100%
- Minimum resolution: 1/65536
- Interrupt generation of PWM output and capture input

2.2.10.8. Low Rate ADC (LRADC)

- One LRADC input channel
- 6-bit resolution
- Sample rate up to 2 kHz
- Supports hold Key and general Key
- Supports normal, continue and single work mode
- Power supply voltage: 1.8 V, power reference voltage: 1.35 V, analog input and detected voltage range: 0~LEVELB (the maximum value is 1.266 V)

2.2.10.9. TSC

- Supports SPI/SSI interface, interface timing parameters are configurable
- 32 channels PID filter for each TSF
- Supports multiple transport stream packet (188, 192, 204) format
- Hardware packet synchronous byte error detecting
- Hardware PCR packet detecting
- 64x16-bits FIFO for TSG, 64x32-bits FIFO for TSF
- Configurable SPI transport stream generator for streams in DRAM memory
- Supports DVB-CSA V1.1, DVB-CSA V2.1 Descrambler

2.2.11. Package

- TFBGA284 balls, 0.65 mm ball pitch, 0.35 mm ball size, 14 mm x 12 mm body

2.3. Block Diagram

Figure 2-1 shows the system block diagram of the H616.

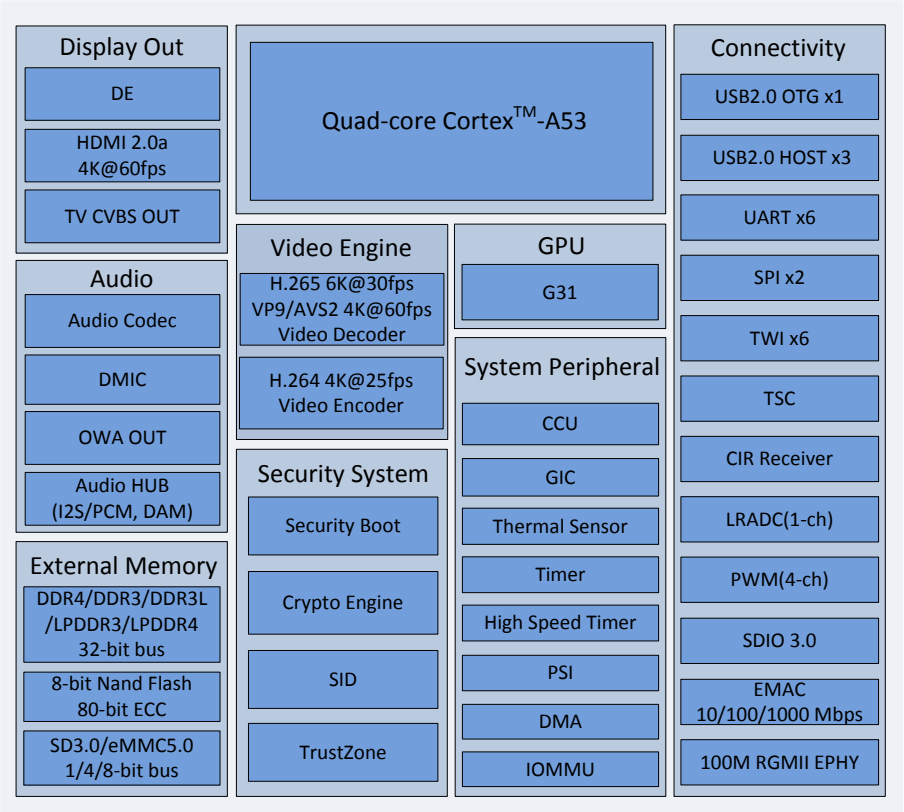

Figure 2- 1. H616 System Block Diagram

Figure 2-2 shows the OTT Box solution of the H616.

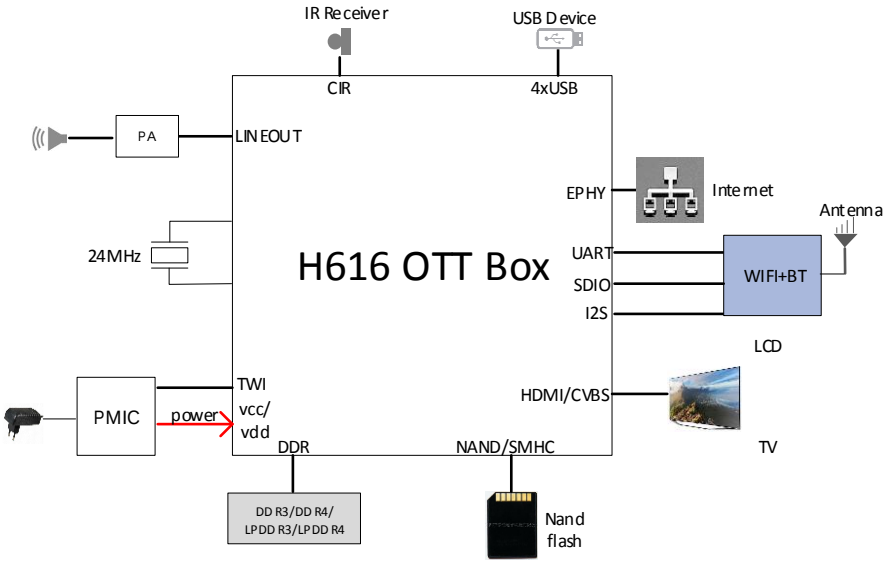

Figure 2- 2. H616 OTT Box Solution

Contents

Figures

Tables

Chapter 3 System

3.1. Memory Mapping

Module	Address(It is for Cluster CPU)	Size(Bytes)
BROM	0x0000 0000---0x0000 FFFF	64K
SRAM A1	0x0002 0000---0x0002 7FFF	32K(support Byte operation, clock source is AHB1)
SRAM C	0x0002 8000---0x0005 7FFF	Borrow VE 128K, DE 64K, supports Byte operation, clock source is AHB1
Accelerator		
DE	0x0100 0000---0x013F FFFF	4M
DI0	0x0142 0000---0x0145 FFFF	256K
G2D	0x0148 0000---0x014B FFFF	256K
GPU	0x0180 0000---0x0183 FFFF	256K
CE_NS	0x0190 4000---0x0190 47FF	2K
CE_S	0x0190 4800---0x0190 4FFF	2K
CE_KEY_SRAM	0x0190 8000---0x0190 8FFF	4K
VE SRAM	0x01A0 0000---0x01BF FFFF	2M
VE	0x01C0 E000---0x01C0 FFFF	8K
System Resources		
SYS_CFG	0x0300 0000---0x0300 0FFF	4K
CCU	0x0300 1000---0x0300 1FFF	4K
DMA	0x0300 2000---0x0300 2FFF	4K
HSTIMER	0x0300 5000---0x0300 5FFF	4K
SID	0x0300 6000---0x0300 6FFF	4K
SMC	0x0300 7000---0x0300 7FFF	4K
SPC	0x0300 8000---0x0300 83FF	1K
TIMER	0x0300 9000---0x0300 93FF	1K
PWM	0x0300 A000---0x0300 A3FF	1K
GPIO	0x0300 B000---0x0300 B3FF	1K
PSI	0x0300 C000---0x0300 C3FF	1K
GIC	0x0302 0000---0x0302 FFFF	64K
IOMMU	0x030F 0000---0x030F FFFF	64K
RTC	0x0700 0000---0x0700 03FF	1K
PRCM	0x0701 0000---0x0701 03FF	1K
TWD	0x0702 0800 – 0x0702 0BFF	1K
Memory		

NAND0	0x0401 1000---0x0401 1FFF	4K
SMHC0	0x0402 0000---0x0402 0FFF	4K
SMHC1	0x0402 1000---0x0402 1FFF	4K
SMHC2	0x0402 2000---0x0402 2FFF	4K
MSI_CTRL	0x047F A000---0x047F AFFF	4K
DRAM_CTRL	0x047F B000---0x047F FFFF	20K
PHY_CTRL	0x0480 0000---0x04FF FFFF	8M
Interfaces		
UART0	0x0500 0000---0x0500 03FF	1K
UART1	0x0500 0400---0x0500 07FF	1K
UART2	0x0500 0800---0x0500 0BFF	1K
UART3	0x0500 0C00---0x0500 0FFF	1K
UART4	0x0500 1000---0x0500 13FF	1K
UART5	0x0500 1400---0x0500 17FF	1K
TWI0	0x0500 2000---0x0500 23FF	1K
TWI1	0x0500 2400---0x0500 27FF	1K
TWI2	0x0500 2800---0x0500 2BFF	1K
TWI3	0x0500 2C00---0x0500 2FFF	1K
TWI4	0x0500 3000---0x0500 33FF	1K
S_TWI0	0x0708 1400---0x0708 17FF	1K
SPI0	0x0501 0000---0x0501 0FFF	4K
SPI1	0x0501 1000---0x0501 1FFF	4K
EMAC0	0x0502 0000---0x0502 FFFF	64K
EMAC1	0x0503 0000---0x0503 FFFF	64K
TS0	0x0506 0000---0x0506 0FFF	4K
THS	0x0507 0400---0x0507 07FF	1K
LRADC	0x0507 0800---0x0507 0BFF	1K
OWA	0x0509 3000---0x0509 33FF	1K
DMIC	0x0509 5000---0x0509 53FF	1K
Audio Codec	0x0509 6000---0x0509 6FFF	4K
Audio HUB	0x0509 7000---0x0509 7FFF	4K
USB0(USB2.0_OTG)	0x0510 0000---0x051F FFFF	1M
USB1(USB2.0_HOST1)	0x0520 0000---0x052F FFFF	1M
USB2(USB2.0_HOST2)	0x0531 0000---0x0531 0FFF	4K
USB3(USB2.0_HOST3)	0x0531 1000---0x0531 1FFF	4K
CIR_RX	0x0704 0000---0x0704 03FF	1K
Display		
HDMI_TX0(1.4/2.0)	0x0600 0000---0x060F FFFF	1M
DISP_IF_TOP	0x0651 0000---0x0651 0FFF	4K
TCON_TV0	0x0651 5000---0x0651 5FFF	4K
TCON_TV1	0x0651 6000---0x0651 6FFF	4K
TVE_TOP	0x0652 0000---0x0652 3FFF	16K
TVE0	0x0652 4000---0x0652 7FFF	16K
CPUX Related		

CPU_SUBSYS_CFG	0x0810 0000---0x0810 03FF	1K
TIMESTAMP_STU	0x0811 0000---0x0811 0FFF	4K
TIMESTAMP_CTRL	0x0812 0000---0x0812 0FFF	4K
IDC	0x0813 0000---0x0813 0FFF	3K
C0_CPUX_CFG	0x0901 0000---0x0901 03FF	1K
C0_CPUX_MBIST	0x0902 0000---0x0902 0FFF	4K
DRAM		
DRAM	0x4000 0000---0x13FFF FFFF	4G

3.2. CPUX Configuration

3.2.1. Overview

The C0_CPUX_CFG module is used for configuring cluster0, such as reset, control, cache, debug, CPU status.
The CPU_SUBSYS_CTRL module is used for the system resource control of CPU sub-system, such as GIC-400, JTAG.

The CPUX_CFG includes the following features:

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

3.2.2. Operations and Functional Descriptions

3.2.2.1. Signal Description

For the detail of CPUX signal, please refer to ***ARM Cortex-A53 TRM***, such as DDI0464F_cortex_A53_mpcore_r0p5_trm.pdf

3.2.2.2. L2 Idle Mode

When the L2 cache of Cluster needs to enter WFI mode, firstly make sure that the CPU[3:0] of Cluster enters WFI mode, which can be checked through the bit[19:16] of **Cluster CPU Status Register**, and then pull high the **ACINACTM** of Cluster by writing 1 to the bit0 of **Cluster Control Register1**, and then check whether L2 enters idle status by checking whether the **STANDBYWFIL2** is high. Note that set the **ACINACTM** to low when exiting the L2 idle mode.

3.2.2.3. CPUX Reset System

The CPUX reset includes **core reset**, **power-on reset** and **H_Reset**. And their scopes rank: **core reset** < **power-on Reset** < **H_Reset**. The description of all reset signal in CPUX reset system is as follows.

Table 3- 1. Reset Signal Description

Reset Signal	Description
CORE_RST	This is the primary reset signal which can reset the corresponding core logic including NEON, VFP, Debug, ETM, breakpoint and watchpoint logic. It maps to a warm reset that covers 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 reset of the processor logic and the integrated debug functionality. This does not reset debug logic in the debug power domain. Including CORE_RST/ETM_RST/DBG_RST.
AXI2MBUS_RST	Reset 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	Reset ETM debug logic circuit.
DBG_RST	Reset only the debug, and breakpoint and watchpoint logic in the processor power domain. It also resets the debug logic for each processor in the debug power domain.
SOC_DBG_RST	Reset all the debug logic including DBG_RST.
MBIST_RST	Reset all resettable registers in the cluster, for entry into, and exit from, MBIST mode.
H_RST	Including PWRON_RST/L2_RST/MBIST_RST/SOC_DBG_RST/C0_CPUX_CFG.
CPU_SUBSYS_RST	Including C0_H_RST/GIC-400/CPU_SUBSYS_CTRL.

3.2.2.4. CPUX Power Block Diagram

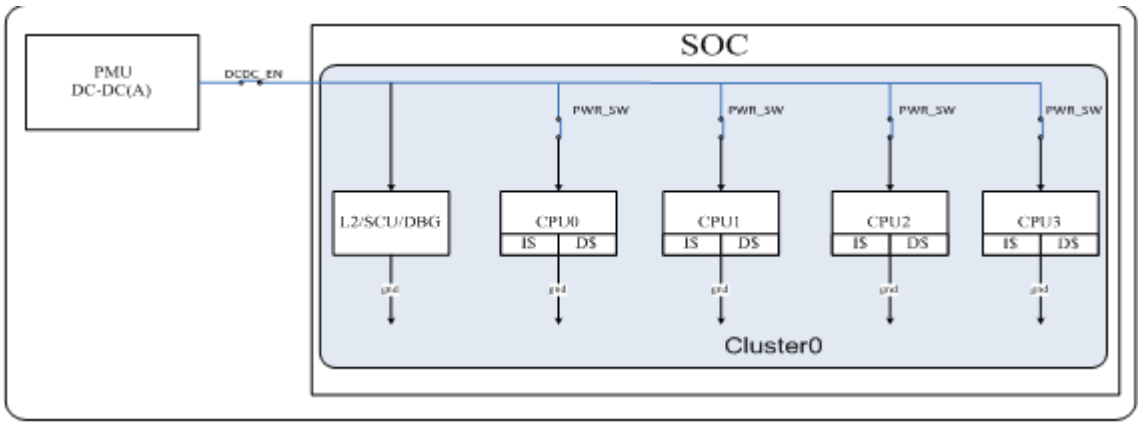

Figure 3- 1. CPUX Power Domain Block Diagram

Figure 3-1 lists the power domain of cluster in default. The power switch of all CPU core are power-on, the pwron_rst of all CPU core are de-asserted, the core reset of CPU0 is de-asserted, the core reset of CPU [3:1] is asserted.

Since each CPU core and its appended circuits have the same power domain, the processor and related L1 cache, neon and vfp should be taken as a whole core.

C0_CPUX_CFG and cluster0 belong to the same power domain, within opening and closing cluster0 process, when cluster0 starts to power on again from power-off state, C0_CPUX_CFG holds in default state, at this time software need initial C0_CPUX_CFG after C0_H_RST is de-asserted.

CPU_SUBSYS_CTRL belongs to system power domain. The power domains of CPU related module are as follows.

Power Domain	Modules	Description
Cluster0	Cluster0/C0_CPUX_CFG/C0_MBIST	Cluster0 circuit, C0_CPUX_CFG module and CPU reset/power/mbist

System	Timestamp/GIC/CPU_SUBSYS_CTRL/Clock	Provide system source of CPU sub-system

3.2.2.5. Operation Principle

The CPU-related operations (such as open/close core, cluster switch, status query) need proper configuration of C0_CPUX_CFG module, as well as the combination of related system control resources including BUS, clock.

3.2.3. Programming Guidelines

For CPU core and cluster operation, please see the ***H616_CPU_AP_Note***.

3.2.4. 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
RVBARADDR0_L	0x0040	Reset Vector Base Address Register0_L
RVBARADDR0_H	0x0044	Reset Vector Base Address Register0_H
RVBARADDR1_L	0x0048	Reset Vector Base Address Register1_L
RVBARADDR1_H	0x004C	Reset Vector Base Address Register1_H
RVBARADDR2_L	0x0050	Reset Vector Base Address Register2_L
RVBARADDR2_H	0x0054	Reset Vector Base Address Register2_H
RVBARADDR3_L	0x0058	Reset Vector Base Address Register3_L
RVBARADDR3_H	0x005C	Reset Vector Base Address Register3_H
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

3.2.5. Cluster 0 Configuration Register Description

3.2.5.1. 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
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[3:0] Reset Assert 0: assert 1: de-assert

3.2.5.2. 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 of barriers onto system bus 0: Barriers are broadcasted onto system bus, this requires an AMBA4 interconnect 1: Barriers are not broadcasted onto the system bus. This is compatible with an AXI3 interconnect
30	R/W	0x0	BROADCAST_INNER Enable broadcasting of inner shareable transactions 0: Inner shareable transactions are not broadcasted externally 1: Inner shareable transactions are broadcasted externally
29	R/W	0x0	BROADCAST_OUTER Enable broadcasting of 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 of 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
27:24	R/W	0x0	AA64Naa32 Register width state 0 : AArch32 1 : AArch64 This pin is sampled only during reset of the processor
23:12	/	/	/
11:8	R/W	0x0	CP15S_DISABLE Disable write access to some secure CP15 register.
7:5	/	/	/
4	R/W	0x0	L2_RST_DISABLE Disable automatic L2 cache invalidate 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[3:0] L1 cache invalidate at reset: 0: L1 cache is reset by hardware. 1: L1 cache is not reset by hardware.

3.2.5.3. 0x0014 Cluster 0 Control Register1(Default Value: 0x0000_0000)

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

1	R/W	0x0	CRM auto select slow frequency enable 0: disable auto select 1: enable auto select
0	R/W	0x0	ACINACTM Snoop interface is inactive and no longer accepts request 0: Snoop interface is active 1: Snoop interface is inactive

3.2.5.4. 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 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[3:0] Clear the status of interface.
19:0	/	/	/

3.2.5.5. 0x0024 Cache Configuration Register(Default Value: 0x001A_001A)

Offset: 0x0024			Register Name: CACHE_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21:19	R/W	0x3	EMA_L2D L2 Cache SRAM EMA control port
18:17	R/W	0x1	EMAW_L2D L2 Cache SRAM EMAW control port
16	R/W	0x0	EMAS_L2D L2 Cache SRAM EMAS control port
15:6	/	/	/
5:3	R/W	0x3	EMA Cache SRAM EMA control port
2:1	R/W	0x1	EMAW Cache SRAM EMAW control port
0	R/W	0x0	EMAS Cache SRAM EMAS control port

3.2.5.6. 0x0040 Reset Vector Base Address Register0_L(Default Value: 0x0000_0000)

Offset: 0x0040			Register Name: RVBARADDR0_L
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	R/W	0x0	RVBARADDR[31:2] Reset Vector Base Address[31:2] for executing in 64-bit state (AArch64) of CPU0.
1:0	/	/	/

3.2.5.7. 0x0044 Reset Vector Base Address Register0_H(Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: RVBARADDR0_H
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	RVBARADDR[39:32] Reset Vector Base Address[39:32] for executing in 64-bit state (AArch64) of CPU0.

3.2.5.8. 0x0048 Reset Vector Base Address Register1_L(Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: RVBARADDR1_L
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	R/W	0x0	RVBARADDR[31:2] Reset Vector Base Address[31:2] for executing in 64-bit state (AArch64) of CPU1.
1:0	/	/	/

3.2.5.9. 0x004C Reset Vector Base Address Register1_H(Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: RVBARADDR1_H
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	RVBARADDR[39:32] Reset Vector Base Address[39:32] for executing in 64-bit state (AArch64) of CPU1.

3.2.5.10. 0x0050 Reset Vector Base Address Register2_L(Default Value: 0x0000_0000)

Offset: 0x0050			Register Name: RVBARADDR2_L
Bit	**Read/Write**	**Default/Hex**	**Description**

31:2	R/W	0x0	RVBARADDR[31:2] Reset Vector Base Address[31:2] for executing in 64-bit state (AArch64) of CPU2.
1:0	/	/	/

3.2.5.11. 0x0054 Reset Vector Base Address Register2_H(Default Value: 0x0000_0000)

Offset: 0x0054			Register Name: RVBARADDR2_H
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	RVBARADDR[39:32] Reset Vector Base Address[39:32] for executing in 64-bit state (AArch64) of CPU2.

3.2.5.12. 0x0058 Reset Vector Base Address Register3_L(Default Value: 0x0000_0000)

Offset: 0x0058			Register Name: RVBARADDR3_L
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	R/W	0x0	RVBARADDR[31:2] Reset Vector Base Address[31:2] for executing in 64-bit state (AArch64) of CPU3.
1:0	/	/	/

3.2.5.13. 0x005C Reset Vector Base Address Register3_H(Default Value: 0x0000_0000)

Offset: 0x005C			Register Name: RVBARADDR3_H
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	RVBARADDR[39:32] Reset Vector Base Address[39:32] for executing in 64-bit state (AArch64) of CPU3.

3.2.5.14. 0x0080 Cluster0 CPU Status Register(Default Value: 0x0001_0000)

Offset: 0x0080			Register Name: C0_CPU_STATUS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R	0x0	SMP_AMP CPU[3:0] is in Symmetric Multiprocessing mode or Asymmetric Multiprocessing mode.

			0: AMP mode 1: SMP mode
23:20	/	/	/
19:16	R	0x1	STANDBYWFI Indicates if Cluster CPU[3:0] is in WFI standby mode 0: Processor not in WFI standby mode. 1: Processor in WFI standby mode
15:12	/	/	/
11:8	R	0x0	STANDBYWFE Indicates if Cluster CPU[3:0] is in the WFE standby mode 0: Processor not in WFE standby mode 1: Processor in WFE standby mode
7:1	/	/	/
0	R	0x0	STANDBYWFIL2 Indicates if the Cluster L2 memory system is in WFI standby mode 0: Cluster L2 not in WFI standby mode 1: Cluster L2 in WFI standby mode

3.2.5.15. 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	/	/	/

3.2.5.16. 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[3:0] External restart requests.
7:4	/	/	/
3:0	R/W	0x1	C_DBGPWRDUP[3:0] Cluster Powered-up 0: Core is powered down 1: Core is powered up

3.2.5.17. 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[3:0] Handshake for DBGRESTART.
11:8	/	/	/
7:4	R	0x0	C_DBGNOPWRDWN No power-down request. Debugger has requested that processor is not powered down. Debug no power down[3:0].
3:0	R	0x0	C_DBGPWRUPREQ Power up request Debug power up request[3:0] 0: Do not request that the core is powered up 1: Request that the core is powered up

3.2.6. CPU Subsystem Control Register List

Module Name	Base Address
CPU_SUBSYS_CTRL	0x08100000

Register Name	Offset	Description
GENER_CTRL_REG0	0x0000	General Control Register0
GENER_CTRL_REG1	0x0004	General Control Register1
GIC_JTAG_RST_CTRL	0x000C	GIC and Jtag Reset Control Register
C0_INT_EN	0x0010	Cluster0 Interrupt Enable Control Register
IRQ_FIQ_STATUS	0x0014	IRQ/FIQ Status Register
GENER_CTRL_REG2	0x0018	General Control Register2
DBG_STATE	0x001C	Debug State Register

3.2.7. CPU Subsystem Control Register Description

3.2.7.1. 0x0000 General Control Register0(Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: GENER_CTRL_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1	R/W	0x0	IDC clock enable 0: disable IDC clock

			1: enable IDC clock
0	R/W	0x0	GIC_CFGSDISABLE Disables write access to some secure GIC registers.

3.2.7.2. 0x0004 General Control Register1(Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: GENER_CTRL_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	AXI to MBUS Clock Gating disable, the priority of this bit is higher than bit[6]
6	R/W	0x0	AXI to MBUS Clock Gating enable
5:0	/	/	/

3.2.7.3. 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[3:0] Clear the status of interface, for debug
15:12	/	/	/
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 Jtag portrst 0: assert 1: de-assert
8	R/W	0x1	TRST Jtag trst 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

3.2.7.4. 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	/	/	/
15:0	R/W	0xFFFF	C0_GIC_EN Interrupt enable control register. Mask irq_out/firq_out to system domain.

3.2.7.5. 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]

3.2.7.6. 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
15:1	/	/	/
0	R/W	0x0	C0_TSCLKCHANGE Cluster 0 Time Stamp change bit

3.2.7.7. 0x001C Debug State Register(Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: DBG_STATE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R	0x0	C0_DBG_STATE Cluster 0 is in debug mode or normal mode

3.3. CCU

3.3.1. Overview

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

The CCU includes the following features:

- 12 PLLs
- Bus source and divisions
- Clock output control
- PLL bias control
- PLL tuning control
- PLL pattern control
- Configuring modules clock
- Bus clock gating
- Bus software reset
- PLL lock control

3.3.2. Operations and Functional Descriptions

3.3.2.1. System Bus Tree

Figure 3-2 shows a block diagram of the System Bus Tree.

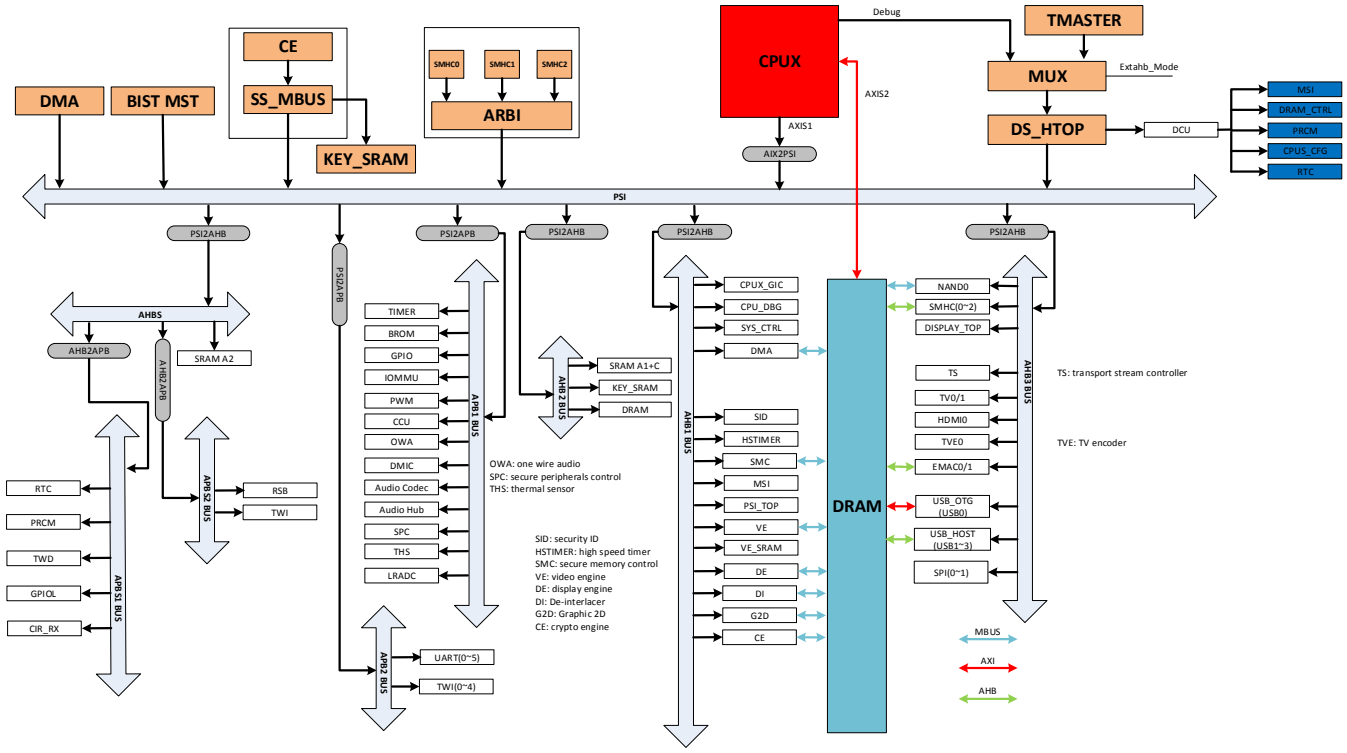

Figure 3- 2. System Bus Tree

3.3.2.2. Bus Clock Generation

Figure 3-3 describes bus clock generation.

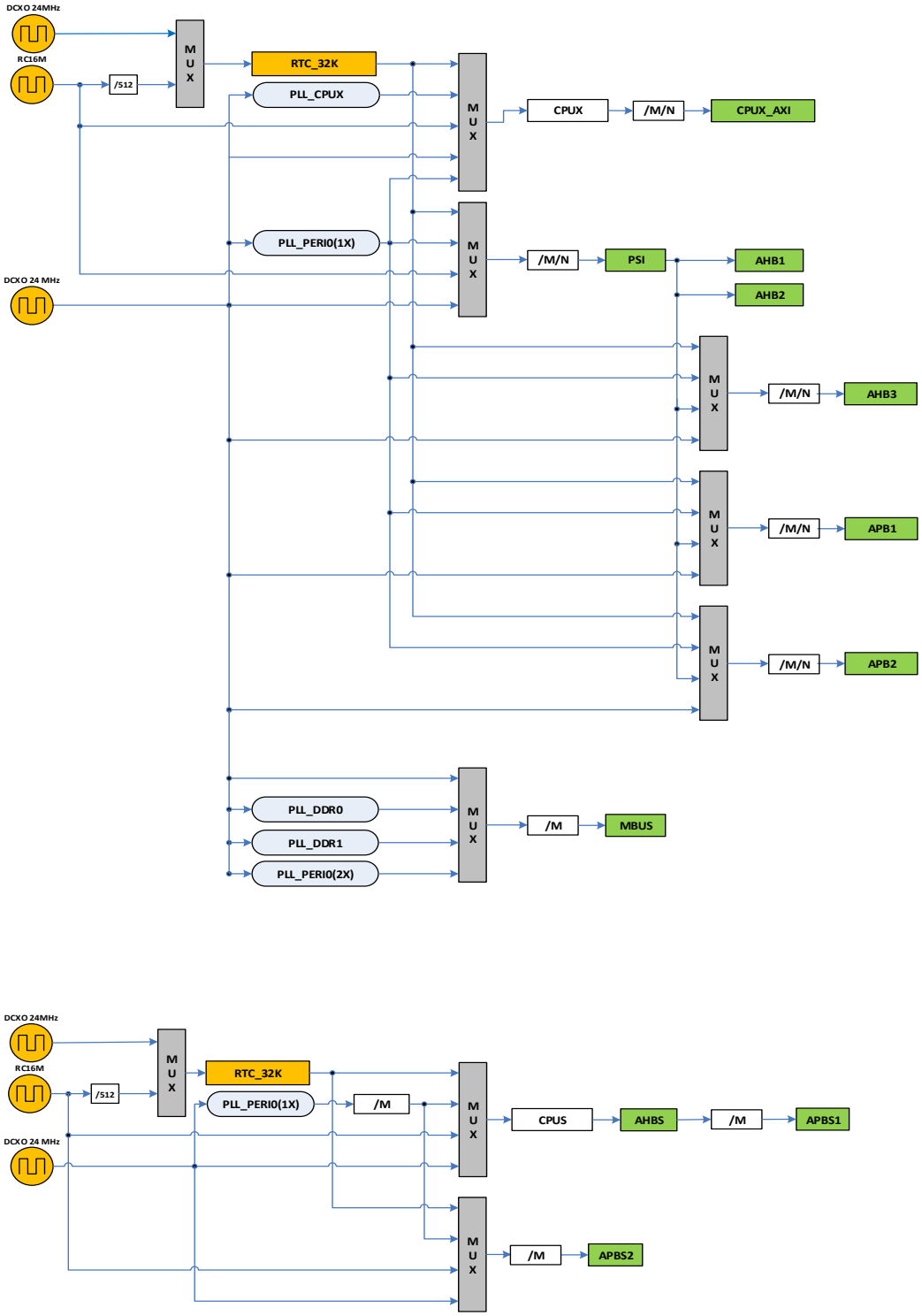

Figure 3- 3. Bus Clock Generation

3.3.2.3. Module Clock Generation

Figure 3-4 describes module clock generation. The frequencies in parantheses are the default typical frequencies.

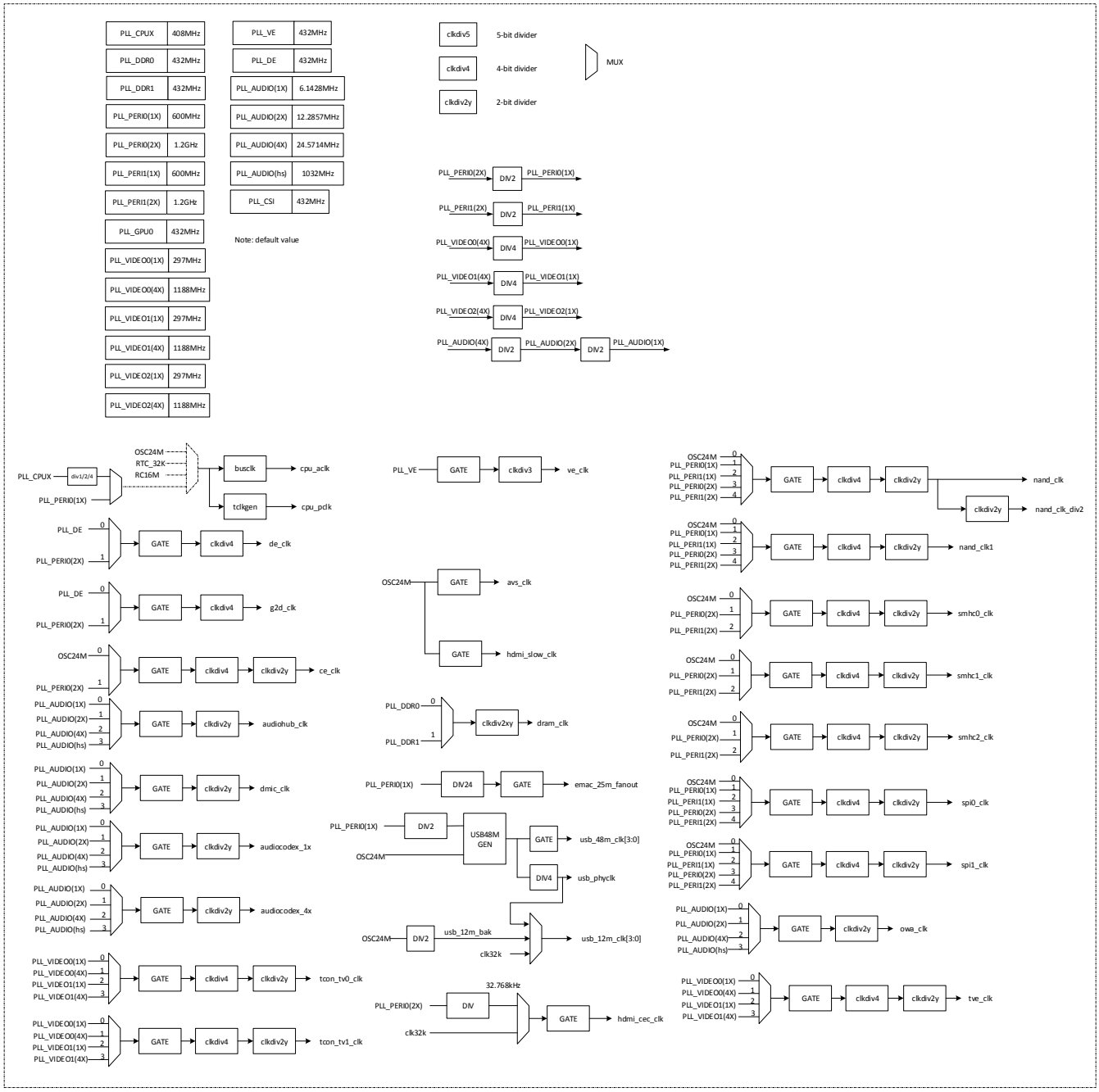

Figure 3- 4. Module Clock Generation

3.3.2.4. PLL Distribution

Figure 3-5 shows the block diagram of PLL distribution.

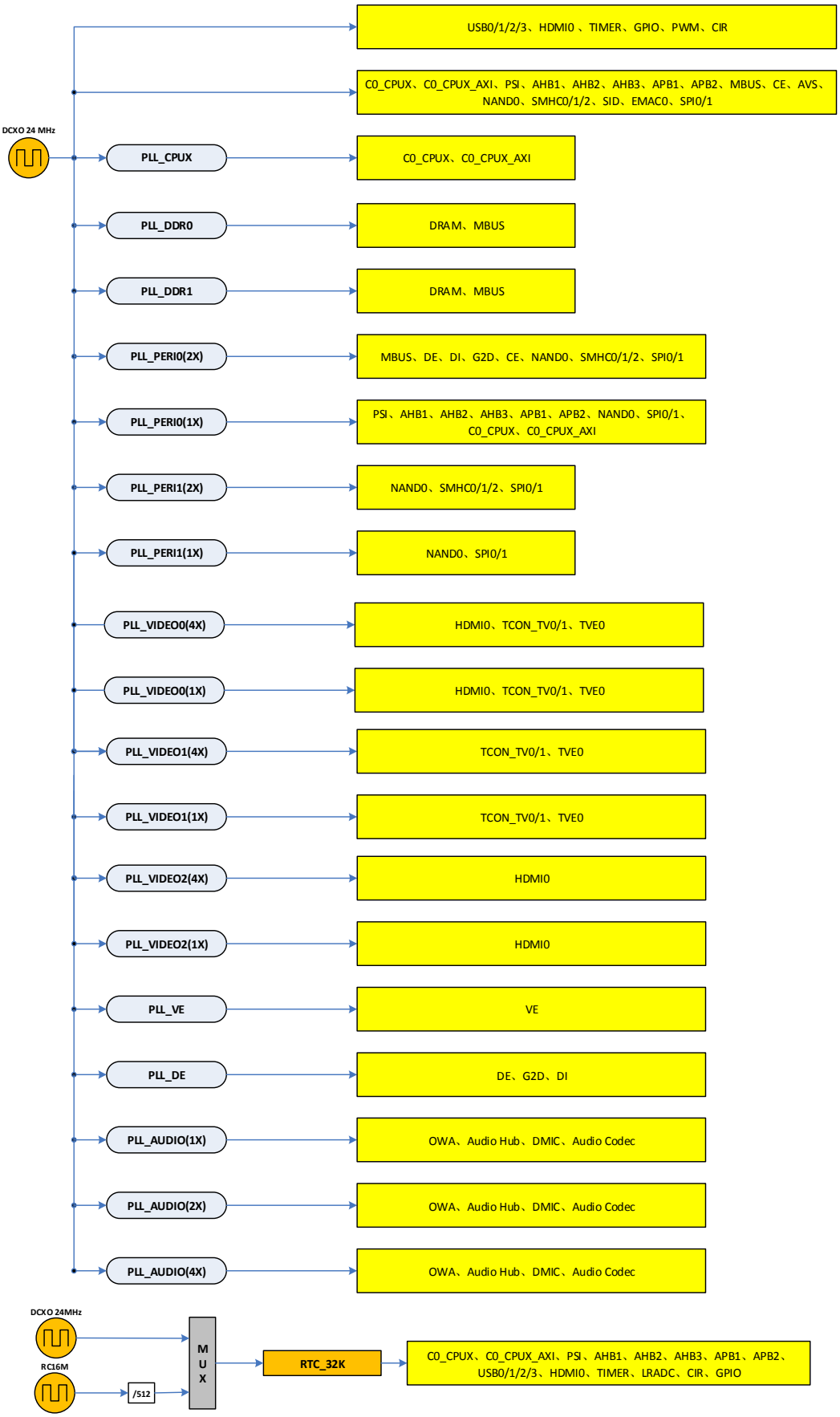

Figure 3- 5. Module Clock Tree

3.3.2.5. PLL Features

Table 3- 2. PLL Features

PLL	Stable Operating Frequency	Actual Operating Frequency	Spread Spectrum	Linear FM	Rate Control	Pk-Pk	Lock Time
PLL_CPUX	288 MHz~5.0 GHz (24*N/div1)	288 MHz~1.8 GHz	No	No	No	<200 ps	1.5 ms
PLL_AUDIO	24.576 MHz, 22.5792 MHz, (24*N/div1/div2)	24.576 MHz, 22.5792 MHz, (24.576 * 8) MHz, (22.5792 * 8) MHz	Yes(fractional frequency division)	No	No	<200 ps	500 us
PLL_PERI0(2X)	180 MHz~3.0 GHz (24*N/div1/div2)	1.2 GHz	Yes	No	No	<200 ps	500 us
PLL_PERI1(2X)	180 MHz~3.0 GHz (24*N/div1/div2)	1.2G Hz	Yes	No	No	<200 ps	500 us
PLL_VIDEO0(4X)	252 MHz~3.0 GHz (24*N/div)	192 MHz~2.4 GHz	Yes	No	No	<200 ps	500 us
PLL_VIDEO1(4X)	252 MHz~3.0 GHz (24*N/div)	192 MHz~2.4 GHz	Yes	No	No	<200 ps	500 us
PLL_VIDEO2(4X)	252 MHz~3.0 GHz (24*N/div)	192 MHz~2.4 GHz	Yes	No	No	<200 ps	500 us
PLL_VE	180 MHz~3.0 GHz (24*N/div1/div2)	192 MHz~600 MHz	Yes	No	No	<200 ps	500 us
PLL_DDR0	180 MHz~3.0 GHz (24*N/div1/div2)	192 MHz~2 GHz	Yes	No	No	200 MHz~800 MHz(<200 ps) 800 MHz~1.3 GHz(<140 ps) 1.3 GHz~2.0 GHz(<100 ps)	500 us
PLL_DDR1	180 MHz~3.0 GHz (24*N/div1/div2)	192 MHz~2 GHz	Yes	No	No	200 MHz~800 MHz(<200 ps) 800 MHz~1.3 GHz(<140 ps) 1.3 GHz~2.0 GHz(<100ps)	500 us
PLL_DE	180 MHz~3.0 GHz (24*N/div1/div2)	192 MHz~600 MHz	Yes	No	No	<200 ps	500 us
PLL_GPU	180 MHz~3.0 GHz (24*N/div1/div2)	300 MHz~600 MHz	Yes	No	No	<200 ps	500 us

3.3.3. Programming Guidelines

3.3.3.1. Frequency Adjustment of PLL_CPUX

The frequency configuration formula of PLL_CPUX: PLL_CPUX = 24 MHz*N/P, where, the N parameter is frequency-doubling factor of PLL, the next parameter configuration can proceed after PLL relock; the P parameter is digital post-frequency division, which can be dynamically switched in real time, and it does not affect the normal work of PLL.

The CPU PLL supports dynamic frequency configuration (modify the value of N). The CPU should first switch to a lower intermediate frequency and then adjust to the target frequency when switching the frequency. The process is as follows.

(1) Before you configure PLL_CPU, switch the clock source of CPU to PLL_PERI0(1X).
(2) Modify the N, P parameter of PLL_CPU.
(3) Write the PLL Lock Enable bit to 0 and then write it to 1.
(4) Wait the Lock bit (bit28) of PLL_CPUX_CTRL to 1.
(5) Switch the clock source of CPU to PLL_CPU.

3.3.3.2. Frequency Adjustment of PLL_AUDIO

The frequency configuration formula of PLL_AUDIO: PLL_AUDIO = 24 MHz*N/M0/M1/P. Changing any parameter of N, M0, M1 and P will affect the normal work of PLL, which needs to be relocked. Therefore, dynamic adjustment is not supported.

For PLL_AUDIO, two frequency points usually are needed: 24.576 MHz and 22.5792 MHz. There are generally specific recommended configuration factors for the two frequencies. To implement the desired frequency point of PLL_AUDIO, you need to use the decimal frequency division function. The process is as follows.

(1) Configure the N, M1, M0, P factor.
(2) Configure the PLL_SDM_ENABLE bit of PLL_AUDIO_CTRL to 1.
(3) Configure PLL_AUDIO_PAT0_CTRL to enable digital spread spectrum.
(4) Write the PLL Lock Enable bit of PLL_AUDIO_CTRL to 0 and then write it to 1.
(5) Wait the Lock bit (bit28) of PLL_AUDIO_CTRL to 1.

NOTE

The P factor of PLL_AUDIO is odd number, the clock output is non-equal duty.

3.3.3.3. Frequency Adjustment of PLL_DDR

For the clock of DDR, the switch of the clock source and the frequency division coefficient is burrless, but the frequency adjustment of the module should follow the following rules.

- From high frequency to low frequency: switch the clock source first, and then set the frequency division coefficient;
- From low frequency to high frequency: switch the frequency division coefficient first, and then modify clock source.

The controller has 2 PLL_DDR, the adjustment process is as follows.

(1) If the SDRAM controller uses PLL_DDR0, when a new frequency adjustment is needed, first configure the target frequency of PLL_DDR1(refer to the configuration process in section 3.3.3.4), wait PLL_DDR1 locking.

(2) Configure 0x800 DRAM_CLK_REG, switch the clock source of PLL_DDR0 to PLL_DDR1.

3.3.3.4. Frequency Adjustment of General PLL

(1) At present, the PLL should be enabled. If the PLL is not enabled, refer to the PLL process from disable to enable in section 3.3.3.5. For PLL, it is not suggested to switch during PLL using. When clock is not needed, it is suggested to configure the PLL_OUTPUT_EN bit of PLL_CTRL to disable the output gate of PLL.

(2) General PLL cannot be used in the process of frequency modulation. It is suggested to configure the PLL_OUTPUT_EN bit of PLL_CTRL to 0 in the process of PLL adjustment.

(3) Configure the N, M1, M0 factor. (It is not suggested to configure M1 factor, configure according to <<PLL recommended configuration table>>)

(4) Write the PLL Lock Enable bit (bit29) of PLL_CTRL to 0 and then write it to 1.

(5) Wait the Lock bit (bit28) of PLL_CTRL to 1.

(6) Configure PLL_OUTPUT_EN to 1.

3.3.3.5. PLL Disable to PLL Enable

(1) Configure the N, M1, M0 factor of PLL_CTRL_REG.

(2) Write the Enable bit of PLL_CTRL_REG to 1.

(3) Write the Lock Enable bit of PLL_CTRL_REG to 1.

(4) Wait the status of Lock to 1.

(5) Delay 20us, the PLL can be used.

3.3.3.6. PLL Enable to PLL Disable

(1) Write the Enable bit of PLL to 0.

(2) Write the Lock Enable bit (bit29) of PLL_CTRL_REG to 0.

CAUTION

In the normal using of PLL, it is not recommended to switch PLL frequently, because the switch of PLL will cause mutual interference between PLL, which will affect the stability of the system. Therefore, it is recommended to turn off PLL by configuring the PLL_OUTPUT_EN bit of PLL_CTRL to 0, instead of writing 0 to the enable bit.

3.3.3.7. Bus Configuration

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

- From high frequency to low frequency: switch the clock source first, and then set the frequency division factor;
- From low frequency to high frequency: switch the frequency division factor first, and then switch clock source.

3.3.3.8. Module Clock Configuration

For the bus gating and reset register of modules, the reset is de-asserted first, and then the CLK gating is enabled, to ensure that no problem will occur due to the module not being reset synchronously released.

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

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

3.3.3.9. Spread Spectrum Function

The configuration of spread spectrum follows the following steps.

Step1: Configure PLL_CTRL Register

- According to PLL frequency and PLL frequency formula f = [(N+1)/(M0+1)/(M1+1)+X] * 24 MHz, suppose the value of divisor M0 and divisor M1, calculate factor N and decimal value X, and write M0、M1、N and PLL frequency to the PLL_CTRL register.
- Configure the SDM_Enable bit of the PLL_CTRL register to 1 to enable spread spectrum function.

NOTE

Having different PLL calculate formula for different PLL, please refer to each PLL_CTRL register.

Step 2: Configure PLL_PAT Register

- According to decimal value X and spread spectrum frequency(the bit[18:17] of the PLL_PAT register), calculate WAVE_BOT (= 2^{17} * X1) and WAVE_STEP (= 2^{17} * (X2-X1) / (24 MHz/PREQ) * 2).
- Configure spread spectrum mode(SPR_FREQ_MODE) to 2 or 3.
- Configure the spread spectrum clock source select bit(SDM_CLK_SEL) to 0 by default. But if the PLL_INPUT_DIV_M1 bit of the PLL_CTRL register is 1, the bit should set to 1.
- Write WAVE_BOT、WAVE_STEP、PREQ、SPR_FREQ_MODE and SDM_CLK_SEL to the PLL_PAT register, and configure SIG_DELT_PAT_EN to 1.

Step 3: Delay 20us

3.3.4. Register List

Module Name	Base Address
CCU	0x03001000

Register Name	Offset	Description
PLL_CPUX_CTRL_REG	0x0000	PLL_CPUX Control Register
PLL_DDR0_CTRL_REG	0x0010	PLL_DDR0 Control Register
PLL_DDR1_CTRL_REG	0x0018	PLL_DDR1 Control Register
PLL_PERI0_CTRL_REG	0x0020	PLL_PERI0 Control Register
PLL_PERI1_CTRL_REG	0x0028	PLL_PERI1 Control Register
PLL_GPU0_CTRL_REG	0x0030	PLL_GPU0 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_DE_CTRL_REG	0x0060	PLL_DE Control Register
PLL_AUDIO_CTRL_REG	0x0078	PLL_AUDIO Control Register
PLL_DDR0_PAT_CTRL_REG	0x0110	PLL_DDR0 Pattern Control Register
PLL_DDR1_PAT_CTRL_REG	0x0118	PLL_DDR1 Pattern Control Register
PLL_PERI0_PAT0_CTRL_REG	0x0120	PLL_PERI0 Pattern0 Control Register
PLL_PERI0_PAT1_CTRL_REG	0x0124	PLL_PERI0 Pattern1 Control Register
PLL_PERI1_PAT0_CTRL_REG	0x0128	PLL_PERI1 Pattern0 Control Register
PLL_PERI1_PAT1_CTRL_REG	0x012C	PLL_PERI1 Pattern1 Control Register
PLL_GPU0_PAT0_CTRL_REG	0x0130	PLL_GPU0 Pattern0 Control Register
PLL_GPU0_PAT1_CTRL_REG	0x0134	PLL_GPU0 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_DE_PAT0_CTRL_REG	0x0160	PLL_DE Pattern0 Control Register
PLL_DE_PAT1_CTRL_REG	0x0164	PLL_DE Pattern1 Control Register
PLL_AUDIO_PAT0_CTRL_REG	0x0178	PLL_AUDIO Pattern0 Control Register
PLL_AUDIO_PAT1_CTRL_REG	0x017C	PLL_AUDIO Pattern1 Control Register
PLL_CPUX_BIAS_REG	0x0300	PLL_CPUX Bias Register
PLL_DDR0_BIAS_REG	0x0310	PLL_DDR0 Bias Register
PLL_DDR1_BIAS_REG	0x0318	PLL_DDR1 Bias Register
PLL_PERI0_BIAS_REG	0x0320	PLL_PERI0 Bias Register

PLL_PERI1_BIAS_REG	0x0328	PLL_PERI1 Bias Register
PLL_GPU0_BIAS_REG	0x0330	PLL_GPU0 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_DE_BIAS_REG	0x0360	PLL_DE Bias Register
PLL_AUDIO_BIAS_REG	0x0378	PLL_AUDIO Bias Register
PLL_CPUX_TUN_REG	0x0400	PLL_CPUX Tuning Register
CPUX_AXI_CFG_REG	0x0500	CPUX_AXI Configuration Register
PSI_AHB1_AHB2_CFG_REG	0x0510	PSI_AHB1_AHB2 Configuration Register
AHB3_CFG_REG	0x051C	AHB3 Configuration Register
APB1_CFG_REG	0x0520	APB1 Configuration Register
APB2_CFG_REG	0x0524	APB2 Configuration Register
MBUS_CFG_REG	0x0540	MBUS Configuration 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
GPU_CLK_REG	0x0670	GPU Clock Register
GPU_CLK1_REG	0x0674	GPU Clock1 Register
GPU_BGR_REG	0x067C	GPU 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
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
PSI_BGR_REG	0x079C	PSI 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
NAND0_0_CLK_REG	0x0810	NAND0_0 Clock Register
NAND0_1_CLK_REG	0x0814	NAND0_1 Clock Register
NAND_BGR_REG	0x082C	NAND 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
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
EPHY_25M_CLK_REG	0x0970	EPHY_25M Clock Register
EMAC_BGR_REG	0x097C	EMAC Bus Gating Reset Register
TS_CLK_REG	0x09B0	TS Clock Register
TS_BGR_REG	0x09BC	TS Bus Gating Reset Register
THS_BGR_REG	0x09FC	THS Bus Gating Reset Register
OWA_CLK_REG	0x0A20	OWA 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_1X_CLK_REG	0x0A50	AUDIO CODEC 1X Clock Register
AUDIO_CODEC_4X_CLK_REG	0x0A54	AUDIO CODEC 4X Clock Register
AUDIO_CODEC_BGR_REG	0x0A5C	AUDIO CODEC Bus Gating Reset Register
AUDIO_HUB_CLK_REG	0x0A60	AUDIO_HUB Clock Register
AUDIO_HUB_BGR_REG	0x0A6C	AUDIO_HUB Bus Gating Reset Register
USB0_CLK_REG	0x0A70	USB0 Clock Register
USB1_CLK_REG	0x0A74	USB1 Clock Register
USB2_CLK_REG	0x0A78	USB2 Clock Register
USB_BGR_REG	0x0A8C	USB Bus Gating Reset Register
HDMI0_CLK_REG	0x0B00	HDMI0 Clock Register
HDMI0_SLOW_CLK_REG	0x0B04	HDMI0 Slow Clock Register
HDMI_CEC_CLK_REG	0x0B10	HDMI CEC Clock Register
HDMI_BGR_REG	0x0B1C	HDMI Bus Gating Reset Register
DISPLAY_IF_TOP_BGR_REG	0x0B5C	DISPLAY_IF_TOP BUS GATING RESET Register
TCON_TV0_CLK_REG	0x0B80	TCON TV0 Clock Register
TCON_TV1_CLK_REG	0x0B80	TCON TV1 Clock Register
TCON_TV_BGR_REG	0x0B9C	TCON TV GATING RESET Register
TVE0_CLK_REG	0x0BB0	TVE0 Clock Register
TVE_BGR_REG	0x0BBC	TVE BUS GATING RESET Register
HDMI_HDCP_CLK_REG	0x0C40	HDMI HDCP Clock Register
HDMI_HDCP_BGR_REG	0x0C4C	HDMI HDCP Bus Gating Reset Register
CCU_SEC_SWITCH_REG	0x0F00	CCU Security Switch 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
24M_27M_CLK_OUTPUT_REG	0x0F20	24M or 27M Clock Output Register

3.3.5. Register Description

3.3.5.1. 0x0000 PLL_CPUX Control Register (Default Value: 0x0A00_1000)

Offset: 0x0000			Register Name: PLL_CPUX_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable PLL_CPUX=24 MHz*N/P **The PLL_CPUX output frequency must be in the range from 200 MHz to 3 GHz. And the default value of PLL_CPUX is 408 MHz.**
30	/	/	/
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.)
27	R/W	0x1	PLL_OUTPUT_ENABLE 0:Disable 1:Enable The bit is used to control the output enable of PLL.
26:24	R/W	0x0	PLL_LOCK_TIME. PLL lock time The bit indicates the step amplitude from one frequency to another.
23:18	/	/	/
17:16	R/W	0x0	PLL_OUT_EXT_DIVP PLL Output External Divider P 00: 1 01: 2 10: 4 11: / When output clock is less than 288 MHz, clock frequency is output by dividing P.
15:8	R/W	0x10	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 11.
7:2	/	/	/
1:0	R/W	0x0	PLL_FACTOR_M PLL Factor M M = PLL_FACTOR_M + 1

			PLL_FACTOR_M is from 0 to 3. **Note: The bit is only for testing.**

3.3.5.2. 0x0010 PLL_DDR0 Control Register (Default Value: 0x0800_2301)

Offset: 0x0010			**Register Name: PLL_DDR0_CTRL_REG**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable PLL_DDR= 24 MHz*N/M0/M1 **The default value of PLL_DDR0 is 432 MHz.**
30	/	/	/
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.)
27	R/W	0x1	PLL_OUTPUT_ENABLE 0:Disable 1:Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_ENABLE 0:Disable 1:Enable
23:16	/	/	/
15:8	R/W	0x23	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 11.
7:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV_M1. PLL Input Div M1. M1=PLL_INPUT_DIV_M1 + 1 PLL_INPUT_DIV_M1 is from 0 to 1.
0	R/W	0x1	PLL_OUTPUT_DIV_M0. PLL Output Div M0. M0=PLL_OUTPUT_DIV_M0 + 1 PLL_OUTPUT_DIV_M0 is from 0 to 1.

3.3.5.3. 0x0018 PLL_DDR1 Control Register (Default Value: 0x0800_2301)

Offset: 0x0018			Register Name: PLL_DDR1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable PLL_DDR= 24 MHz*N/M0/M1 **The default value of PLL_DDR1 is 432 MHz.**
30	/	/	/
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.)
27	R/W	0x1	PLL_OUTPUT_ENABLE 0:Disable 1:Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_ENABLE 0:Disable 1:Enable
23:16	/	/	/
15:8	R/W	0x23	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 11.
7:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV_M1. PLL Input Div M1. M1=PLL_INPUT_DIV_M1 + 1 PLL_INPUT_DIV_M1 is from 0 to 1.
0	R/W	0x1	PLL_OUTPUT_DIV_M0. PLL Output Div M0. M0=PLL_OUTPUT_DIV_M0 + 1 PLL_OUTPUT_DIV_M0 is from 0 to 1.

3.3.5.4. 0x0020 PLL_PERI0 Control Register (Default Value: 0x0800_3100)

Offset: 0x0020			Register Name: PLL_PERI0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable PLL_PERI0(2X) = 24 MHz*N/M0/M1 PLL_PERI0(1X) = 24 MHz*N/M0/M1/2 **The default value of PLL_PERI0(2X) is 1.2 GHz. It is not recommended to modify the value.**
30	/	/	/
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.)
27	R/W	0x1	PLL_OUTPUT_ENABLE 0:Disable 1:Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_ENABLE 0: Disable 1: Enable
23:16	/	/	/
15:8	R/W	0x31	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 11.
7:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV_M1. PLL Input Div M1. M1=PLL_INPUT_DIV_M1 + 1 PLL_INPUT_DIV_M1 is from 0 to 1.
0	R/W	0x0	PLL_OUTPUT_DIV_M0. PLL Output Div M0. M0=PLL_OUTPUT_DIV_M0 + 1 PLL_OUTPUT_DIV_M0 is from 0 to 1.

3.3.5.5. 0x0028 PLL_PERI1 Control Register (Default Value: 0x0800_3100)

Offset: 0x0028			Register Name: PLL_PERI1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable PLL_PERI1(2X) = 24 MHz*N/M0/M1. PLL_PERI1(1X) = 24 MHz*N/M0/M1/2. **The default value of PLL_PERI1(2X) is 1.2 GHz. It is not recommended to modify the value.**
30	/	/	/
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.)
27	R/W	0x1	PLL_OUTPUT_ENABLE 0:Disable 1:Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_ENABLE 0: Disable 1: Enable
23:16	/	/	/
15:8	R/W	0x31	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 11.
7:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV_M1 PLL Input Div M1 M1=PLL_INPUT_DIV_M1 + 1 PLL_INPUT_DIV_M1 is from 0 to 1.
0	R/W	0x0	PLL_OUTPUT_DIV_M0 PLL Output Div M0 M0=PLL_OUTPUT_DIV_M0 + 1 PLL_OUTPUT_DIV_M0 is from 0 to 1.

3.3.5.6. 0x0030 PLL_GPU0 Control Register (Default Value: 0x0800_2301)

Offset: 0x0030			Register Name: PLL_GPU0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable PLL_GPU0 = 24 MHz*N/M0/M1. **The default value of PLL_GPU0 is 432 MHz.**
30	/	/	/
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.)
27	R/W	0x1	PLL_OUTPUT_ENABLE 0:Disable 1:Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_ENABLE 0: Disable 1: Enable
23:16	/	/	/
15:8	R/W	0x23	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 11.
7:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV_M1. PLL Input Div M1. M1=PLL_INPUT_DIV_M1 + 1 PLL_INPUT_DIV_M1 is from 0 to 1.
0	R/W	0x1	PLL_OUTPUT_DIV_M0. PLL Output Div M0. M0=PLL_OUTPUT_DIV_M0 + 1 PLL_OUTPUT_DIV_M0 is from 0 to 1.

3.3.5.7. 0x0040 PLL_VIDEO0 Control Register (Default Value: 0x0800_6203)

Offset: 0x0040			Register Name: PLL_VIDEO0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable For application, PLL_VIDEO0(4X)= 24 MHz*N/M. PLL_VIDEO0(1X)=24 MHz*N/M/4. **The default value of PLL_VIDEO0(4X) is 1188 MHz.**
30	/	/	/
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.)
27	R/W	0x1	PLL_OUTPUT_ENABLE 0: Disable 1: Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_ENABLE 0: Disable 1: Enable
23:16	/	/	/
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 11.
7:2	/	/	/
1	R/W	0x1	PLL_INPUT_DIV_M 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_DIV_D PLL Output Div D M0=PLL_OUTPUT_DIV_D + 1 PLL_OUTPUT_DIV_D is from 0 to 1. The bit is only for testing. For test, PLL_VIDEO0(4X) =24 MHz*N/M/D

3.3.5.8. 0x0048 PLL_VIDEO1 Control Register (Default Value: 0x0800_6203)

Offset: 0x0048			Register Name: PLL_VIDEO1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable For application, PLL_VIDEO1(4X)= 24 MHz*N/M. PLL_VIDEO1(1X)=24 MHz*N/M/4. **The default value of PLL_VIDEO1(4X) is 1188 MHz.**
30	/	/	/
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.)
27	R/W	0x1	PLL_OUTPUT_ENABLE 0: Disable 1: Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_ENABLE 0: Disable 1: Enable
23:16	/	/	/
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 11.
7:2	/	/	/
1	R/W	0x1	PLL_INPUT_DIV_M 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_DIV_D PLL Output Div D M0=PLL_OUTPUT_DIV_D + 1 PLL_OUTPUT_DIV_D is from 0 to 1. The bit is only for testing. For test, PLL_VIDEO0(4X) =24 MHz*N/M/D

3.3.5.9. 0x0050 PLL_VIDEO2 Control Register (Default Value: 0x0800_6203)

Offset: 0x0050			Register Name: PLL_VIDEO2_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable For application, PLL_VIDEO2(4X)= 24 MHz*N/M. PLL_VIDEO2(1X)=24 MHz*N/M/4. **The default value of PLL_VIDEO2(4X) is 1188 MHz.**
30	/	/	/
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.)
27	R/W	0x1	PLL_OUTPUT_ENABLE 0: Disable 1: Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_ENABLE 0: Disable 1: Enable
23:16	/	/	/
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 11.
7:2	/	/	/
1	R/W	0x1	PLL_INPUT_DIV_M 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_DIV_D PLL Output Div D M0=PLL_OUTPUT_DIV_D + 1 PLL_OUTPUT_DIV_D is from 0 to 1. The bit is only for testing. For test, PLL_VIDEO0(4X) =24 MHz*N/M/D

3.3.5.10. 0x0058 PLL_VE Control Register (Default Value: 0x0800_2301)

Offset: 0x0058			Register Name: PLL_VE_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable The PLL_VE = 24 MHz*N/M0/M1. The default value of PLL_VE is 432 MHz.
30	/	/	/
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.)
27	R/W	0x1	PLL_OUTPUT_ENABLE 0: Disable 1: Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_ENABLE 0: Disable 1: Enable
23:16	/	/	/
15:8	R/W	0x23	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 11.
7:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV_M1. PLL Input Div M1. M1=PLL_INPUT_DIV_M1 + 1 PLL_INPUT_DIV_M1 is from 0 to 1.
0	R/W	0x1	PLL_OUTPUT_DIV_M0. PLL Output Div M0. M0=PLL_OUTPUT_DIV_M0 + 1 PLL_OUTPUT_DIV_M0 is from 0 to 1.

3.3.5.11. 0x0060 PLL_DE Control Register (Default Value: 0x0800_2301)

Offset: 0x0060			Register Name: PLL_DE_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable PLL_DE = 24 MHz*N/M0/M1. The default value of PLL_DE is 432 MHz.
30	/	/	/
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.)
27	R/W	0x1	PLL_OUTPUT_ENABLE 0: Disable 1: Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_ENABLE 0: Disable 1: Enable
23:16	/	/	/
15:8	R/W	0x23	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 11.
7:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV_M1. PLL Input Div M1. M1=PLL_INPUT_DIV_M1 + 1 PLL_INPUT_DIV_M1 is from 0 to 1.
0	R/W	0x1	PLL_OUTPUT_DIV_M0. PLL Output Div M0. M0=PLL_OUTPUT_DIV_M0 + 1 PLL_OUTPUT_DIV_M0 is from 0 to 1.

3.3.5.12. 0x0078 PLL_AUDIO Control Register (Default Value: 0x0814_2A01)

Offset: 0x0078			Register Name: PLL_AUDIO_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable This PLL is for Audio. PLL_AUDIO(hs)=24 MHz*N/M1 PLL_AUDIO(4X) = 24 MHz*N/M0/M1/P PLL_AUDIO(2X) = 24 MHz*N/M0/M1/P/2 PLL_AUDIO(1X) = 24 MHz*N/M0/M1/P/4 **7.5≤N/M0/M1≤125 and 12≤N** **The range of 24 MHz*N/M0/M1 is from 180 MHz to 3 GHz.** **The default value of PLL_AUDIO(4X) is 24.5714 MHz.** **Common configuration:** **When PLL_AUDIO(1X) is 24.576 MHz, PLL_AUDIO_CTRL_REG is recommended to set to 0xA8010F01, PLL_AUDIO_PAR0_CTRL_REG is recommended to set to 0xE000C49B.** **When PLL_AUDIO(1X) is 22.5792 MHz, PLL_AUDIO_CTRL_REG is recommended to set to 0xA8021501, PLL_AUDIO_PAR0_CTRL_REG is recommended to set to 0xE001288C.**
30	/	/	/
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_ENABLE 0:Disable 1:Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_ENABLE Spread Spectrum and Decimal Frequency Division 0: Disable 1: Enable
23:22	/	/	/
21:16	R/W	0x14	PLL_POST_DIV_P PLL Post-div P P= PLL_POST_DIV_P +1 PLL_POST_DIV_P is from 0 to 63.

15:8	R/W	0x2A	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 11.
7:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV_M1 PLL Input Div M1 M1=PLL_INPUT_DIV_M1 + 1 PLL_INPUT_DIV_M1 is from 0 to 1.
0	R/W	0x1	PLL_OUTPUT_DIV_M0 PLL Output Div M0 M0=PLL_OUTPUT_DIV_M0 + 1 PLL_OUTPUT_DIV_M0 is from 0 to 1.

3.3.5.13. 0x0110 PLL_DDR0 Pattern Control Register (Default Value: 0x0000_0000)

Offset: 0x0110			**Register Name: PLL_DDR0_PAT_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(1bit) 11: Triangular(nbit)
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 **When PLL_INPUT_DIV_M1 is 1, the bit is 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

3.3.5.14. 0x0118 PLL_DDR1 Pattern Control Register (Default Value: 0x0000_0000)

Offset: 0x0118			Register Name: PLL_DDR1_PAT_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(1bit) 11: Triangular(nbit)
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 **When PLL_INPUT_DIV_M1 is 1, the bit is 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

3.3.5.15. 0x0120 PLL_PERI0 Pattern0 Control Register (Default Value: 0x0000_0000)

Offset: 0x0120			Register Name: PLL_PERI0_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(1bit) 11: Triangular(nbit)
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 PLL_INPUT_DIV_M1 is 1, the bit is 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

3.3.5.16. 0x0124 PLL_PERI0 Pattern1 Control Register (Default Value: 0x0000_0000)

Offset: 0x0124			**Register Name: PLL_PERI0_PAT1_CTRL_REG**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	DITHER_EN
23:21	/	/	/
20	R/W	0x0	FRAC_EN
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN

3.3.5.17. 0x0128 PLL_PERI1 Pattern0 Control Register (Default Value: 0x0000_0000)

Offset: 0x0128			**Register Name: PLL_PERI1_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 PLL_INPUT_DIV_M1 is 1, the bit is 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

3.3.5.18. 0x012C PLL_PERI1 Pattern1 Control Register (Default Value: 0x0000_0000)

Offset: 0x012C			Register Name: PLL_PERI1_PAT1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	DITHER_EN
23:21	/	/	/
20	R/W	0x0	FRAC_EN
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN

3.3.5.19. 0x0130 PLL_GPU0 Pattern0 Control Register (Default Value: 0x0000_0000)

Offset: 0x0130			Register Name: PLL_GPU0_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 PLL_INPUT_DIV_M1 is 1, the bit is 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

3.3.5.20. 0x0134 PLL_GPU0 Pattern1 Control Register (Default Value: 0x0000_0000)

Offset: 0x0134			Register Name: PLL_GPU0_PAT1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	DITHER_EN
23:21	/	/	/
20	R/W	0x0	FRAC_EN
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN

3.3.5.21. 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 PLL_INPUT_DIV_M1 is 1, the bit is 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

3.3.5.22. 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
23:21	/	/	/
20	R/W	0x0	FRAC_EN
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN

3.3.5.23. 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
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 PLL_INPUT_DIV_M1 is 1, the bit is 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

3.3.5.24. 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
23:21	/	/	/
20	R/W	0x0	FRAC_EN
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN

3.3.5.25. 0x0150 PLL_VIDEO2 Pattern0 Control Register (Default Value: 0x0000_0000)

Offset: 0x0150			Register Name: PLL_VIDEO2_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 PLL_INPUT_DIV_M1 is 1, the bit is 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

3.3.5.26. 0x0154 PLL_VIDEO2 Pattern1 Control Register (Default Value: 0x0000_0000)

Offset: 0x0154	Register Name: PLL_VIDEO2_PAT1_CTRL_REG

Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	HERSHEY EN 0: triangular 1: hershey
30:25	/	/	/
24	R/W	0x0	DITHER_EN
23:21	/	/	/
20	R/W	0x0	FRAC_EN
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN

3.3.5.27. 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 PLL_INPUT_DIV_M1 is 1, the bit is 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

3.3.5.28. 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
23:21	/	/	/
20	R/W	0x0	FRAC_EN
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN

3.3.5.29. 0x0160 PLL_DE Pattern0 Control Register (Default Value: 0x0000_0000)

Offset: 0x0160			Register Name: PLL_DE_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 PLL_INPUT_DIV_M1 is 1, the bit is 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

3.3.5.30. 0x0164 PLL_DE Pattern1 Control Register (Default Value: 0x0000_0000)

Offset: 0x0164			Register Name: PLL_DE_PAT1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	DITHER_EN

23:21	/	/	/
20	R/W	0x0	FRAC_EN
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN

3.3.5.31. 0x0178 PLL_AUDIO Pattern0 Control Register (Default Value: 0x0000_0000)

Offset: 0x0178			Register Name: PLL_AUDIO_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 PLL_INPUT_DIV_M1 is 1, the bit is 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

3.3.5.32. 0x017C PLL_AUDIO Pattern1 Control Register (Default Value: 0x0000_0000)

Offset: 0x017C			Register Name: PLL_AUDIO_PAT1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	DITHER_EN
23:21	/	/	/
20	R/W	0x0	FRAC_EN
19:17	/	/	/

16:0	R/W	0x0	FRAC_IN

3.3.5.33. 0x0300 PLL_CPUX Bias Register (Default Value: 0x8010_0000)

Offset: 0x0300			Register Name: PLL_CPUX_BIAS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x1	VCO_RST VCO reset in
30:21	/	/	/
20:16	R/W	0x10	PLL_BIAS_CURRENT PLL current bias control [4:0], CPU_CP.
15:0	/	/	/

3.3.5.34. 0x0310 PLL_DDR0 Bias Register (Default Value: 0x0003_0000)

Offset: 0x0310			Register Name: PLL_DDR0_BIAS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20:16	R/W	0x3	PLL_BIAS_CTRL. PLL bias control [4:0].
15:0	/	/	/

3.3.5.35. 0x0320 PLL_PERI0 Bias Register (Default Value: 0x0003_0000)

Offset: 0x0320			Register Name: PLL_PERI0_BIAS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20:16	R/W	0x3	PLL_BIAS_CTRL PLL bias control [4:0]
15:0	/	/	/

3.3.5.36. 0x0328 PLL_PERI1 Bias Register (Default Value: 0x0003_0000)

Offset: 0x0328			Register Name: PLL_PERI1_BIAS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20:16	R/W	0x3	PLL_BIAS_CTRL PLL bias control [4:0]
15:0	/	/	/

3.3.5.37. 0x0330 PLL_GPU0 Bias Register (Default Value: 0x0003_0000)

Offset: 0x0330			Register Name: PLL_GPU0_BIAS_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20:16	R/W	0x3	PLL_BIAS_CTRL PLL bias control [4:0]
15:0	/	/	/

3.3.5.38. 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_BIAS_CTRL PLL bias control [4:0]
15:0	/	/	/

3.3.5.39. 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_BIAS_CTRL PLL bias control [4:0]
15:0	/	/	/

3.3.5.40. 0x0350 PLL_VIDEO2 Bias Register (Default Value: 0x0003_0000)

Offset: 0x0350			Register Name: PLL_VIDEO2_BIAS_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20:16	R/W	0x3	PLL_BIAS_CTRL PLL bias control [4:0]
15:0	/	/	/

3.3.5.41. 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_BIAS_CTRL PLL bias control [4:0]
15:0	/	/	/

3.3.5.42. 0x0360 PLL_DE Bias Register (Default Value: 0x0003_0000)

Offset: 0x0360			Register Name: PLL_DE_BIAS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20:16	R/W	0x3	PLL_BIAS_CTRL PLL bias control [4:0]
15:0	/	/	/

3.3.5.43. 0x0378 PLL_AUDIO Bias Register (Default Value: 0x0003_0000)

Offset: 0x0378			Register Name: PLL_AUDIO_BIAS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20:16	R/W	0x3	PLL_BIAS_CTRL PLL bias control [4:0]
15:0	/	/	/

3.3.5.44. 0x0400 PLL_CPUX Tuning Register (Default Value: 0x4440_4000)

Offset: 0x0400			Register Name: PLL_CPUX_TUN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30:28	R/W	0x4	VCO_RNG_CTRL VCO range control [2:0]
27	/	/	/
26:24	R/W	0x4	KVCO_GAIN_CTRL KVCO gain control [2:0]
23	/	/	/
22:16	R/W	0x40	CNT_INIT_CTRL Counter initial control [6:0]
15	R/W	0x0	C_OD0 C-REG-OD0 for verify
14:8	R/W	0x40	C_B_IN C-B-IN [6:0] for verify
7	R/W	0x0	C_OD1

			C-REG-OD1 for verify
6:0	RO	0x0	C_B_OUT C-B-OUT [6:0] for verify

3.3.5.45. 0x0500 CPUX_AXI Configuration Register (Default Value: 0x0000_0301)

Offset: 0x0500			Register Name: CPUX_AXI_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: OSC24M 001: RTC_32K 010: RC16M 011: PLL_CPUX 100: PLL_PERI0(1X) 101: reserved 110: reserved 111: reserved CPUX Clock = Clock Source CPUX_AXI Clock = Clock Source/M CPUX_APB Clock = Clock Source/N
23:10	/	/	/
9:8	R/W	0x3	CPUX_APB_FACTOR_N Factor N.(N = FACTOR_N +1) FACTOR_N is from 0 to 3.
7:2	/	/	/
1:0	R/W	0x1	FACTOR_ M Factor M.(M = FACTOR_M +1) FACTOR_M is from 0 to 3.

3.3.5.46. 0x0510 PSI_AHB1_AHB2 Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0510			Register Name: PSI_AHB1_AHB2_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: OSC24M 01: RTC_32K 10: RC16M 11: PLL_PERI0(1X)

			PSI_AHB1_AHB2 CLK = Clock Source/M/N.
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N. Factor N 00: 1 01: 2 10: 4 11: 8
7:2	/	/	/
1:0	R/W	0x0	FACTOR_M Factor M.(M = FACTOR_M +1) FACTOR_M is from 0 to 3.

3.3.5.47. 0x051C AHB3 Configuration Register (Default Value: 0x0000_0000)

Offset: 0x051C			Register Name: AHB3_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: OSC24M 01: RTC_32K 10: PSI 11: PLL_PERI0(1X) AHB3 CLK = Clock Source/M/N.
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:2	/	/	/
1:0	R/W	0x0	FACTOR_M Factor M.(M = FACTOR_M +1) FACTOR_M is from 0 to 3.

3.3.5.48. 0x0520 APB1 Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0520			Register Name: APB1_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/

25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: OSC24M 01: RTC_32K 10: PSI 11: PLL_PERI0(1X) APB1 CLK = Clock Source/M/N.
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:2	/	/	/
1:0	R/W	0x0	FACTOR_M Factor M.(M = FACTOR_M +1) FACTOR_M is from 0 to 3.

3.3.5.49. 0x0524 APB2 Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0524			Register Name: APB2_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: OSC24M 01: RTC_32K 10: PSI 11: PLL_PERI0(1X) APB2 CLK = Clock Source/M/N.
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:2	/	/	/
1:0	R/W	0x0	FACTOR_M Factor M.(M = FACTOR_M +1) FACTOR_M is from 0 to 3.

3.3.5.50. 0x0540 MBUS Configuration Register (Default Value: 0xC000_0000)

Offset: 0x0540			Register Name: MBUS_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x1	CLK_GATING. Gating Special Clock 0: Clock is OFF 1: Clock is ON MBUS CLK = Clock Source/M.
30	R/W	0x1	MBUS_RST. MBUS Reset 0: Assert 1: De-assert
29:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL. Clock Source Select 00: OSC24M 01: PLL_PERI0(2X) 10: PLL_DDR0 11: PLL_DDR1
23:3	/	/	/
2:0	R/W	0x0	FACTOR_M Factor M.(M = FACTOR_M +1) FACTOR_M is from 0 to 7.

3.3.5.51. 0x0600 DE Clock Register (Default Value: 0x0000_0000)

Offset: 0x0600			Register Name: DE_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.
30:25	/	/	/
24	R/W	0x0	CLK_SRC_SEL. Clock Source Select 0: PLL_DE 1: PLL_PERI0(2X)
23:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M.(M = FACTOR_M +1) FACTOR_M is from 0 to 15.

3.3.5.52. 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 Gating Clock For DE 0: Mask 1: Pass

3.3.5.53. 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	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M.
30:25	/	/	/
24	R/W	0x0	CLK_SRC_SEL. Clock Source Select 0: PLL_DE 1: PLL_PERI0(2X)
23:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M.(M = FACTOR_M +1) FACTOR_M is from 0 to 15.

3.3.5.54. 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 Gating Clock For DI 0: Mask 1: Pass

3.3.5.55. 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	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M.
30:25	/	/	/
24	R/W	0x0	CLK_SRC_SEL. Clock Source Select 0: PLL_DE 1: PLL_PERI0(2X)
23:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M.(M = FACTOR_M +1) FACTOR_M is from 0 to 15.

3.3.5.56. 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 Gating Clock For G2D 0: Mask 1: Pass

3.3.5.57. 0x0670 GPU Clock0 Register (Default Value: 0x0000_0000)

Offset: 0x0670			Register Name: GPU_CLK0_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.
30:25	/	/	/
24	R/W	0x0	CLK_SRC_SEL Clock Source Select 0: PLL_GPU0 1: PLL_PERI_BAK_CLK(PLL_PERI_BAK_CLK is from GPU_CLK1_REG) **Note: The switch needs to be a burr-free switch.**
23:2	/	/	/
1:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 3. Burr-free divider.

3.3.5.58. 0x0674 GPU Clock1 Register (Default Value: 0x0000_0000)

Offset: 0x0674			Register Name: GPU_CLK1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_PERI_BAK_CLK_GATING. Gating Special Clock 0: Clock is OFF 1: Clock is ON PLL_PERI_BAK = Clock Source/M. Clock Source is from PLL_PERI0(2X). Burr-free divider.
23:3	/	/	/
1:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 3. Burr-free divider.

3.3.5.59. 0x067C GPU Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x067C			Register Name: GPU_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**

31:17	/	/	/
16	R/W	0x0	GPU_RST. GPU Reset. 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	GPU_GATING. Gating Clock For GPU 0: Mask 1: Pass

3.3.5.60. 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	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:25	/	/	/
24	R/W	0x0	CLK_SRC_SEL Clock Source Select 0: OSC24M 1: PLL_PERI0(2X)
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.

3.3.5.61. 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
15:1	/	/	/
0	R/W	0x0	CE_GATING Gating Clock for CE 0: Mask 1: Pass

3.3.5.62. 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	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/Divider M.
30:25	/	/	/
24	R/W	0x0	CLK_SRC_SEL. Clock Source Select 0: PLL_VE 1: /
23:3	/	/	/
2:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 7.

3.3.5.63. 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	/	/	/
16	R/W	0x0	VE_RST VE Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	VE_GATING Gating Clock for VE 0: Mask

			1: Pass

3.3.5.64. 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 Gating Clock for DMA 0: Mask 1: Pass **Note: The working clock of DMA is from AHB1.**

3.3.5.65. 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 **Note: The working clock of HSTIMER is from AHB1.**

3.3.5.66. 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	SCLK_GATING Gating Special Clock 0: Clock is OFF

			1: Clock is ON SCLK = OSC24M.
30:0	/	/	/

3.3.5.67. 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	/	/	/
0	R/W	0x0	DBGSYS_GATING Gating Clock for DBGSYS 0: Mask 1: Pass **Note: The working clock of DBGSYS is from OSC24M.**

3.3.5.68. 0x079C PSI Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x079C			Register Name: PSI_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	PSI_RST PSI Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	PSI_GATING Gating Clock for PSI 0: Mask 1: Pass **Note: The working clock of PSI is from PSI clock.**

3.3.5.69. 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 **Note: The working clock of PWM is from APB1 or OSC24M.**

3.3.5.70. 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

3.3.5.71. 0x0800 DRAM Clock Register (Default Value: 0x0000_0000)

Offset: 0x0800			**Register Name: DRAM_CLK_REG**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R/W	0x0	MODULE_RST Module Reset 0: Assert 1: De-assert SCLK = Clock Source/M.
29:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_DDR0 01: PLL_DDR1 1X: /
23:2	/	/	/
1:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 3.

3.3.5.72. 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_GATING Gating MBUS Clock For G2D 0: Mask 1: Pass
9	/	/	/
8	R/W	0x0	Reserved
7:6	/	/	/
5	R/W	0x0	NAND0_MCLK_GATING Gating MBUS Clock For NAND0 0: Mask 1: Pass
4	/	/	/
3	R/W	0x0	TS0_MCLK_GATING Gating MBUS Clock For TS0 0: Mask 1: Pass
2	R/W	0x0	CE_MCLK_GATING Gating MBUS Clock For CE 0: Mask 1: Pass
1	R/W	0x0	VE_MCLK_GATING Gating MBUS Clock For VE 0: Mask 1: Pass
0	R/W	0x0	DMA_MCLK_GATING Gating MBUS Clock For DMA 0: Mask 1: Pass

NOTE

DE MCLK is put in DE module to control. DI MCLK is put in DI module to control.

3.3.5.73. 0x080C DRAM Bus Gating Reset Register (Default Value: 0x0000_0000)

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	0x0	DRAM_GATING Gating Clock for DRAM 0: Mask 1: Pass

3.3.5.74. 0x0810 NAND0_0 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0810			Register Name: NAND0_0_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	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL. Clock Source Select 000: OSC24M 001: PLL_PERI0(1X) 010: PLL_PERI1(1X) 011: PLL_PERI0(2X) 100: PLL_PERI1(2X) 1XX:/
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.

3.3.5.75. 0x0814 NAND0_1 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0814	Register Name: NAND0_1_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/N/M.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: OSC24M 001: PLL_PERI0(1X) 010: PLL_PERI1(1X) 011: PLL_PERI0(2X) 100: PLL_PERI1(2X) 1XX:/
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.

3.3.5.76. 0x082C NAND Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x082C			Register Name: NAND_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	NAND0_RST NAND0 Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	NAND0_GATING Gating Clock For NAND0 0: Mask 1: Pass

3.3.5.77. 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	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: OSC24M 01: PLL_PERI0(2X) 10: PLL_PERI1(2X) 11: /
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.

3.3.5.78. 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	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: OSC24M 01: PLL_PERI0(2X) 10: PLL_PERI1(2X) 11: /

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.

3.3.5.79. 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	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: OSC24M 01: PLL_PERI0(2X) 10: PLL_PERI1(2X) 11: /
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.

3.3.5.80. 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
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

3.3.5.81. 0x090C UART Bus Gating Reset Register (Default Value: 0x0000_0000)

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
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
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

NOTE

The working clock of UART is APB2.

3.3.5.82. 0x091C TWI Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x091C			Register Name: TWI_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20	R/W	0x0	TWI4_RST TWI4 Reset 0: Assert 1: De-assert
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
16	R/W	0x0	TWI0_RST TWI0 Reset 0: Assert 1: De-assert
15:5	/	/	/
4	R/W	0x0	TWI4_GATING Gating Clock for TWI4 0: Mask 1: Pass
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

NOTE

The working clock of TWI is APB2.

3.3.5.83. 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	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: OSC24M 001: PLL_PERI0(1X) 010: PLL_PERI1(1X) 011: PLL_PERI0(2X) 100: PLL_PERI1(2X) 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.

3.3.5.84. 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	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: OSC24M 001: PLL_PERI0(1X) 010: PLL_PERI1(1X) 011: PLL_PERI0(2X) 100: PLL_PERI1(2X) 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.

3.3.5.85. 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	/	/	/
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

3.3.5.86. 0x0970 EPHY_25M Clock Register (Default Value: 0x0000_0000)

Offset: 0x0970			Register Name: EPHY_25M_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 = PLL_PERI0(1X)/24 = 25M.
30	R/W	0x0	PLL_PERI0_GATING Gating PLL_PERI0 Clock 0: Clock is OFF 1: Clock is ON
29:0	/	/	/

3.3.5.87. 0x097C EMAC Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x097C			Register Name: EMAC_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R/W	0x0	EMAC1_RST EMAC1 Reset 0: Assert 1: De-assert
16	R/W	0x0	EMAC0_RST EMAC0 Reset 0: Assert 1: De-assert
15:2	/	/	/
1	R/W	0x0	EMAC1_GATING Gating Clock for EMAC1 0: Mask 1: Pass
0	R/W	0x0	EMAC0_GATING Gating Clock for EMAC0 0: Mask 1: Pass

NOTE

The working clock of EMAC is from AHB3.

3.3.5.88. 0x09B0 TS Clock Register (Default Value: 0x0000_0000)

Offset: 0x09B0			Register Name:TS_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.
30:25	/	/	/
24	R/W	0x0	CLK_SRC_SEL Clock Source Select 0: OSC24M 1: PLL_PERI0(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.

3.3.5.89. 0x09BC TS Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x09BC			Register Name: TS_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	TS_RST TS Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	TS_GATING Gating Clock For TS

			0: Mask 1: Pass

3.3.5.90. 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	/	/	/
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 **Note: The working clock of THS is from OSC24M.**

3.3.5.91. 0x0A20 OWA Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A20			Register Name: OWA_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/N
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO(1X) 01: PLL_AUDIO(2X) 10: PLL_AUDIO(4X) 11: PLL_AUDIO(hs)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8

7:0	/	/	/

3.3.5.92. 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

3.3.5.93. 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	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/N
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO(1X) 01: PLL_AUDIO(2X) 10: PLL_AUDIO(4X) 11: PLL_AUDIO(hs)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:0	/	/	/

3.3.5.94. 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

3.3.5.95. 0x0A50 AUDIO CODEC 1X Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A50			Register Name: AUDIO_CODEC_1X_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.
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO(1X) 01: PLL_AUDIO(2X) 10: PLL_AUDIO(4X) 11: PLL_AUDIO(hs)
23:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 15.

3.3.5.96. 0x0A54 AUDIO CODEC 4X Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A54			Register Name: AUDIO_CODEC_4X_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.
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO(1X) 01: PLL_AUDIO(2X) 10: PLL_AUDIO(4X) 11: PLL_AUDIO(hs)
23:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 15.

3.3.5.97. 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

3.3.5.98. 0x0A60 AUDIO_HUB Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A60			Register Name: AUDIO_HUB_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/N.
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select

			00: PLL_AUDIO(1X) 01: PLL_AUDIO(2X) 10: PLL_AUDIO(4X) 11: PLL_AUDIO(hs)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:0	/	/	/

3.3.5.99. 0x0A6C AUDIO_HUB Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0A6C			Register Name: AUDIO_HUB_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	AUDIO_HUB_RST AUDIO_HUB Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	AUDIO_HUB_GATING Gating Clock For AUDIO_HUB 0: Mask 1: Pass

3.3.5.100. 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	SCLK_GATING_OHCI0 Gating Special Clock For OHCI0 0: Clock is OFF 1: Clock is ON
30	R/W	0x0	USBPHY0_RST USB PHY0 Reset 0: Assert 1: De-assert
29	R/W	0x0	SCLK_GATING_USBPHY0 Gating Special Clock For USBPHY0

			0: Clock is OFF 1: Clock is ON SCLK is from OSC24M.
28:26	/	/	/
25:24	R/W	0x0	OHCI0_12M_SRC_SEL OHCI0 12M Source Select 00: 12M divided from 48 MHz 01: 12M divided from 24 MHz 10: LOSC 11: /
23:0	/	/	/

3.3.5.101. 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	SCLK_GATING_OHCI1 Gating Special Clock For OHCI1 0: Clock is OFF 1: Clock is ON
30	R/W	0x0	USBPHY1_RST USB PHY1 Reset 0: Assert 1: De-assert
29	R/W	0x0	SCLK_GATING_USBPHY1 Gating Special Clock For USBPHY1 0: Clock is OFF 1: Clock is ON SCLK is from OSC24M.
28:26	/	/	/
25:24	R/W	0x0	OHCI1_12M_SRC_SEL OHCI1 12M Source Select 00: 12M divided from 48 MHz 01: 12M divided from 24 MHz 10: LOSC 11: /
23:0	/	/	/

3.3.5.102. 0x0A78 USB2 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A78			Register Name: USB2_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**

31	R/W	0x0	SCLK_GATING_OHCI2 Gating Special Clock For OHCI2 0: Clock is OFF 1: Clock is ON
30	R/W	0x0	USBPHY2_RST USB PHY2 Reset 0: Assert 1: De-assert
29	R/W	0x0	SCLK_GATING_USBPHY2 Gating Special Clock For USBPHY2 0: Clock is OFF 1: Clock is ON SCLK is from OSC24M.
28:26	/	/	/
25:24	R/W	0x0	OHCI2_12M_SRC_SEL OHCI2 12M Source Select 00: 12M divided from 48 MHz 01: 12M divided from 24 MHz 10: LOSC 11: /
23:0	/	/	/

3.3.5.103. 0x0A7C USB3 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0A7C			Register Name: USB3_CLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	SCLK_GATING_OHCI3 Gating Special Clock For OHCI3 0: Clock is OFF 1: Clock is ON
30	R/W	0x0	USBPHY3_RST USB PHY3 Reset 0: Assert 1: De-assert
29	R/W	0x0	SCLK_GATING_USBPHY3 Gating Special Clock For USBPHY3 0: Clock is OFF 1: Clock is ON SCLK is from OSC24M.
28:26	/	/	/
25:24	R/W	0x0	OHCI3_12M_SRC_SEL OHCI3 12M Source Select 00: 12M divided from 48 MHz 01: 12M divided from 24 MHz

			10: LOSC 11: /
23:0	/	/	/

3.3.5.104. 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	USBOTG_RST USBOTG Reset 0: Assert 1: De-assert
23	R/W	0x0	USBEHCI3_RST USBEHCI3 Reset 0: Assert 1: De-assert
22	R/W	0x0	USBEHCI2_RST USBEHCI2 Reset 0: Assert 1: De-assert
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	R/W	0x0	USBOHCI3_RST. USBOHCI3 Reset. 0: Assert 1: De-assert
18	R/W	0x0	USBOHCI2_RST. USBOHCI2 Reset. 0: Assert 1: De-assert
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
15:9	/	/	/
8	R/W	0x0	USBOTG_GATING Gating Clock For USBOTG 0: Mask 1: Pass
7	R/W	0x0	USBEHCI3_GATING Gating Clock For USBEHCI3 0: Mask 1: Pass
6	R/W	0x0	USBEHCI2_GATING Gating Clock For USBEHCI2 0: Mask 1: Pass
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	R/W	0x0	USBOHCI3_GATING Gating Clock For USBOHCI3 0: Mask 1: Pass
2	R/W	0x0	USBOHCI2_GATING Gating Clock For USBOHCI2 0: Mask 1: Pass
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

3.3.5.105. 0x0A9C LRADC Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0A9C			Register Name: LRADC_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/

16	R/W	0x0	LRADC_RST LRADC Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	LRADC_GATING Gating Clock For LRADC 0: Mask 1: Pass

3.3.5.106. 0x0B00 HDMI0 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0B00			Register Name: HDMI0_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.
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_VIDEO0(1X) 01: PLL_VIDEO0(4X) 10:PLL_VIDEO2(1X) 11:PLL_VIDEO2(4X)
23:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 15.

3.3.5.107. 0x0B04 HDMI0 Slow Clock Register (Default: 0x0000_0000)

Offset: 0x0B04			Register Name: HDMI0_SLOW_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 = OSC24M.
30:0	/	/	/

3.3.5.108. 0x0B10 HDMI CEC Clock Register(Default Value: 0x0000_0000)

Offset: 0x0B10			Register Name: HDMI_CEC_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
30	R/W	0x0	PLL_PERI_GATING Gating PLL_PERI Clock 0:Clock is OFF 1:Clock is ON
29:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: CCMU_32K 01: PLL_PERI0(2X)/36621 = 32.768kHz 1X:/
23:0	/	/	/

3.3.5.109. 0x0B1C HDMI Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0B1C			Register Name: HDMI_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R/W	0x0	HDMI0_SUB_RST HDMI0_SUB Reset 0: Assert 1: De-assert
16	R/W	0x0	HDMI0_MAIN_RST HDMI0_MAIN Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	HDMI0_GATING Gating Clock For HDMI0 0: Mask 1: Pass

3.3.5.110. 0x0B5C DISPLAY_IF_TOP Bus Gating Reset Register(Default Value: 0x0000_0000)

Offset: 0x0B5C	Register Name: DISPLAY_IF_TOP_BGR_REG

Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	DISPLAY_IF_TOP_RST DISPLAY_IF_TOP Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	DISPLAY_IF_TOP_GATING Gating Clock For DISPLAY_IF_TOP 0: Mask 1: Pass

3.3.5.111. 0x0B80 TCON TV0 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0B80			Register Name: TCON_TV0_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	/	/	/
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: / 101: /
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.

3.3.5.112. 0x0B84 TCON TV1 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0B84			Register Name: TCON_TV1_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	/	/	/
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: / 101: /
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.

3.3.5.113. 0x0B9C TCON TV Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0B9C			Register Name: TCON_TV_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R/W	0x0	TCON_TV1_RST TCON_TV1 Reset 0: Assert 1: De-assert
16	R/W	0x0	TCON_TV0_RST TCON_TV0 Reset 0: Assert 1: De-assert

15:2	/	/	/
1	R/W	0x0	TCON_TV1_GATING Gating Clock For TCON_TV1 0: Mask 1: Pass
0	R/W	0x0	TCON_TV0_GATING Gating Clock For TCON_TV0 0: Mask 1: Pass

3.3.5.114. 0x0BB0 TVE0 Clock Register (Default Value: 0x0000_0000)

Offset: 0x0BB0			Register Name: TVE0_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	/	/	/
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: / 101: /
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.

3.3.5.115. 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	TVE0_RST TVE0 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	TVE0_GATING Gating Clock For TVE0 0: Mask 1: Pass
0	R/W	0x0	TVE_TOP_GATING Gating Clock For TCON_TV1 0: Mask 1: Pass

3.3.5.116. 0x0C40 HDMI HDCP Clock Register (Default Value: 0x0000_0000)

Offset: 0x0C40			Register Name: HDMI_HDCP_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.
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_PERI0(1X) 01: PLL_PERI1(1X) Others:/
23:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 15.

3.3.5.117. 0x0C4C HDMI HDCP Bus Gating Reset Register (Default Value: 0x0000_0000)

Offset: 0x0C4C			Register Name: HDMI_HDCP_BGR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	HDMI_HDCP_RST HDMI_HDCP Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	HDMI_HDCP_GATING Gating Clock For HDMI_HDCP 0: Mask 1: Pass

3.3.5.118. 0x0F00 CCU Security Switch Register (Default Value: 0x0000_0000)

Offset: 0x0F00			Register Name: CCMU_SEC_SWITCH_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x0	MBUS_SEC MBUS clock registers' security 0: Secure 1: Non-secure
1	R/W	0x0	BUS_SEC Bus relevant registers' security 0: Secure 1: Non-secure
0	R/W	0x0	PLL_SEC PLL relevant registers' security 0: Secure 1: Non-secure

NOTE

If the secure bit in SID module has not been programed, the register is invalid.

3.3.5.119. 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	DBG_EN

			Debug Enable 0: Disable 1: Enable
30:25	/	/	/
24:20	R/W	0x0	DBG_SEL Debug Select 00000: PLL_C0_CPUX 00001: / 00010: PLL_DDR0 00011: PLL_DDR1 00100: PLL_PERI0 00101: PLL_PERI1 00110: PLL_GPU 00111: / 01000: PLL_VIDEO0 01001: PLL_VIDEO1 01010: / 01011: PLL_VE 01100: PLL_DE 01101: / 01110: / 01111: PLL_AUDIO 10000: / 10001: / 10010: / 10011: / 10100: / 10101: / 10110: / 10111: / 11000: / 11001: / 11010: / 11011: / 11100: / Others: /
19	/	/	/
18:17	R/W	0x0	UNLOCK_LEVEL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles
16	R/W	0x0	LOCK_LEVEL Lock Level 0: 24-26 Clock Cycles

			1: 23-27 Clock Cycles
15:0	/	/	/

3.3.5.120. 0x0F08 Frequency Detect Control Register(Default Value: 0x0000_0020)

Offset: 0x0F08			Register Name: FRE_DET_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/WAC	0x0	ERROR_FLAG Error Flag 0: Write 0 to clear 1: Error
30:2	/	/	/
8:4	R/W	0x2	Detect Time Time=1/32k*(2^RegValue) **Note: 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

3.3.5.121. 0x0F0C Frequency 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 register must be an integral multiple of 32. The unit is kHz.

3.3.5.122. 0x0F10 Frequency 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 register must be an integral multiple of 32. The unit is kHz.

3.3.5.123. 0x0F20 24M or 27M Clock Output Register (Default Value: 0x0000_0000)

Offset: 0x0F20			Register Name: 24M_27M_CLK_OUTPUT_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	27M_CLK_OUTPUT_EN 27M Clock Output enable 0: Disable 1: Enable
30	R/w	0x0	24M_27M_SEL 0:27M 1:24M When selecting 24M, the clock is from crystal. When selecting 27M, the clock is from PLL_CSI, if div_sel is valid, 27M clock can be output by configuring PLL_CSI and div_sel.
29:2	/	/	/
1:0	R/W	0x0	DIV_SEL 00:Div2 01:Div4 10:Div8 11:Div16 Only for 27M clock.

3.4. BROM System

3.4.1. Overview

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

The BROM system divides into two parts: FEL and Media Boot. The task of FEL is to write the external data to the local NVM, the task of the Media Boot is to load an effective and legitimate BOOT0 from NVM and running.

The BROM system includes the following features:

- CPU0 boot process
- Mandatory upgrade process through SMHC0, USB
- Supports GPIO pin or eFuse to select the kind of boot media to boot
- Supports normal boot and secure boot
- Supports loading only-certified firmware
- Ensure that Secure Boot is a trusted environment

3.4.2. Operations and Functional Descriptions

3.4.2.1. Boot Media Select

The BROM system supports the following boot media:

- SD/eMMC
- Nand Flash
- SPI NOR Flash
- SPI NAND Flash

There are two ways of Boot Select: **GPIO Pin Select** and **eFuse Select**. On startup, the BROM will read the state of BOOT_MODE, according to the state of BOOT_MODE to decide whether GPIO pin or eFuse to select the kind of boot media to boot. The BOOT_MODE is actually a bit at SID. Table 3-3 shows BOOT_MODE Setting.

Table 3- 3. BOOT_MODE Setting

BOOT_MODE(BROM_CONFIG[0] at SID)	Boot Select Type
0	GPIO pin select
1	eFuse select

3.4.2.1.1. GPIO Pin Select

If the state of the BOOT_MODE is 0, that is to choose GPIO pin. And in GPIO pin mode, there are 5 bits to select which boot media to boot. Table 3-4 shows boot media devices in GPIO pin mode.

Table 3- 4. GPIO Pin Boot Select Configuration

Pin_Boot_Select[4:0](at 0x03000024[13:9])	Boot Media	Note
01111	SMHC0->SPI_NAND	
10111	SMHC0->SPI_NOR	
11011	SMHC0->EMMC_BOOT->EMMC_USER	
11101	SMHC0->EMMC_USER->EMMC_BOOT	
11110	SMHC0->SLC_NAND	No support
11111	SMHC0->MLC_NAND	

NOTE

For H616 package, Boot_Select[0] is fixed at 1, so the SLC_NAND media is not supported.

3.4.2.1.2. eFuse Select

If the state of the BOOT_MODE is 1, that is to choose the eFuse type. The eFuse type has one 12 bits configuration, every 3 bits is divided into a group of the Boot Select, so it has four groups of boot_select. Table 3-5 shows eFuse Boot Select Configuration.

Table 3- 5. eFuse Boot Select Configuration

eFuse_Boot_Select_Cfg[11:0] (at BROM_TRY[11:0] of SID module)	**Description**
eFuse_Boot_Select[2:0]	eFuse_Boot_Select_1
eFuse_Boot_Select[5:3]	eFuse_Boot_Select_2
eFuse_Boot_Select[8:6]	eFuse_Boot_Select_3
eFuse_Boot_Select[11:9]	eFuse_Boot_Select_4

Table 3-6 describes each group of the eFuse Boot Select Setting. The first group to the third group are the same settings, but the fourth group need to be careful. If eFuse_Boot_Select_7 is set to 111, that means the way of the *Try*. The way of *Try* is followed by SMHC0, SMHC2, Nand Flash, SPI NOR Flash, SPI NAND Flash.

Table 3- 6. eFuse Boot Select Setting

eFuse_Boot_Select_n	Boot Media
000	Try
001	SLC NAND Flash
010	SMHC2
011	SPI NOR Flash
100	SPI NAND Flash

101	MLC NAND Flash
110	Reserved
111	The next a group of the eFuse_Boot_Select, but when the n is equal to 7, it will be a way of Try.

3.4.2.2. BROM Process

In Normal boot mode, the system boot will start from CPU0. BROM will read the Hotplug Flag Regsiter, according to the flag whether to go through the appropriate process. Finally, BROM will read the state of the FEL Pin, if the FEL Pin signal is detected pulling to high level, then the system will jump to the Try Media Boot process, or jump to the mandatory upgrade process. Figure 3-6 shows the BROM Process.

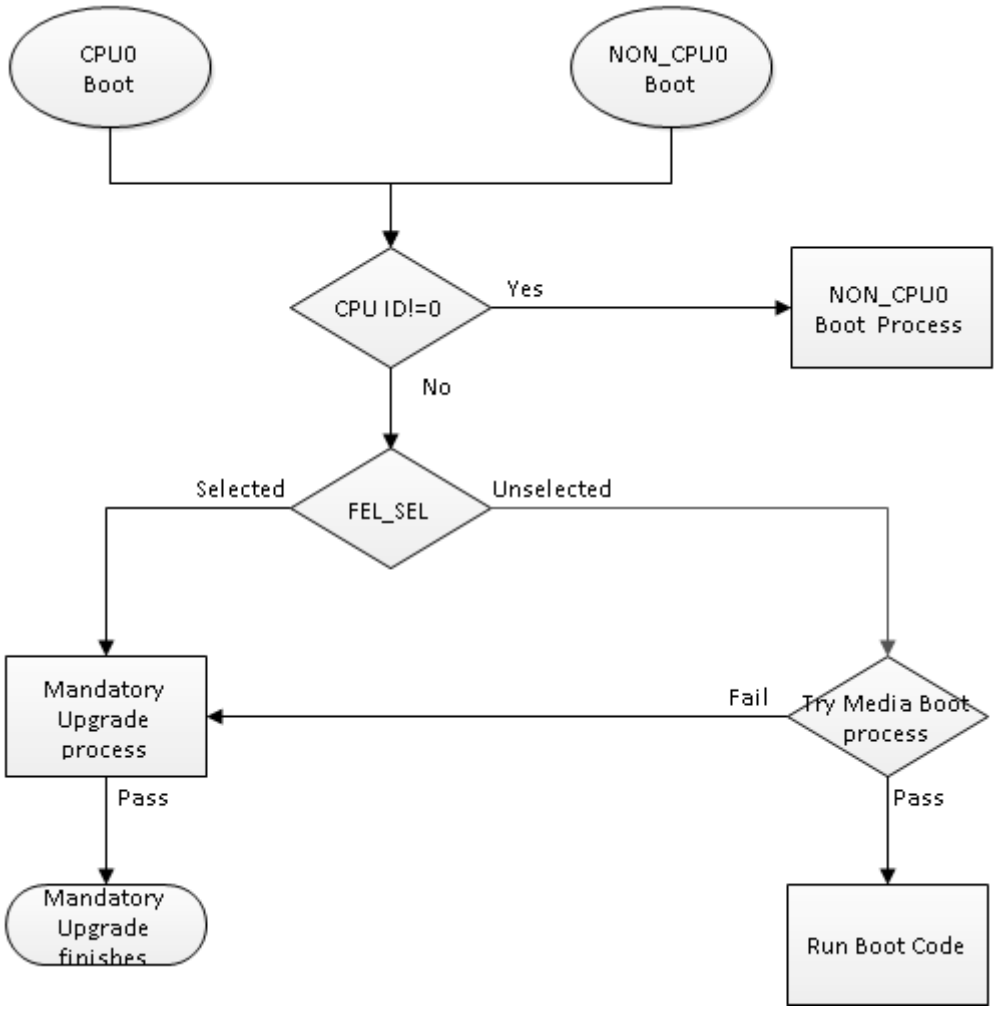

Figure 3- 6. Boot Process Diagram

3.4.2.3. Secure BROM Process

The Secure BROM supports the following features:

- Supports X509 certificate
- Supports cryptographic algorithms: AES-128, SHA-256, RSA-2048, DES
- Supports OTP/eFuse

Before runing Security Boot, software must check whether it has been modified or replaced, so the system will check and verify the integrity of the certificate, because the certificate has been using the RSA algorithm signature. The system also uses the Crypto Engine (CE) hardware module to accelerate the speed of encryption and decryption. Using standard cryptography ensure that the firmware images can be trusted, so the Secure BROM ensure the system security state is as expected.

In Security boot mode, by comparison witch Normal BROM, after the Try Media Boot process finishes, the system will go to run Security BROM software. Figure 3-7 shows the Secure BROM Process.

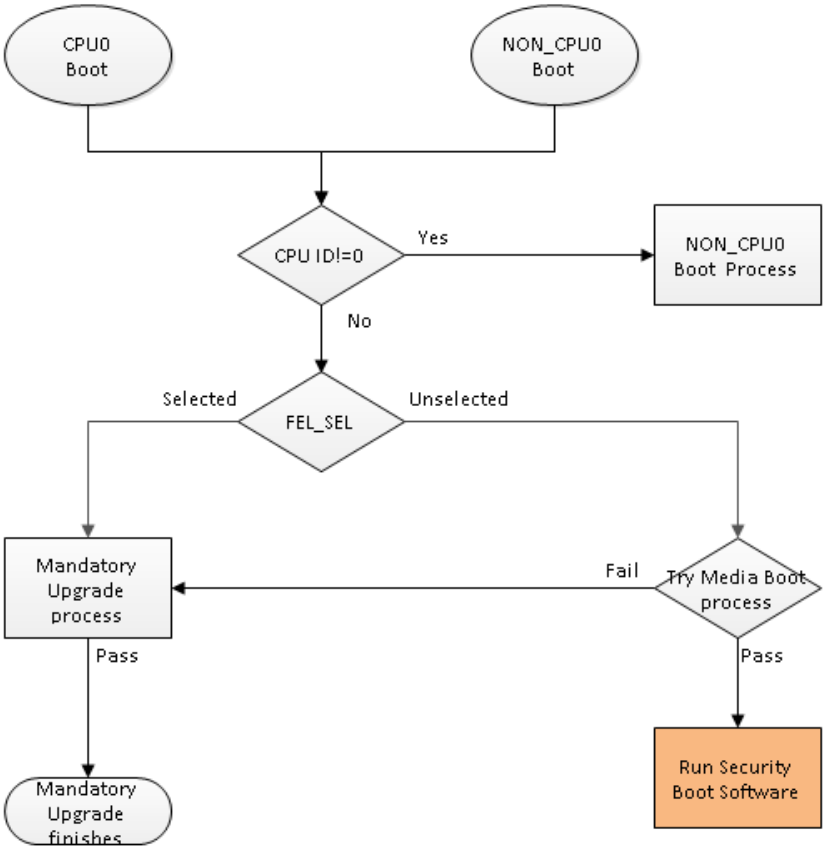

Figure 3- 7. Security BROM Process Diagram

3.4.2.4. BROM Process Description

3.4.2.4.1. Mandatory Upgrade Process

When the system chooses to whether enter Mandatory Upgrade Processor, if the FEL signal is detected pulling to low level, then the system will jump to the Mandatory Upgrade Process. Figure 3-8 shows the mandatory upgrade process.

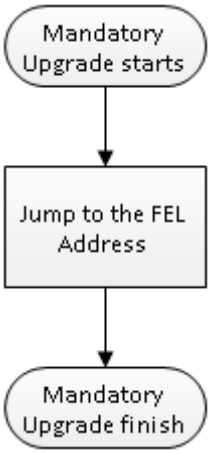

Figure 3- 8. Mandatory Upgrade Process

NOTE

The FEL address of the Normal BROM is 0x20.

The FEL address of the Secure BROM is 0x64.

3.4.2.4.2. FEL Process

When the system chooses to enter Mandatory Upgrade Process, then the system will jump to the FEL process. Figure 3-9 shows the FEL upgrade process.

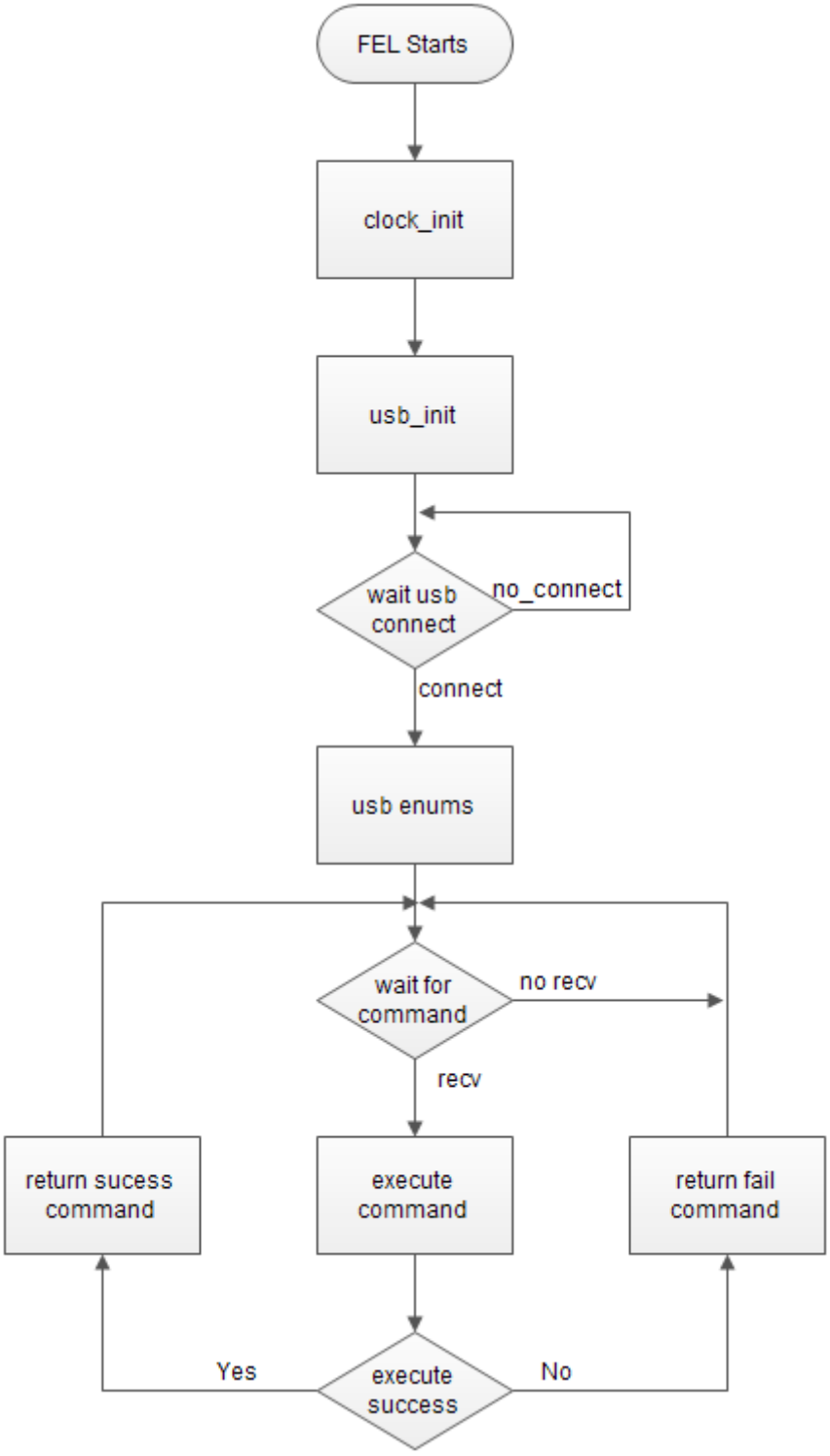

Figure 3- 9. USB FEL Process

3.4.2.4.3. Try Media Boot Process

When the system chooses to whether enters mandatory upgrade process, if the FEL pin signal is detected pulling high, then the system will jump to the try media boot process.

Try Media Boot Process will read the state of BOOT_MODE register, the state of BOOT_MODE decides whether to boot

from GPIO pin or efuse. Figure 3-10 shows GPIO pin boot select process. Figure 3-11 shows efuse boot select process.

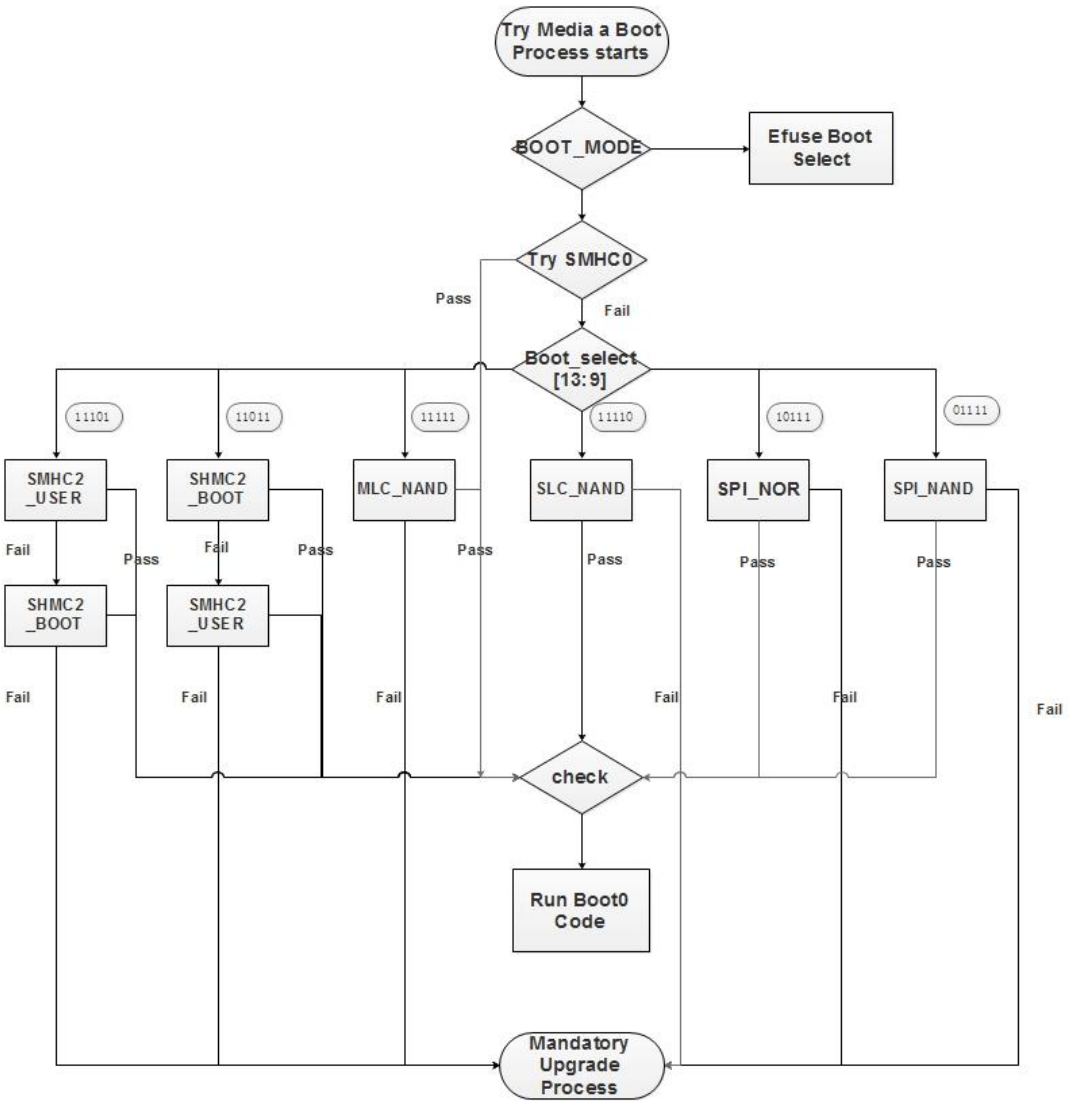

Figure 3- 10. GPIO Pin Boot Select Process

NOTE

SMHC0 usually is external SD/TF Card.

SMHC2 usually is external eMMC.

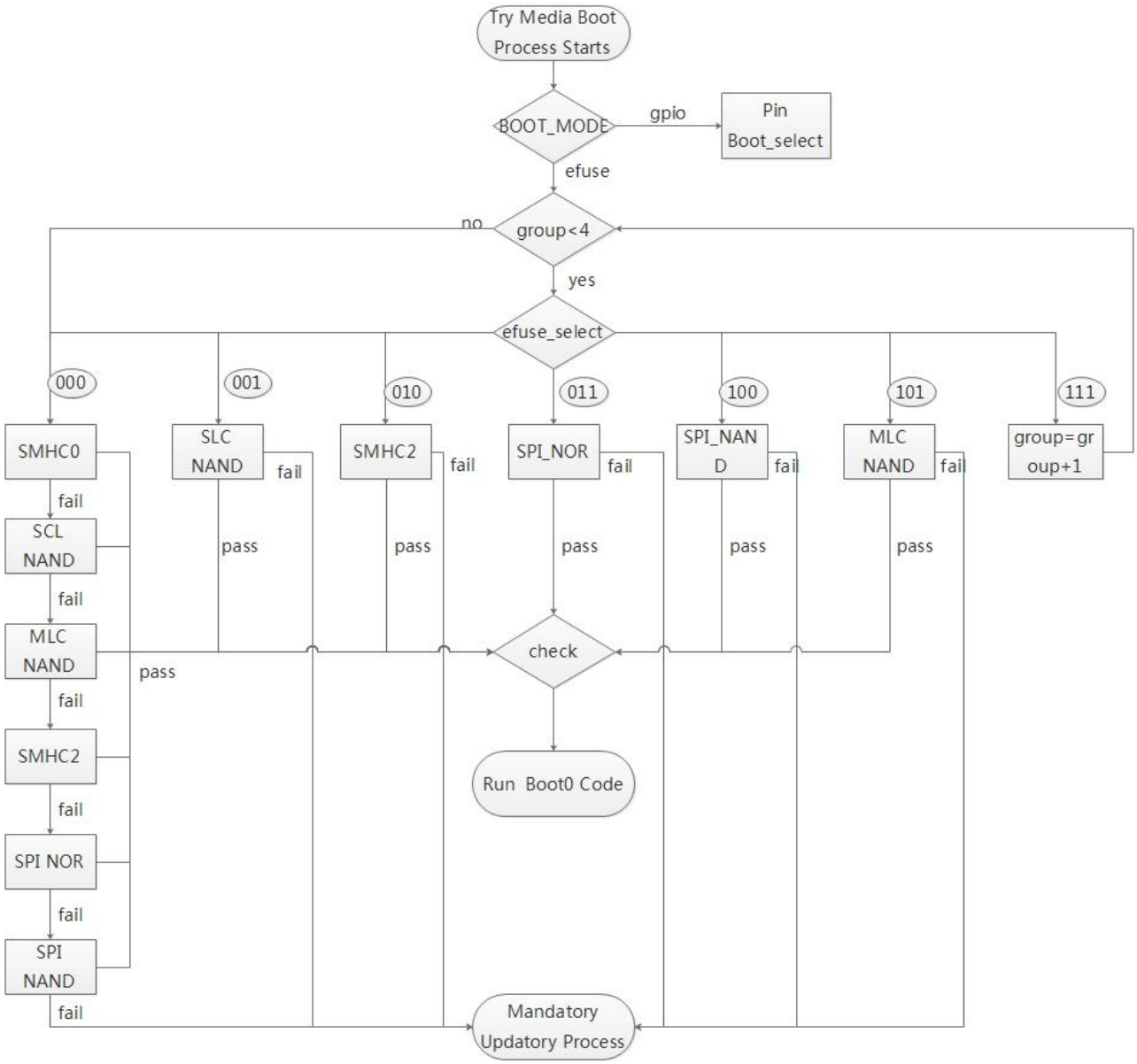

Figure 3- 11. eFuse Boot Select Process

3.5. System Configuration

3.5.1. Overview

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

The address range of SRAM is as follows.

Area	Address	Size
SRAM A1	0x0002 0000---0x0002 7FFF	32KB(Supports Byte operation, the clock source is AHB1)
SRAM C	0x0002 8000---0x0005 7FFF	Borrows 128KB from VE, borrows 64KB from DE, supports Byte operation, the clock source can be switched to AHB1)

3.5.2. Operations and Functional Descriptions

3.5.2.1. SRAM

The system SRAM includes SRAM A1 and SRAM C. The address between SRAM A1 and SRAM C is continuous.

The SRAM A1 is used in System area, the SRAM C is a memory which system borrows from specific module(such as DE, VE), only in special scene(such as BOOT, STANDBY, etc), the SRAM C will switch to system to use.

When the SRAM of the module switchs to SRAM C, then the clock of the SRAM switchs to AHB1, if using SRAM C, the switch needs be opened, and the bus gating of the module needs be opened, the SRAM can be accessed.

3.5.3. Register List

Module Name	Base Address
SYS_CFG	0x0300 0000

Register Name	Offset	Description
VER_REG	0x0024	Version Register
EMAC_EPHY_CLK_REG0	0x0030	EMAC-EPHY Clock Register 0

3.5.4. Register Description

3.5.4.1. 0x0024 Version Register

<table>
<tr><td colspan="3">Offset: 0x0024</td><td>Register Name: VER_REG</td></tr>
<tr><td>Bit</td><td>Read/Write</td><td>Default/Hex</td><td>Description</td></tr>
<tr><td>31:14</td><td>/</td><td>/</td><td>/</td></tr>
<tr><td>13:9</td><td>R</td><td>UDF</td><td>BOOT_SEL_PAD_STA
<table>
<tr><th>Bit[9]</th><th>Bit[10]</th><th>Bit[11]</th><th>Bit[12]</th><th>Bit[13]</th><th>Media</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>MLC NAND</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>SLC NAND</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>eMMC USER</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>eMMC_BOOT</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>SPI_NOR</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>SPI_NAND</td></tr>
</table>
Bit[9] --> BOOT SELECT0 (For H616 package, Bit[9] is fixed at 1)
Bit[10] --> PC3
Bit[11] --> PC4
Bit[12] --> PC5
Bit[13] --> PC6</td></tr>
<tr><td>8</td><td>R</td><td>UDF</td><td>FEL_SEL_PAD_STA
Fel_Select_Pin_Status
0: Run_FEL
1:Try Media Boot</td></tr>
<tr><td>7:0</td><td>/</td><td>/</td><td>/</td></tr>
</table>

3.5.4.2. 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	EMAC0 BPS_EFFUSE
27	R/W	0x0	EMAC0 XMII_SEL 0: Internal SMI and MII 1: External SMI and MII
26:25	R/W	0x0	EMAC0 EPHY_MODE Operation Mode Selection 00 : Normal Mode 01 : Sim Mode 10 : AFE Test Mode 11 : /
24:20	R/W	0x0	EMAC0 PHY_ADDR PHY Address
19	/	/	/

18	R/W	0x1	EMAC0 CLK_SEL 0 : 25 MHz 1 : 24 MHz
17	R/W	0x0	EMAC0 LED_POL 0 : High active 1 : Low active
16	R/W	0x1	EMAC0 SHUTDOWN 0 : Power up 1 : Shutdown
15	R/W	0x1	EMAC0 PHY_SELECT 0 : External PHY 1 : Internal PHY
14	/	/	/
13	R/W	0x0	EMAC0 RMII_EN 0 : Disable RMII Module 1 : Enable RMII Module When this bit is asserted, MII or RGMII interface is disabled(This means bit13 is prior to bit2)
12:10	R/W	0x0	EMAC0 ETXDC Configure EMAC Transmit Clock Delay Chain
9:5	R/W	0x0	EMAC0 ERXDC Configure EMAC Receive Clock Delay Chain
4	R/W	0x0	EMAC0 ERXIE Enable EMAC Receive Clock Invertor 0: Disable 1: Enable
3	R/W	0x0	EMAC0 ETXIE Enable EMAC Transmit Clock Invertor 0: Disable 1: Enable
2	R/W	0x0	EMAC0 EPIT EMAC PHY Interface Type 0: MII 1: RGMII
1:0	R/W	0x0	EMAC0 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

NOTE

When configuring RMII interface, the bit13 should be written to 1, and the bit2 should be written to 0. Select TXCLK as the clock source of RMII, the bit0 can be written to 0.

3.6. Timer

3.6.1. Overview

The timer module implements the timing and counting functions. The timer module includes timer0, timer1, watchdog and AVS0, AVS1.

The timer 0 and timer 1 are completely consistent. The timer 0 and timer 1 have the following features:

- Configurable count clock: LOSC and OSC24M. LOSC is the internal low-frequency clock or the external low-frequency clock by setting LOSC_SRC_SEL. The external low-frequency has much accuracy.
- Configurable 8 prescale factor
- Programmable 32-bit down timer
- Two working modes: continue mode and single count 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 after an exception occurs in the system. The watchdog has the following features:

- Single clock source: OSC24M/750
- 12 initial values to configure
- Generation of timeout interrupts
- Generation of reset signal
- Watchdog restart the timing

The AVS is used to the synchronization of audio and video. The AVS module includes AVS0 and AVS1, the AVS0 and AVS1 are completely consistent. The AVS has the following features:

- Single clock source: OSC24M
- Programmable 33-bit up timer
- Initial value can be updated anytime
- 12-bit frequency divider factor
- Pause/Start function

3.6.2. Block Diagram

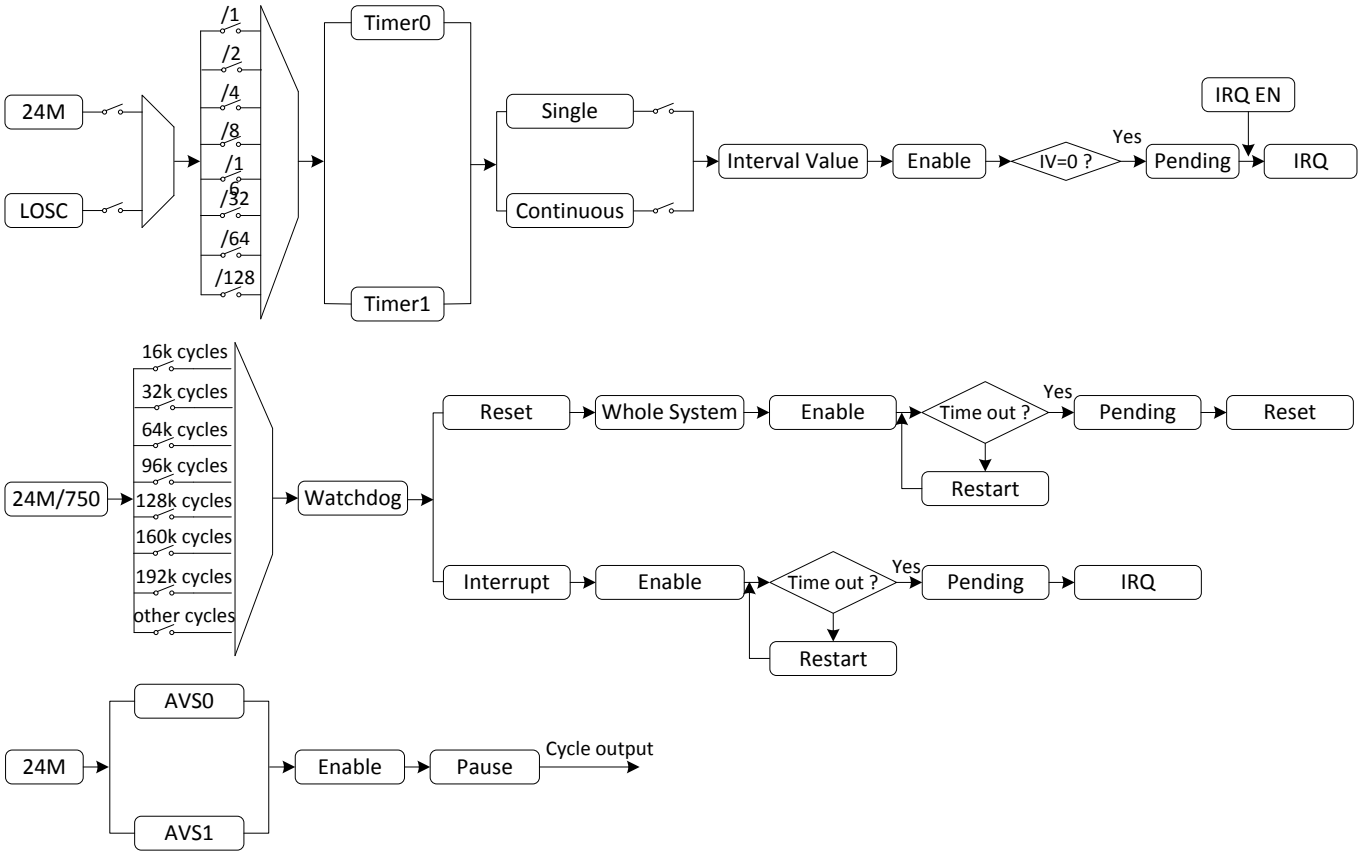

Figure 3- 12. Timer Block Diagram

3.6.3. Operations and Functional Descriptions

3.6.3.1. Timer Formula

Using Timer0 as an example.

$$T_{timer0} = \frac{TMR0\_INTV\_VALUE\_REG - TMR0\_CUR\_VALUE\_REG}{TMR0\_CLK\_SRC} \times TMR0\_CLK\_PRES$$

TMR0_INTV_VALUE_REG: timer initial value;
TMR0_CUR_VALUE_REG: timer current counter;
TMR0_CLK_SRC: timer clock source;
TMR0_CLK_PRES: timer clock prescale ratio.

3.6.3.2. Typical Application

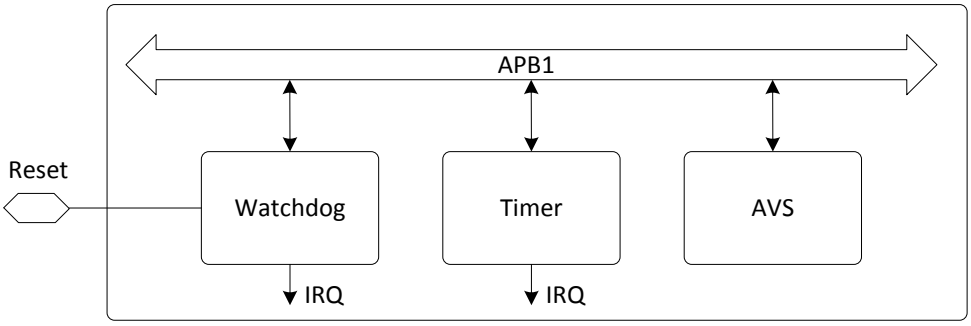

Figure 3- 13. Timer Application Diagram

Timer, watchdog and AVS configure register by APB1 bus.
Timer and watchdog have interrupt mode.
The system configures the time of watchdog, if the system has no timing for restart watchdog(such as bus hang dead), then watchdog sends out watchdog reset external signal to reset system; meanwhile watchdog outputs signal to RESET pad to reset PMIC.

3.6.3.3. Function Implemention

3.6.3.3.1. Timer

The timer is a 32-bit down counter, the counter value is decreased by 1 on each rising edge of the count clock. Each timer has independent interrupt.

The timer has two operating modes.

- **Continuous mode**

The bit7 of the TMRn_CTRL_REG is set to the continuous mode, when the count value is decreased to 0, the timer module reloads data from TMRn_INTV_VALUE_REG then continues to count.

- **Single mode**

The bit7 of the TMRn_CTRL_REG is set to the single mode, when the count value is decreased to 0, the timer stops counting. The timer starts to count again only when a new initial value is loaded.

Each timer has a prescaler that divides the working clock frequency of each timer by 1,2,4,8,16,32,64,128.

3.6.3.3.2. Watchdog

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

The watchdog has two operating modes.

- **Interrupt mode**

The WDOG0_CFG_REG is set to 0x2, when the counter value reaches 0 and WDOG0_IRQ_EN_REG is enabled, the watchdog generates an interrupt, the watchdog enters into interrupt mode.

- **Reset mode**

The WDOG0_CFG_REG is set to 0x1, when the counter value reaches 0, the watchdog generates a reset signal to reset the entire system.

The clock source of the watchdog is OSC24M/750. There are 12 configurable initial count values.
The watchdog can restart to count by setting the WDOG0_CTRL_REG: write 0xA57 to bit[12:1], then write 1 to bit[0].

3.6.3.3.3. AVS

The AVS is a 33-bit up counter. The counter value is increased by 1 on each rising edge of the count clock.
The AVS can be operated after its clock gating in CCU module is opened.
The AVS has an OSC24M clock source and a 12-bit division factor(N1 or N2). When the timer increases to N1 or N2 from 0, AVS counter adds 1; when the counter reaches 33-bit upper limit, the AVS will start to count from initial value again.
In counter working process, the division factor and initial counter of the AVS can be changed anytime. And the AVS can stop or start to operate counter anytime.

3.6.3.4. Operating Mode

3.6.3.4.1. Timer Initial

(1) Configure the timer parameters: clock source, prescale factor, working mode. The configuration of these parameters have no sequence, and can be implemented by writing **TMRn_CTRL_REG**.
(2) Write the initial value: write **TMRn_INTV_VALUE_REG** to provide an initial value for the timer; write the bit[1] of **TMRn_CTRL_REG** to load the initial value to the timer, if the bit[1] is 1, writing operation cannot perform; if is 0, this indicates successful loading.
(3) Enable timer: write the bit[0] of **TMRn_CTRL_REG** to enable timer count; read **TMRn_CUR_VALUE_REG** to get the current count value.

3.6.3.4.2. Timer Interrupt

(1) Enable interrupt: write corresponding interrupt enable bit of **TMR_IRQ_EN_REG**, when timer counter time reaches, the corresponding interrupt generates.
(2) After enter interrupt process, write **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.

3.6.3.4.3. Watchdog Initial

(1) Write **WDOG0_CFG_REG** to configure the generation of the interrupts and the output of reset signal.
(2) Write **WDOG0_MODE_REG** to configure the initial count value.
(3) Write **WDOG0_MODE_REG** to enable the watchdog.

3.6.3.4.4. Watchdog Interrupt

Watchdog interrupt is only used for the counter.
(1) Write **WDOG0_IRQ_EN_REG** to enable the interrupt.
(2) After enter the interrupt process, write **WDOG0_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.

3.6.3.4.5. AVS Start/Pause

(1) Write **AVS_CNT_DIV_REG** to configure the division factor.
(2) Write **AVS_CNT_REG** to configure the initial count value.
(3) Write **AVS_CNT_CTL_REG** to enable AVS counter. AVS counter can be paused at any time.

3.6.4. Programming Guidelines

3.6.4.1. Timer

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

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

3.6.4.2. Watchdog Reset

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

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



```
writel(readl(WDOG_MODE)|(1<<0), WDOG_MODE);     //Enable Watchdog
```

3.6.4.3. Watchdog Restart

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

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

3.6.5. Register List

Module Name	Base Address
Timer	0x03009000

Register Name	Offset	Description
TMR_IRQ_EN_REG	0x0000	Timer IRQ Enable Register
TMR_IRQ_STA_REG	0x0004	Timer Status Register
TMR0_CTRL_REG	0x0010	Timer 0 Control Register
TMR0_INTV_VALUE_REG	0x0014	Timer 0 Interval Value Register
TMR0_CUR_VALUE_REG	0x0018	Timer 0 Current Value Register
TMR1_CTRL_REG	0x0020	Timer 1 Control Register
TMR1_INTV_VALUE_REG	0x0024	Timer 1 Interval Value Register
TMR1_CUR_VALUE_REG	0x0028	Timer 1 Current Value Register
WDOG_IRQ_EN_REG	0x00A0	Watchdog IRQ Enable Register
WDOG_IRQ_STA_REG	0x00A4	Watchdog Status Register
WDOG_CTRL_REG	0x00B0	Watchdog Control Register
WDOG_CFG_REG	0x00B4	Watchdog Configuration Register
WDOG_MODE_REG	0x00B8	Watchdog Mode 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 Timer 1 Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	TMR0_IRQ_EN Timer 0 Interrupt Enable 0: Disable 1: Enable

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 Timer 1 IRQ Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending, timer 1 interval value is reached
0	R/W1C	0x0	TMR0_IRQ_PEND Timer 0 IRQ Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending, timer 0 interval value is reached

3.6.6.3. 0x0010 Timer 0 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 Timer 0 mode 0: Continuous mode. When interval value reached, the timer will not disable automatically. 1: Single mode. When interval value reached, the timer will disable automatically.
6:4	R/W	0x0	TMR0_CLK_PRES

			Select the pre-scale of timer 0 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 00:LOSC 01: OSC24M 10: / 11: /
1	R/W	0x0	TMR0_RELOAD Timer 0 Reload 0: No effect 1: Reload timer 0 Interval value After the bit is set, it can not be written again before it is cleared automatically.
0	R/W	0x0	TMR0_EN Timer 0 Enable 0: Stop/Pause 1: Start If the timer starts, it will reload the interval value to internal register, and the current counter will count from interval value to 0. If the current counter does not reach the zero, the timer enable bit is set to "0", the current value counter will pause. At least wait for 2 cycles, the start bit can be set to 1. In timer pause state, the interval value register can be modified. If the timer starts again, and the software hopes the current value register to down-count from the new interval value, the reload bit and the enable bit should be set to 1 at the same time.

3.6.6.4. 0x0014 Timer 0 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 Timer 0 Interval Value

NOTE

The value setting should consider the system clock and the timer clock source.

3.6.6.5. 0x0018 Timer 0 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 Timer 0 Current Value Timer 0 current value is a 32-bit down-counter (from interval value to 0).

3.6.6.6. 0x0020 Timer 1 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 Timer 1 mode 0: Continuous mode. When interval value reached, the timer will not disable automatically. 1: Single mode. When interval value reached, the timer will disable automatically.
6:4	R/W	0x0	TMR1_CLK_PRES Select the pre-scale of timer 1 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 00: LOSC 01: OSC24M 10: / 11: /
1	R/W	0x0	TMR1_RELOAD Timer 1 Reload 0: No effect 1: Reload timer 1 Interval value After the bit is set, it can not be written again before it is cleared automatically.
0	R/W	0x0	TMR1_EN Timer 1 Enable 0: Stop/Pause 1: Start If the timer starts, it will reload the interval value to internal register, and the

			current counter will count from interval value to 0. If the current counter does not reach the zero, the timer enable bit is set to "0", the current value counter will pause. At least wait for 2 cycles, the start bit can be set to 1. In timer pause state, the interval value register can be modified. If the timer starts again, and the software hopes the current value register to down-count from the new interval value, the reload bit and the enable bit should be set to 1 at the same time.

3.6.6.7. 0x0024 Timer 1 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 Timer 1 Interval Value

NOTE

The value should consider the system clock and the timer clock source.

3.6.6.8. 0x0028 Timer 1 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 Timer 1 Current Value Timer 1 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/W1S	0x0	WDOG_IRQ_EN Watchdog Interrupt Enable 0: No effect 1: Watchdog interrupt enable

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 Watchdog IRQ Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending. Watchdog interval value is reached.

3.6.6.11. 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	R/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.
0	R/W1S	0x0	WDOG_RESTART Watchdog Restart 0: No effect 1: Restart the Watchdog

3.6.6.12. 0x00B4 Watchdog Configuration Register (Default Value: 0x0000_0001)

Offset: 0x00B4			Register Name:WDOG_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1:0	R/W	0x1	WDOG_CONFIG 00: / 01: To whole system 10: Only interrupt 11: /

3.6.6.13. 0x00B8 Watchdog Mode Register (Default Value: 0x0000_0000)

Offset: 0x00B8			Register Name:WDOG_MODE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:4	R/W	0x0	WDOG_INTV_VALUE Watchdog Interval Value Watchdog clock source is OSC24M/750. If the clock source is turned off, Watchdog will not work.

			0000: 16000 cycles (0.5s) 0001: 32000 cycles (1s) 0010: 64000 cycles (2s) 0011: 96000 cycles (3s) 0100: 128000 cycles (4s) 0101: 160000 cycles (5s) 0110: 192000 cycles (6s) 0111: 256000 cycles (8s) 1000: 320000 cycles (10s) 1001: 384000 cycles (12s) 1010: 448000 cycles (14s) 1011: 512000 cycles (16s) Others: Reserved
3:1	/	/	/
0	R/W1S	0x0	WDOG_EN Watchdog Enable 0: No effect 1: Enable the Watchdog

3.6.6.14. 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: Not pause 1: Pause Counter 1
8	R/W	0x0	AVS_CNT0_PS Audio/Video Sync Counter 0 Pause Control 0: Not pause 1: Pause Counter 0
7:2	/	/	/
1	R/W	0x0	AVS_CNT1_EN Audio/Video Sync Counter 1 Enable/Disable The counter source is OSC24M. 0: Disable 1: Enable
0	R/W	0x0	AVS_CNT0_EN Audio/Video Sync Counter 0 Enable/Disable The counter source is OSC24M. 0: Disable 1: Enable

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

3.6.6.16. 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 Counter 1 for Audio/Video Sync Application The high 32 bits of the internal 33-bit counter register. The initial value of the internal 33-bit counter register can be set by software. The LSB bit of the 33-bit counter register should be zero when the initial value is updated. It will count from the initial value. The initial value can be updated at any time. It can also be paused by setting AVS_CNT1_PS to '1'. When it is paused, the counter will not increase.

3.6.6.17. 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	/	/	/
27:16	R/W	0x5DB	AVS_CNT1_D Divisor N for AVS Counter 1 AVS CN1 CLK=24 MHz/Divisor_N1. Divisor N1 = Bit [27:16] + 1. The number N is from 1 to 0x7ff. The zero value is reserved. The internal 33-bit counter engine will maintain another 12-bit counter. The 12-bit counter is used for counting the cycle number of one 24 MHz clock. When the 12-bit counter reaches (>= N) the divisor value, the internal 33-bit counter register will increase 1 and the 12-bit counter will reset to zero and restart again. It can be configured by software at any time.
15:12	/	/	/

11:0	R/W	0x5DB	AVS_CNT0_D Divisor N for AVS Counter 0 AVS CN0 CLK=24 MHz/Divisor_N0. Divisor N0 = Bit [11:0] + 1 The number N is from 1 to 0x7ff. The zero value is reserved. The internal 33-bit counter engine will maintain another 12-bit counter. The 12-bit counter is used for counting the cycle number of one 24 MHz clock. When the 12-bit counter reaches (>= N) the divisor value, the internal 33-bit counter register will increase 1 and the 12-bit counter will reset to zero and restart again. It can be configured by software at any time.

3.7. High Speed Timer

3.7.1. Overview

The high speed timer(HSTimer) module implements more precise timing and counting functions.

The HSTimer has the following features:

- Timing clock is AHB1 that can provides more precise timing clock
- Configurable 5 prescale factor
- Configurable 56-bit down timer
- Supports 2 working modes: continuous mode and single mode
- Supports test mode
- Generates an interrupt when the count is decreased to 0

3.7.2. Block Diagram

Figure 3-14 shows a block diagram of the HSTimer.

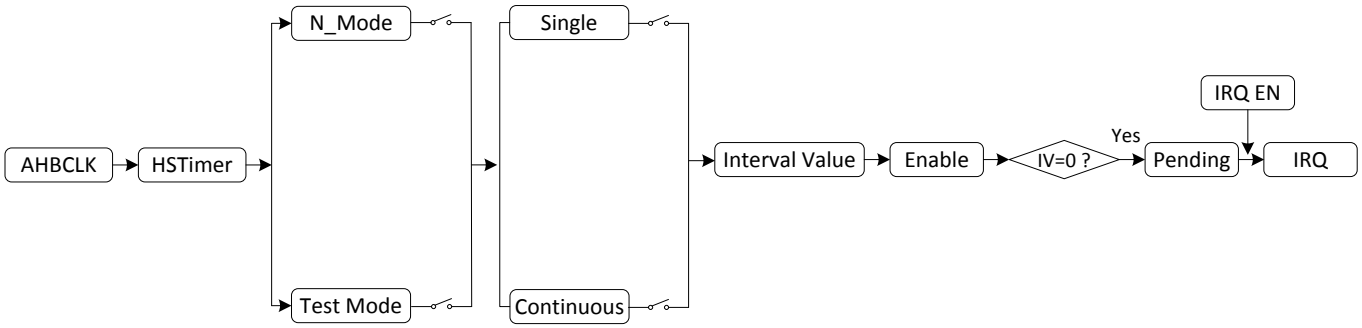

Figure 3- 14. HSTimer Block Diagram

3.7.3. Operations and Functional Descriptions

3.7.3.1. HSTimer Formula

$$\frac{(HS\_TMR\_INTV\_HI\_REG<<32 + HS\_TMR\_INTV\_LO\_REG) - (HS\_TMR\_CURNT\_HI\_REG<<32 + HS\_TMR\_CURNT\_LO\_REG)}{AHB1CLK} \times HS\_TMR\_CLK$$

HS_TMR_INTV_HI_REG: Initial of Counter Higher Bit
HS_TMR_INTV_LO_REG: Initial of Counter Lower Bit
HS_TMR_CURNT_HI_REG: Current Value of Counter Higher Bit
HS_TMR_CURNT_LO_REG: Current Value of Counter Lower Bit

AHB1CLK: AHB1 Clock Frequency
HS_TMR_CLK: Time Prescale Ratio of Counter

3.7.3.2. Typical Application

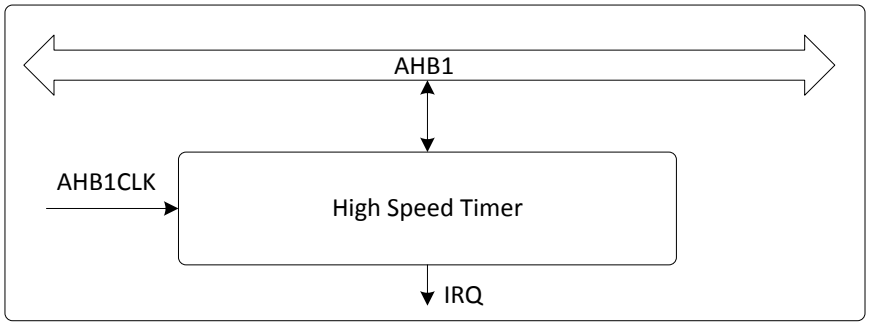

Figure 3- 15. HSTimer Application Diagram

The high speed timer is on AHB1, and the high speed timer controls registers by AHB1.
The high speed timer has single clock source: AHB. The high speed timer can generate interrupt.

3.7.3.3. Function Implemention

The high speed timer is a 56-bit down counter, the counter value is decreased by 1 on each rising edge of the count clock.

The high speed timer has two timing modes.

- Continuous mode: The bit7 of **HS_TMR0_CTRL_REG** is set to the continuous mode, when the count value is decreased to 0, the high speed timer module reloads data from **HS_TMR_INTV_LO_REG** and **HS_TMR_INTV_HI_REG,** then continues to count.
- Single mode: The bit7 of **HS_TMR0_CTRL_REG** is set to the single mode, when the count value is decreased to 0, the high speed timer stops counting. The high speed timer starts to count again only when a new initial value is loaded.

The high speed timer has two operating modes.

- Normal mode: When the bit31 of **HS_TMR0_CTRL_REG** is set to the normal mode, the high speed timer is used as 56-bit down counter, which can finish continuous timing and single timing.
- Test mode: When the bit31 of **HS_TMR0_CTRL_REG** is set to the test mode, then **HS_TMR_INTV_LO_REG** must be set to 0x1, the high speed timer is used as 24-bit down counter, and **HS_TMR_INTV_HI_REG** is the initial value of the high speed timer.

Each high speed timer has a prescaler that divides the working clock frequency of each working timer by 1,2,4,8,16.

3.7.3.4. Operating Mode

3.7.3.4.1. HSTimer Initial

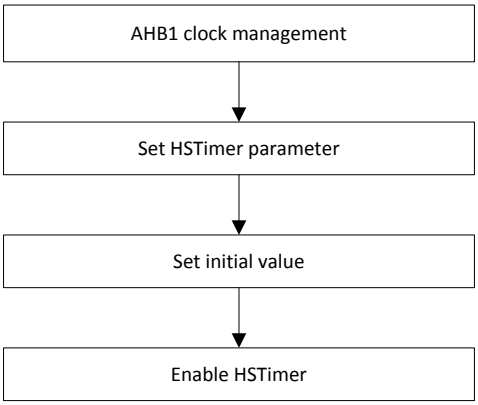

Figure 3- 16. HSTimer Initialization Process

(1) AHB1 clock management: Open the clock gating of AHB1 and de-assert the soft reset of AHB1 in CCU.
(2) Configure the corresponding parameters of the high speed timer: clock source, prescaler factor, working mode, counting mode. These parameters that are written to **HS_TMR0_CTRL_REG** have no sequences.
(3) Write the initial value: Firstly write the low-bit register **(HS_TMR_INTV_LO_REG)**, then write the high-bit register **(HS_TMR_INTV_HI_REG)**. Write the bit1 of **HS_TMR0_CTRL_REG** to load the initial value. If in timing stop stage of high speed timer, write the bit1 and bit0 of **HS_TMR0_CTRL_REG** to reload the initial value.
(4) Enable high speed timer: Write the bit[0] of **HS_TMR0_CTRL_REG** to enable high speed timer to count.
(5) Reading **HS_TMR_CURNT_LO_REG** and **HS_TMR_CURNT_HI_REG** can get current counting value.

3.7.3.4.2. HSTimer Interrupt

(1) Enable interrupt: Write the corresponding interrupt enable bit of **HS_TMR_IRQ_EN_REG**, when the counting time of high speed timer reaches, the corresponding interrupt generates.
(2) After enter 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

Take making a 1us delay using HSTimer0 for an instance as follows, AHB1CLK will be configured as 100 MHz and n_mode,single mode and 2 pre-scale will be selected in this instance.

```
writel(0x32, HS_TMR0_INTV_LO);                  //Set interval value Lo 0x32
writel(0x0, HS_TMR0_INTV_HI);                   //Set interval value Hi 0x0
writel(0x90, HS_TMR0_CTRL);                     //Select n_mode,2 pre-scale,single mode
writel(readl(HS_TMR0_CTRL)|(1<<1), HS_TMR0_CTRL);   //Set 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
High Speed Timer	0x03005000

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 Timer 0 Control Register
HS_TMR0_INTV_LO_REG	0x0024	HS Timer 0 Interval Value Low Register
HS_TMR0_INTV_HI_REG	0x0028	HS Timer 0 Interval Value High Register
HS_TMR0_CURNT_LO_REG	0x002C	HS Timer 0 Current Value Low Register
HS_TMR0_CURNT_HI_REG	0x0030	HS Timer 0 Current Value High Register
HS_TMR1_CTRL_REG	0x0040	HS Timer 1 Control Register
HS_TMR1_INTV_LO_REG	0x0044	HS Timer 1 Interval Value Low Register
HS_TMR1_INTV_HI_REG	0x0048	HS Timer 1 Interval Value High Register
HS_TMR1_CURNT_LO_REG	0x004C	HS Timer 1 Current Value Low Register
HS_TMR1_CURNT_HI_REG	0x0050	HS Timer 1 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/W1S	0x0	HS_TMR1_INT_EN High Speed Timer 1 Interrupt Enable 0: No effect 1: High Speed Timer1 interval value reached interrupt enable
0	R/W1S	0x0	HS_TMR0_INT_EN High Speed Timer 0 Interrupt Enable 0: No effect 1: High Speed Timer0 interval value reached interrupt enable

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 High Speed Timer 1 IRQ Pending Setting 1 to the bit will clear it. 0: No effect 1: Pending, High speed timer 1 interval value is reached.
0	R/W1C	0x0	HS_TMR0_IRQ_PEND High Speed Timer 0 IRQ Pending Setting 1 to the bit will clear it. 0: No effect 1: Pending, High speed timer 0 interval value is reached.

3.7.6.3. 0x0020 HS Timer 0 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 High Speed Timer 0 Test Mode In test mode, the low register should be set to 0x1, the high register will down counter. The counter needs to be reloaded. 0: Normal mode 1: Test mode
30:8	/	/	/
7	R/W	0x0	HS_TMR0_MODE High Speed Timer 0 Mode 0: Continuous mode. When interval value reached, the timer will not disable automatically. 1: Single mode. When interval value reached, the timer will disable automatically.
6:4	R/W	0x0	HS_TMR0_CLK Select the pre-scale of the high speed timer 0 clock sources. 000: /1 001: /2 010: /4 011: /8 100: /16 101: / 110: / 111: /
3:2	/	/	/

1	R/W1S	0x0	HS_TMR0_RELOAD High Speed Timer 0 Reload 0: No effect 1: Reload High Speed Timer 0 Interval Value
0	R/W	0x0	HS_TMR0_EN High Speed Timer 0 Enable 0: Stop/Pause 1: Start If the timer starts, it will reload the interval value to internal register, and the current counter will count from interval value to 0. If the current counter does not reach the zero, the timer enable bit is set to "0", the current value counter will pause. At least wait for 2 cycles, the start bit can be set to 1. In timer pause state, the interval value register can be modified. If the timer starts again, and the software hopes the current value register to down-count from the new interval value, the reload bit and the enable bit should be set to 1 at the same time.

3.7.6.4. 0x0024 HS Timer 0 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 High Speed Timer 0 Interval Value [31:0]

3.7.6.5. 0x0028 HS Timer 0 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 High Speed Timer 0 Interval Value [55:32]

NOTE

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

3.7.6.6. 0x002C HS Timer 0 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

			High Speed Timer 0 Current Value [31:0]

3.7.6.7. 0x0030 HS Timer 0 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 High Speed Timer 0 Current Value [55:32]

NOTE

HS timer current value is a 56-bit down-counter (from interval value to 0).

The current value register is a 56-bit register. When read or write the current value, the Low register should be read or written first.

3.7.6.8. 0x0040 HS Timer 1 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 High Speed Timer 1 Test Mode In test mode, the low register should be set to 0x1, the high register will down counter. The counter needs to be reloaded. 0: Normal mode 1: Test mode
30:8	/	/	/
7	R/W	0x0	HS_TMR1_MODE High Speed Timer 1 Mode 0: Continuous mode. When interval value reached, the timer will not disable automatically. 1: Single mode. When interval value reached, the timer will disable automatically.
6:4	R/W	0x0	HS_TMR1_CLK Select the pre-scale of the high speed timer 1 clock sources. 000: /1 001: /2 010: /4 011: /8 100: /16 101: / 110: / 111: /
3:2	/	/	/

1	R/W1S	0x0	HS_TMR1_RELOAD High Speed Timer 1 Reload 0: No effect 1: Reload High Speed Timer 1 Interval Value
0	R/W	0x0	HS_TMR1_EN High Speed Timer 1 Enable 0: Stop/Pause 1: Start If the timer starts, it will reload the interval value to internal register, and the current counter will count from interval value to 0. If the current counter does not reach the zero, the timer enable bit is set to "0", the current value counter will pause. At least wait for 2 cycles, the start bit can be set to 1. In timer pause state, the interval value register can be modified. If the timer starts again, and the software hopes the current value register to down-count from the new interval value, the reload bit and the enable bit should be set to 1 at the same time.

3.7.6.9. 0x0044 HS Timer 1 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 High Speed Timer 1 Interval Value [31:0]

3.7.6.10. 0x0048 HS Timer 1 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 High Speed Timer 1 Interval Value [55:32]

NOTE

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

3.7.6.11. 0x004C HS Timer 1 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

			High Speed Timer 1 Current Value [31:0]

3.7.6.12. 0x0050 HS Timer 1 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 High Speed Timer 1 Current Value [55:32]

NOTE

HS timer current value is a 56-bit down-counter (from interval value to 0).

The current value register is a 56-bit register. When read or write the current value, the Low register should be read or written first.

3.8. GIC

3.8.1. Interrupt Source

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
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	UART0	0x0080	UART0 interrupt
33	UART1	0x0084	UART1 interrupt
34	UART2	0x0088	UART2 interrupt
35	UART3	0x008C	UART3 interrupt

Interrupt Number	Interrupt Source	Interrupt Vector	Description
36	UART4	0x0090	UART4 interrupt
37	UART5	0x0094	UART5 interrupt
38	TWI0	0x0098	TWI0 interrupt
39	TWI1	0x009C	TWI1 interrupt
40	TWI2	0x00A0	TWI2 interrupt
41	TWI3	0x00A4	TWI3 interrupt
42	TWI4	0x00A8	TWI4 interrupt
43	/	0x00AC	/
44	SPI0	0x00B0	SPI0 interrupt
45	SPI1	0x00B4	SPI1 interrupt
46	EMAC0	0x00B8	EMAC0 interrupt
47	EMAC1	0x00BC	EMAC1 interrupt
48	PWM	0x00C0	PWM interrupt
49	TS	0x00C4	Transport stream interrupt
50	/	0x00C8	/
51	THS	0x00CC	Thermal sensor control interrupt
52	LRADC	0x00D0	LRADC interrupt
53	OWA	0x00D4	OWA interrupt
54	DMIC	0x00D8	DMIC interrupt
55	AudioCodec_ADC	0x00DC	AudioCodec_ADC interrupt
56	AudioHub	0x00E0	AudioHub interrupt
57	USB2.0_OTG_DEVICE	0x00E4	USB2.0_OTG_DEVICE interrupt
58	USB2.0_OTG_EHCI	0x00E8	USB2.0_OTG_EHCI interrupt
59	USB2.0_OTG_OHCI	0x00EC	USB2.0_OTG_OHCI interrupt
60	USB2.0_HOST1_EHCI	0x00F0	USB2.0_HOST1_EHCI interrupt
61	USB2.0_HOST1_OHCI	0x00F4	USB2.0_HOST1_OHCI interrupt
62	USB2.0_HOST2_EHCI	0x00F8	USB2.0_HOST2_EHCI interrupt
63	USB2.0_HOST2_OHCI	0x00FC	USB2.0_HOST2_OHCI interrupt
64	USB2.0_HOST3_EHCI	0x0100	USB2.0_HOST3_EHCI interrupt
65	USB2.0_HOST3_OHCI	0x0104	USB2.0_HOST3_OHCI interrupt
Memory			
66	NAND0	0x0108	NAND0 interrupt
67	SMHC0	0x010C	SMHC0 interrupt
68	SMHC1	0x0110	SMHC1 interrupt
69	SMHC2	0x0114	SMHC2 interrupt
70	MSI	0x0118	MSI interrupt
71	/	0x011C	/
72	DRAM_PHY	0x0120	DRAM_PHY interrupt
System			
73	CLK_DET	0x0124	Clock detect interrupt
74	DMA	0x0128	DMA interrupt
75	GPIOE	0x012C	GPIOE interrupt

Interrupt Number	Interrupt Source	Interrupt Vector	Description
76	/	0x0130	/
77	HSTIMER0	0x0134	High speed timer0 interrupt
78	HSTIMER1	0x0138	High speed timer1 interrupt
79	SMC	0x013C	Secure memory control interrupt
80	TIMER0	0x0140	Timer0 interrupt
81	TIMER1	0x0144	Timer1 interrupt
82	WDOG	0x0148	Watchdog interrupt
83	/	0x014C	/
84	GPIOC	0x0150	GPIOC interrupt
85	/	0x0154	/
86	GPIOF	0x0158	GPIOF interrupt
87	GPIOG	0x015C	GPIOG interrupt
88	GPIOH	0x0160	GPIOH interrupt
89	GPIOI	0x0164	GPIOI interrupt
90	AudioCodec_DAC	0x0168	AudioCodec_DAC interrupt
91	PSI	0x016C	PSI interrupt
92	BUS_TIMEOUT	0x0170	BUS timeout interrupt
93	IOMMU	0x0174	IOMMU interrupt
94	/	0x0178	/
Display Interface			
95	HDMI_TX0	0x017C	HDMI_TX0 interrupt
96	/	0x0180	/
97	/	0x0184	/
98	TCON_TV0	0x0188	TCON_TV0 interrupt
99	TCON_TV1	0x018C	TCON_TV1 interrupt
100	TVE	0x0190	TV encoder interrupt
101	/	0x0194	/
102	/	0x0198	/
103	/	0x019C	/
104	/	0x01A0	/
105	/	0x01A4	/
106	/	0x01A8	/
107	/	0x01AC	/
108	/	0x01B0	/
109	/	0x01B4	/
110	/	0x01B8	/
111	/	0x01BC	/
112	/	0x01C0	/
113	/	0x01C4	/
114	/	0x01C8	/
115	/	0x01CC	/
116	/	0x01D0	/

Interrupt Number	Interrupt Source	Interrupt Vector	Description
117	/	0x01D4	/
118	/	0x01D8	/
119	/	0x01DC	/
Accelerator			
120	DE	0x01E0	DE interrupt
121	DI	0x01E4	DI interrupt
122	G2D	0x01E8	G2D interrupt
123	CE_NS	0x01EC	CE_NS interrupt
124	CE	0x01F0	CE interrupt
125	VE	0x01F4	VE interrupt
126	GPU_EVENT	0x01F8	GPU_EVENT interrupt
127	GPU_JOB	0x01FC	GPU_JOB interrupt
128	GPU_MMU	0x0200	GPU_MMU interrupt
129	GPU	0x0204	GPU interrupt
130	/	0x0208	/
131	/	0x020C	/
132	/	0x0210	/
133	/	0x0214	/
134	/	0x0218	/
Other			
135	NMI	0x021C	NMI interrupt
136	R_Alarm0	0x0220	R_Alarm0 interrupt
137	S_TWI0	0x0224	S_TWI0 interrupt
138	CIR_RX	0x0228	CIR_RX interrupt
139	R_CPU_IDLE	0x022C	R_CPU_IDLE interrupt
140	TWD	0x0230	Trust watchdog interrupt
141	/	0x0234	/
142	/	0x0238	/
143	/	0x023C	/
144	/	0x0240	/
145	/	0x0244	/
146	/	0x0248	/
147	/	0x024C	/
CPUX Related			
160	C0_CTI0	0x0280	C0_CTI0 interrupt
161	C0_CTI1	0x0284	C0_CTI1 interrupt
162	C0_CTI2	0x0288	C0_CTI2 interrupt
163	C0_CTI3	0x028C	C0_CTI3 interrupt
164	C0_COMMTX0	0x0290	C0_COMMTX0 interrupt
165	C0_COMMTX1	0x0294	C0_COMMTX1 interrupt
166	C0_COMMTX2	0x0298	C0_COMMTX2 interrupt
167	C0_COMMTX3	0x029C	C0_COMMTX3 interrupt

Interrupt Number	Interrupt Source	Interrupt Vector	Description
168	C0_COMMRX0	0x02A0	C0_COMMRX0 interrupt
169	C0_COMMRX1	0x02A4	C0_COMMRX1 interrupt
170	C0_COMMRX2	0x02A8	C0_COMMRX2 interrupt
171	C0_COMMRX3	0x02AC	C0_COMMRX3 interrupt
172	C0_PMU0	0x02B0	C0_PMU0 interrupt
173	C0_PMU1	0x02B4	C0_PMU1 interrupt
174	C0_PMU2	0x02B8	C0_PMU2 interrupt
175	C0_PMU3	0x02BC	C0_PMU3 interrupt
176	C0_AXI_ERROR	0x02C0	C0_AXI_ERROR interrupt
177	AXI_WR_IRQ	0x02C4	AXI_WR_IRQ interrupt
178	AXI_RD_IRQ	0x02C8	AXI_RD_IRQ interrupt
179	DBGRSTREQ0	0x02CC	DBGRSTREQ0 interrupt
180	DBGRSTREQ1	0x02D0	DBGRSTREQ1 interrupt
181	DBGRSTREQ2	0x02D4	DBGRSTREQ2 interrupt
182	DBGRSTREQ3	0x02D8	DBGRSTREQ3 interrupt
183	nVCPUMNTIRQ0	0x02DC	nVCPUMNTIRQ0 interrupt
184	nVCPUMNTIRQ1	0x02E0	nVCPUMNTIRQ1 interrupt
185	nVCPUMNTIRQ2	0x02E4	nVCPUMNTIRQ2 interrupt
186	nVCPUMNTIRQ3	0x02E8	nVCPUMNTIRQ3 interrupt
187	nCOMMIRQ0	0x02EC	nCOMMIRQ0 interrupt
188	nCOMMIRQ1	0x02F0	nCOMMIRQ1 interrupt
189	nCOMMIRQ2	0x02F4	nCOMMIRQ2 interrupt
190	nCOMMIRQ3	0x02F8	nCOMMIRQ3 interrupt
191	DBGPWRUPREQ_out	0x02FC	DBGPWRUPREQ_out interrupt

For complete GIC information, refer to the ***GIC PL400 technical reference manual*** and ***ARM GIC Architecture Specification V2.0***.

3.9. DMA

3.9.1. Overview

The direction memory access (DMA) is used to transfer data between a peripheral and a memory, between peripherals, or between memories. DMA is a high-speed data transfer operation that reduces the CPU resources.

The DMA has the following features:

- 16 channels DMA
- Provides 32 peripheral DMA requests for data read and 32 peripheral DMA requests for data write
- Transfer with linked list
- Programmable 8-,16-,32-,64-bit data width
- Programmable DMA burst length
- DRQ response includes wait mode and handshake mode
- Memory devices support non-aligned transform
- DMA channel supports pause function

3.9.2. Block Diagram

The following figure shows a block diagram of DMA.

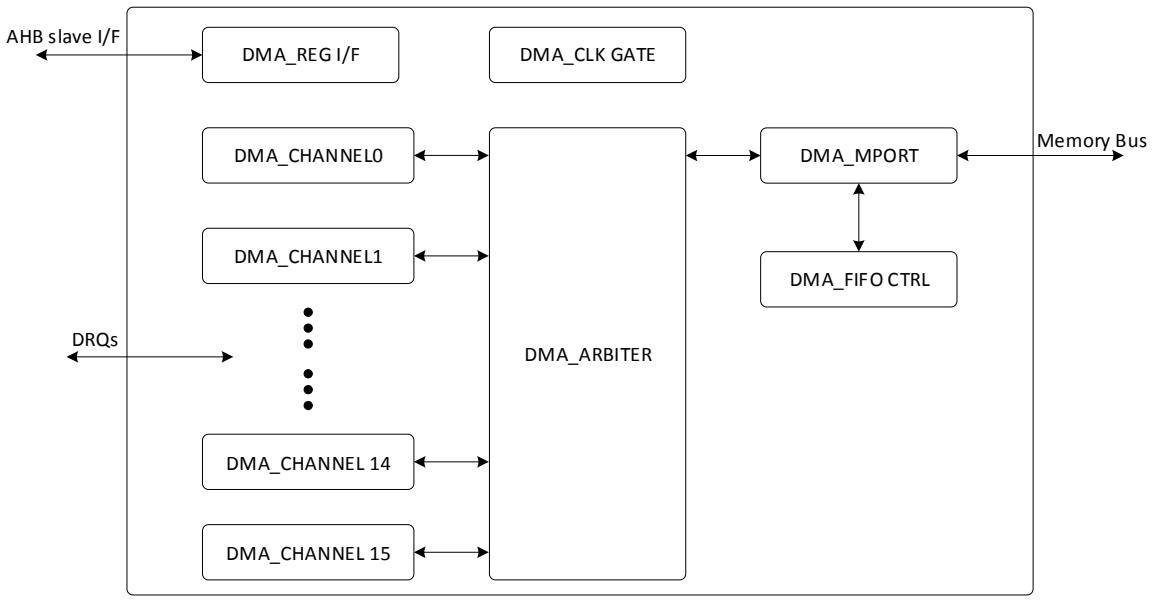

Figure 3- 17. DMA Block Diagram

DMA_ARBITER: Arbitrate DMA read/write requirement of each channel, and convert to read/write requirement of each port.

DMA_CHANNEL: DMA transform engine. Each channel is independent. The priorities of DMA channels use polling mechanism. When the DMA requests from two peripherals are valid simultaneously, if DMA_ARBITER is non-idle, the

next channel of the current channel has the higher priority; if DMA_ARBITER is idle, the channel0 has the highest priority, whereas the channel 15 has the lowest priority.
DMA_MPORT: Receive read/write requirement of DMA_ARBITER, and convert to the corresponding MBUS access.
DMA_FIFOCTL: Internal FIFO cell control module.
DMA_REGIF: Common register module, mainly used to resolve AHB1 demand.
DMA_CLKGATE: Hardware auto clock gating control module.

DMA integrates 16 independent DMA channels. When DMA channel starts, DMA gets DMA descriptor by DMA_DESC_ADDR_REG to use for the configuration information of the current DMA package transfer, and DMA can transfer data between the specified peripherals through the configuration information. When a package transfer finished, DMA judges whether the current channel transfer finished or continues to obtain/transfer the descriptor of the next package through the linked information in descriptor. When the chained address information of the descriptor indicates the current channel transfer is completed, DMA will close chain-transfer and the channel.

3.9.3. Operations and Functional Descriptions

3.9.3.1. Clock and Reset

DMA is on AHB1. The clock of AHB1 influences the transfer efficiency of DMA.

3.9.3.2. Typical Application

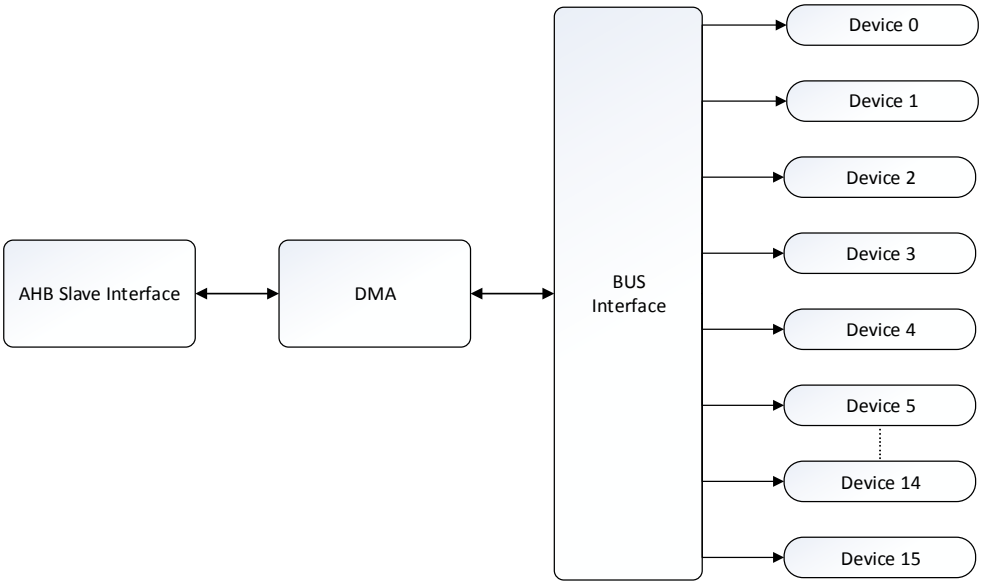

Figure 3- 18. DMA Typical Application Diagram

3.9.3.3. DRQ Type

Table 3- 7. DMA DRQ Table

Source DRQ Type		Destination DRQ Type	
port0	SRAM	port0	SRAM
port1	DRAM	port1	DRAM
port2		port2	OWA
port3	AHUB_drqr0	port3	AHUB_drqt0
port4	AHUB_drqr1	port4	AHUB_drqt1
port5	AHUB_drqr2	port5	AHUB_drqt2
port6		port6	Audio Codec
port7	DMIC	port7	
port8		port8	
port9		port9	
port10	NAND0	port10	NAND0
port11		port11	
port12		port12	
port13		port13	
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	OTG_EP1	Port30	OTG_EP1
Port31	OTG_EP2	Port31	OTG_EP2
Port32	OTG_EP3	Port32	OTG_EP3
Port33	OTG_EP4	Port33	OTG_EP4
Port34	OTG_EP5	Port34	OTG_EP5
Port35		Port35	
Port36		Port36	
Port37		Port37	
Port38		Port38	
Port39		Port39	
Port40		Port40	

Port41		Port41	
Port42		Port42	
Port43	TWI0	Port43	TWI0
Port44	TWI1	Port44	TWI1
Port45	TWI2	Port45	TWI2
Port46	TWI3	Port46	TWI3
Port47	TWI4	Port47	TWI4
Port48	S_TWI0	Port48	S_TWI0

3.9.3.4. DMA Descriptor

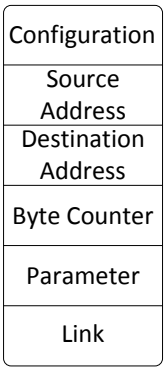

Figure 3- 19. DMA Descriptor

DMA descriptor is the configuration information of DMA transfer that decides the DMA working mode. Each descriptor includes 6 words, in turn, configuration, source address, destination address, byte counter, parameter, link.

(1) Configuration : Configure the following information by DMA_CFG_REG.

- **DRQ type of source and destination:** the DRQ signal of devices is as driving signal of DMA transfer.
- **Transferred address count mode**: IO mode indicates the address is fixed during transfer; linear mode indicates the address is increasing during transfer.
- **Transferred block length**: block length is the amount of DMA transferred data in one-shot valid DRQ. The block length supports 1-bit,4-bit,8-bit or 16-bit mode.
- **Transferred data width**: data width indicates the data width of every operation, and supports 8-bit,16-bit,32-bit or 64-bit mode.

(2) Source Address: Configure the transferred source address.

(3) Destination Address: Configure the transferred destination address.

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

(4) Byte counter: Configure the amount of a package. The maximum package is not more than (2^25-1) bytes. If the amount of the package reaches the maximum value, even if DRQ is valid, DMA should stop the current transfer.

(5) Parameter: Configure the interval between data block. The parameter is valid for non-memory peripherals. When DMA detects that DRQ is high level, DMA transfers block cycle. And during time, the changing of DRQ is ignored. After transferred, DMA waits the setting cycle(WAIT_CYC), then executes the next DRQ detection.

(6) Link: If the value of the link is 0xFFFFF800, the current package is at the end of the linked list. DMA will stop transfer after the package is transferred; if the value of the link is not 0xFFFFF800, the value of the link is considered the descriptor address of the next package.

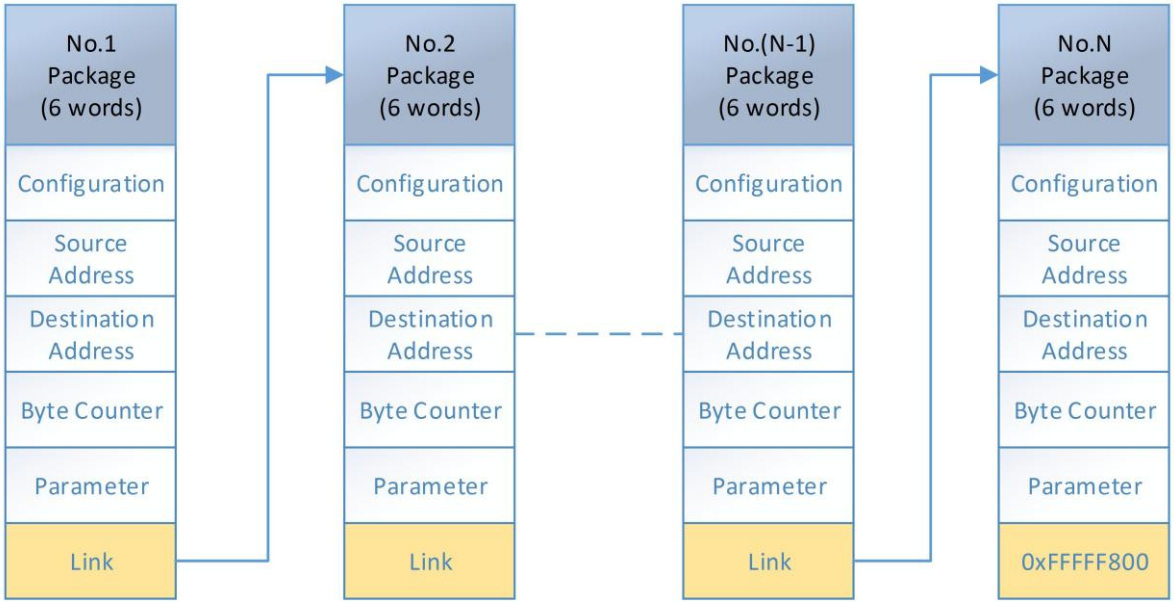

Figure 3- 20. DMA Chain Transfer

3.9.3.5. Interrupt

The half package interrupt is enabled, DMA sends half package interrupt after the half package transfer completes. The total package interrupt is enabled, DMA sends package end interrupt after the total package transfer completes. The total queue interrupt is enabled, DMA sends queue end interrupt after the total queue completes.
Notice that when CPU does not respond to the interrupts timely, or two DMA interrupts generate very closely, the later interrupt may override the former one. So For CPU, the DMA has only a system interrupt source.

3.9.3.6. Security

DMA supports system Trustzone, and supports DMA channel secure mode. Each DMA channel is secure by default. When system Trustzone is enabled, DMA is secure, only the secure devices can access DMA.

When DMA channel is configured to non-secure, then the channel can only access the non-secure memory area. DMA cannot write data to secure memory area, the read-back data from secure memory area is 0.

3.9.3.7. Clock Gating

DMA CLK GATE module is the clock module of auto-controlled by hardware. DMA CLK GATE module is mainly used to generate the clock of DMA sub-module and the local circuit in module, including clock gating of channel and clock gating of public part.

The clock gating of the channel indicates DMA clock can auto-open when the system accesses the current DMA channel register and DMA channel is enabled. When DMA transfer is completed, DMA channel clock can auto-close after 16 HCLK delay, meanwhile the clock of the corresponding channel control and FIFO control will be closed.

The clock gating of the common part indicates the clock of the common circuit can auto-close when all DMA channels are opened. The common circuit includes the common circuit of FIFO control module, MPORT module and memory bus clock.

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

3.9.3.8. Transfer Mode

DMA supports two data transfer modes: wait mode and handshake mode.

(1) Wait Mode

When device request signal enters DMA, the device request signal is transformed into the internal DRQ signal through block and wait counter. The transformed principle is as follows.

- When DMA detects the external request signal valid, DMA starts to operate the device, the internal DRQ always holds high level before the block operating amount reaches.
- When the transfer amount of DMA reaches the block operating amount, the internal DRQ pulls low automatically.
- After the internal DRQ holds low automatically at the DMA cycle of wait counter times, DMA restarts to detect the external request, if the external request signal is valid, then the next transfer starts.

(2) Handshake Mode

- When DMA detects the external request signal valid, DMA starts to operate the device, the internal DRQ always holds high level before the block operating amount reaches.
- When the transfer amount of DMA reaches the block operating amount, the internal DRQ pulls low automatically; meanwhile within the last DMA operation before reaching block amount, DMA follows the operating demand to send DMA last signal simultaneously.
- The DMA last signal that is used as a part of DMA demand transmits at BUS, when the device receives the operating demand of DMA last at BUS, the device can judge DMA transfer block length finished, that is before transmit the request again, DMA operation cannot appear, and a DMA active signal is generated to the DMA controller. Notice that each DRQ signal of device corresponds to an active signal, if the device has many DRQ signals, then DMA returns different active signal through different bus operation.
- When DMA receives the transmitted active signal of devices, DMA ACK signal is returned to devices.
- After the device receives DMA ACK signal, if all operations of devices are completed , FIFO status and DRQ status are refreshed, then active signal is set as invalid.
- When DMA detects the falling edge of active signal, then the corresponding ACK signal is set as invalid, and DMA restarts to detect the external request signal. If the request signal is valid, then the next transfer starts.

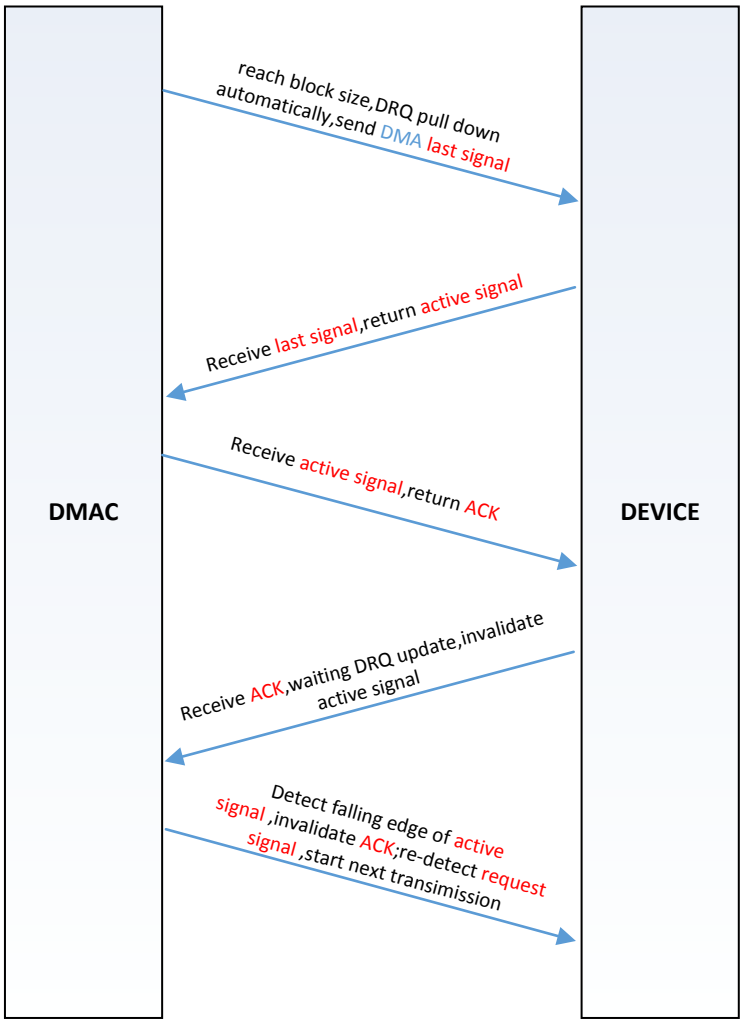

Figure 3- 21. DMA Transfer Mode

3.9.3.9. Auto-alignment Function

The DMA supports address alignment of non-IO devices, that is when the start address of non-IO devices is non 32-byte aligned, DMA firstly aligns the burst transfer within 32-byte to 32-byte. If the device of a DMA channel is configured to non-IO type, and the start address is 0x86, then DMA firstly aligns 26-byte burst transfer to 0xA0, then DMA transfers by 64-byte burst(maximum transfer amount of MBUS allowed). The address 32-byte alignment helps to improve the DRAM access efficiency.

IO devices do not support address alignment, so the bit width of IO devices must match the address offset, or not DMA ignores the non-consistency and indirectly transmits data of the corresponding bit width to the address.

The DMA descriptor address does not support auto-aligned function. The address must ensure word-aligned, or not DMA cannot identify descriptor.

3.9.3.10. Operating Mode

3.9.3.10.1. DMA Clock Control

- The DMA clock is synchronous with AHB1 clock. Make sure that open the DMA gating bit of AHB1 clock before access DMA register.
- The reset input signal of DMA is asynchronous with AHB1, and is low valid by default. Make sure that de-assert the reset signal of DMA before access DMA register.
- To avoid indefinite state within registers , firstly de-assert the reset signal, secondly open the gating bit of AHB1.
- DMA has the function of clock auto gating, DMA clock can be disabled in DMA idle state using software to reduce power consumption. DMA enables clock auto gating by default.

3.9.3.10.2. DMA Transfer Process

The DMA transfer process is as follows.

(1) Request DMA channel, and judge the idle state of the channel by whether DMA channel is enabled.
(2) Write the descriptor(6 words) into memory, the descriptor must be word-aligned. Refer to ***3.9.3.4 DMA descriptor*** in detail.
(3) Write the start address of storing descriptor to **DMA_DESC_ADDR_REG**.
(4) Enable DMA channel, and write the corresponding channel to **DMA_EN_REG**.
(5) DMA obtains the descriptor information.
(6) Start to transmit a package, when half package is completed, DMA sends **Half Package Transfer Interrupt**; when total package is completed, DMA sends **Package End Transfer Interrupt**. These interrupt status can be read by **DMA_IRQ_PEND_REG**.
(7) Set **DMA_PAU_REG** to pause or resume the data transfer.
(8) After completed the total package transfer, 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; if the link is other value, the next package starts to transmit. When the transfer ends, DMA sends **Queue End Transfer Interrupt**.
(9) Disable the DMA channel.

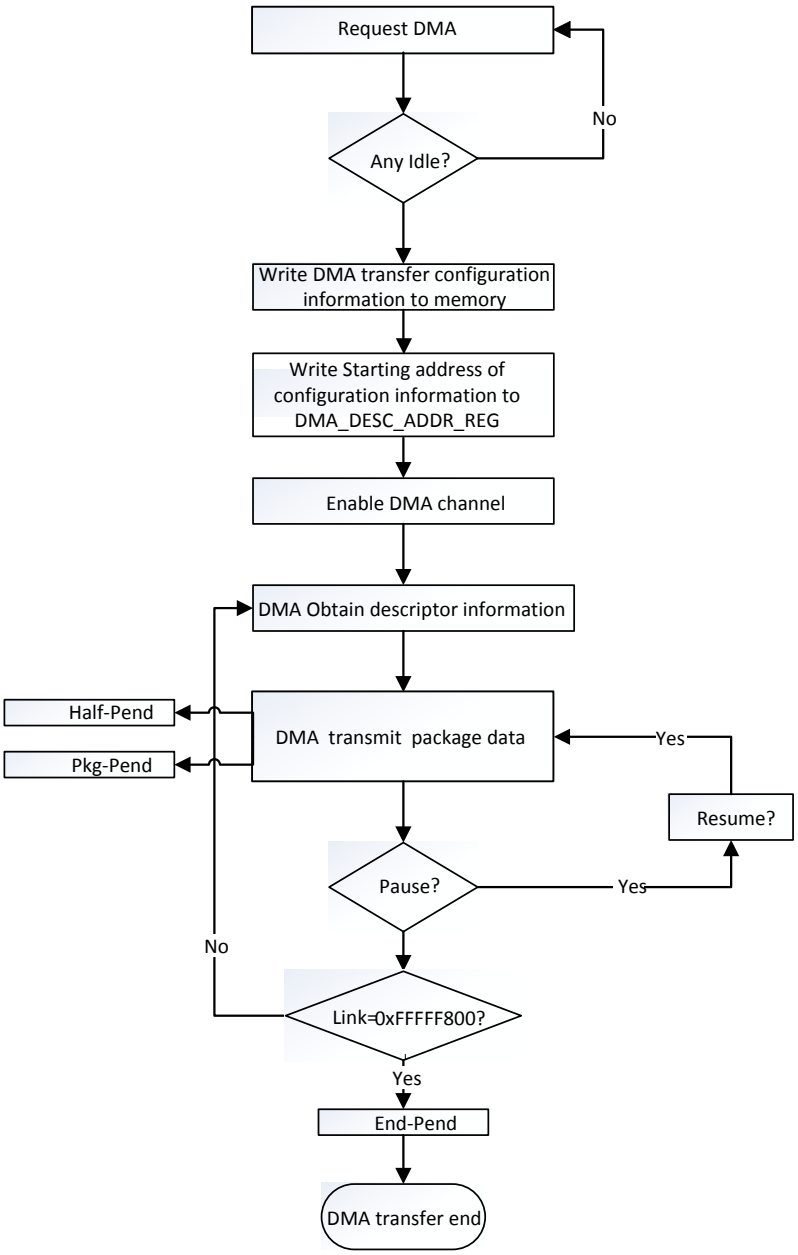

Figure 3- 22. DMA Transfer Process

3.9.3.10.3. DMA Interrupt

(1) Enable interrupt: write the corresponding interrupt enable of **DMA_IRQ_EN_REG**, when the corresponding interrupt condition is satisfied, the corresponding interrupt generates.

(2) After enter the interrupt process, write **DMA_IRQ_PEND_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.

3.9.4. Programming Guidelines

(1) The transfer width of IO type device is consistent with the offset of start address.

(2) MBUS protocol does not support read operation of non-integer word, so for non-integer word read operation, device must ignore redundant inconsistent data between data width and configuration, that is, the device of non-integer word must interpret DMA demand through its FIFO width instead of read demand width.

(3) When the DMA transfer is paused, this is equivalent to invalid DRQ. Because DMA transfer command has a certain time delay, DMA will not stop transfer immediately until the current command and the command in Arbiter finished, at most 32-byte data.

DMA application example :

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

DMA supports increasing data package in transfer, there are a few points to note here.

- When the value of **DMA Channel Descriptor Address Register** is 0xFFFFF800, it indicates that DMA channel has got back the descriptor of the last package. When DMA channel completed the package data transfer, DMA channel will stop automatically data transfer.
- If data packages are needed to increase, then at first it is essential to judge that whether DMA channel has got back the descriptor of the last package, if DMA channel has got back the descriptor of the last package, then this is impossible for increasing data package, DMA channel need start again. If DMA is not transmitting the last package, then the last descriptor address 0xFFFFF800 can be changed to the start address of the next descriptor.
- To ensure that the data changed valid, we can read again the value of **DMA Channel Descriptor Address Register** after changed the data. If there is not 0xFFFFF800, then it indicates that increasing data package is succeed, and fail otherwise. Because the process of increasing data package needs some time, during this time, DMA channel may get back the descriptor of the last package. At the moment we can read again **DMA Channel Current Source Address Register** and **DMA Channel Current Destination Address Register**, if the increasing memory address accords with the information of the increasing data package, then the increasing data package is succeed, and fail otherwise.
- To ensure the higher success rate, it is suggested that increase data package before half package interrupt of penultimate data package.

3.9.5. Register List

Module Name	Base Address
DMA	0x03002000

Register Name	Offset	Description
DMA_IRQ_EN_REG0	0x0000	DMA IRQ Enable Register 0
DMA_IRQ_EN_REG1	0x0004	DMA IRQ Enable Register 1
DMA_IRQ_PEND_REG0	0x0010	DMA IRQ Pending Register 0
DMA_IRQ_PEND_REG1	0x0014	DMA IRQ Pending Register 1
DMA_SEC_REG	0x0020	DMA Security Register
DMA_AUTO_GATE_REG	0x0028	DMA Auto Gating Register
DMA_STA_REG	0x0030	DMA Status Register
DMA_EN_REG	0x0100+N*0x0040	DMA Channel Enable Register (N=0~15)
DMA_PAU_REG	0x0100+N*0x0040+0x0004	DMA Channel Pause Register(N=0~15)
DMA_DESC_ADDR_REG	0x0100+N*0x0040+0x0008	DMA Channel Start Address Register(N=0~15)
DMA_CFG_REG	0x0100+N*0x0040+0x000C	DMA Channel Configuration Register(N=0~15)
DMA_CUR_SRC_REG	0x0100+N*0x0040+0x0010	DMA Channel Current Source Register(N=0~15)
DMA_CUR_DEST_REG	0x0100+N*0x0040+0x0014	DMA Channel Current Destination Register(N=0~15)
DMA_BCNT_LEFT_REG	0x0100+N*0x0040+0x0018	DMA Channel Byte Counter Left Register(N=0~15)
DMA_PARA_REG	0x0100+N*0x0040+0x001C	DMA Channel Parameter Register(N=0~15)
DMA_MODE_REG	0x0100+N*0x0040+0x0028	DMA Mode Register(N=0~15)
DMA_FDESC_ADDR_REG	0x0100+N*0x0040+0x002C	DMA Former Descriptor Address Register(N=0~15)
DMA_PKG_NUM_REG	0x0100+N*0x0040+0x0030	DMA Package Number Register(N=0~15)

3.9.6. Register Description

3.9.6.1. 0x0000 DMA IRQ Enable Register 0 (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: DMA_IRQ_EN_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R/W	0x0	DMA7_QUEUE_IRQ_EN DMA 7 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
29	R/W	0x0	DMA7_PKG_IRQ_EN DMA 7 Package End Transfer Interrupt Enable 0: Disable 1: Enable
28	R/W	0x0	DMA7_HLAF_IRQ_EN DMA 7 Half Package Transfer Interrupt Enable

			0: Disable 1: Enable
27	/	/	/
26	R/W	0x0	DMA6_QUEUE_IRQ_EN DMA 6 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
25	R/W	0x0	DMA6_PKG_IRQ_EN DMA 6 Package End Transfer Interrupt Enable 0: Disable 1: Enable
24	R/W	0x0	DMA6_HLAF_IRQ_EN DMA 6 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
23	/	/	/
22	R/W	0x0	DMA5_QUEUE_IRQ_EN DMA 5 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
21	R/W	0x0	DMA5_PKG_IRQ_EN DMA 5 Package End Transfer Interrupt Enable 0: Disable 1: Enable
20	R/W	0x0	DMA5_HLAF_IRQ_EN DMA 5 Half package Transfer Interrupt Enable 0: Disable 1: Enable
19	/	/	/
18	R/W	0x0	DMA4_QUEUE_IRQ_EN DMA 4 Queue End Transfer Interrupt Enable. 0: Disable 1: Enable
17	R/W	0x0	DMA4_PKG_IRQ_EN DMA 4 Package End Transfer Interrupt Enable 0: Disable 1: Enable
16	R/W	0x0	DMA4_HLAF_IRQ_EN DMA 4 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
15	/	/	/
14	R/W	0x0	DMA3_QUEUE_IRQ_EN DMA 3 Queue End Transfer Interrupt Enable 0: Disable

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

			DMA 0 Half Package Transfer Interrupt Enable 0: Disable 1: Enable

3.9.6.2. 0x0004 DMA IRQ Enable Register 1 (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: DMA_IRQ_EN_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R/W	0x0	DMA15_QUEUE_IRQ_EN DMA 15 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
29	R/W	0x0	DMA15_PKG_IRQ_EN DMA 15 Package End Transfer Interrupt Enable 0: Disable 1: Enable
28	R/W	0x0	DMA15_HLAF_IRQ_EN DMA 15 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
27	/	/	/
26	R/W	0x0	DMA14_QUEUE_IRQ_EN DMA 14 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
25	R/W	0x0	DMA14_PKG_IRQ_EN DMA 14 Package End Transfer Interrupt Enable 0: Disable 1: Enable
24	R/W	0x0	DMA14_HLAF_IRQ_EN DMA 14 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
23	/	/	/
22	R/W	0x0	DMA13_QUEUE_IRQ_EN DMA 13 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
21	R/W	0x0	DMA13_PKG_IRQ_EN DMA 13 Package End Transfer Interrupt Enable 0: Disable 1: Enable

20	R/W	0x0	DMA13_HLAF_IRQ_EN DMA 13 Half package Transfer Interrupt Enable 0: Disable 1: Enable
19	/	/	/
18	R/W	0x0	DMA12_QUEUE_IRQ_EN DMA 12 Queue End Transfer Interrupt Enable. 0: Disable 1: Enable
17	R/W	0x0	DMA12_PKG_IRQ_EN DMA 12 Package End Transfer Interrupt Enable 0: Disable 1: Enable
16	R/W	0x0	DMA12_HLAF_IRQ_EN DMA 12 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
15	/	/	/
14	R/W	0x0	DMA11_QUEUE_IRQ_EN DMA 11 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
13	R/W	0x0	DMA11_PKG_IRQ_EN DMA 11 Package End Transfer Interrupt Enable 0: Disable 1: Enable
12	R/W	0x0	DMA11_HLAF_IRQ_EN DMA 11 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
11	/	/	/
10	R/W	0x0	DMA10_QUEUE_IRQ_EN DMA 10 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
9	R/W	0x0	DMA10_PKG_IRQ_EN DMA 10 Package End Transfer Interrupt Enable 0: Disable 1: Enable
8	R/W	0x0	DMA10_HLAF_IRQ_EN DMA 10 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
7	/	/	/
6	R/W	0x0	DMA9_QUEUE_IRQ_EN

			DMA 9 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
5	R/W	0x0	DMA9_PKG_IRQ_EN DMA 9 Package End Transfer Interrupt Enable. 0: Disable 1: Enable
4	R/W	0x0	DMA9_HLAF_IRQ_EN DMA 9 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
3	/	/	/
2	R/W	0x0	DMA8_QUEUE_IRQ_EN DMA 8 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
1	R/W	0x0	DMA8_PKG_IRQ_EN DMA 8 Package End Transfer Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	DMA8_HLAF_IRQ_EN DMA 8 Half Package Transfer Interrupt Enable 0: Disable 1: Enable

3.9.6.3. 0x0010 DMA IRQ Pending Status Register 0 (Default Value: 0x0000_0000)

Offset:0x0010			**Register Name: DMA_IRQ_PEND_REG0**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R/W1C	0x0	DMA7_QUEUE_IRQ_PEND DMA 7 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
29	R/W1C	0x0	DMA7_PKG_IRQ_PEND DMA 7 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
28	R/W1C	0x0	DMA7_HLAF_IRQ_PEND DMA 7 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect

			1: Pending
27	/	/	/
26	R/W1C	0x0	DMA6_QUEUE_IRQ_PEND DMA 6 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
25	R/W1C	0x0	DMA6_PKG_IRQ_ PEND DMA 6 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
24	R/W1C	0x0	DMA6_HLAF_IRQ_PEND DMA 6 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
23	/	/	/
22	R/W1C	0x0	DMA5_QUEUE_IRQ_PEND DMA 5 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
21	R/W1C	0x0	DMA5_PKG_IRQ_ PEND DMA 5 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
20	R/W1C	0x0	DMA5_HLAF_IRQ_PEND DMA 5 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
19	/	/	/
18	R/W1C	0x0	DMA4_QUEUE_IRQ_PEND DMA 4 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
17	R/W1C	0x0	DMA4_PKG_IRQ_ PEND DMA 4 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
16	R/W1C	0x0	DMA4_HLAF_IRQ_PEND

			DMA 4 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
15	/	/	/
14	R/W1C	0x0	DMA3_QUEUE_IRQ_PEND DMA 3 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending.
13	R/W1C	0x0	DMA3_PKG_IRQ_PEND DMA 3 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
12	R/W1C	0x0	DMA3_HLAF_IRQ_PEND DMA 3 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
11	/	/	/
10	R/W1C	0x0	DMA2_QUEUE_IRQ_PEND DMA 2 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
9	R/W1C	0x0	DMA2_PKG_IRQ_PEND DMA 2 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
8	R/W1C	0x0	DMA2_HLAF_IRQ_PEND DMA 2 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
7	/	/	/
6	R/W1C	0x0	DMA1_QUEUE_IRQ_PEND DMA 1 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
5	R/W1C	0x0	DMA1_PKG_IRQ_PEND DMA 1 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it.

			0: No effect 1: Pending
4	R/W1C	0x0	DMA1_HLAF_IRQ_PEND DMA 1 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
3	/	/	/
2	R/W1C	0x0	DMA0_QUEUE_IRQ_PEND DMA 0 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
1	R/W1C	0x0	DMA0_PKG_IRQ_ PEND DMA 0 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
0	R/W1C	0x0	DMA0_HLAF_IRQ_PEND DMA 0 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending

3.9.6.4. 0x0014 DMA IRQ Pending Status Register 1 (Default Value: 0x0000_0000)

Offset:0x0014			**Register Name: DMA_IRQ_PEND_REG1**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R/W1C	0x0	DMA15_QUEUE_IRQ_PEND DMA 15 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
29	R/W1C	0x0	DMA15_PKG_IRQ_PEND DMA 15 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
28	R/W1C	0x0	DMA15_HLAF_IRQ_PEND DMA 15 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending

27	/	/	/
26	R/W1C	0x0	DMA14_QUEUE_IRQ_PEND DMA 14 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
25	R/W1C	0x0	DMA14_PKG_IRQ_ PEND DMA 14 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
24	R/W1C	0x0	DMA14_HLAF_IRQ_PEND DMA 14 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
23	/	/	/
22	R/W1C	0x0	DMA13_QUEUE_IRQ_PEND DMA 13 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
21	R/W1C	0x0	DMA13_PKG_IRQ_ PEND DMA 13 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
20	R/W1C	0x0	DMA13_HLAF_IRQ_PEND DMA 13 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
19	/	/	/
18	R/W1C	0x0	DMA12_QUEUE_IRQ_PEND DMA 12 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
17	R/W1C	0x0	DMA12_PKG_IRQ_ PEND DMA 12 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
16	R/W1C	0x0	DMA12_HLAF_IRQ_PEND DMA 12 Half Package Transfer Interrupt Pending. Setting 1 to the bit

			will clear it. 0: No effect 1: Pending
15	/	/	/
14	R/W1C	0x0	DMA11_QUEUE_IRQ_PEND DMA 11 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending.
13	R/W1C	0x0	DMA11_PKG_IRQ_PEND DMA 11 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
12	R/W1C	0x0	DMA11_HLAF_IRQ_PEND DMA 11 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
11	/	/	/
10	R/W1C	0x0	DMA10_QUEUE_IRQ_PEND DMA 10 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
9	R/W1C	0x0	DMA10_PKG_IRQ_PEND DMA 10 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
8	R/W1C	0x0	DMA10_HLAF_IRQ_PEND DMA 10 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
7	/	/	/
6	R/W1C	0x0	DMA9_QUEUE_IRQ_PEND DMA 9 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
5	R/W1C	0x0	DMA9_PKG_IRQ_PEND DMA 9 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect

			1: Pending
4	R/W1C	0x0	DMA9_HLAF_IRQ_PEND DMA 9 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
3	/	/	/
2	R/W1C	0x0	DMA8_QUEUE_IRQ_PEND DMA 8 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
1	R/W1C	0x0	DMA8_PKG_IRQ_ PEND DMA 8 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
0	R/W1C	0x0	DMA8_HLAF_IRQ_PEND DMA 8 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending

3.9.6.5. 0x0020 DMA Security Register (Default Value: 0x0000_0000)

Offset:0x0020			**Register Name: DMA_SEC_REG**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15	R/W	0x0	DMA15_SEC DMA channel 15 security 0: Secure 1: Non-secure
14	R/W	0x0	DMA14_SEC DMA channel 14 security 0: Secure 1: Non-secure
13	R/W	0x0	DMA13_SEC DMA channel 13 security 0: Secure 1: Non-secure
12	R/W	0x0	DMA12_SEC DMA channel 12 security 0: Secure 1: Non-secure

11	R/W	0x0	DMA11_SEC DMA channel 11 security 0: Secure 1: Non-secure
10	R/W	0x0	DMA10_SEC DMA channel 10 security 0: Secure 1: Non-secure
9	R/W	0x0	DMA9_SEC DMA channel 9 security 0: Secure 1: Non-secure
8	R/W	0x0	DMA8_SEC DMA channel 8 security 0: Secure 1: Non-secure
7	R/W	0x0	DMA7_SEC DMA channel 7 security 0: Secure 1: Non-secure
6	R/W	0x0	DMA6_SEC DMA channel 6 security 0: Secure 1: Non-secure
5	R/W	0x0	DMA5_SEC DMA channel 5 security 0: Secure 1: Non-secure
4	R/W	0x0	DMA4_SEC DMA channel 4 security 0: Secure 1: Non-secure
3	R/W	0x0	DMA3_SEC DMA channel 3 security 0: Secure 1: Non-secure
2	R/W	0x0	DMA2_SEC DMA channel 2 security 0: Secure 1: Non-secure
1	R/W	0x0	DMA1_SEC DMA channel 1 security 0: Secure 1: Non-secure
0	R/W	0x0	DMA0_SEC

			DMA channel 0 security 0: Secure 1: Non-secure

3.9.6.6. 0x0028 DMA Auto Gating Register (Default Value: 0x0000_0000)

Offset:0x0028			Register Name: DMA_AUTO_GATE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x0	DMA_MCLK_CIRCUIT DMA MCLK interface circuit auto gating bit 0: Auto gating enable 1: Auto gating disable
1	R/W	0x0	DMA_COMMON_CIRCUIT DMA common circuit auto gating bit 0: Auto gating enable 1: Auto gating disable
0	R/W	0x0	DMA_CHAN_CIRCUIT DMA channel circuit auto gating bit 0: Auto gating enable 1: Auto gating disable

NOTE

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

3.9.6.7. 0x0030 DMA Status Register (Default Value: 0x0000_0000)

Offset:0x0030			Register Name: DMA_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R	0x0	MBUS FIFO Status 0: Empty 1: Not Empty
29:16	/	/	/
15	R	0x0	DMA15_STATUS DMA Channel 15 Status 0: Idle 1: Busy
14	R	0x0	DMA14_STATUS DMA Channel 14 Status 0: Idle 1: Busy

13	R	0x0	DMA13_STATUS DMA Channel 13 Status 0: Idle 1: Busy
12	R	0x0	DMA12_STATUS DMA Channel 12 Status 0: Idle 1: Busy
11	R	0x0	DMA11_STATUS DMA Channel 11 Status 0: Idle 1: Busy
10	R	0x0	DMA10_STATUS DMA Channel 10 Status 0: Idle 1: Busy
9	R	0x0	DMA9_STATUS DMA Channel 9 Status 0: Idle 1: Busy
8	R	0x0	DMA8_STATUS DMA Channel 8 Status 0: Idle 1: Busy
7	R	0x0	DMA7_STATUS DMA Channel 7 Status 0: Idle 1: Busy
6	R	0x0	DMA6_STATUS DMA Channel 6 Status 0: Idle 1: Busy
5	R	0x0	DMA5_STATUS DMA Channel 5 Status 0: Idle 1: Busy
4	R	0x0	DMA4_STATUS DMA Channel 4 Status 0: Idle 1: Busy
3	R	0x0	DMA3_STATUS DMA Channel 3 Status 0: Idle 1: Busy
2	R	0x0	DMA2_STATUS

			DMA Channel 2 Status 0: Idle 1: Busy
1	R	0x0	DMA1_STATUS DMA Channel 1 Status 0: Idle 1: Busy
0	R	0x0	DMA0_STATUS DMA Channel 0 Status 0: Idle 1: Busy

3.9.6.8. 0x0100+N*0x0040 DMA Channel Enable Register (Default Value: 0x0000_0000)

Offset:0x0100+N*0x0040 (N=0~15)			Register Name: DMA_EN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	DMA_EN DMA Channel Enable 0: Disable 1: Enable

3.9.6.9. 0x0104+N*0x0040 DMA Channel Pause Register (Default Value: 0x0000_0000)

Offset:0x0100+N*0x0040+0x0004(N=0~15)			Register Name: DMA_PAU_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	DMA_PAUSE Pausing DMA Channel Transfer Data 0: Resume Transferring 1: Pause Transferring

3.9.6.10. 0x0108+N*0x0040 DMA Channel Descriptor Address Register (Default Value: 0x0000_0000)

Offset:0x0100+N*0x0040+0x0008(N=0~15)			Register Name: DMA_DESC_ADDR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	R/W	0x0	DMA_DESC_LOW_ ADDR DMA Channel Descriptor Word Address, Low 30bits. The Descriptor Address must be word-aligned.
1:0	R/W	0x0	DMA_DESC_HIGH_ ADDR DMA Channel Descriptor High Address, High 2bits

			The real address is as below: DMA Channel Descriptor Address = {bit[1:0],bit[31:2],2'b00};

3.9.6.11. 0x010C+N*0x0040 DMA Channel Configuration Register (Default Value: 0x0000_0000)

Offset:0x0100+N*0x0040+0x000C(N=0~15)			Register Name: DMA_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:27	/	/	/
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
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.

3.9.6.12. 0x0110+N*0x0040 DMA Channel Current Source Address Register (Default Value: 0x0000_0000)

Offset:0x0100+N*0x0040+0x0010(N=0~15)			Register Name: DMA_CUR_SRC_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DMA_CUR_SRC DMA Channel Current Source Address, read only.

3.9.6.13. 0x0114+N*0x0040 DMA Channel Current Destination Address Register (Default Value: 0x0000_0000)

Offset:0x0100+N*0x0040+0x0014(N=0~15)			Register Name: DMA_CUR_DEST_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DMA_CUR_DEST DMA Channel Current Destination Address, read only.

3.9.6.14. 0x0118+N*0x0040 DMA Channel Byte Counter Left Register (Default Value: 0x0000_0000)

Offset:0x0100+N*0x0040+0x0018(N=0~15)			Register Name: DMA_BCNT_LEFT_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24:0	R	0x0	DMA_BCNT_LEFT DMA Channel Byte Counter Left, read only.

3.9.6.15. 0x011C+N*0x0040 DMA Channel Parameter Register (Default Value: 0x0000_0000)

Offset:0x0100+N*0x0040+0x001C(N=0~15)			Register Name: DMA_PARA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R	0x0	WAIT_CYC Wait Clock Cycles

3.9.6.16. 0x0128+N*0x0040 DMA Mode Register (Default Value: 0x0000_0000)

Offset:0x0100+N*0x0040+0x0028(N=0~15)			Register Name: DMA_MODE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**

31:4	/	/	/
3	R/W	0x0	DMA_DST_MODE 0: Wait mode 1: Handshake mode
2	R/W	0x0	DMA_SRC_MODE 0: Wait mode 1: Handshake mode
1:0	/	/	/

3.9.6.17. 0x012C+N*0x0040 DMA Former Descriptor Address Register (Default Value: 0x0000_0000)

Offset:0x0100+N*0x0040+0x002C(N=0~15)			Register Name: DMA_FDESC_ADDR_REG
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.

3.9.6.18. 0x0130+N*0x0040 DMA Package Number Register (Default Value: 0x0000_0000)

Offset:0x0100+N*0x0040+0x0030(N=0~15)			Register Name: DMA_PKG_NUM_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DMA_PKG_NUM This register will record the number of packages which has been completed in one transmission.

3.10. Thermal Sensor Controller

3.10.1. Overview

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

The Thermal Sensor Controller(THS) embeds four thermal sensors, sensor0 is located in GPU, sensor1 is located in VE, sensor2 is located in CPU, sensor3 is located in DDR. The thermal sensor can generate interrupt to SW to lower temperature via DVFS, on reaching a certain thermal threshold.

The THS has the following features:

- Temperature Accuracy : ±3°C from 0°C to +100°C, ±5°C from -25°C to +125°C
- Power supply voltage: 1.8V
- Averaging filter for thermal sensor reading
- Supports over-temperature protection interrupt and over-temperature alarm interrupt

3.10.2. Block Diagram

Figure 3-23 shows a block diagram of the Thermal Sensor Controller.

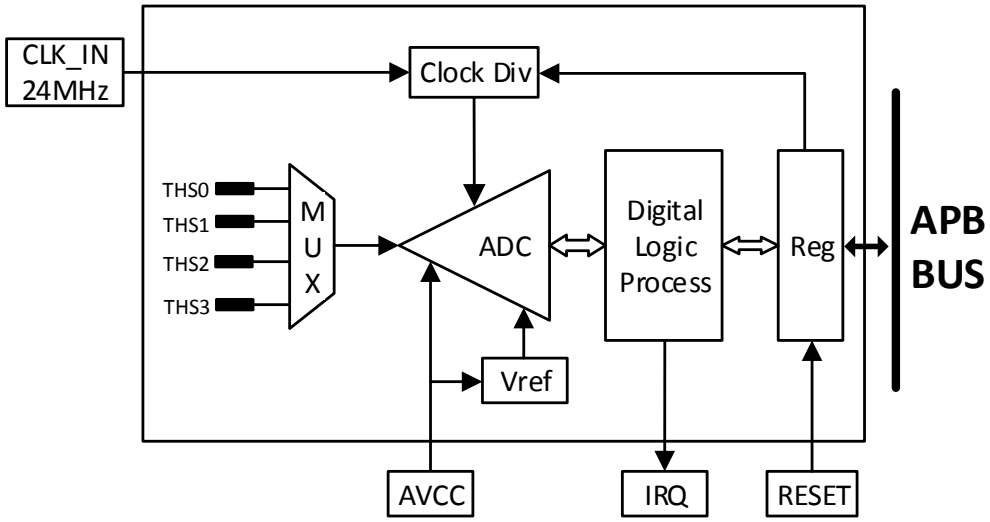

Figure 3- 23. Thermal Sensor Controller Block Diagram

3.10.3. Operations and Functional Descriptions

3.10.3.1. Clock Sources

The THS gets one clock source. Table 3-9 describes the clock source for Thermal Sensor Controller. Users can see **Clock Controller Unit(CCU)** for clock setting, configuration and gating information.

Table 3- 9. Thermal Sensor Controller Clock Sources

Clock Sources	Description
OSC24M	24M OSC

3.10.3.2. Timing Requirements

CLK_IN = 24 MHz
CONV_TIME(Conversion Time) = 1/(24 MHz/14Cycles) =0.583 (us)
TACQ> 1/(24 MHz/24 Cycles)
THERMAL_PER > ADC Sample Frequency > TACQ+CONV_TIME

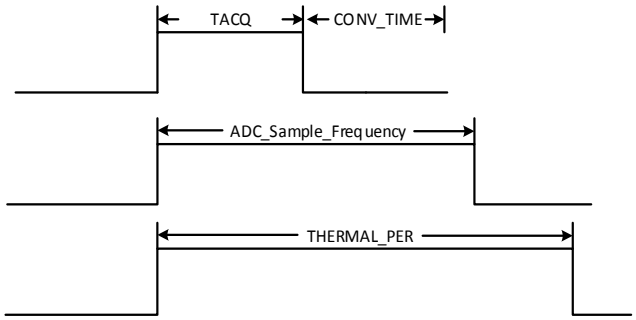

Figure 3- 24. Thermal Sensor Time Requirement

3.10.3.3. Interrupt

The THS has four interrupt sources, such as DATA_IRQ, SHUTDOWN_IRQ, ALARM_IRQ and ALARM_OFF_IRQ. Figure 3-25 shows the thermal sensor interrupt sources.

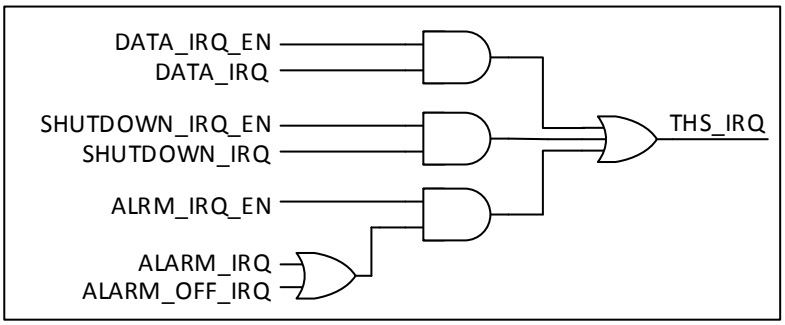

Figure 3- 25. Thermal Sensor Controller Interrupt Source

When temperature is higher than Alarm_Threshold, ALARM_IRQ is generated. When temperature is lower than

Alarm_Off_Thershold, ALARM_OFF_IRQ is generated. ALARM_OFF_IRQ is fall edge trigger.

3.10.3.4. THS Temperature Conversion Formula

T= (sensor_data - 3255)/(-12.401), the unit of T is Celsius.
sensor_data: read from sensor data register.

3.10.4. Programming Guidelines

The initial process of the THS is as follows.

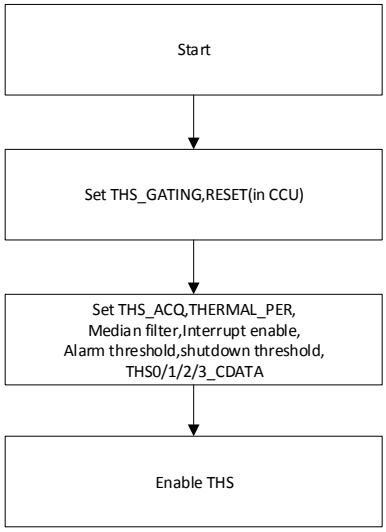

Figure 3- 26. THS Initial Process

The formula of THS is y=-ax+b. In FT stage, THS is calibrated through ambient temperature, the calibration value is written in eFUSE. Please refer to SID Spec about eFUSE information.

Before enabling THS, read eFUSE value and write the value to **THS_CDATA**.

(1).Query Mode

Step1: Write 0x1 to the bit16 of **THS_BGR_REG** to dessert reset.
Step2: Write 0x1 to the bit0 of **THS_BGR_REG** to open THS clock.
Step3: Write 0x2F to the bit[15:0] of **THS_CTRL** to set ADC acquire time.
Step4: Write 0x1DF to the bit[31:16] of **THS_CTRL** to set ADC sample frequency divider.
Step5: Write 0x3A to the bit[31:12] of **THS_PER** to set THS work period.
Step6: Write 0x1 to the bit2 of **THS_FILTER** to enable temperature convert filter.
Step7: Write 0x1 to the bit[1:0] of **THS_FILTER** to select filter type.
Step8: Read THS efuse value from SID, then write the efuse value to **THS_CDATA** to calibrate THS.
Step9: Write 0x1 to the bit[0] 0f **THS_EN** to enable THS.
Step10: Read the bit[0] of **THS_DATA_INTS**, if is 1, temperature conversion is complete.

Step11: Read the bit[11:0] of **THS_DATA**, calculate THS temperature based on THS Temperature Conversion Formula in Section 3.10.3.4.

(2). Interrupt Mode

Step1: Write 0x1 to the bit16 of **THS_BGR_REG** to dessert reset.

Step2: Write 0x1 to the bit0 of **THS_BGR_REG** to open THS clock.

Step3: Write 0x2F to the bit[15:0] of **THS_CTRL** to set ADC acquire time.

Step4: Write 0x1DF to the bit[31:16] of **THS_CTRL** to set ADC sample frequency divider.

Step5: Write 0x3A to the bit[31:12] of **THS_PER** to set THS work period.

Step6: Write 0x1 to the bit2 of **THS_FILTER** to enable temperature convert filter.

Step7: Write 0x1 to the bit[1:0] of **THS_FILTER** to select filter type.

Step8: Read THS efuse value from SID, then write the efuse value to **THS_CDATA** to calibrate THS.

Step9: Write 0x1 to the bit[0] of **THS_DATA_INTC** to enable the interrupt of THS.

Step10: Set GIC interface based on IRQ 51, write the bit[19] of the **0x03021104** register to 0x1.

Step11: Put interrupt handler address into interrupt vector table.

Step12: Write 0x1 to the bit[0] 0f **THS_EN** to enable THS.

Step13: Read the bit[0] of **THS_DATA_INTS**, if is 1, temperature conversion is complete.

Step14: Read the bit[11:0] of **THS_DATA**, calculate THS temperature based on THS Temperature Conversion Formula in Section 3.10.3.4.

3.10.5. Register List

Module Name	Base Address
Thermal Sensor	0x05070400

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
THS0_ALARM_CTRL	0x0040	THS0 Alarm Threshold Control Register
THS1_ALARM_CTRL	0x0044	THS1 Alarm Threshold Control Register
THS2_ALARM_CTRL	0x0048	THS2 Alarm Threshold Control Register
THS3_ALARM_CTRL	0x004C	THS3 Alarm Threshold Control Register
THS01_SHUTDOWN_CTRL	0x0080	THS0 & THS1 Shutdown Threshold Control Register

THS23_SHUTDOWN_CTRL	0x0084	THS2 & THS3 Shutdown Threshold Control Register
THS01_CDATA	0x00A0	THS0 & THS1 Calibration Data
THS23_CDATA	0x00A4	THS2 & THS3 Calibration Data
THS0_DATA	0x00C0	THS0 Data Register
THS1_DATA	0x00C4	THS1 Data Register
THS2_DATA	0x00C8	THS2 Data Register
THS3_DATA	0x00CC	THS3 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	**Rear/Write**	**Default/Hex**	**Description**
31:16	R/W	0x1DF	FS_DIV ADC Sample Frequency Divider CLK_IN/(N+1) , N > 0x17 The default value indicates 50 kHz.
15:0	R/W	0x2F	TACQ ADC Acquire Time CLK_IN/(n+1) The default value indicates 2us.

3.10.6.2. 0x0004 THS Enable Register(Default Value : 0x0000_0000)

Offset: 0x0004			Register Name: THS_EN
Bit	**Rear/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3	R/W	0x0	THS3_EN Enable temperature measurement sensor3 0:Disable 1:Enable
2	R/W	0x0	THS2_EN Enable temperature measurement sensor2 0:Disable 1:Enable
1	R/W	0x0	THS1_EN Enable temperature measurement sensor1 0:Disable 1:Enable
0	R/W	0x0	THS0_EN Enable temperature measurement sensor0

			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 4096*(n+1)/CLK_IN The default value indicates 10ms.
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**
31:4	/	/	/
3	R/W	0x0	THS3_DATA_IRQ_EN Selects temperature measurement data of sensor3 0:Disable 1:Enable
2	R/W	0x0	THS2_DATA_IRQ_EN Selects temperature measurement data of sensor2 0:Disable 1:Enable
1	R/W	0x0	THS1_DATA_IRQ_EN Selects temperature measurement data of sensor1 0:Disable 1:Enable
0	R/W	0x0	THS0_DATA_IRQ_EN Selects temperature measurement data of sensor0 0:Disable 1:Enable

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**
31:4	/	/	/
3	R/W	0x0	SHUT_INT3_EN Selects shutdown interrupt for sensor3

			0:Disable 1:Enable
2	R/W	0x0	SHUT_INT2_EN Selects shutdown interrupt for sensor2 0:Disable 1:Enable
1	R/W	0x0	SHUT_INT1_EN Selects shutdown interrupt for sensor1 0:Disable 1:Enable
0	R/W	0x0	SHUT_INT0_EN Selects shutdown interrupt for sensor0 0:Disable 1:Enable

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**
31:4	/	/	/
3	R/W	0x0	ALARM_INT3_EN Selects alarm interrupt for sensor3 0:Disable 1:Enable
2	R/W	0x0	ALARM_INT2_EN Selects alarm interrupt for sensor2 0:Disable 1:Enable
1	R/W	0x0	ALARM_INT1_EN Selects alarm interrupt for sensor1 0:Disable 1:Enable
0	R/W	0x0	ALARM_INT0_EN Selects alarm interrupt for sensor0 0:Disable 1:Enable

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**
31:4	/	/	/

3	R/W1C	0x0	THS3_DATA_IRQ _STS Data interrupt status for sensor3 Write ‘1’ to clear this interrupt.
2	R/W1C	0x0	THS2_DATA_IRQ _STS Data interrupt status for sensor2 Write ‘1’ to clear this interrupt.
1	R/W1C	0x0	THS1_DATA_IRQ _STS Data interrupt status for sensor1 Write ‘1’ to clear this interrupt.
0	R/W1C	0x0	THS0_DATA_IRQ _STS Data interrupt status for sensor0 Write ‘1’ to clear this interrupt.

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**
31:4	/	/	/
3	R/W1C	0x0	SHUT_INT3_STS Shutdown interrupt status for sensor3 Write ‘1’ to clear this interrupt.
2	R/W1C	0x0	SHUT_INT2_STS Shutdown interrupt status for sensor2 Write ‘1’ to clear this interrupt.
1	R/W1C	0x0	SHUT_INT1_STS Shutdown interrupt status for sensor1 Write ‘1’ to clear this interrupt.
0	R/W1C	0x0	SHUT_INT0_STS Shutdown interrupt status for sensor0 Write ‘1’ to clear this interrupt.

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**
31:4	/	/	/
3	R/W1C	0x0	ALARM_ OFF3_STS Alarm interrupt off pending for sensor3 Write ‘1’ to clear this interrupt.
2	R/W1C	0x0	ALARM_ OFF2_STS Alarm interrupt off pending for sensor2 Write ‘1’ to clear this interrupt.

1	R/W1C	0x0	ALARM_ OFF1_STS Alarm interrupt off pending for sensor1 Write ‘1’ to clear this interrupt.
0	R/W1C	0x0	ALARM_ OFF0_STS Alarm interrupt off pending for sensor0 Write ‘1’ to clear this interrupt.

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**
31:4	/	/	/
3	R/W1C	0x0	ALARM_INT3_STS Alarm interrupt pending for sensor3 Write ‘1’ to clear this interrupt.
2	R/W1C	0x0	ALARM_INT2_STS Alarm interrupt pending for sensor2 Write ‘1’ to clear this interrupt.
1	R/W1C	0x0	ALARM_INT1_STS Alarm interrupt pending for sensor1 Write ‘1’ to clear this interrupt.
0	R/W1C	0x0	ALARM_INT0_STS Alarm interrupt pending for sensor0 Write ‘1’ to clear this interrupt.

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: Disable 1: Enable
1:0	R/W	0x1	FILTER_TYPE Average Filter Type 00: 2 01: 4 10: 8 11: 16

3.10.6.12. 0x0040 THS0 Alarm Threshold Control Register(Default Value: 0x05A0_0684)

Offset: 0x0040			Register Name: THS0_ALARM_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0x5A0	ALARM0_T_HOT Thermal Sensor0 alarm threshold for hot temperature
15:12	/	/	/
11:0	R/W	0x684	ALARM0_T_HYST Thermal Sensor0 alarm threshold for hysteresis temperature

3.10.6.13. 0x0044 THS1 Alarm Threshold Control Register(Default Value: 0x05A0_0684)

Offset: 0x0044			Register Name: THS1_ALARM_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0x5A0	ALARM1_T_HOT Thermal Sensor1 alarm threshold for hot temperature
15:12	/	/	/
11:0	R/W	0x684	ALARM1_T_HYST Thermal Sensor1 alarm threshold for hysteresis temperature

3.10.6.14. 0x0048 THS2 Alarm Threshold Control Register(Default Value: 0x05A0_0684)

Offset: 0x0048			Register Name: THS2_ALARM_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0x5A0	ALARM2_T_HOT Thermal Sensor2 alarm threshold for hot temperature
15:12	/	/	/
11:0	R/W	0x684	ALARM2_T_HYST Thermal Sensor2 alarm threshold for hysteresis temperature

3.10.6.15. 0x004C THS3 Alarm Threshold Control Register(Default Value: 0x05A0_0684)

Offset: 0x004C			Register Name: THS3_ALARM_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0x5A0	ALARM3_T_HOT Thermal Sensor3 alarm threshold for hot temperature
15:12	/	/	/

11:0	R/W	0x684	ALARM3_T_HYST Thermal Sensor3 alarm threshold for hysteresis temperature

3.10.6.16. 0x0080 THS0&1 Shutdown Threshold Control Register (Default Value: 0x04E9_04E9)

Offset: 0x0080			Register Name: THS01_SHUTDOWN_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0x4E9	SHUT1_T_HOT Thermal Sensor1 shutdown threshold for hot temperature
15:12	/	/	/
11:0	R/W	0x4E9	SHUT0_T_HOT Thermal Sensor0 shutdown threshold for hot temperature

3.10.6.17. 0x0084 THS2&3 Shutdown Threshold Control Register (Default Value: 0x04E9_04E9)

Offset: 0x0084			Register Name: THS23_SHUTDOWN_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0x4E9	SHUT3_T_HOT Thermal Sensor3 shutdown threshold for hot temperature
15:12	/	/	/
11:0	R/W	0x4E9	SHUT2_T_HOT Thermal Sensor2 shutdown threshold for hot temperature

3.10.6.18. 0x00A0 THS0&1 Calibration Data Register (Default Value: 0x0800_0800)

Offset: 0x00A0			Register Name: THS01_CDATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0x800	THS1_CDATA Thermal Sensor1 calibration data
15:12	/	/	/
11:0	R/W	0x800	THS0_CDATA Thermal Sensor0 calibration data

3.10.6.19. 0x00A4 THS2&3 Calibration Data Register (Default Value: 0x0800_0800)

Offset: 0x00A4			Register Name: THS23_CDATA
Bit	**Read/Write**	**Default/Hex**	**Description**

31:28	/	/	/
27:16	R/W	0x800	THS3_CDATA Thermal Sensor3 calibration data
15:12	/	/	/
11:0	R/W	0x800	THS2_CDATA Thermal Sensor2 calibration data

3.10.6.20. 0x00C0 THS0 Data Register(Default Value: 0x0000_0000)

Offset: 0x00C0			Register Name: THS0_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11:0	R	0x0	THS0_DATA Temperature measurement data of sensor0

3.10.6.21. 0x00C4 THS1 Data Register(Default Value: 0x0000_0000)

Offset: 0x00C4			Register Name: THS1_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11:0	R	0x0	THS1_DATA Temperature measurement data of sensor1

3.10.6.22. 0x00C8 THS2 Data Register(Default Value: 0x0000_0000)

Offset: 0x00C8			Register Name: THS2_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11:0	R	0x0	THS2_DATA Temperature measurement data of sensor2

3.10.6.23. 0x00CC THS3 Data Register(Default Value: 0x0000_0000)

Offset: 0x00CC			Register Name: THS3_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11:0	R	0x0	THS3_DATA Temperature measurement data of sensor3

3.11. PSI

3.11.1. Overview

PSI(Peripheral System Interconnect) is a peripheral bus interconnect device based on AHB and APB protocol, which supports 16 AHB master and 16 slave bus. The type of slave bus can be AHB bus or APB bus. Each bus supports 64 slave devices.

3.11.2. Block Diagram

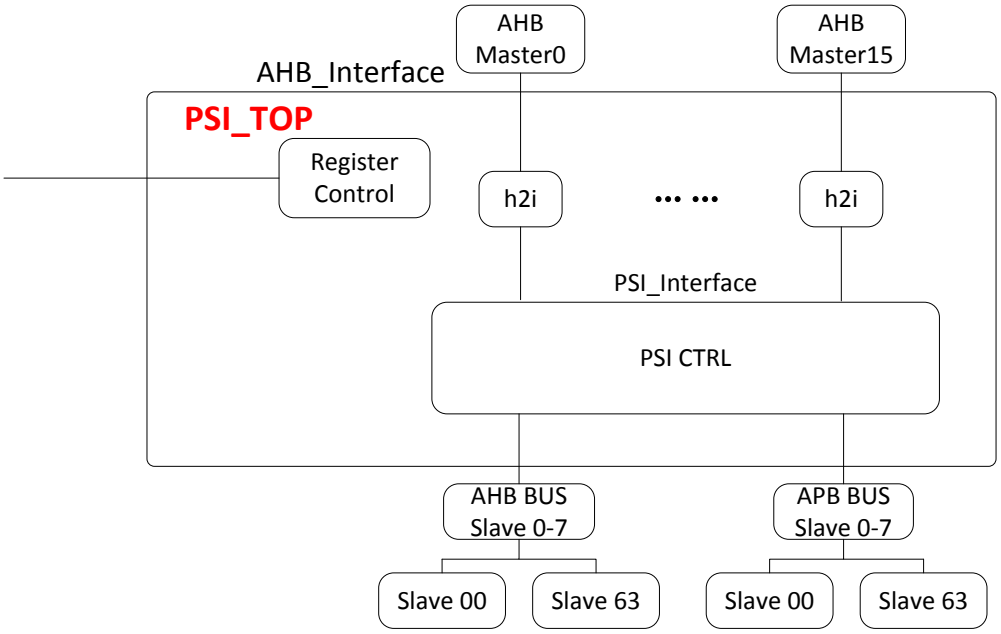

Figure 3- 27. PSI Block Diagram

3.12. IOMMU

3.12.1. Overview

IOMMU(I/O Memory management unit) is designed for product specific memory requirements. It maps the virtual address(sent by peripheral access memory) to the physical address. 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 DE, DI, VE_R, VE, G2D parallel address mapping
- Supports DE, DI, VE_R, VE, G2D bypass function independently
- Supports DE, DI, VE_R, VE, G2D prefetch independently
- Supports DE, DI, VE_R, VE, G2D 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

3.12.2. Block Diagram

IOMMU internal module 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.

Figure 3-27 shows the internal block diagram of IOMMU.

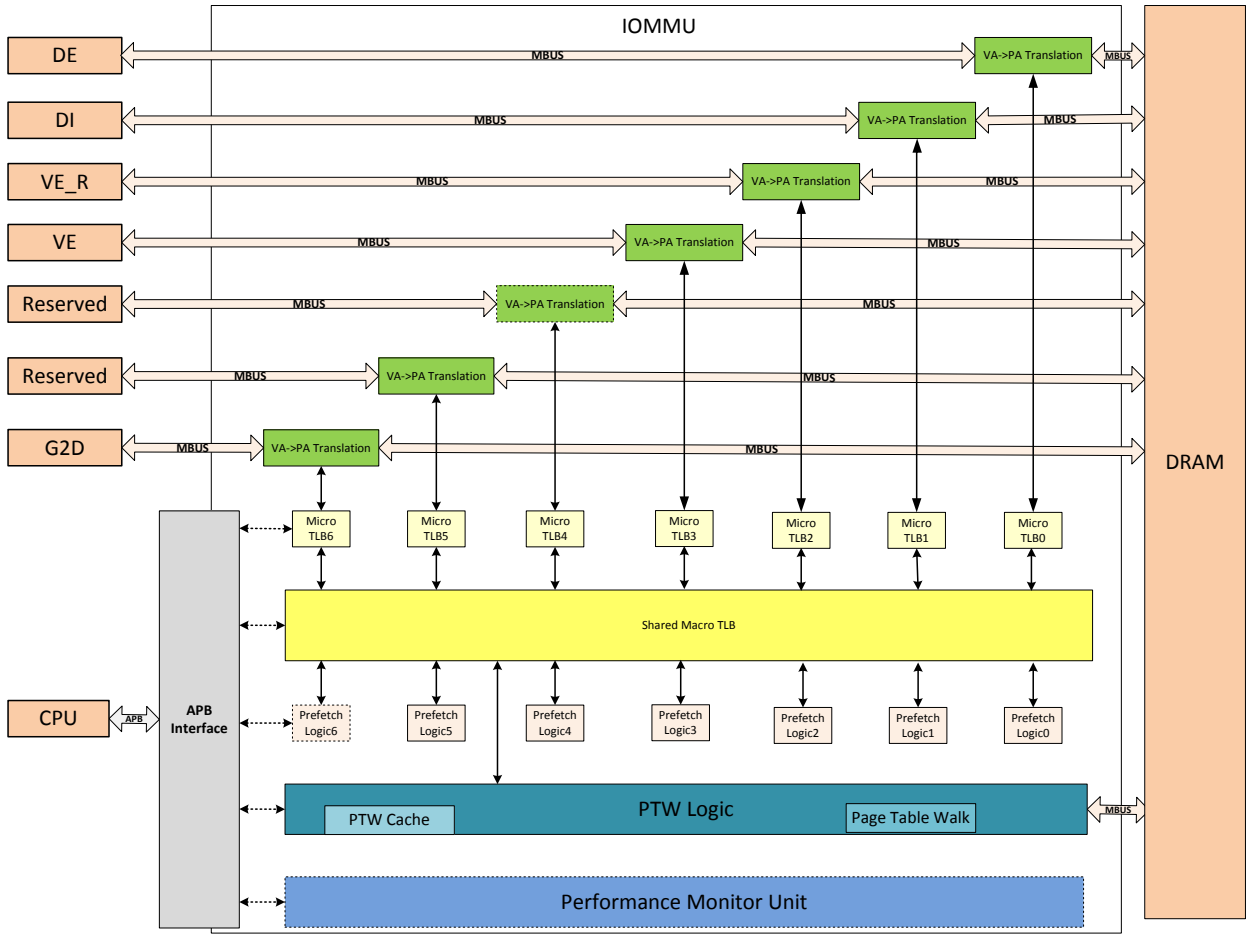

Figure 3- 28. IOMMU Block Diagram

Table 3- 8. Correspondence Relation between Master and Module

Master Number	Module
Master0	DE
Master1	DI
Master2	VE_R
Master3	VE
Master4	Reserved
Master5	Reserved
Master6	G2D

3.12.3. Operations and Functional Descriptions

3.12.3.1. Clock Sources

IOMMU contains two clock domains in the module. Address mapping is generated by MBUS clock domain, and Register and interrupt processing are generated by APB clock domain. The two domains are asynchronous, and they are from different clock sources.

3.12.3.2. Operation Modes

3.12.3.2.1. Initialization

- Release the IOMMU reset signal by writing 1 to the bit[31] of the ***IOMMU Reset Register***;
- Write the base address of the first TLB to the ***IOMMU Translation Table Base Register***;
- Set up the ***IOMMU Interrupt Enable Register***;
- Enable the IOMMU by configuring the ***IOMMU Enable Register*** in the final.

3.12.3.2.2. 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 application is as follows.

- **Micro TLB hit**

a). The master device sends a transfer command, and also sends the address to the corresponding Micro TLB, and searches virtual address corresponding to the Level2 page table;
b). If Micro TLB hits, it will return a corresponding physical addresses and the Level2 page table of permission Index;
c). Address transform module converts the virtual address into physical address, and checks the permissions at the same time. If pass, transfer is completed.

- **Micro TLB miss, Macro TLB hit**

a). The master device sends a transfer command, and also sends the address to the corresponding Micro TLB, and searches virtual address corresponding to the Level2 page table;
b). If Micro TLB misses, then continue to search Macro TLB;
c). If Macro TLB hits, it will return the Level2 page table to Micro TLB;
d). Micro TLB receives the page table, and puts it to Micro TLB(if this Micro TLB is full, there has replace activities), at the same time, sends page table entries to address translation module;
e). Address transform module converts the virtual address into physical address, and checks the permissions at the same time. If pass, transfer is completed.

- **Micro TLB miss, Macro TLB miss, PTW Cache hit**

a). The master device sends a transfer command, and also sends the address to the corresponding Micro TLB, and searches virtual address corresponding to the Level2 page table;
b). If Micro TLB misses, then continue to search Macro TLB;
c). If Macro TLB misses, then it will send the request to the PTW to return the corresponding page table;
d). PTW first accesses PTW Cache, confirms that the required Level1 page table exists in the PTW Cache, sends the page table to PTW logic;
e). 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;

f). Macro TLB stores the Level2 page table (may happen replace activities), and will return the Level2 page table to Micro TLB;

g). Micro TLB receives the page table entries, puts in the Micro TLB (if this Micro TLB is full, there will happen replace activities), and sends page table entries to address translation module;

h). Address transform module converts the virtual address into physical address, and checks the permissions at the same time. If pass, transfer is completed.

- **Micro TLB miss, Macro TLB miss, PTW Cache miss**

a). The master device sends a transfer command, and also sends the address to the corresponding Micro TLB, and searches virtual address corresponding to the Level2 page table;

b). If Micro TLB misses, then continue to search Macro TLB;

c). If Macro TLB misses, there will send the request to the PTW to return the corresponding page table;

d). PTW accesses PTW Cache, there is no necessary Level1 page table;

e). PTW accesses memory, gets the corresponding Level1 page table and stores in the PTW Cache; (may happen replace activities)

f). 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;

g). Macro TLB stores the Level2 page table (may happen replace activities), and will return the Level 2 page table to Micro TLB;

h). Micro TLB receives the page table entries, puts in the Micro TLB (if this Micro TLB is full, there will happen replace activities), and sends page table entries to address translation module;

i). Address transform module converts the virtual address into physical address, and checks the permissions at the same time. If pass, transfer is completed.

- **Permission error**

a). Permission checking always performs in the address conversion;

b). Once the permission checking makes mistake, the new access of the master suspends, before this visit continues;

c). Set the error status register;

d). Trigger interrupt.

- **Invalid Level1 page table**

a). Invalid Level1 page table is checked when PTW logic reads the new level page table from memory;

b). The PTW read sequentially two page table entries from the memory (64-bit data, a complete cache line), and stores in the PTW cache;

c). 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 total cache line(that is two page tables) need to be invalidated.

- **Invalid Level2 page table**

a). Invalid Level2 page table checks when Macro TLB reads the new level page table from memory;

b). The Macro TLB read sequentially two page table entries from the memory (64-bit data, a complete cache line), and stores in the Macro TLB;

c). 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 total cache line(that is two page tables) need to be invalidated.

The internal address switch process shows in Figure 3-28.

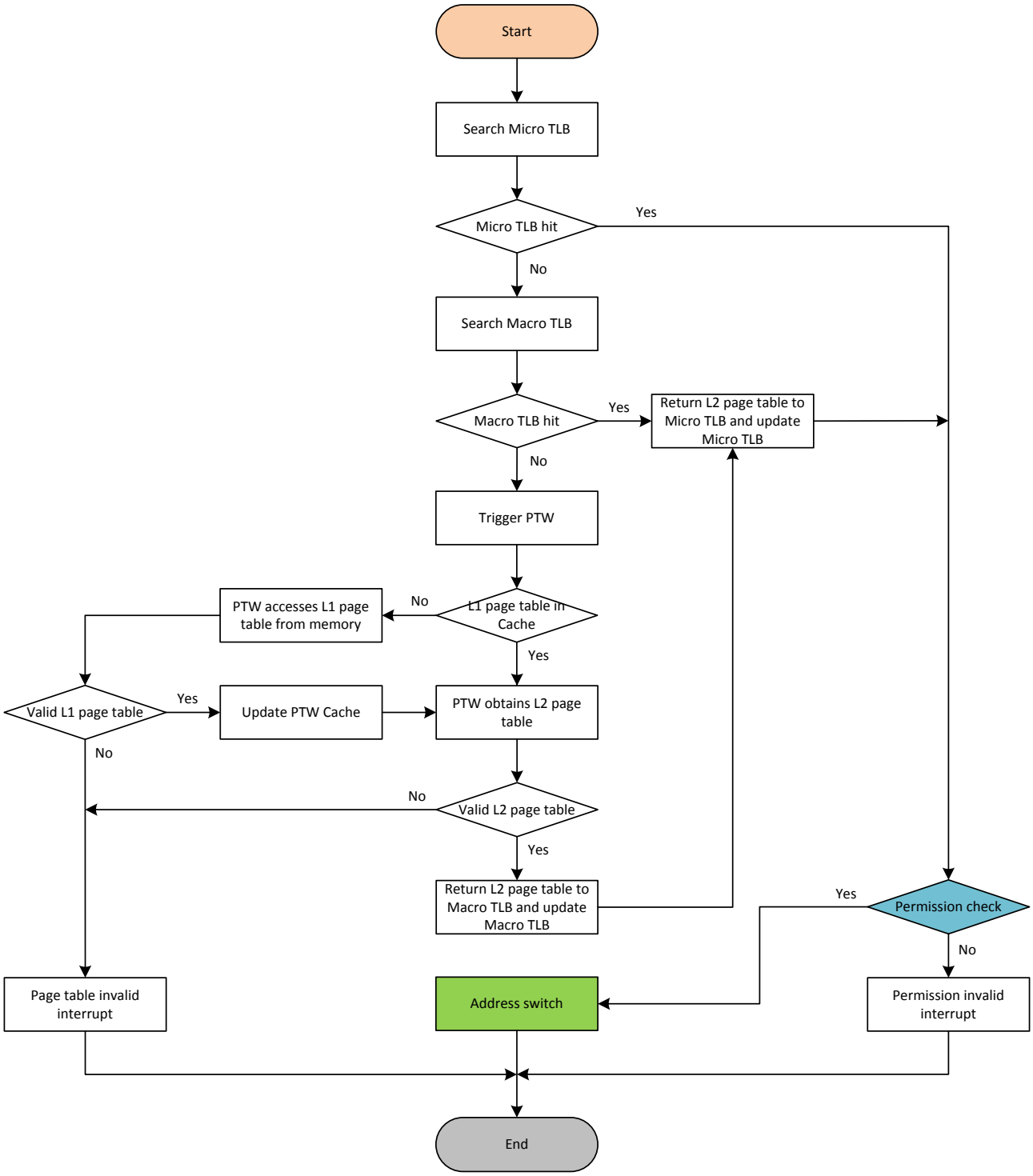

Figure 3- 29. Internal Switch Process

3.12.3.2.3. 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 that 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. A complete VA-PA address translation process is shown in Figure 3-30.

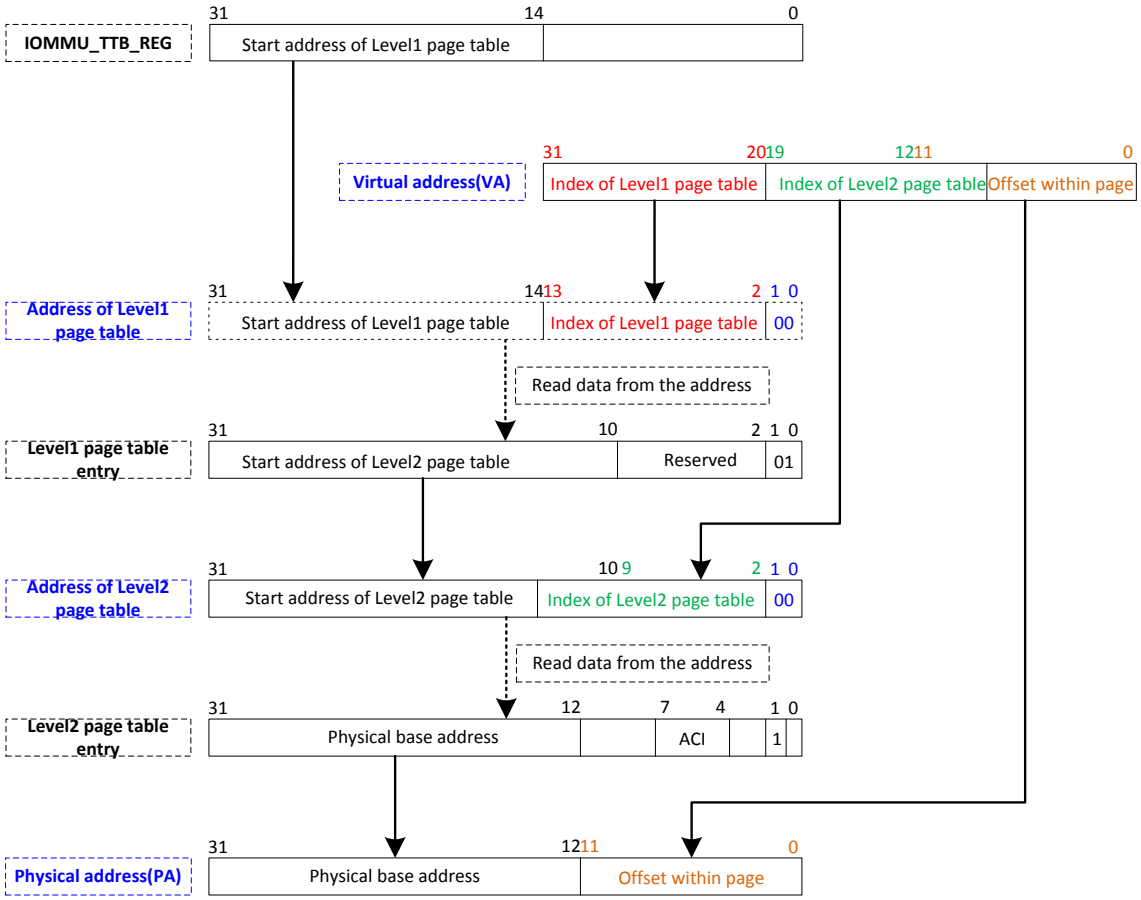

Figure 3- 30. VA-PA Switch Process

3.12.3.2.4. Clear and Invalidate TLB

When multi page table content refresh, or table address 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 access to TLB or Cache, then configure the corresponding Flush bit of **IOMMU TLB Flush Enable Register** , after operation takes effect, 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 TLB VA-PA mapping pairs. The invalid TLB supports two modes.

(1) Mode0

Firstly, set **IOMMU TLB Invalidation Mode Select Register** to 0 to select mode0;

Secondly, write target address to **IOMMU TLB Invalidation Address Register**;

Thirdly, 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, 0xffffc000, 0xffff8000, 0xffff0000 belongs to the legal value; and 0xffffd000, 0xffffb000, 0xfffa000, 0xffff9000, 0xffff7000 belongs to illegal values.

Finally, configure **IOMMU TLB Invalidation Enable Register** to enable 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 process is shown as follows.

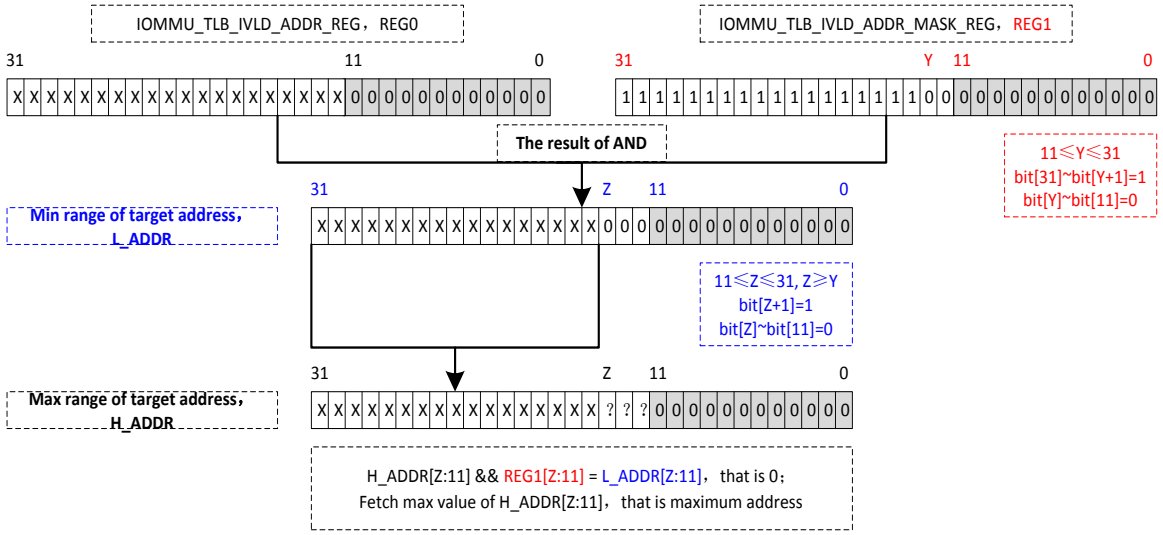

Figure 3- 31. Invalid TLB Address Range

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 target address range 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 target address range 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 target address range 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 target address range is from 0xEEEE0000 to 0xEEEE3000.

(2) Mode1

Firstly, set **IOMMU TLB Invalidation Mode Select Register** to 1 to select mode1;

Secondly, 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;**

Finally, configure **IOMMU TLB Invalidation Enable Register** to enable invalid operation, then invalid related TLB

operation can be completed.

3.12.3.3. Page Table Format

3.12.3.3.1. Level1 Page Table

The format of Level1 page table is as follows.

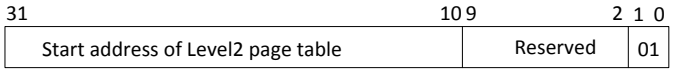

Figure 3- 32. Level1 Page Table Format

Bit[31:10]: Base address of Level2 page table;

Bit[9:2]: Reserved;

Bit[1:0]: 01 is valid page table; others are fault;

3.12.3.3.2. Level2 Page Table

The format of Level2 page table is as follows.

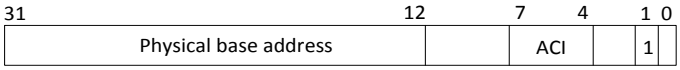

Figure 3- 33. Level1 Page Table Format

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

3.12.3.3.3. Permission Index

The read/write access control of series register such as **IOMMU Domain Authority Control Register** is as follows.

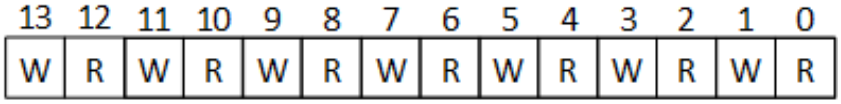

Figure 3- 34. Read/Write Permission Control

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- 9. 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
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**.

3.12.4. Programming Guidelines

3.12.4.1. IOMMU Reset

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.

3.12.4.2. IOMMU Enable

Before opening the IOMMU address mapping function, 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.

3.12.4.3. Configure TTB

Operating the register must close IOMMU address mapping function, namely IOMMU_ENABLE_REG [0] is 0; or Bypass

function of all masters is set to 1, or no the state of transfer bus commands.

3.12.4.4. Clear 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.

3.12.4.5. Read/Write VA Data

For virtual address, read/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, after the operation is finished, check if the results are as expected.

3.12.4.6. PMU Statistics

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

3.12.5. Register List

Module Name	Base Address
IOMMU	0x030F0000

Register Name	Offset	Description
IOMMU_RESET _REG	0x0010	IOMMU ResetRegister
IOMMU_ENABLE_REG	0x0020	IOMMU EnableRegister

IOMMU_BYPASS_REG	0x0030	IOMMU BypassRegister
IOMMU_AUTO_GATING_REG	0x0040	IOMMU Auto GatingRegister
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
IOMMU_TTB_REG	0x0050	IOMMU Translation Table BaseRegister
IOMMU_TLB_ENABLE_REG	0x0060	IOMMU TLB EnableRegister
IOMMU_TLB_PREFETCH_REG	0x0070	IOMMU TLB PrefetchRegister
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_ADDR_REG	0x00A0	IOMMU PC Invalidation Address Register
IOMMU_PC_IVLD_ENABLE_REG	0x00A8	IOMMU PC Invalidation Enable Register
IOMMU_DM_AUT_CTRL_REG0	0x00B0	IOMMU Domain Authority Control Register 0
IOMMU_DM_AUT_CTRL_REG1	0x00B4	IOMMU Domain Authority Control Register 1
IOMMU_DM_AUT_CTRL_REG2	0x00B8	IOMMU Domain Authority Control Register 2
IOMMU_DM_AUT_CTRL_REG3	0x00BC	IOMMU Domain Authority Control Register 3
IOMMU_DM_AUT_CTRL_REG4	0x00C0	IOMMU Domain Authority Control Register 4
IOMMU_DM_AUT_CTRL_REG5	0x00C4	IOMMU Domain Authority Control Register 5
IOMMU_DM_AUT_CTRL_REG6	0x00C8	IOMMU Domain Authority Control Register 6
IOMMU_DM_AUT_CTRL_REG7	0x00CC	IOMMU Domain Authority Control Register 7
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_ADDR_REG0	0x0110	IOMMU Interrupt Error Address Register 0
IOMMU_INT_ERR_ADDR_REG1	0x0114	IOMMU Interrupt Error Address Register 1
IOMMU_INT_ERR_ADDR_REG2	0x0118	IOMMU Interrupt Error Address Register 2
IOMMU_INT_ERR_ADDR_REG3	0x011C	IOMMU Interrupt Error Address Register 3
IOMMU_INT_ERR_ADDR_REG4	0x0120	IOMMU Interrupt Error Address Register 4
IOMMU_INT_ERR_ADDR_REG5	0x0124	IOMMU Interrupt Error Address Register 5
IOMMU_INT_ERR_ADDR_REG6	0x0128	IOMMU Interrupt Error Address Register 6
IOMMU_INT_ERR_ADDR_REG7	0x0130	IOMMU Interrupt Error Address Register 7
IOMMU_INT_ERR_ADDR_REG8	0x0134	IOMMU Interrupt Error Address Register 8
IOMMU_INT_ERR_DATA_REG0	0x0150	IOMMU Interrupt Error Data Register 0
IOMMU_INT_ERR_DATA_REG1	0x0154	IOMMU Interrupt Error Data Register 1
IOMMU_INT_ERR_DATA_REG2	0x0158	IOMMU Interrupt Error Data Register 2
IOMMU_INT_ERR_DATA_REG3	0x015C	IOMMU Interrupt Error Data Register 3
IOMMU_INT_ERR_DATA_REG4	0x0160	IOMMU Interrupt Error Data Register 4
IOMMU_INT_ERR_DATA_REG5	0x0164	IOMMU Interrupt Error Data Register 5

IOMMU_INT_ERR_DATA_REG6	0x0168	IOMMU Interrupt Error Data Register 6
IOMMU_INT_ERR_DATA_REG7	0x0170	IOMMU Interrupt Error Data Register 7
IOMMU_INT_ERR_DATA_REG8	0x0174	IOMMU Interrupt Error Data Register 8
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_LOW_REG0	0x0230	IOMMU PMU Access Low Register 0
IOMMU_PMU_ACCESS_HIGH_REG0	0x0234	IOMMU PMU Access High Register 0
IOMMU_PMU_HIT_LOW_REG0	0x0238	IOMMU PMU Hit Low Register 0
IOMMU_PMU_HIT_HIGH_REG0	0x023C	IOMMU PMU Hit High Register 0
IOMMU_PMU_ACCESS_LOW_REG1	0x0240	IOMMU PMU Access Low Register 1
IOMMU_PMU_ACCESS_HIGH_REG1	0x0244	IOMMU PMU Access High Register 1
IOMMU_PMU_HIT_LOW_REG1	0x0248	IOMMU PMU Hit Low Register 1
IOMMU_PMU_HIT_HIGH_REG1	0x024C	IOMMU PMU Hit High Register 1
IOMMU_PMU_ACCESS_LOW_REG2	0x0250	IOMMU PMU Access Low Register 2
IOMMU_PMU_ACCESS_HIGH_REG2	0x0254	IOMMU PMU Access High Register 2
IOMMU_PMU_HIT_LOW_REG2	0x0258	IOMMU PMU Hit Low Register 2
IOMMU_PMU_HIT_HIGH_REG2	0x025C	IOMMU PMU Hit High Register 2
IOMMU_PMU_ACCESS_LOW_REG3	0x0260	IOMMU PMU Access Low Register 3
IOMMU_PMU_ACCESS_HIGH_REG3	0x0264	IOMMU PMU Access High Register 3
IOMMU_PMU_HIT_LOW_REG3	0x0268	IOMMU PMU Hit Low Register 3
IOMMU_PMU_HIT_HIGH_REG3	0x026C	IOMMU PMU Hit High Register 3
IOMMU_PMU_ACCESS_LOW_REG4	0x0270	IOMMU PMU Access Low Register 4
IOMMU_PMU_ACCESS_HIGH_REG4	0x0274	IOMMU PMU Access High Register 4
IOMMU_PMU_HIT_LOW_REG4	0x0278	IOMMU PMU Hit Low Register 4
IOMMU_PMU_HIT_HIGH_REG4	0x027C	IOMMU PMU Hit High Register 4
IOMMU_PMU_ACCESS_LOW_REG5	0x0280	IOMMU PMU Access Low Register 5
IOMMU_PMU_ACCESS_HIGH_REG5	0x0284	IOMMU PMU Access High Register 5
IOMMU_PMU_HIT_LOW_REG5	0x0288	IOMMU PMU Hit Low Register 5
IOMMU_PMU_HIT_HIGH_REG5	0x028C	IOMMU PMU Hit High Register 5
IOMMU_PMU_ACCESS_LOW_REG6	0x0290	IOMMU PMU Access Low Register 6
IOMMU_PMU_ACCESS_HIGH_REG6	0x0294	IOMMU PMU Access High Register 6
IOMMU_PMU_HIT_LOW_REG6	0x0298	IOMMU PMU Hit Low Register 6
IOMMU_PMU_HIT_HIGH_REG6	0x029C	IOMMU PMU Hit High Register 6
IOMMU_PMU_ACCESS_LOW_REG7	0x02D0	IOMMU PMU Access Low Register 7
IOMMU_PMU_ACCESS_HIGH_REG7	0x02D4	IOMMU PMU Access High Register 7
IOMMU_PMU_HIT_LOW_REG7	0x02D8	IOMMU PMU Hit Low Register 7
IOMMU_PMU_HIT_HIGH_REG7	0x02DC	IOMMU PMU Hit High Register 7
IOMMU_PMU_ACCESS_LOW_REG8	0x02E0	IOMMU PMU Access Low Register 8
IOMMU_PMU_ACCESS_HIGH_REG8	0x02E4	IOMMU PMU Access High Register 8

IOMMU_PMU_HIT_LOW_REG8	0x02E8	IOMMU PMU Hit Low Register 8
IOMMU_PMU_HIT_HIGH_REG8	0x02EC	IOMMU PMU Hit High Register 8
IOMMU_PMU_TL_LOW_REG0	0x0300	IOMMU Total Latency Low Register 0
IOMMU_PMU_TL_HIGH_REG0	0x0304	IOMMU Total Latency High Register 0
IOMMU_PMU_ML_REG0	0x0308	IOMMU Max Latency Register 0
IOMMU_PMU_TL_LOW_REG1	0x0310	IOMMU Total Latency Low Register 1
IOMMU_PMU_TL_HIGH_REG1	0x0314	IOMMU Total Latency High Register 1
IOMMU_PMU_ML_REG1	0x0318	IOMMU Max Latency Register 1
IOMMU_PMU_TL_LOW_REG2	0x0320	IOMMU Total Latency Low Register 2
IOMMU_PMU_TL_HIGH_REG2	0x0324	IOMMU Total Latency High Register 2
IOMMU_PMU_ML_REG2	0x0328	IOMMU Max Latency Register 2
IOMMU_PMU_TL_LOW_REG3	0x0330	IOMMU Total Latency Low Register 3
IOMMU_PMU_TL_HIGH_REG3	0x0334	IOMMU Total Latency High Register 3
IOMMU_PMU_ML_REG3	0x0338	IOMMU Max Latency Register 3
IOMMU_PMU_TL_LOW_REG4	0x0340	IOMMU Total Latency Low Register 4
IOMMU_PMU_TL_HIGH_REG4	0x0344	IOMMU Total Latency High Register 4
IOMMU_PMU_ML_REG4	0x0348	IOMMU Max Latency Register 4
IOMMU_PMU_TL_LOW_REG5	0x0350	IOMMU Total Latency Low Register 5
IOMMU_PMU_TL_HIGH_REG5	0x0354	IOMMU Total Latency High Register 5
IOMMU_PMU_ML_REG5	0x0358	IOMMU Max Latency Register 5
IOMMU_PMU_TL_LOW_REG6	0x0360	IOMMU Total Latency Low Register 6
IOMMU_PMU_TL_HIGH_REG6	0x0364	IOMMU Total Latency High Register 6
IOMMU_PMU_ML_REG6	0x0368	IOMMU Max Latency Register 6

3.12.6. Register Description

3.12.6.1. 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	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	MACRO_TLB_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.
15:7	/	/	/
6	R/W	0x1	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.
5	R/W	0x1	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	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	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	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.
1	R/W	0x1	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	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.

3.12.6.2. 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)

3.12.6.3. 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	MASTER6_BYPASS Master6 bypass switch 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
5	R/W	0x1	MASTER5_BYPASS 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	MASTER4_BYPASS 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	MASTER3_BYPASS 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	MASTER2_BYPASS 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	MASTER1_BYPASS Master1 bypass switch 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	MASTER0_BYPASS 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 after the 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 .

3.12.6.4. 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 decreasing power consumption of the module. 0: Disable auto gating function 1: Enable auto gating function

3.12.6.5. 0x0044 IOMMU Write Buffer Control Register (Default Value: 0x0000_0039)

Offset: 0x0044			Register Name: IOMMU_WBUF_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:6	/	/	/
5	R/W	0x1	MASTER5_WBUF_CTRL Master5 write buffer control bit 0: Disable write buffer 1: Enable write buffer
4	R/W	0x1	MASTER4_WBUF_CTRL Master4 write buffer control bit 0: Disable write buffer 1: Enable write buffer
3	R/W	0x1	MASTER3_WBUF_CTRL Master3 write buffer control bit 0: Disable write buffer 1: Enable write buffer
2:1	/	/	/
0	R/W	0x1	MASTER0_WBUF_CTRL Master0 write buffer control bit 0: Disable write buffer 1: Enable write buffer

3.12.6.6. 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	MASTER6_OOO_CTRL Master6 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order
5	R/W	0x1	MASTER5_OOO_CTRL Master5 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order
4	R/W	0x1	MASTER4_OOO_CTRL Master4 out-of-order control bit

			0: Disable out-of-order 1: Enable out-of-order
3	R/W	0x1	MASTER3_OOO_CTRL Master3 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order
2	R/W	0x1	MASTER2_OOO_CTRL Master2 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order **Note: AI does not support out-of-order, the bit is invalid.**
1	R/W	0x1	MASTER1_OOO_CTRL Master1 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order
0	R/W	0x1	MASTER0_OOO_CTRL Master0 out-of-order control bit 0: Disable out-of-order 1: Enable out-of-order

3.12.6.7. 0x004C IOMMU 4KB Boundary Protect Control Register (Default Value: 0x0000_007F)

Offset: 0x004C			**Register Name: IOMMU_4KB_BDY_PRT_CTRL_REG**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6	R/W	0x1	MASTER6_4KB_BDY_PRT_CTRL Master6 4KB boundary protect control bit 0: Disable 4KB boundary protect 1: Enable 4KB boundary protect
5	R/W	0x1	MASTER5_4KB_BDY_PRT_CTRL Master4 4KB boundary protect control bit 0: Disable 4KB boundary protect 1: Enable 4KB boundary protect
4	R/W	0x1	MASTER4_4KB_BDY_PRT_CTRL Master4 4KB boundary protect control bit 0: Disable 4KB boundary protect 1: Enable 4KB boundary protect
3	R/W	0x1	MASTER3_4KB_BDY_PRT_CTRL Master3 4KB boundary protect control bit 0: Disable 4KB boundary protect 1: Enable 4KB boundary protect
2	R/W	0x1	MASTER2_4KB_BDY_PRT_CTRL Master2 4KB boundary protect control bit 0: Disable 4KB boundary protect

			1: Enable 4KB boundary protect
1	R/W	0x1	MASTER1_4KB_BDY_PRT_CTRL Master1 4KB boundary protect control bit 0: Disable 4KB boundary protect 1: Enable 4KB boundary protect
0	R/W	0x1	MASTER0_4KB_BDY_PRT_CTRL Master0 4KB boundary protect control bit 0: Disable 4KB boundary protect 1: Enable 4KB boundary protect

NOTE

When the virtual address sent by master is over the 4KB boundary, 4KB protection unit will split it into two serial access.

3.12.6.8. 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 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	/	/	/

3.12.6.9. 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
0	R/W	0x1	MICRO_TLB0_ENABLE Micro TLB0 enable bit 0: Disable 1: Enable

3.12.6.10. 0x0070 IOMMU TLB Prefetch Register (Default Value: 0x0000_0000)

Offset: 0x0070			**Register Name: IOMMU_TLB_PREFETCH_REG**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6	R/W	0x0	MICRO_TLB6_PREFETCH Micro TLB6 prefetch enable 0: Disable 1: Enable **Note: When G2D accesses DDR, the prefetch function is suggested to disable.**
5	R/W	0x0	MICRO_TLB5_PREFETCH Micro TLB5 prefetch enable 0: Disable 1: Enable
4	R/W	0x0	MICRO_TLB4_PREFETCH Micro TLB4 prefetch enable

			0: Disable 1: Enable
3	R/W	0x0	MICRO_TLB3_PREFETCH Micro TLB3 prefetch enable 0: Disable 1: Enable
2	R/W	0x0	MICRO_TLB2_PREFETCH Micro TLB2 prefetch enable 0: Disable 1: Enable
1	R/W	0x0	MICRO_TLB1_PREFETCH Micro TLB1 prefetch enable 0: Disable 1: Enable
0	R/W	0x0	MICRO_TLB0_PREFETCH Micro TLB0 prefetch enable 0: Disable 1: Enable

3.12.6.11. 0x0080 IOMMU TLB Flush Enable Register (Default Value: 0x0000_0000)

Offset: 0x0080			Register Name: IOMMU_TLB_FLUSH_ENABLE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R/WAC	0x0	PTW_CACHE_FLUSH Clear PTW Cache 0: No clear operation or clear operation completed 1: Enable is cleared After Flush operation completes, the bit can clear 0 automatically.
16	R/WAC	0x0	MACRO_TLB_FLUSH Clear Macro TLB 0: No clear operation or clear operation completed 1: Enable is cleared After Flush operation completes, the bit can clear 0 automatically.
15:7	/	/	/
6	R/WAC	0x0	MICRO_TLB6_FLUSH Clear Micro TLB6 0: No clear operation or clear operation completed 1: Enable is cleared After Flush operation completes, the bit can clear 0 automatically.
5	R/WAC	0x0	MICRO_TLB5_FLUSH Clear Micro TLB5 0: No clear operation or clear operation completed 1: Enable is cleared

			After Flush operation completes, the bit can clear 0 automatically.
4	R/WAC	0x0	MICRO_TLB4_FLUSH Clear Micro TLB4 0: No clear operation or clear operation completed 1: Enable is cleared After Flush operation completes, the bit can clear 0 automatically.
3	R/WAC	0x0	MICRO_TLB3_FLUSH Clear Micro TLB3 0: No clear operation or clear operation completed 1: Enable is cleared After Flush operation completes, the bit can clear 0 automatically.
2	R/WAC	0x0	MICRO_TLB2_FLUSH Clear Micro TLB2 0: No clear operation or clear operation completed 1: Enable is cleared After Flush operation completes, the bit can clear 0 automatically.
1	R/WAC	0x0	MICRO_TLB1_FLUSH Clear Micro TLB1 0: No clear operation or clear operation completed 1: Enable is cleared After Flush operation completes, the bit can clear 0 automatically.
0	R/WAC	0x0	MICRO_TLB0_FLUSH Clear Micro TLB1 0: No clear operation or clear operation completed 1: Enable is cleared After Flush operation completes, the bit can clear 0 automatically.

NOTE

When performing flush operation, all TLB/Cache access will be paused.

Before flush starts, the operation that has entered TLB continues to complete.

3.12.6.12. 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: Use invalid TLB with Mask mode 1: Use invalid TLB with Start and End mode

3.12.6.13. 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 address, 4KB aligned.
11:0	/	/	/

3.12.6.14. 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 address, 4KB aligned.
11:0	/	/	/

3.12.6.15. 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, 4KB aligned.
11:0	/	/	/

Operation:

1) Set the virtual address which needs to be operated in **IOMMU_TLB_IVLD_ADDR_REG**.
2) Set the mask of virtual address which 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 starts, there is no absolute relationship between same address switch operation and Invalidation operation.

3.12.6.16. 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, 4KB aligned.
11:0	/	/	/

3.12.6.17. 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 invalid operation 0: No-operation or operation completed 1: Enable is invalid After invalidation operation completed, the bit can clear 0 automatically. When operating Invalidation, TLB/Cache operation has not affected. After or Before Invalidation starts, there is no absolute relationship between same address switch operation and Invalidation operation.

3.12.6.18. 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, 1MB aligned.
19:0	/	/	/

3.12.6.19. 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 invalid operation 0: No-operation or operation completed 1: Enable is invalid After invalidation operation completed, the bit can clear 0 automatically. After or Before Invalidation starts, there is no absolute relationship between same address switch operation and Invalidation operation.

3.12.6.20. 0x00B0 IOMMU Domain Authority Control Register 0 (Default Value: 0x0000_0000)

Offset: 0x00B0			Register Name: IOMMU_DM_AUT_CTRL_REG0
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
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
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 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
12	R	0x0	DM0_M6_RD_AUT_CTRL Domain0 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited
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 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

NOTE

Software can be set up 15 different permission control types , which are set in IOMMU_DM_AUT_CTRL_REG0 ~ 7. As well as a default access control type, domain0. The read/write operation of DOMIAN1 ~ 15 is unlimited by default.

Software needs to set the corresponding permission control domain index of the page table item in the secondary page table entries[7:4], the default value is 0, use domian0, namely 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.

3.12.6.21. 0x00B4 IOMMU Domain Authority Control Register 1 (Default Value: 0x0000_0000)

Offset: 0x00B4			Register Name: IOMMU_DM_AUT_CTRL_REG1
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
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
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 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 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
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
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 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

3.12.6.22. 0x00B8 IOMMU Domain Authority Control Register 2 (Default Value: 0x0000_0000)

Offset: 0x00B8			**Register Name: IOMMU_DM_AUT_CTRL_REG2**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29	R/W	0x0	DM5_M6_WT_AUT_CTRL Domain5 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited
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
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
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
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
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 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 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

3.12.6.23. 0x00BC IOMMU Domain Authority Control Register 3 (Default Value: 0x0000_0000)

Offset: 0x00BC			Register Name: IOMMU_DM_AUT_CTRL_REG3
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
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
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 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	/	/	/
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
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
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 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

3.12.6.24. 0x00C0 IOMMU Domain Authority Control Register 4 (Default Value: 0x0000_0000)

Offset: 0x00C0			**Register Name: IOMMU_DM_AUT_CTRL_REG4**
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
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
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 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
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
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
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

3.12.6.25. 0x00C4 IOMMU Domain Authority Control Register 5 (Default Value: 0x0000_0000)

Offset: 0x00C4			**Register Name: IOMMU_DM_AUT_CTRL_REG5**
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
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
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 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
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
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 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

3.12.6.26. 0x00C8 IOMMU Domain Authority Control Register 6 (Default Value: 0x0000_0000)

Offset: 0x00C8			Register Name: IOMMU_DM_AUT_CTRL_REG6
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 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 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
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

3.12.6.27. 0x00CC IOMMU Domain Authority Control Register 7 (Default Value: 0x0000_0000)

Offset: 0x00CC			Register Name: IOMMU_DM_AUT_CTRL_REG7
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
28	R/W	0x0	DM15_M6_RD_AUT_CTRL Domain15 read permission control for master6 0: The read-operation is permitted

			1: The read-operation is prohibited
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
19	R/W	0x0	DM15_M1_WT_AUT_CTRL Domain15 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited
18	R/W	0x0	DM15_M1_RD_AUT_CTRL Domain15 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
17	R/W	0x0	DM15_M0_WT_AUT_CTRL Domain15 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited

16	R/W	0x0	DM15_M0_RD_AUT_CTRL Domain15 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited
15:14	/	/	/
13	R/W	0x0	DM14_M6_WT_AUT_CTRL Domain14 write permission control for master6 0: The write-operation is permitted 1: The write-operation is prohibited
12	R/W	0x0	DM14_M6_RD_AUT_CTRL Domain14 read permission control for master6 0: The read-operation is permitted 1: The read-operation is prohibited
11	R/W	0x0	DM14_M5_WT_AUT_CTRL Domain14 write permission control for master5 0: The write-operation is permitted 1: The write-operation is prohibited
10	R/W	0x0	DM14_M5_RD_AUT_CTRL Domain14 read permission control for master5 0: The read-operation is permitted 1: The read-operation is prohibited
9	R/W	0x0	DM14_M4_WT_AUT_CTRL Domain14 write permission control for master4 0: The write-operation is permitted 1: The write-operation is prohibited
8	R/W	0x0	DM14_M4_RD_AUT_CTRL Domain14 read permission control for master4 0: The read-operation is permitted 1: The read-operation is prohibited
7	R/W	0x0	DM14_M3_WT_AUT_CTRL Domain14 write permission control for master3 0: The write-operation is permitted 1: The write-operation is prohibited
6	R/W	0x0	DM14_M3_RD_AUT_CTRL Domain14 read permission control for master3 0: The read-operation is permitted 1: The read-operation is prohibited
5	R/W	0x0	DM14_M2_WT_AUT_CTRL Domain14 write permission control for master2 0: The write-operation is permitted 1: The write-operation is prohibited
4	R/W	0x0	DM14_M2_RD_AUT_CTRL Domain14 read permission control for master2 0: The read-operation is permitted 1: The read-operation is prohibited

3	R/W	0x0	DM14_M1_WT_AUT_CTRL Domain14 write permission control for master1 0: The write-operation is permitted 1: The write-operation is prohibited
2	R/W	0x0	DM14_M1_RD_AUT_CTRL Domain14 read permission control for master1 0: The read-operation is permitted 1: The read-operation is prohibited
1	R/W	0x0	DM14_M0_WT_AUT_CTRL Domain14 write permission control for master0 0: The write-operation is permitted 1: The write-operation is prohibited
0	R/W	0x0	DM14_M0_RD_AUT_CTRL Domain14 read permission control for master0 0: The read-operation is permitted 1: The read-operation is prohibited

3.12.6.28. 0x00D0 IOMMU Domain Authority Overwrite Register (Default Value: 0x0000_0000)

Offset: 0x00D0			Register Name: IOMMU_DM_AUT_OVWT_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	DM_AUT_OVWT_ENABLE Domain write/read permission overwrite enable 0: Disable 1: Enable
30:14	/	/	/
13	R/W	0x0	M6_WT_AUT_OVWT_CTRL Master6 write permission overwrite control 0: The write-operation is permitted 1: The write-operation is prohibited
12	R/W	0x0	M6_RD_AUT_OVWT_CTRL Master6 read permission overwrite control 0: The read-operation is permitted 1: The read-operation is prohibited
11	R/W	0x0	M5_WT_AUT_OVWT_CTRL Master5 write permission overwrite control 0: The write-operation is permitted 1: The write-operation is prohibited
10	R/W	0x0	M5_RD_AUT_OVWT_CTRL Master5 read permission overwrite control 0: The read-operation is permitted 1: The read-operation is prohibited
9	R/W	0x0	M4_WT_AUT_OVWT_CTRL Master5 write permission overwrite control

			0: The write-operation is permitted 1: The write-operation is prohibited
8	R/W	0x0	M4_RD_AUT_OVWT_CTRL Master5 read permission overwrite control 0: The read-operation is permitted 1: The read-operation is prohibited
7	R/W	0x0	M3_WT_AUT_OVWT_CTRL Master3 write permission overwrite control 0: The write-operation is permitted 1: The write-operation is prohibited
6	R/W	0x0	M3_RD_AUT_OVWT_CTRL Master3 read permission overwrite control 0: The read-operation is permitted 1: The read-operation is prohibited
5	R/W	0x0	M2_WT_AUT_OVWT_CTRL Master2 write permission overwrite control 0: The write-operation is permitted 1: The write-operation is prohibited
4	R/W	0x0	M2_RD_AUT_OVWT_CTRL Master2 read permission overwrite control 0: The read-operation is permitted 1: The read-operation is prohibited
3	R/W	0x0	M1_WT_AUT_OVWT_CTRL Master1 write permission overwrite control 0: The write-operation is permitted 1: The write-operation is prohibited
2	R/W	0x0	M1_RD_AUT_OVWT_CTRL Master1 read permission overwrite control 0: The read-operation is permitted 1: The read-operation is prohibited
1	R/W	0x0	M0_WT_AUT_OVWT_CTRL Master0 write permission overwrite control 0: The write-operation is permitted 1: The write-operation is prohibited
0	R/W	0x0	M0_RD_AUT_OVWT_CTRL Master0 read permission overwrite control 0: The read-operation is permitted 1: The read-operation is prohibited

NOTE

Setting the REG_ARD_OVWT can mask the Domain control defined by IOMMU_DM_AUT_CTRL_REG0~7. All the property of Level2 are covered by the property defined in REG_ARD_OVWT. Allow read and write for all by default.

3.12.6.29. 0x0100 IOMMU Interrupt Enable Register (Default Value: 0x0000_0000)

Offset: 0x0100			Register Name: IOMMU_INT_ENABLE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R/W	0x0	L2_PAGE_TABLE_INVALID_EN Level2 page table invalid interrupt enable 0: Mask interrupt 1: Enable interrupt
16	R/W	0x0	L1_PAGE_TABLE_INVALID_EN Level1 page table invalid interrupt enable 0: Mask interrupt 1: Enable interrupt
15:7	/	/	/
6	R/W	0x0	MICRO_TLB6_INVALID_EN Micro TLB6 permission invalid interrupt enable 0: Mask interrupt 1: Enable interrupt
5	R/W	0x0	MICRO_TLB5_INVALID_EN Micro TLB5 permission invalid interrupt enable 0: Mask interrupt 1: Enable interrupt
4	R/W	0x0	MICRO_TLB4_INVALID_EN Micro TLB4 permission invalid interrupt enable 0: Mask interrupt 1: Enable interrupt
3	R/W	0x0	MICRO_TLB3_INVALID_EN Micro TLB3 permission invalid interrupt enable 0: Mask interrupt 1: Enable interrupt
2	R/W	0x0	MICRO_TLB2_INVALID_EN Micro TLB2 permission invalid interrupt enable 0: Mask interrupt 1: Enable interrupt
1	R/W	0x0	MICRO_TLB1_INVALID_EN Micro TLB1 permission invalid interrupt enable 0: Mask interrupt 1: Enable interrupt
0	R/W	0x0	MICRO_TLB0_INVALID_EN Micro TLB0 permission invalid interrupt enable 0: Mask interrupt 1: Enable interrupt

NOTE

Invalid page table and permission error can not make one device or multi-devices in system work normally.

Permission error usually happens in MicroTLB. The error generates interrupt, and waits for processing through software.

Invalid page table usually happens in MacroTLB. The error can not influence the access of other devices. So the error page table needs go back the way it comes, but the error should not be written in each level TLB.

3.12.6.30. 0x0104 IOMMU Interrupt Clear Register (Default Value: 0x0000_0000)

Offset: 0x0104			Register Name: IOMMU_INT_CLR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	W	0x0	L2_PAGE_TABLE_INVALID_CLR Level2 page table invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt
16	W	0x0	L1_PAGE_TABLE_INVALID_CLR Level1 page table invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt
15:7	/	/	/
6	W	0x0	MICRO_TLB6_INVALID_CLR Micro TLB6 permission invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt
5	W	0x0	MICRO_TLB5_INVALID_CLR Micro TLB5 permission invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt
4	W	0x0	MICRO_TLB4_INVALID_CLR Micro TLB4 permission invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt
3	W	0x0	MICRO_TLB3_INVALID_CLR Micro TLB3 permission invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt
2	W	0x0	MICRO_TLB2_INVALID_CLR Micro TLB2 permission invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt
1	W	0x0	MICRO_TLB1_INVALID_CLR Micro TLB1 permission invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt
0	W	0x0	MICRO_TLB0_INVALID_CLR

			Micro TLB0 permission invalid interrupt clear bit 0: Invalid operation 1: Clear interrupt

3.12.6.31. 0x0108 IOMMU Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0108			Register Name: IOMMU_INT_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R	0x0	L2_PAGE_TABLE_INVALID_STA Level2 page table invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens
16	R	0x0	L1_PAGE_TABLE_INVALID_STA Level1 page table invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens
15:7	/	/	/
6	R	0x0	MICRO_TLB6_INVALID_STA Micro TLB6 permission invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens
5	R	0x0	MICRO_TLB5_INVALID_STA Micro TLB5 permission invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens
4	R	0x0	MICRO_TLB4_INVALID_STA Micro TLB4 permission invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens
3	R	0x0	MICRO_TLB3_INVALID_STA Micro TLB3 permission invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens
2	R	0x0	MICRO_TLB2_INVALID_STA Micro TLB2 permission invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens
1	R	0x0	MICRO_TLB1_INVALID_STA Micro TLB1 permission invalid interrupt status bit 0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens
0	R	0x0	MICRO_TLB0_INVALID_STA Micro TLB0 permission invalid interrupt status bit

			0: Interrupt does not happen or interrupt is cleared 1: Interrupt happens

3.12.6.32. 0x0110 IOMMU Interrupt Error Address Register 0 (Default Value: 0x0000_0000)

Offset: 0x0110			Register Name: IOMMU_INT_ERR_ADDR_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_ADDR0 Virtual address that caused Micro TLB0 to interrupt

3.12.6.33. 0x0114 IOMMU Interrupt Error Address Register 1 (Default Value: 0x0000_0000)

Offset: 0x0114			Register Name: IOMMU_INT_ERR_ADDR_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_ADDR1 Virtual address that caused Micro TLB1 to interrupt

3.12.6.34. 0x0118 IOMMU Interrupt Error Address Register 2 (Default Value: 0x0000_0000)

Offset: 0x0118			Register Name: IOMMU_INT_ERR_ADDR_REG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_ADDR2 Virtual address that caused Micro TLB2 to interrupt

3.12.6.35. 0x011C IOMMU Interrupt Error Address Register 3 (Default Value: 0x0000_0000)

Offset: 0x011C			Register Name: IOMMU_INT_ERR_ADDR_REG3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_ADDR3 Virtual address that caused Micro TLB3 to interrupt

3.12.6.36. 0x0120 IOMMU Interrupt Error Address Register 4 (Default Value: 0x0000_0000)

Offset: 0x0120			Register Name: IOMMU_INT_ERR_ADDR_REG4
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_ADDR4 Virtual address that caused Micro TLB4 to interrupt

3.12.6.37. 0x0124 IOMMU Interrupt Error Address Register 5 (Default Value: 0x0000_0000)

Offset: 0x0124			Register Name: IOMMU_INT_ERR_ADDR_REG5
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_ADDR5 Virtual address that caused Micro TLB5 to interrupt

3.12.6.38. 0x0128 IOMMU Interrupt Error Address Register 6 (Default Value: 0x0000_0000)

Offset: 0x0128			Register Name: IOMMU_INT_ERR_ADDR_REG6
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_ADDR6 Virtual address that caused Micro TLB6 to interrupt

3.12.6.39. 0x0130 IOMMU Interrupt Error Address Register 7 (Default Value: 0x0000_0000)

Offset: 0x0130			Register Name: IOMMU_INT_ERR_ADDR_REG7
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_ADDR7 Virtual address that caused L1 page table to interrupt

3.12.6.40. 0x0134 IOMMU Interrupt Error Address Register 8 (Default Value: 0x0000_0000)

Offset: 0x0134			Register Name: IOMMU_INT_ERR_ADDR_REG8
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_ADDR8 Virtual address that caused L2 page table to interrupt

3.12.6.41. 0x0150 IOMMU Interrupt Error Data Register 0 (Default Value: 0x0000_0000)

Offset: 0x0150			Register Name: IOMMU_INT_ERR_DATA_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_DATA0 Corresponding page table of virtual address that caused Micro TLB0 to interrupt

3.12.6.42. 0x0154 IOMMU Interrupt Error Data Register 1 (Default Value: 0x0000_0000)

Offset: 0x0154	Register Name: IOMMU_INT_ERR_DATA_REG1

Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	INT_ERR_DATA1 Corresponding page table of virtual address that caused Micro TLB1 to interrupt

3.12.6.43. 0x0158 IOMMU Interrupt Error Data Register 2 (Default Value: 0x0000_0000)

Offset: 0x0158			Register Name: IOMMU_INT_ERR_DATA_REG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_DATA2 Corresponding page table of virtual address that caused Micro TLB2 to interrupt

3.12.6.44. 0x015C IOMMU Interrupt Error Data Register 3 (Default Value: 0x0000_0000)

Offset: 0x015C			Register Name: IOMMU_INT_ERR_DATA_REG3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_DATA3 Corresponding page table of virtual address that caused Micro TLB3 to interrupt

3.12.6.45. 0x0160 IOMMU Interrupt Error Data Register 4 (Default Value: 0x0000_0000)

Offset: 0x0160			Register Name: IOMMU_INT_ERR_DATA_REG4
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_DATA4 Corresponding page table of virtual address that caused Micro TLB4 to interrupt

3.12.6.46. 0x0164 IOMMU Interrupt Error Data Register 5 (Default Value: 0x0000_0000)

Offset: 0x0164			Register Name: IOMMU_INT_ERR_DATA_REG5
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_DATA5 Corresponding page table of virtual address that caused Micro TLB5 to interrupt

3.12.6.47. 0x0168 IOMMU Interrupt Error Data Register 6 (Default Value: 0x0000_0000)

Offset: 0x0168			Register Name: IOMMU_INT_ERR_DATA_REG6
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_DATA6 Corresponding page table of virtual address that caused Micro TLB6 to interrupt

3.12.6.48. 0x0170 IOMMU Interrupt Error Data Register 7 (Default Value: 0x0000_0000)

Offset: 0x0170			Register Name: IOMMU_INT_ERR_DATA_REG7
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_DATA7 Corresponding page table of virtual address that caused L1 page table to interrupt

3.12.6.49. 0x0174 IOMMU Interrupt Error Data Register 8 (Default Value: 0x0000_0000)

Offset: 0x0174			Register Name: IOMMU_INT_ERR_DATA_REG8
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	INT_ERR_DATA8 Corresponding page table of virtual address that caused L2 page table to interrupt

3.12.6.50. 0x0180 IOMMU L1 Page Table Interrupt Register (Default Value: 0x0000_0000)

Offset: 0x0180			Register Name: IOMMU_L1PG_INT_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x0	DBG_MODE_L1PG_INT Debug mode address switch causes L1 page table to occur interrupt.
30:7	/	/	/
6	R	0x0	MASTER6_L1PG_INT Master6 address switch causes L1 page table to occur interrupt.
5	R	0x0	MASTER5_L1PG_INT Master5 address switch causes L1 page table to occur interrupt.
4	R	0x0	MASTER4_L1PG_INT Master4 address switch causes L1 page table to occur interrupt.
3	R	0x0	MASTER3_L1PG_INT Master3 address switch causes L1 page table to occur interrupt.
2	R	0x0	MASTER2_L1PG_INT Master2 address switch causes L1 page table to occur interrupt.

1	R	0x0	MASTER1_L1PG_INT Master1 address switch causes L1 page table to occur interrupt.
0	R	0x0	MASTER0_L1PG_INT Master0 address switch causes L1 page table to occur interrupt.

3.12.6.51. 0x0184 IOMMU L2 Page Table Interrupt Register (Default Value: 0x0000_0000)

Offset: 0x0184			Register Name: IOMMU_L2PG_INT_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x0	DBG_MODE_L2PG_INT Debug mode address switch causes L2 page table to occur interrupt.
30:7	/	/	/
6	R	0x0	MASTER6_L2PG_INT Master6 address switch causes L2 page table to occur interrupt.
5	R	0x0	MASTER5_L2PG_INT Master5 address switch causes L2 page table to occur interrupt.
4	R	0x0	MASTER4_L2PG_INT Master4 address switch causes L2 page table to occur interrupt.
3	R	0x0	MASTER3_L2PG_INT Master3 address switch causes L2 page table to occur interrupt.
2	R	0x0	MASTER2_L2PG_INT Master2 address switch causes L2 page table to occur interrupt.
1	R	0x0	MASTER1_L2PG_INT Master1 address switch causes L2 page table to occur interrupt.
0	R	0x0	MASTER0_L2PG_INT Master0 address switch causes L2 page table to occur interrupt.

3.12.6.52. 0x0190 IOMMU Virtual Address Register (Default Value: 0x0000_0000)

Offset: 0x0190			Register Name: IOMMU_VA _REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	VA Virtual address of read/write

3.12.6.53. 0x0194 IOMMU Virtual Address Data Register (Default Value: 0x0000_0000)

Offset: 0x0194			Register Name: IOMMU_VA_DATA _REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	VA_DATA Data of read/write virtual address

3.12.6.54. 0x0198 IOMMU Virtual Address Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0198			Register Name: IOMMU_VA_CONFIG _REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MODE_SEL 0: Prefetch 1: Debug Mode It is used to chose prefetch mode or Debug mode.
31:9	/	/	/
8	R/W	0x0	VA_CONFIG 0: Read operation 1: Write operation
7:1	/	/	/
0	R/WAC	0x0	VA_CONFIG_START 0: No operation or operation completes 1: Start After the operation completes, the bit can clear to 0 automatically.

Read operation process:

a) Write IOMMU_VA_REG[31:0];

b) Write IOMMU_VA_CONFIG_REG[8] to 0;

c) Write IOMMU_VA_CONFIG_REG[0] to 1 to start read-process;

d) Query IOMMU_VA_CONFIG_REG[0] until it is 0;

e) Read IOMMU_VA_DATA_REG[31:0];

Write operation process:

a) Write IOMMU_VA _REG[31:0];

b) Write IOMMU_VA_DATA_REG[31:0];

c) Write IOMMU_VA_CONFIG_REG[8] to 1;

d) Write IOMMU_VA_CONFIG_REG[0] to 1 to start write-process;

e) Query IOMMU_VA_CONFIG_REG[0] until it is 0;

3.12.6.55. 0x0200 IOMMU PMU Enable Register (Default Value: 0x0000_0000)

Offset: 0x0200			Register Name: IOMMU_PMU_ENABLE _REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	PMU_ENABLE 0: Disable statistical function 1: Enable statistical function

3.12.6.56. 0x0210 IOMMU PMU Clear Register (Default Value: 0x0000_0000)

Offset: 0x0210	Register Name: IOMMU_PMU_CLR _REG

Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/WAC	0x0	PMU_CLR 0: No clear operation or clear operation completes 1: Clear counter data After the operation completes, the bit can clear to 0 automatically.

3.12.6.57. 0x0230 IOMMU PMU Access Low Register 0 (Default Value: 0x0000_0000)

Offset: 0x0230			Register Name: IOMMU_PMU_ACCESS_LOW_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ACCESS_LOW0 Record total number of Micro TLB0 access , lower 32-bit register

3.12.6.58. 0x0234 IOMMU PMU Access High Register 0 (Default Value: 0x0000_0000)

Offset: 0x0234			Register Name: IOMMU_PMU_ACCESS_HIGH_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH0 Record total number of Micro TLB0 access , higher 11-bit register

3.12.6.59. 0x0238 IOMMU PMU Hit Low Register 0 (Default Value: 0x0000_0000)

Offset: 0x0238			Register Name: IOMMU_PMU_HIT_LOW_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_HIT_LOW0 Record total number of Micro TLB0 hit , lower 32-bit register

3.12.6.60. 0x023C IOMMU PMU Hit High Register 0 (Default Value: 0x0000_0000)

Offset: 0x023C			Register Name: IOMMU_PMU_HIT_HIGH_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH0 Record total number of Micro TLB0 hit , higher 11-bit register

3.12.6.61. 0x0240 IOMMU PMU Access Low Register 1 (Default Value: 0x0000_0000)

Offset: 0x0240			Register Name: IOMMU_PMU_ACCESS_LOW_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ACCESS_LOW1 Record total number of Micro TLB1 access , lower 32-bit register

3.12.6.62. 0x0244 IOMMU PMU Access High Register 1 (Default Value: 0x0000_0000)

Offset: 0x0244			Register Name: IOMMU_PMU_ACCESS_HIGH_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH1 Record total number of Micro TLB1 access , higher 11-bit register

3.12.6.63. 0x0248 IOMMU PMU Hit Low Register 1 (Default Value: 0x0000_0000)

Offset: 0x0248			Register Name: IOMMU_PMU_HIT_LOW_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_HIT_LOW1 Record total number of Micro TLB1 hit , lower 32-bit register

3.12.6.64. 0x024C IOMMU PMU Hit High Register 1 (Default Value: 0x0000_0000)

Offset: 0x024C			Register Name: IOMMU_PMU_HIT_HIGH_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH1 Record total number of Micro TLB1 hit , higher 11-bit register

3.12.6.65. 0x0250 IOMMU PMU Access Low Register 2 (Default Value: 0x0000_0000)

Offset: 0x0250			Register Name: IOMMU_PMU_ACCESS_LOW_REG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ACCESS_LOW2 Record total number of Micro TLB2 access , lower 32-bit register

3.12.6.66. 0x0254 IOMMU PMU Access High Register 2 (Default Value: 0x0000_0000)

Offset: 0x0254			Register Name: IOMMU_PMU_ACCESS_HIGH_REG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH2 Record total number of Micro TLB2 access , higher 11-bit register

3.12.6.67. 0x0258 IOMMU PMU Hit Low Register 2 (Default Value: 0x0000_0000)

Offset: 0x0258			Register Name: IOMMU_PMU_HIT_LOW_REG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_HIT_LOW2 Record total number of Micro TLB2 hit , lower 32-bit register

3.12.6.68. 0x025C IOMMU PMU Hit High Register 2 (Default Value: 0x0000_0000)

Offset: 0x025C			Register Name: IOMMU_PMU_HIT_HIGH_REG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH2 Record total number of Micro TLB2 hit , higher 11-bit register

3.12.6.69. 0x0260 IOMMU PMU Access Low Register 3 (Default Value: 0x0000_0000)

Offset: 0x0260			Register Name: IOMMU_PMU_ACCESS_LOW_REG3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ACCESS_LOW3 Record total number of Micro TLB3 access, lower 32-bit register

3.12.6.70. 0x0264 IOMMU PMU Access High Register 3 (Default Value: 0x0000_0000)

Offset: 0x0264			Register Name: IOMMU_PMU_ACCESS_HIGH_REG3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH3 Record total number of Micro TLB3 access , higher 11-bit register

3.12.6.71. 0x0268 IOMMU PMU Hit Low Register 3 (Default Value: 0x0000_0000)

Offset: 0x0268			Register Name: IOMMU_PMU_HIT_LOW_REG3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_HIT_LOW3 Record total number of Micro TLB3 hit, lower 32-bit register

3.12.6.72. 0x026C IOMMU PMU Hit High Register 3 (Default Value: 0x0000_0000)

Offset: 0x026C			Register Name: IOMMU_PMU_HIT_HIGH_REG3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH3 Record total number of Micro TLB3 hit , higher 11-bit register

3.12.6.73. 0x0270 IOMMU PMU Access Low Register 4 (Default Value: 0x0000_0000)

Offset: 0x0270			Register Name: IOMMU_PMU_ACCESS_LOW_REG4
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ACCESS_LOW4 Record total number of Micro TLB4 access, lower 32-bit register

3.12.6.74. 0x0274 IOMMU PMU Access High Register 4 (Default Value: 0x0000_0000)

Offset: 0x0274			Register Name: IOMMU_PMU_ACCESS_HIGH_REG4
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH4 Record total number of Micro TLB4 access, higher 11-bit register

3.12.6.75. 0x0278 IOMMU PMU Hit Low Register 4 (Default Value: 0x0000_0000)

Offset: 0x0278			Register Name: IOMMU_PMU_HIT_LOW_REG4
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_HIT_LOW4 Record total number of Micro TLB4 hit, lower 32-bit register

3.12.6.76. 0x027C IOMMU PMU Hit High Register 4 (Default Value: 0x0000_0000)

Offset: 0x027C			Register Name: IOMMU_PMU_HIT_HIGH_REG4
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH4 Record total number of Micro TLB4 hit, higher 11-bit register

3.12.6.77. 0x0280 IOMMU PMU Access Low Register 5 (Default Value: 0x0000_0000)

Offset: 0x0280			Register Name: IOMMU_PMU_ACCESS_LOW_REG5
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ACCESS_LOW5 Record total number of Micro TLB5 access, lower 32-bit register

3.12.6.78. 0x0284 IOMMU PMU Access High Register 5 (Default Value: 0x0000_0000)

Offset: 0x0284			Register Name: IOMMU_PMU_ACCESS_HIGH_REG5
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH5 Record total number of Micro TLB5 access, higher 11-bit register

3.12.6.79. 0x0288 IOMMU PMU Hit Low Register 5 (Default Value: 0x0000_0000)

Offset: 0x0288			Register Name: IOMMU_PMU_HIT_LOW_REG5
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_HIT_LOW5 Record total number of Micro TLB5 hit, lower 32-bit register

3.12.6.80. 0x028C IOMMU PMU Hit High Register 5 (Default Value: 0x0000_0000)

Offset: 0x028C			Register Name: IOMMU_PMU_HIT_HIGH_REG5
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH5 Record total number of Micro TLB5 hit, higher 11-bit register

3.12.6.81. 0x0290 IOMMU PMU Access Low Register6 (Default Value: 0x0000_0000)

Offset: 0x0290			Register Name: IOMMU_PMU_ACCESS_LOW_REG6
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ACCESS_LOW6 Record total number of Micro TLB6 access, lower 32-bit register

3.12.6.82. 0x0294 IOMMU PMU Access High Register 6 (Default Value: 0x0000_0000)

Offset: 0x0294			Register Name: IOMMU_PMU_ACCESS_HIGH_REG6
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH6 Record total number of Micro TLB6 access, higher 11-bit register

3.12.6.83. 0x0298 IOMMU PMU Hit Low Register 6 (Default Value: 0x0000_0000)

Offset: 0x0298			Register Name: IOMMU_PMU_HIT_LOW_REG6
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_HIT_LOW6 Record total number of Micro TLB6 hit, lower 32-bit register

3.12.6.84. 0x029C IOMMU PMU Hit High Register 6 (Default Value: 0x0000_0000)

Offset: 0x029C			Register Name: IOMMU_PMU_HIT_HIGH_REG6
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH6 Record total number of Micro TLB6 hit, higher 11-bit register

3.12.6.85. 0x02D0 IOMMU PMU Access Low Register 7 (Default Value: 0x0000_0000)

Offset: 0x02D0			Register Name: IOMMU_PMU_ACCESS_LOW_REG7
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ACCESS_LOW7 Record total number of Micro TLB7 access, lower 32-bit register

3.12.6.86. 0x02D4 IOMMU PMU Access High Register 7 (Default Value: 0x0000_0000)

Offset: 0x02D4			Register Name: IOMMU_PMU_ACCESS_HIGH_REG7
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH7 Record total number of Micro TLB7 access, higher 11-bit register

3.12.6.87. 0x02D8 IOMMU PMU Hit Low Register 7 (Default Value: 0x0000_0000)

Offset: 0x02D8			Register Name: IOMMU_PMU_HIT_LOW_REG7
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_HIT_LOW7 Record total number of Micro TLB7 hit, lower 32-bit register

3.12.6.88. 0x02DC IOMMU PMU Hit High Register 7 (Default Value: 0x0000_0000)

Offset: 0x02DC			Register Name: IOMMU_PMU_HIT_HIGH_REG7
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH7 Record total number of Micro TLB7 hit, higher 11-bit register

3.12.6.89. 0x02E0 IOMMU PMU Access Low Register 8 (Default Value: 0x0000_0000)

Offset: 0x02E0			Register Name: IOMMU_PMU_ACCESS_LOW_REG8
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ACCESS_LOW8 Record total number of PTW Cache access, lower 32-bit register

3.12.6.90. 0x02E4 IOMMU PMU Access High Register 8 (Default Value: 0x0000_0000)

Offset: 0x02E4			Register Name: IOMMU_PMU_ACCESS_HIGH_REG8
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_ACCESS_HIGH8 Record total number of PTW Cache access, higher 11-bit register

3.12.6.91. 0x02E8 IOMMU PMU Hit Low Register 8 (Default Value: 0x0000_0000)

Offset: 0x02E8			Register Name: IOMMU_PMU_HIT_LOW_REG8
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_HIT_LOW8 Record total number of PTW Cache hit, lower 32-bit register

3.12.6.92. 0x02EC IOMMU PMU Hit High Register 8 (Default Value: 0x0000_0000)

Offset: 0x02EC			Register Name: IOMMU_PMU_HIT_HIGH_REG8
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R	0x0	PMU_HIT_HIGH8 Record total number of PTW Cache hit, higher 11-bit register

3.12.6.93. 0x0300 IOMMU Total Latency Low Register 0 (Default Value: 0x0000_0000)

Offset: 0x0300			Register Name: IOMMU_PMU_TL_LOW_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_TL_LOW0 Record total latency of Master0, lower 32-bit register

3.12.6.94. 0x0304 IOMMU Total Latency High Register 0 (Default Value: 0x0000_0000)

Offset: 0x0304			Register Name: IOMMU_PMU_TL_HIGH_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17:0	R	0x0	PMU_TL_HIGH0 Record total latency of Master0, higher 18-bit register

3.12.6.95. 0x0308 IOMMU Max Latency Register 0 (Default Value: 0x0000_0000)

Offset: 0x0308			Register Name: IOMMU_PMU_ML_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ ML0 Record the max latency of Master0.

3.12.6.96. 0x0310 IOMMU Total Latency Low Register 1(Default Value: 0x0000_0000)

Offset: 0x0310			Register Name: IOMMU_PMU_TL_LOW_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_TL_LOW1 Record total latency of Master1, lower 32-bit register

3.12.6.97. 0x0314 IOMMU Total Latency High Register 1 (Default Value: 0x0000_0000)

Offset: 0x0314			Register Name: IOMMU_PMU_TL_HIGH_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17:0	R	0x0	PMU_TL_HIGH1 Record total latency of Master1, higher 18-bit register

3.12.6.98. 0x0318 IOMMU Max Latency Register 1 (Default Value: 0x0000_0000)

Offset: 0x0318			Register Name: IOMMU_PMU_ML_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ML1 Record the max latency of Master1.

3.12.6.99. 0x0320 IOMMU Total Latency Low Register 2 (Default Value: 0x0000_0000)

Offset: 0x0320			Register Name: IOMMU_PMU_TL_LOW_REG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_TL_LOW2 Record total latency of Master2, lower 32-bit register

3.12.6.100. 0x0324 IOMMU Total Latency High Register 2 (Default Value: 0x0000_0000)

Offset: 0x0324			Register Name: IOMMU_PMU_TL_HIGH_REG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17:0	R	0x0	PMU_TL_HIGH2 Record total latency of Master2, higher 18-bit register

3.12.6.101. 0x0328 IOMMU Max Latency Register 2 (Default Value: 0x0000_0000)

Offset: 0x0328			Register Name: IOMMU_PMU_ML_REG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ ML2 Record the max latency of Master2.

3.12.6.102. 0x0330 IOMMU Total Latency Low Register 3 (Default Value: 0x0000_0000)

Offset: 0x0330			Register Name: IOMMU_PMU_TL_LOW_REG3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ TL_LOW3 Record total latency of Master3, lower 32-bit register

3.12.6.103. 0x0334 IOMMU Total Latency High Register 3 (Default Value: 0x0000_0000)

Offset: 0x0334			Register Name: IOMMU_PMU_TL_HIGH_REG3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17:0	R	0x0	PMU_TL_HIGH3 Record total latency of Master3, higher 18-bit register

3.12.6.104. 0x0338 IOMMU Max Latency Register 3 (Default Value: 0x0000_0000)

Offset: 0x0338			Register Name: IOMMU_PMU_ML_REG3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ ML3 Record the max latency of Master3.

3.12.6.105. 0x0340 IOMMU Total Latency Low Register 4 (Default Value: 0x0000_0000)

Offset: 0x0340			Register Name: IOMMU_PMU_TL_LOW_REG4
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_TL_LOW4 Record total latency of Master4, lower 32-bit register

3.12.6.106. 0x0344 IOMMU Total Latency High Register 4 (Default Value: 0x0000_0000)

Offset: 0x0344	Register Name: IOMMU_PMU_TL_HIGH_REG4

Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17:0	R	0x0	PMU_TL_HIGH4 Record total latency of Master4, higher 18-bit register

3.12.6.107. 0x0348 IOMMU Max Latency Register 4 (Default Value: 0x0000_0000)

Offset: 0x0348			Register Name: IOMMU_PMU_ML_REG4
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ ML4 Record the max latency of Master4.

3.12.6.108. 0x0350 IOMMU Total Latency Low Register 5 (Default Value: 0x0000_0000)

Offset: 0x0350			Register Name: IOMMU_PMU_TL_LOW_REG5
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_TL_LOW5 Record total latency of Master5, lower 32-bit register

3.12.6.109. 0x0354 IOMMU Total Latency High Register 5 (Default Value: 0x0000_0000)

Offset: 0x0354			Register Name: IOMMU_PMU_TL_HIGH_REG5
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17:0	R	0x0	PMU_TL_HIGH5 Record total latency of Master5, higher 18-bit register

3.12.6.110. 0x0358 IOMMU Max Latency Register 5 (Default Value: 0x0000_0000)

Offset: 0x0358			Register Name: IOMMU_PMU_ML_REG5
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ ML5 Record the max latency of Master5.

3.12.6.111. 0x0360 IOMMU Total Latency Low Register 6 (Default Value: 0x0000_0000)

Offset: 0x0360			Register Name: IOMMU_PMU_TL_LOW_REG6
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_TL_LOW6

			Record total latency of Master6, lower 32-bit register

3.12.6.112. 0x0364 IOMMU Total Latency High Register 6 (Default Value: 0x0000_0000)

Offset: 0x0364			Register Name: IOMMU_PMU_TL_HIGH_REG6
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17:0	R	0x0	PMU_TL_HIGH6 Record total latency of Master6, higher 18-bit register

3.12.6.113. 0x0368 IOMMU Max Latency Register 6 (Default Value: 0x0000_0000)

Offset: 0x0368			Register Name: IOMMU_PMU_ML_REG6
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PMU_ML6 Record the max latency of Master6.

3.13. RTC

3.13.1. Overview

The RTC(Real Time Clock) is used to display the real time and periodically wakeup. The RTC can display the year, month, day, week, hour, minute, second in real time. The RTC has the independent power to continue to work in system power-off.

The RTC has the following features:

- 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
- Supports one solution without low-frequency crystal, a precise 32.768 kHz counter clock can be generated by using HOSC to calibrate the internal RC clock
- Configurable initial value by software anytime
- Periodically alarm to wakeup the external devices
- 16 general purpose registers for storing power-off information

3.13.2. Clock Tree Diagram

The clock tree diagram of RTC is shown in Figure 3-35.

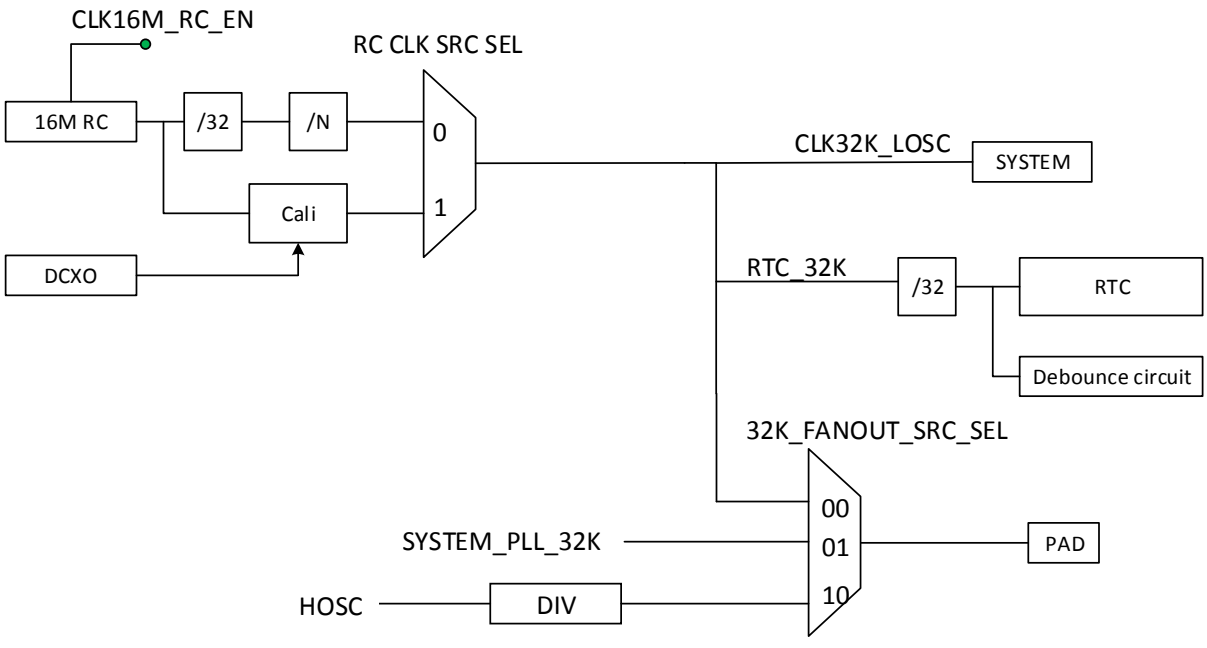

Figure 3- 35. RTC Clock Tree

RTC clock tree can be selected by corresponding switch, there are 2 options: 32K obtained by frequency division of RC, Cali 32K after calibrated.
Clock source: internal 16 MHz RC oscillator, if using calibration output, the high-frequency crystal of DCXO is needed.
Output clock: CLK32K_LOSC and RTC_32K.
Fanout: The clock source of fanout can select RTC_32K, or 32K divided by PLL_PERI(2X), or 32K divided by HOSC.

3.13.3. Operations and Functional Descriptions

3.13.3.1. External Signals

Table 3- 10. RTC External Signals

Signal	Description
X32KFOUT	32.768 kHz clock fanout, provides low frequency clock to the external device

3.13.3.2. Clock and Reset

The RTC module has the independent reset signal, the signal follows VCC_RTC. When VCC_RTC powers on, the reset signal resets the RTC module; after VCC_RTC reaches stable, the reset signal always holds high level. Watchdog Reset cannot reset RTC.
The RTC module accesses its register by APBS1.

3.13.3.3. Typical Application

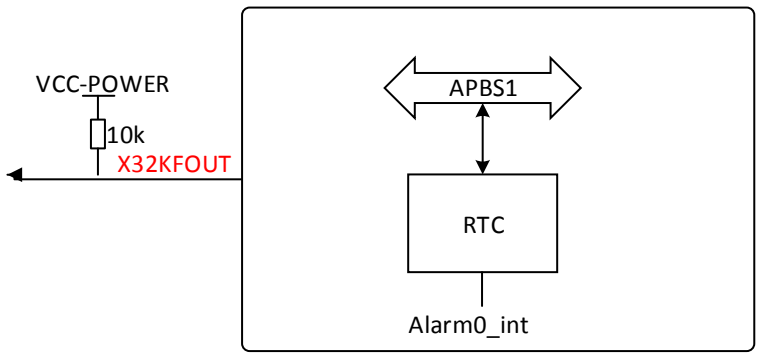

Figure 3- 36. RTC Application Diagram

The system accesses RTC register by APBS1 to generate the real time.
If the external device needs low frequency oscillator, which can be provided by X32KFOUT.

3.13.3.4. Function Implementation

3.13.3.4.1. Clock Sources

The RTC has 1 clock source: internal RC.

When using internal RC, the clock of RTC can be changed by changing division ratio, or a precise 32K clock can be output by enabling calibration circuit.

3.13.3.4.2. Real Time Clock

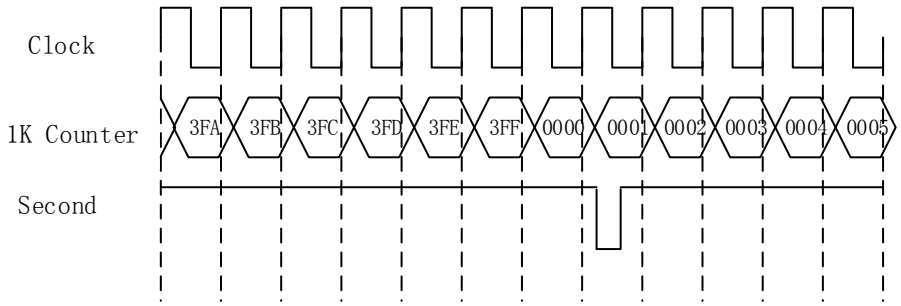

Figure 3- 37. RTC Counter

The 1K counter adds 1 on each rising edge of the clock. When the clock number reaches 0x3FF, 1K counter starts to count again from 0, and the second counter adds 1. The step structure of 1KHz counter is as follows.

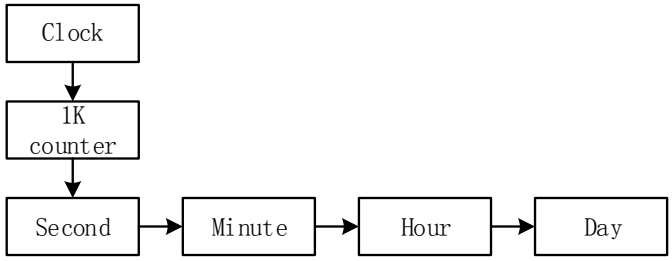

Figure 3- 38. RTC 1KHz Counter Step Structure

According to above implementation, the changing range of each counter is as follows.

Table 3- 11. RTC Counter Changing Range

Counter	Range
Second	0~59
Minute	0~59
Hour	0~23
Day	0~65535 (The year, month, day need be transformed by software according to day counter)

CAUTION

Because there is no error correction mechanism in the hardware, note that each counter configuration should not exceed a reasonable counting range.

3.13.3.4.3. Alarm 0

The principle of alarm0 is a comparator. When RTC timer reaches scheduled time, the RTC generates the interrupt.
The RTC only generates one interrupt when RTC timer reached the scheduled day, hour, minute and second counter, then the RTC needs set a new scheduled time, the next interrupt can be generated.

3.13.3.4.4. Power-off Storage

The RTC provides sixteen 32-bit general purpose register to store power-off information.
When the system is in shutdown or standby scene, CPU can judge software process by the storing information.

3.13.3.4.5. RTC_VIO

The RTC module has a LDO, the input source of the LDO is VCC_RTC, the output of the LDO is RTC_VIO. But VCC_RTC of H616 package is connected internally to VCC_PLL, and RTC_VIO is not available in H616 package.

3.13.3.4.6. RC Calibration

The basic circuit of RC calibration is shown in Figure 3-39. Whether to output the calibrated RC clock can be selected by the RC_Cali_SEL control bit, the calibration principle is as follows.

As shown in Figure 3-40, with HOSC(24M) as the reference clock, calculate the counter number M of RC clock within 1ms/16ms/128ms to obtain the accurate frequency of internal RC. By dividing the accurate frequency by 32.768 kHz, the frequency divider(K) from RC clock to 32.768 kHz is obtained. Lastly, RC16M is divided into 32.768 kHz frequency by the frequency divider(K).

NOTE

The calibration principle is output 32.768 kHz, not input 16 MHz.

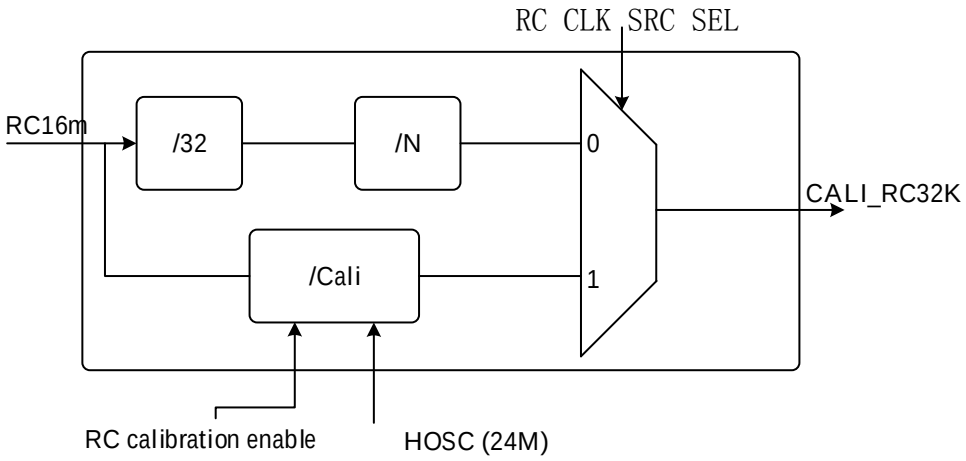

Figure 3- 39. Calibration Circuit

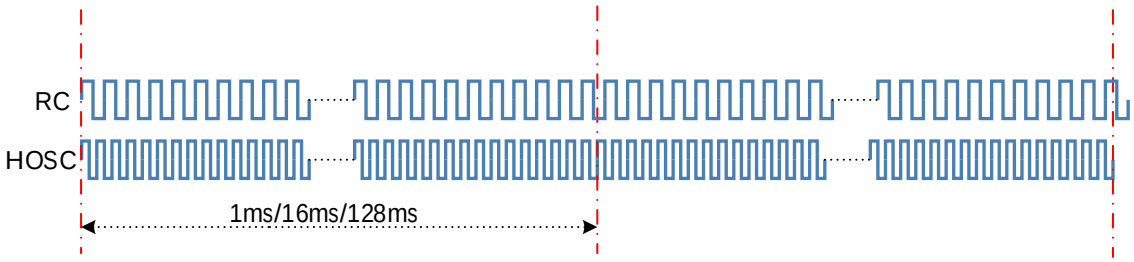

Figure 3- 40. RC and HOSC Waveform

3.13.3.4.7. DCXO Timed Wakeup

The logic of DCXO timed wakeup circuit is relatively simple, including two controls: timed wakeup hardware automatic enable and timed wakeup time length (software configuration). The timed wakeup means that DCXO circuit is required to wakeup the output clock once every second(1s~60s, usually the ambient temperature changes little in a few seconds) for 32K calibration in the super standby or shutdown scenario,after calibration, DCXO circuit is closed, the closed time is timed wakeup time length(software configuration).The time of DCXO circuit from wakeup starting to stable output is 3~4ms. Although the timed wakeup function is closed, DCXO circuit always had worked. The process of timed wakeup is shown in Figure 3-41.

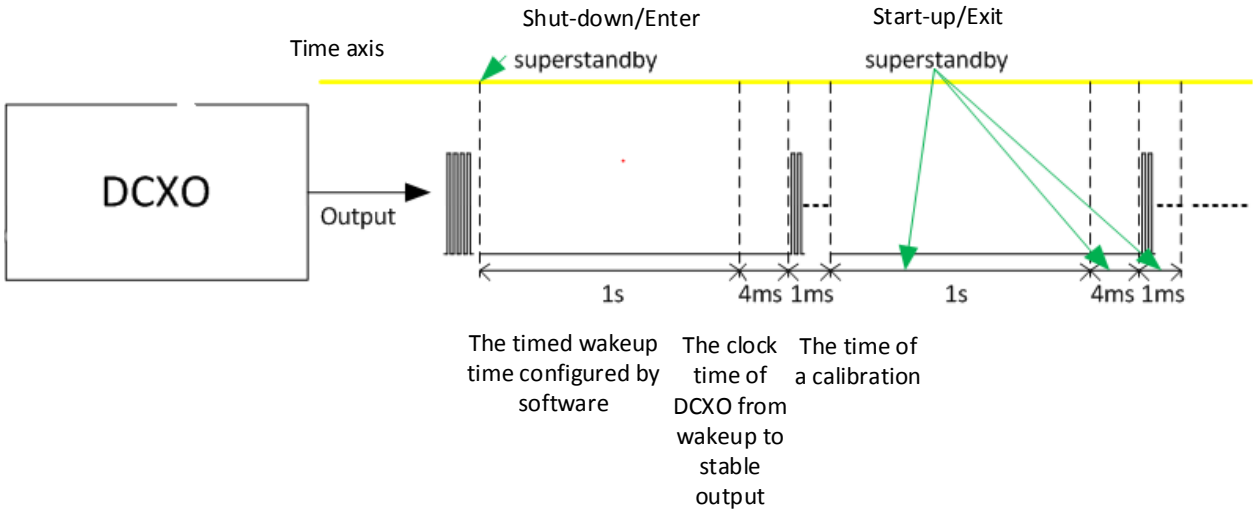

Figure 3- 41. DCXO Timed Wakeup Waveform

The time of a calibration in shutdown or super standby: the timed wakeup time configured by software + the clock time of DCXO from wakeup to stable output + the time of a calibration. The timed wakeup time configured by the software in the figure is 1s, and can be configured by software in application. It is the theoretical maximum value for DCXO from wakeup to stable output clock in 4ms , the specific value is subject to IC measured results. In the any time of these three periods, the startup or exit of the super standby action will not cause DCXO abnormal.

The enable signal of DCXO and the enable signal of timed wakeup DCXO is “OR” logic, and they do not contradict each other.

The interval between continuous DCXO enable operation and disable operation is at least greater than 4us.

3.13.3.5. Operating Mode

3.13.3.5.1. RTC Calendar

(1) Write time initial value: Write the current time to **RTC_DAY_REG** and **RTC_HH_MM_SS_REG**.
(2) After configured time, read the bit[8:7] of **LOSC_CTRL_REG** to ensure that configuration is completed.
(3) After update time, the RTC restarts to count again. The software can read the current time anytime.

NOTE

The RTC can only provide day counter, so the current day counter need be converted to year, month, day and week by software.

After configured time at each time, you need ensure the bit[8:7] of LOSC_CTRL_REG is 0 before the next setting is performed.

3.13.3.5.2. Alarm0

(1) Enable alram0 interrupt by writing **ALARM0_IRQ_EN**.
(2) Set the counter comparator, write the count-down day, hour, minute, second number to **ALARM0_DAY_REG** and **ALARM0_HH_MM_SS_REG**.
(3) Enable alarm0 function by writing **ALARM0_ENABLE_REG**, then the software can query alarm count value in real time by **ALARM0_DAY_REG** and **ALARM0_HH_MM_SS_REG.** When the setting time reaches, **ALARM0_IRQ_STA_REG** is set to 1 to generate interrupt.
(4) After enter the interrupt process, write **ALARM0_IRQ_STA_REG** to clear the interrupt pending, and execute the interrupt process.
(5) Resume the interrupt and continue to execute the interrupted process.

3.13.3.5.3. Fanout

Set the bit0 of **32K_FANOUT_GATING_REG** to 1, and ensure that external pull-up resistor and voltage are normal, then 32.768 kHz fanout square wave can be output.

3.13.3.5.4. DRAM Data Encrypt

If using DRAM data encrypt, the DRAM data read by CPU is the encrypted data. The steps are as follows.
Before write/read **CRY_KEY_REG** and **CRY_EN_REG**, the bit[15:0] of **CRY_CONFIG_REG** should be written to 0x1689.

NOTE

Note that this step needs to be performed before each read and write operation, otherwise the register operation is not successful.

3.13.3.5.5. RC Calibration Usage Scenario

- Power-on: Select non-accurate 32K clock divided by internal RC.
- Normal scenario: Select 32kHz clock divided by 24 MHz.
- Standby or power-off scenario: Use external calibration clock 32kHz.

3.13.4. Programming Guidelines

3.13.4.1. Real Time Clock

For example: set time-- 21’, 07:08:09
RTC_DAY_REG = 0x00000015;
RTC_HH_MM_SS_REG = 0x00070809; //0000 0000 000|0 0000(Hour) 00|00 0000(Minute) 00|00 0000(Second)

Read (RTC_DAY_REG);
Read (RTC_HH_MM_SS_REG);

3.13.4.2. Alarm 0

```
irq_request(GIC_SRC_R_Alarm0, Alm0_handler);
irq_enable(GIC_SRC_R_Alarm0);
writel(1, ALARM0_DAY_SET_REG);
writel(1, RTC_HH_MM_SS_REG);          //set 1 second corresponding to normal mode;
writel(1, ALM0_EN);
writel(1, ALM_CONFIG);          //NMI output
while(!readl(ALM0_IRQ_STA));
writel(1, ALM0_IRQ_EN);
while(readl(ALM0_IRQ_STA));
```

3.13.5. Register List

Module Name	Base Address
RTC	0x07000000

Register Name	Offset	Description
LOSC_CTRL_REG	0x0000	Low Oscillator Control Register
INTOSC_CLK_PRESCAL_REG	0x0008	Internal OSC Clock Prescalar Register
INTOSC_CLK_AUTO_CALI_REG	0x000C	Internal OSC Clock Auto Calibration Register
RTC_DAY_REG	0x0010	RTC Year-Month-Day Register
RTC_HH_MM_SS_REG	0x0014	RTC Hour-Minute-Second Register
ALARM0_COUNTER_REG	0x0020	Alarm 0 Counter Register
ALARM0_CUR_VLU_REG	0x0024	Alarm 0 Counter Current Value Register
ALARM0_ENABLE_REG	0x0028	Alarm 0 Enable Register
ALARM0_IRQ_EN	0x002C	Alarm 0 IRQ Enable Register
ALARM0_IRQ_STA_REG	0x0030	Alarm 0 IRQ Status Register
32K_FANOUT_GATING_REG	0x0060	32k Fanout Output Gating Register
GP_DATA_REG	0x0100 + N*0x04	General Purpose Register (N=0~15)
DCXO_CTRL_REG	0x0160	DCXO Control Register
RTC_VIO_REG	0x0190	RTC_VIO Regulate Register
IC_CHARA_REG	0x01F0	IC Characteristic Register
VDDOFF_GATING_SOF_REG	0x01F4	VDD To RTC Isolation Software Control Register
SP_STDBY_FLAG_REG	0x01F8	Super Standby Flag Register
SP_STDBY_SOFT_ENTRY_REG	0x01FC	Super Standby Software Entry Register
USB_STBY_CTRL_REG	0x0200	USB Standby Control Register
EFUSE_HV_PWRSWT_CTRL_REG	0x0204	Efuse High Voltage Power Switch Control Register

CRY_CONFIG_REG	0x0210	Crypt Configuration Register
CRY_KEY_REG	0x0214	Crypt Key Register
CRY_EN_REG	0x0218	Crypt Enable Register

3.13.6. Register Description

3.13.6.1. 0x0000 LOSC Control Register (Default Value: 0x0000_4010)

Offset:0x0000			Register Name: LOSC_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	W	0x0	Reserved
15	R/W	0x0	Reserved
14	R/W	0x1	Reserved
13:9	/	/	/
8	R/W	0x0	RTC_HHMMSS_ACCE RTC HH-MM-SS access After writing the RTC HH-MM-SS Register, this bit is set and it will be cleared until the real writing operation is finished. After writing the RTC HH-MM-SS Register, the RTC HH-MM-SS Register will be refreshed for at most one second.
7	R/W	0x0	RTC_DAY_ACCE RTC DAY access After writing the RTC DAY register, this bit is set and it will be cleared until the real writing operation is finished. After writing the RTC DAY register, the DAY register will be refreshed for at most one second.
6:5	/	/	/
4	R/W	0x1	Reserved
3:2	R/W	0x0	Reserved
1	/	/	/
0	R/W	0x0	Reserved

NOTE

If the bit[8:7] of LOSC_CTRL_REG is set, the RTC HH-MM-SS, DD and ALARM DD-HH-MM-SS register cannot be written.

3.13.6.2. 0x0008 Internal OSC Clock Prescalar Register (Default Value: 0x0000_000F)

Offset:0x0008			Register Name: INTOSC_CLK_PRESCAL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4:0	R/W	0xF	INTOSC_32K_CLK_PRESCAL.

			Internal OSC 32K Clock Prescalar value N. The clock output = Internal RC/32/N. 00000: 1 00001: 2 00002: 3 ..... 11111: 32

3.13.6.3. 0x000C Internal OSC Clock Auto Calibration Register (Default Value: 0x01E8_0000)

Offset:0x000C			Register Name: INTOSC_CLK_AUTO_CALI_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	RO	0x1e8	32k calibration integer divider factor
21:5	RO	0x0	32k calibration decimal divider factor
4	R/W	0x0	Calibration function Clk16M_RC_enable 0: Auto gating 1: Soft bypass
3:2	R/W	0x0	RC Calibration Precise Selection 00: 1ms calibration precise 01: 16ms calibration precise 10: 128ms calibration precise
1	R/W	0x0	RC Calibration Enable 0: Close Calibration circuit 1: Open Calibration circuit
0	R/W	0x0	RC CLK SRC SEL Select the RTC 32k clock source from normal RC or Calibrated RC 0: Normal RC 1: Calibrated RC

3.13.6.4. 0x0010 RTC DAY Register

Offset:0x0010			Register Name: RTC_DAY_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	UDF	DAY Day Range from 0~65535.

NOTE

Ensure that the bit[7] of LOSC_CTRL_REG is 0 before updating RTC_DAY_REG.

3.13.6.5. 0x0014 RTC HH-MM-SS Register

Offset:0x0014			Register Name: RTC_HH_MM_SS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20:16	R/W	UDF	HOUR Range from 0~23
15:14	/	/	/
13:8	R/W	UDF	MINUTE Range from 0~59
7:6	/	/	/
5:0	R/W	UDF	SECOND Range from 0~59

NOTE

Ensure that the bit[8] of LOSC_CTRL_REG is 0 before updating RTC_HH_MM_SS_REG.

3.13.6.6. 0x0020 Alarm 0 Day Setting Register (Default Value: 0x0000_0000)

Offset:0x0020			Register Name: ALARM0_COUNTER_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0	ALARM0_COUNTER Alarm 0 Counter is based on Day.

3.13.6.7. 0x0024 Alarm 0 HH-MM-SS Setting Register

Offset:0x0024			Register Name: ALARM0_CUR_VLU_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20:16	R/W	x	HOUR Range from 0~23
15:14	/	/	/
13:8	R/W	x	MINUTE Range from 0~59
7:6	/	/	/
5:0	R/W	x	SECOND Range from 0~59

3.13.6.8. 0x0028 Alarm 0 Enable Register (Default Value: 0x0000_0000)

Offset:0x0028			Register Name: ALARM0_ENABLE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	ALM_0_EN Alarm 0 Enable If this bit is set to "1", the valid bits of Alarm 0 Counter Register will down count to zero, and the alarm pending bit will be set to "1". 0: Disable 1: Enable

3.13.6.9. 0x002C Alarm 0 IRQ Enable Register (Default Value: 0x0000_0000)

Offset:0x002C			Register Name: ALARM0_IRQ_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	ALARM0_IRQ_EN Alarm 0 IRQ Enable 0: Disable 1: Enable

3.13.6.10. 0x0030 Alarm 0 IRQ Status Register (Default Value: 0x0000_0000)

Offset:0x0030			Register Name: ALARM0_IRQ_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W1C	0x0	ALARM0_IRQ_PEND Alarm 0 IRQ Pending bit 0: No effect 1: Pending, alarm 0 counter value is reached If alarm 0 irq enable is set to 1, the pending bit will be sent to the interrupt controller.

3.13.6.11. 0x0060 32K FANOUT Gating Register (Default Value: 0x0000_0000)

Offset:0x0060			Register Name: 32K_FANOUT_GATING_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	HOSC_32K_DIVIDER_ENABLE 1: enable the hosc 24m to 32k divider circuit

			0: disable the hosc 24m to 32k divider circuit
15:3	/	/	/
2:1	R/W	0x0	LOSC_OUT_SRC_SEL 00:RTC_32K(select by RC_CLK_SRC_SEL) 01: Peripll divided 32K 10: HOSC divided 32K
0	R/W	0x0	32K_FANOUT_GATING Configuration of 32k output, and no 32k output by default. 0: Mask LOSC output gating 1: Enable LOSC output gating

3.13.6.12. 0x0100+N*0x0004 General Purpose Register (Default Value: 0x0000_0000)

Offset:0x0100+N*0x0004 (N=0~15)			Register Name: GP_DATA_REGN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	GP_DATA Data [31:0]

NOTE

General purpose register 0~15 value can be stored if the RTC-VIO is larger than 0.7V.

3.13.6.13. 0x0160 DCXO Control Register (Default Value: 0x083F_10F3 or 0x083F_F0FC)

Offset:0x0160			Register Name: DCXO_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	DCXO_FANOUT_ENB 0: enable DCXO wake up function 1: disable DCXO wake up function
30:28	/	/	/
27:24	R/W	0x8	DCXO_ICTRL DCXO current control value
23	/	/	/
22:16	R/W	0x3F	DCXO_TRIM DCXO cap array value Capacity cell is 55fF
15:13	/	/	/
12:8	R/W	0x10	DCXO_BG DCXO bandgap output voltage
7	R/W	0x1	DCXO_LDO_INRUSHB DCXO LDO driving capacity signal, active high
6	R/W	0x1	XTAL_MODE Xtal mode enable signal, active high

			0: For external clk input mode 1: For normal mode
5:4	R/W	0x3	DCXO_RFCLK_ENHANCE DCXO rfclk enhance Enhance driving capacity of output OUT_RF_REFCLK, 0x0 for 5pF, 0x1 for 10pF, 0x2 for 15pF, 0x3 for 20pF.
3:2	/	/	/
1	R/W	0x1	DCXO_EN DCXO enable 1: Enable 0: Disable
0	R/W	0x1	CLK16M_RC_EN 1: Enable 0: Disable The related register configuration is necessary to ensure the reset debounce circuit has a stable clock source. The first time SoC starts up, by default, the reset debounce circuit of SoC uses 32K divided by RC16M. In power-off, software reads the related bit to ensure whether EXT32K is working normally, if it is normal, first switch the clock source of debounce circuit to EXT32K, then close RC16M. Without EXT32K scenario or external RTC scenario, software confirms firstly whether EXT32K is working normally before switching, or software does not close RC16M.

3.13.6.14. 0x0164 Calibration Control Register (Default Value: 0x0000_0043)

Offset:0x0164			**Register Name: CALI_CTRL_REG**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	WAKEUP_DCXO_EN Wake up DCXO circuit enable.
30:17	/	/	/
16	R/W	0x0	WAKEUP_READY_SLEEP_MODE Calibration wake up ready sleep mode, it must be set before the WAKEUP_DCXO_EN is set to 1. 0: Disable 1: Enable
15:12	R/W	0x0	TIMER FOR READY SLEEP Total timer for ready sleep 0x00: 15s 0x01: 30s 0x02: 45s 0x03: 60s 0x04: 90s 0x05: 120s

			0x06: 150s Others: /
11:8	R/W	0x0	WAKEUP_CNT FOR READY SLEEP Wake up counter for ready sleep 0x00: 250ms 0x01: 500ms 0x02: 750ms 0x03: 1s 0x04: 1.25s 0x05: 1.5s 0x06: 1.75s 0x07: 2s 0x08: 2.25s 0x09: 2.5s 0x0A: 2.75s 0x0B: 3s 0x0C: 3.25s 0x0D: 3.5s 0x0E: 3.75s 0x0F: 4s
7:4	R/W	0x4	WAKEUP_CNT FOR SLEEP Wake up counter for sleep 0x00: 250ms 0x01: 500ms 0x02: 1s 0x03: 10s 0x04: 60s 0x05: 120s 0x06: 180s 0x07: 240s 0x08: 300s 0x09: 360s 0x0A: 420s 0x0B: 480s 0x0C: 540s 0x0D: 600s 0x0E: 1200s 0x0F: 1800s
3:0	R/W	0x3	WAIT DCXO SEL Select for DCXO active after DCXO enable 0x0: 1ms 0x1:2ms 0x2:3ms 0x3:4ms ...

			0xF: 16ms

3.13.6.15. 0x0190 RTC_VIO Regulation Register (Default Value: 0x0000_0004)

Offset:0x0190			Register Name:RTC_VIO_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4	R/W	0x0	V_SEL 0: resistance divider 1: band gap
3	/	/	/
2:0	R/W	0x4	RTC_VIO_REGU These bits are useful for regulating the RTC_VIO from 0.6V to 1.3V, and the regulation step is 0.1V. 000: 1.0V 001: 0.6V (the configuration can cause RTC reset) 010: 0.7V 011: 0.8V 100: 0.9V 101: 1.1V 110: 1.2V 111: 1.3V RTC-VIO is provided power for RTC digital part, the default value is 0.9V. After power-on, software sets the field to 0.8V to save power-consumption.

3.13.6.16. 0x01F0 IC Characteristic Register (Default Value: 0x0000_0000)

Offset:0x01F0			Register Name: IC_CHARA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	IC_CHARA Key Field Should be written at value 0x16AA. Writing any other value in this field aborts the write operation.
15:0	R/W	0x0	ID_DATA Return 0x16AA only if the KEY_FIELD is set as 0x16AA when read those bits, otherwise return 0x0.

3.13.6.17. 0x01F4 VDD To RTC Isolation Software Control Register (Default Value: 0x0000_0000)

Offset:0x01F4			Register Name: VDDOFF_GATING_SOF_REG
Bit	**Read/Write**	**Default/Hex**	**Description**

31:16	W	0x0	KEY_FIELD Key Field This field should be filled with 0x16AA, and then the bit15 can be configured.
15	WAC	0x0	When use vdd_sys to RTC isolation software control, write this bit to 1, it will only be cleared by resetb release.
14:1	/	/	/
0	R/W	0x0	DRAM_CH_PAD_HOLD Hold the pad of DRAM channel 0:not hold 1:hold dram Pad This bit should be set to 1 before VDD_SYS power off while it should be set to 0 after the VDD_SYS power on.

3.13.6.18. 0x01F8 Super Standby Flag Register (Default Value: 0x0000_0000)

Offset:0x01F8			Register Name: SP_STDBY_FLAG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	SP_STDBY_FLAG Key Field Any value can be written and read back in the key field, but if the values are not appropriate, the lower 16 bits will not change in this register. Only fellow the appropriate process, the super standby flag can be written in the lower 16 bits.
15:0	R/W	0x0	SUP_STANBY_FLAG_DATA When system is turned on, the low 16 bits of the value in the Super Standby Flag Register should be 0x0. If software programmer wants to write correct super standby flag ID in low 16 bits, the high 16 bits should be written with 0x16AA at first. Then, software programmer must write 0xAA16XXXX in the Super Standby Flag Register, the 'XXXX' means the correct super standby flag ID.

3.13.6.19. 0x01FC Super Standby Software Entry Register (Default Value: 0x0000_0000)

Offset:0x01FC			Register Name: SP_STDBY_SOFT_ENTRY_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	CPU software entry register when acting from supper standby.

3.13.6.20. 0x0200 USB Standby Control Register (Default Value: 0x9000_0000)

Offset:0x0200	Register Name: USB_STBY_CTRL_REG

Bit	Read/Write	Default/Hex	Description
31	R/W	0x1	VDD_USB power source sel 1: VDD_USB power from ldo 0: VDD_USB power from pin
30:28	R/W	0x1	USB2 LDO output power level select: 000: 0.87V 001: 0.9V 010: 0.94V 011: 0.97V 100:1.01V 101:1.05V 110:1.08V 111:1.11V
27:26	/	/	/
25	R/W	0x0	RC_CLK_EN_USB 0: Disable 1: Enable
24	R/W	0x0	RC_CLK_SEL_USB SCLK_USBPHY, EHCI_HCLK and OHCI_SCLK Clock Source Select. RC16M clock is selected only in USB standby mode if necessary. 0: SCLK_USBPHY is from OSC24M, EHCI_HCLK and OHCI_HCLK are from Hclk 1: SCLK_USBPHY, EHCI_HCLK and OHCI_SCLK are from RC16M
23:17	/	/	/
16	R/W	0x0	USB POWER OFF GATING Gating the VDD_SYS to VDD_USB signal in USB standby mode. It must be set to 1 before entering USB standby mode and set to 0 when exiting Normal mode. 0: disable 1: enable
15:9	/	/	/
8	R/W	0x0	USB_STBY_IRQ_POWER_OFF_GATING Gating the USB standby irq signal to RTC module in Super Standby mode when USB module is power off. It must be set to 1 in Super Standby mode and must set to 0 in other mode. 0: disable 1: enable
7:6	/	/	/
5	R/W	0x0	SYSTEM STANDBY IRQ OUTPUT GATING Mask the SYS standby irq output to nmi pad when SoC is going to USB standby mode or Super standby mode. 0: disable irq output 1: enable irq output
4	R/W	0x0	USB_STBY_IRQ_OUTPUT_GATING Mask the USB standby irq output to nmi pad. It must be set to 1 in USB standby mode and set to 0 in other mode.

			0: disable irq output 1: enable irq output
3:0	/	/	/

3.13.6.21. 0x0204 EFUSE High Voltage Power Switch Control Register (Default Value: 0x0000_0000)

Offset:0x0204			Register Name: EFUSE_HV_PWRSWT_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	EFUSE_1.8V_POWER_SWITCH_CONTROL 1: open power switch 0: close power switch Before programming efuse, the bit need be set to 1.

3.13.6.22. 0x0210 Crypto Configuration Register (Default Value: 0x0000_0000)

Offset:0x0210			Register Name: CRY_CONFIG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0	KEY_FIELD Key Field If you want to read or write Crypt Key Register/Crypt Enable Register, you should write 0x1689 in these bits.

3.13.6.23. 0x0214 Crypto Key Register (Default Value: 0x0000_0000)

Offset:0x0214			Register Name: CRY_KEY_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	CRY_KEY Crypto Key

3.13.6.24. 0x0218 Crypto Enable Register (Default Value: 0x0000_0000)

Offset:0x0218			Register Name: CRY_EN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	CRY_EN Crypto Enable

Contents

Figures

Chapter 4 Video and Graphics

4.1. DE

The Display Engine(DE) is a hardware composer to transfer image layers from a local bus or a video buffer to the LCD interface. The DE supports four overlay windows to blend, and supports image post-processing in the video channel. The block diagram of DE is shown in Figure 4-1.

The DE has the following features:

- Output size up to 4096 x 2048
- Six configurable alpha blending channels
- Four overlay layers in each channel, and has a independent scaler
- Potter-duff compatible blending operation
- Supports AFBC buffer
- Supports keystone correction
- Input format: semi-planar YUV422/YUV420/YUV411/P010/P210 and planar YUV422/YUV420/YUV411, ARGB8888/XRGB8888/RGB888/ARGB4444/ARGB1555/RGB565
- Frame Packing/Top-and-Bottom/Side-by-Side Full/Side-by-Side Half 3D format data
- Supports 10-bit processing path for HDR video
- Supports SDR/HDR10/Hybrid-log gamma EOTF and color space conversion
- Supports SmartColor™ 3.3 for excellent display experience
  - Adaptive detail/edge enhancement
  - Adaptive color enhancement and fresh tone protection
  - Adaptive contrast enhancement
  - Adaptive de-noising for compression noise or mosquito noise with yuv420/422 input
- Supports write back only for high efficient dual display and miracast
- Supports output format YUV444/YUV422/YUV420/RGB444 for 10/8bit
- Supports Register Configuration Queue for register update function

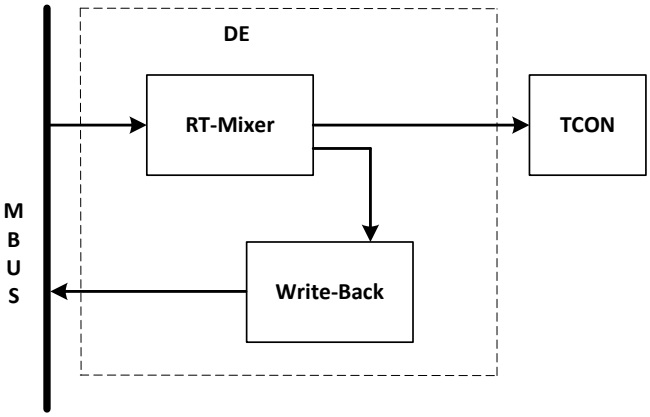

Figure 4- 1. DE Block Diagram

4.2. DI

The De-interlacer300 (DI300) is a module which provides de-interlacing functions. It is an off-line processing module which reading input frame buffer and writing output frame buffer by memory bus. In this version of DI, it also provides Temporal Noise Reduction function to reduce the random noise. And the new-add Film Mode Detection function can detect the pull-down content from video and recover the film with maximum details.

- Supports off-line processing mode only
- Supports 8-bit NV12/NV21/YV12 and planar YUV422/planar YUV422 UV-combined input data format
- Supports 8-bit NV12/NV21/YV12 and planar YUV422/planar YUV422 UV-combined output data format for DIT, and YV12/planar YUV422 output data format for TNR
- Supports video resolution from 32x32 to 2048x1280 pixel
- Supports weave/pixel-motion-adaptive de-interlace method
- Supports temporal noise reduction function
- Supports film mode detection with video-on-film detection
- Performance: module clock 150 MHz for 1080p@60Hz

4.3. G2D

The Graphic 2D(G2D) Engine is hardware accelerator for 2D graphic.

The G2D has the following features:

- Supports layer size up to 2048x2048 pixels
- Supports input/output formats: YUV422(semi-planar and planar format)/YUV420(semi-planar and planar format) /P010/P210/P410/Y8/ARGB8888/XRGB8888/RGB888/ARGB4444/ARGB1555/ARGB2101010 and RGB565
- Supports horizontal and vertical flip, clockwise 0/90/180/270 degree rotate

4.4. Video Decoding

4.4.1. Overview

The Video Decoding consists of Video Control Firmware(VCF) running on ARM processor and embedded hardware Video Engine(VE). VCF gets the bitstream from topper software, parses bitstream, invokes the Video Engine, and generates the decoding image sequence. The decoder image sequence is transmitted by the video output controller to the display device under the control of the topper software.

The Video Decoding has the following features:

- Supports ITU-T H.265 Main10@Level 5.1
  - Maximum video resolution: 6144 x 4320
  - Maximum decoding rate: 150 Mbit/s, 4K@60fps or 6K@30fps
- Supports VP9 Profile2
  - Maximum video resolution: 8192 x 8192
  - Maximum decoding rate: 60 Mbit/s, 4K@60fps
- Supports AVS2 JiZhun 10-bit
  - Maximum video resolution: 8192 x 8192
  - Maximum decoding rate: 60 Mbit/s, 4K@60fps
- Supports ITU-T H.264 Base/Main/High Profile@Level 4.2
  - Maximum video resolution: 4096 x 2048
  - Maximum decoding rate: 60 Mbit/s, 4K@30fps
- Supports ITU-T H.263 Base Profile
  - Maximum video resolution: 1920 x 1080
  - Maximum decoding rate: 60 Mbit/s, 1080p@60fps
- Supports VP8
  - Maximum video resolution: 1920 x 1080
  - Maximum decoding rate: 60 Mbit/s, 1080p@60fps
- Supports MPEG4 SP/ASP@Level 5
  - Maximum video resolution: 3840 x 2160
  - Maximum decoding rate: 60 Mbit/s, 1080p@60fps
- Supports MPEG2 MP/HL
  - Maximum video resolution: 1920 x 1080
  - Maximum decoding rate: 60 Mbit/s, 1080p@60fps
- Supports MPEG1 MP/HL
  - Maximum video resolution: 1920 x 1080
  - Maximum decoding rate: 60 Mbit/s, 1080p@60fps
- Supports VC1 SP/MP/AP
  - Maximum video resolution: 1920 x 1080
  - Maximum decoding rate: 60 Mbit/s, 1080p@60fps
- Supports xvid

- Maximum video resolution: 1920 x 1080
- Maximum decoding rate: 60 Mbit/s, 1080p@60fps

- Supports Sorenson Spark
  - Maximum video resolution: 1920 x 1080
  - Maximum decoding rate: 60 Mbit/s, 1080p@60fps
- Supports AVS/AVS+ JiZhun
  - Maximum video resolution: 1920 x 1080
  - Maximum decoding rate: 30 Mbit/s, 1080p@60fps
- Supports JPEG HFIF file format
  - Maximum video resolution: 16384 x 16384
  - Maximum decoding rate: 45MPPS

4.4.2. Block Diagram

The functional block diagram of the Video Decoding is as follows.

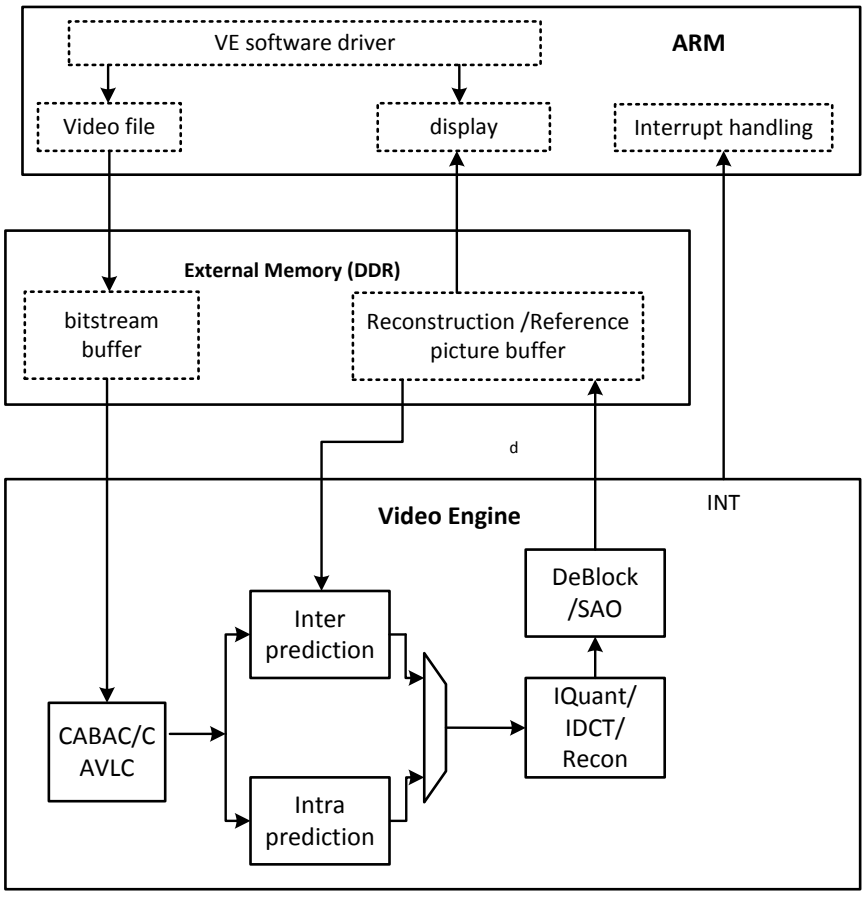

Figure 4- 2. Video Decoding Block Diagram

The Video Engine software driver parses the video file into the corresponding standard video stream, and configures the DDR address of the saved video stream, the DDR address of the reference picture, the DDR address of the reconstructed frame and other necessary information to Video Engine, and starts decoding.

The process of Video Decoding includes reading video stream and parsing syntax, intra-frame prediction, inter-frame prediction, inverse quantization, inverse transform, de-blocking filter, and finally writing the decoded picture into DDR. After the driver software gets the interrupt of Video Decoding, the picture in DDR is sent to the display module.

4.5. Video Encoding

The Video Encoding consists of the video encoding unit(VE) and JPEG encoder(JPGE). The VE supports H.264 encoding, and JPGE supports JPEG/MJPEG encoding.

4.5.1. VE

4.5.1.1. Overview

The VE is a H.264 encoding accelerator implemented by using hardware. It features low CPU usage, short delay and low power consumption.
The VE has the following features:

- Supports ITU-T H.264 high profile/main profile/baseline profile@Level 4.2 encoding
  - Encoding of multiple slice
  - Motion compensation with 1/2 and 1/4 pixel precision
  - Two prediction unit (PU) types of 16x16 and 8x8 for inter-prediction
  - Three prediction unit types of Intra16x16, Intra8x8 and Intra4x4 for intra-prediction
  - Trans4x4 and trans8x8
  - CABAC and CAVLC entropy encoding
  - De-blocking filtering
- Supports the output picture format of semi-planar YCbCr4:2:0
- Supports configurable picture resolutions
- Supports region of interest (ROI) encoding
- Supports on-screen display (OSD) encoding protection that can be enabled or disabled
- Supports OSD front-end overlaying
- Supports three bit rate control modes: constant bit rate (CBR), variable bit rate (VBR) and FIXQP
- Supports the output bit rate ranging from 256 kbit/s to 100 Mbit/s

4.5.1.2. Block Diagram

The functional block diagram of the VE is as follows.

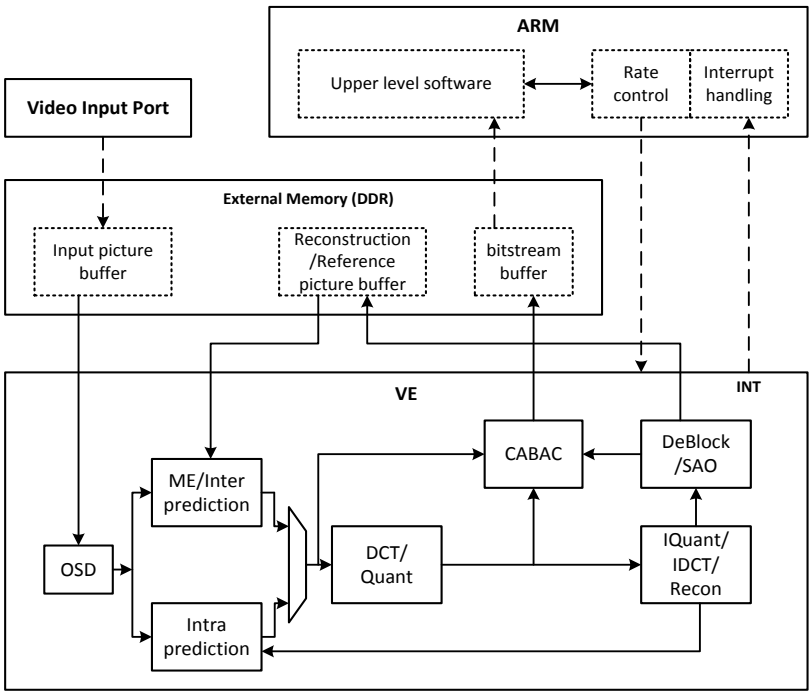

Figure 4- 3. VE Block Diagram

Based on related protocols and algorithms, the VE supports motion estimation/inter-prediction, intra-prediction, transform/quantization, inverse transform/inverse quantization, CABAC encoding/stream generation and DeBlock/SAO. The ARM software controls the bit-rate and handles interrupt.

Before the VE is enabled for video encoding, software allocates three types of buffers mainly in the external DDR SDRAM:

- **Input picture buffer**
  The VE reads the source pictures to be encoded from this buffer during encoding. This buffer is typically written by the Video Input Port module.
- **Reconstruction/Reference picture buffer**
  The VE writes reconstruction pictures to this buffer during encoding. These reconstruction pictures are used as the reference pictures of subsequent pictures. During the encoding of P frames, the reference pictures are read from this buffer.
- **Stream buffer**
  This buffer stores encoded streams. The VE writes streams to this buffer during encoding. This buffer is read by software.

4.5.2. JPGE

4.5.2.1. Overview

The JPGE is a high-performance JPEG encoder implemented by using hardware. It supports 64-megapixel snapshot or HD MJPEG encoding.

The JPGE has the following features:

- Supports ISO/IEC 10918-1 (CCITT T.81) baseline process (DCT sequential) encoding
- Encodes the pictures in the chrominance sampling format of YCbCr4:2:0, YCbCr4:2:2 and YCbCr4:4:4

- Supports multiple input picture formats:
  - Semi-planar YCbCr4:2:0
  - Semi-planar YCbCr4:2:2
  - Semi-planar YCbCr4:4:4
- Supports JPEG encoding with the performance of 1080p@60fps
- Supports configurable picture resolutions
- Supports configurable quantization tables for the Y component, Cb component and Cr component respectively
- Supports OSD front-end overlapping
- Supports the color-to-gray function

4.5.2.2. Block Diagram

The functional block diagram of the JPGE is as follows.

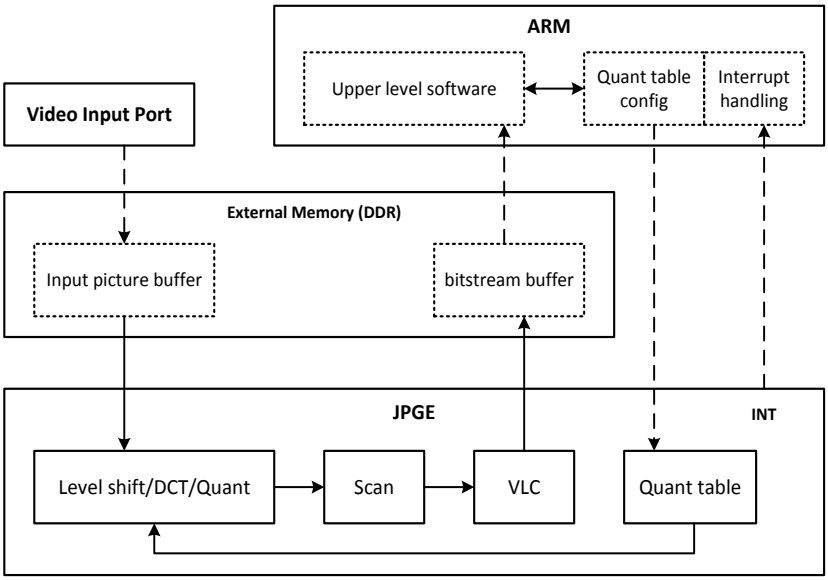

Figure 4- 4. JPGE Block Diagram

The JPGE realizes various protocol processing with large computation such as OSD, level shift, DCT, quantization, scanning, VLC encoding, and stream generation. The ARM software completes the encoding control processing such as quantization table configuration and interrupt processing.

Before the JPGE starts encoding, the software allocates two types of buffers mainly in the external DDR SDRAM:

- **Input picture buffer**

  The JPGE reads the source pictures to be encoded from this buffer during encoding. This buffer is generally written by the Video Input Port module.
- **Stream buffer**

  This buffer stores encoded streams. The JPGE writes streams to this buffer during encoding. This buffer is read by software.

Contents

Figures

Tables

Chapter 5 Memory

5.1. SDRAM Controller(DRAMC)

5.1.1. Overview

The SDRAM Controller (DRAMC) provides a simple, flexible, burst-optimized interface to all in-dusty-standard DDR4/DDR3/DDR3L and Low Power DDR3/4 SDRAM. It supports up to a 32 Gbits memory address space.

The DRAMC automatically handles memory management, initialization, and refresh operations. It gives the host CPU a simple command interface, hiding details of the required address, page, and burst handling procedures. All memory parameters are runtime-configurable, including timing, memory setting, SDRAM type, and Extended-Mode-Register settings.

The SDRAM includes the following features:

- Supports 32-bit one channel
- Supports 2 chip select signals
- Supports DDR4/DDR3/DDR3L/LPDDR3/LPDDR4 SDRAM
- Supports power voltage of different memory device: 1.2V, 1.5V, 1.35V and 1.1V
- Supports memory capacity up to 32 Gbits (4 GB)
- Automatically generates initialization and refresh sequences
- Runtime-configurable parameters setting for application flexibility
- Priority of transferring through multiple ports is programmable
- Random read or write operation is supported

5.2. Nand Flash Controller(NDFC)

5.2.1. Overview

The NDFC is the NAND Flash Controller which supports all NAND flash memory available in the market. New type flash can be supported by software re-configuration.

The On-the-fly error correction code (ECC) is built-in NDFC for enhancing reliability. BCH is implemented and it can detect and correct up to 80 bits error per 1024 bytes data. The on chip ECC and parity checking circuit of NDFC frees CPU for other tasks. The ECC function can be disabled by software.

The data can be transferred by DMA or by CPU memory-mapped IO method. The NDFC provides automatic timing control for reading or writing external Flash. The NDFC maintains the proper relativity for CLE, CE# and ALE control signal lines. There are three different kinds of modes for serial read access, mode0 is for conventional serial access , mode1 is for EDO type and the mode2 is for extension EDO type. NDFC can monitor the status of R/B# signal line.

Block management and wear leveling management are implemented in software.

The NDFC has the following features:

- Supports all SLC/MLC/TLC flash and EF-NAND memory available in the market
- Configure randomize engine seed by using software
- Software configure method for various system and memory types
- Supports 2 chip selects, and 2 ready_busy signals
- Up to 8-bit data bus width
- Supports 1024, 2048, 4096, 8192, 16384, 32768 bytes size per page
- Conventional and EDO serial access method for serial reading Flash
- 80 bits/1 KB On-the-fly BCH code ECC check and error correction
- Output bits number information about corrected error
- ECC automatic disable function for all 0xff data
- NDFC status information is reported by its registers, and interrupt is supported
- One Command FIFO
- Internal DMA controller based on chain-structured descriptor list
- Two 256x32-bit RAM for Pipeline Procession
- Supports SDR, ONFI DDR1.0 , Toggle DDR1.0, ONFI DDR2.0 and Toggle DDR2.0 RAW NAND FLASH
- Maximum IO rate of 50 MHz in SDR mode, and 60 MHz in both DDR1.0 and DDR2.0 mode
- Self-debug for NDFC debug

5.2.2. Block Diagram

The block diagram of the NDFC is shown as follows.

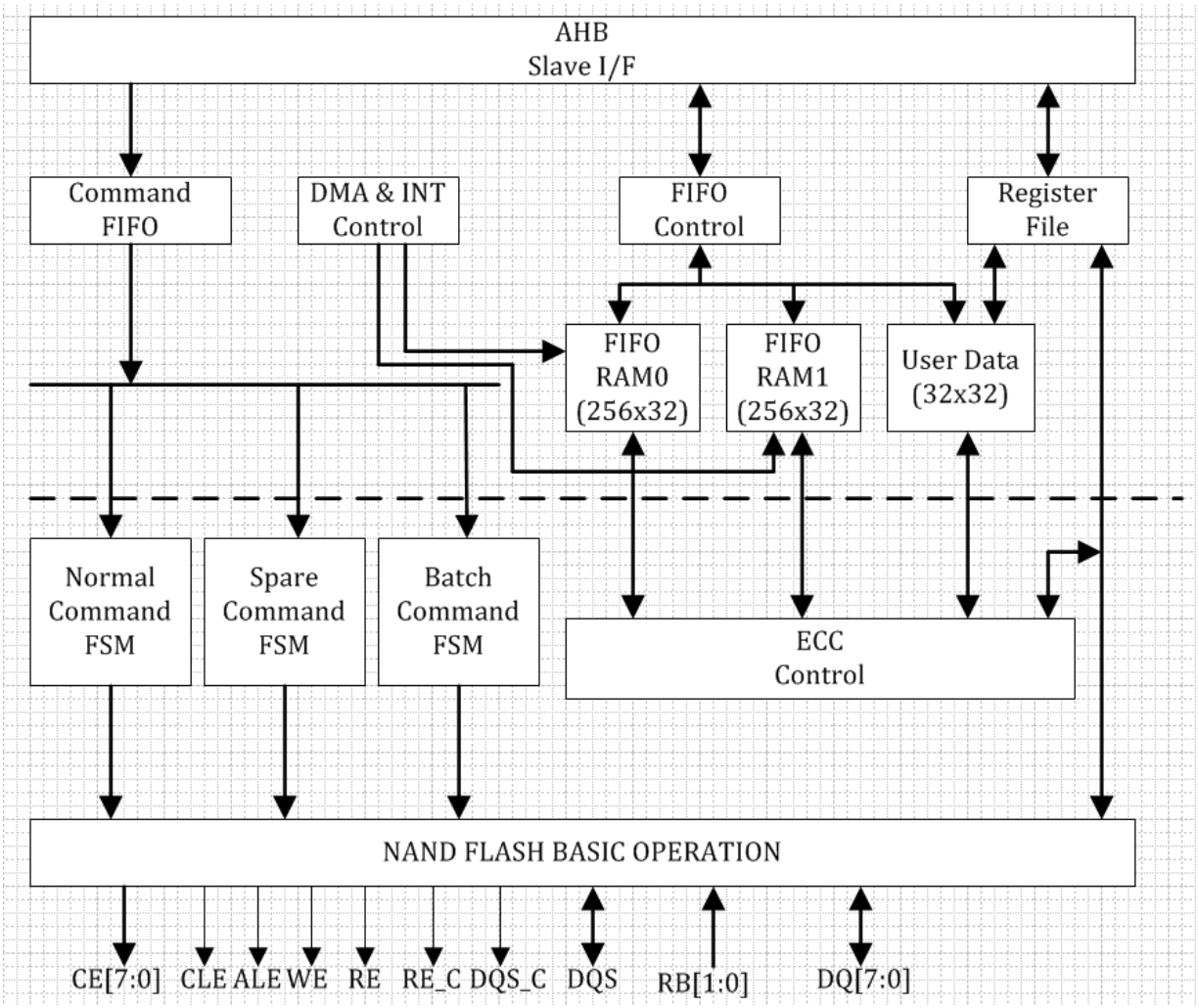

Figure 5- 1. NDFC Block Diagram

5.2.3. Operations and Functional Descriptions

5.2.3.1. External Signals

Table 5-1 describes the external signals of NDFC. DQ0~DQ7 and DQS are bidirectional I/O. WE,ALE,CLE,CE,RE are output pin, RB is input pin. The RB pin in the NAND device is an open-drain driver, which must need a pull-up resistor.

Table 5- 1. NDFC External Signals

Signal	Description	Type
NAND_WE	Write Enable	O
NAND_RE	Read Enable	O
NAND_ALE	Address Latch Enable, High is Active	O
NAND_CLE	Command Latch Enable, High is Active	O
NAND_CE0	Chip Enable, Low is Active	O
NAND_CE1	Chip Enable, Low is Active	O

NAND_RB0	Ready/Busy, Low is Active	I
NAND_RB1	Ready/Busy, Low is Active	I
NAND_DQ0	Data Input / Output	I/O
NAND_DQ1	Data Input / Output	I/O
NAND_DQ2	Data Input / Output	I/O
NAND_DQ3	Data Input / Output	I/O
NAND_DQ4	Data Input / Output	I/O
NAND_DQ5	Data Input / Output	I/O
NAND_DQ6	Data Input / Output	I/O
NAND_DQ7	Data Input / Output	I/O
NAND_DQS	Data Strobe	I/O

5.2.3.2. Clock Sources

To ensure ECC efficiency,ECC engine and NDFC internal logic use different clock. The clock of NDFC internal logic is set by **NAND_0 Clock Register**, the clock of ECC engine is set by **NAND_1 Clock Register** .Note that **NAND_0 Clock Register** set the internal logic clock of NDFC, but the frequency of external Nand Flash device is half of NDFC internal logic clock. That is, if external Nand Flash runs at 40 MHz, then NDFC need set to 80 MHz.

Both ECC engine and NDFC internal logic have five different clock sources. Users can select one of them to make ECC engine or internal logic clock source. Table 5-2 describles the clock sources of NDFC. Users can see CCU in chapter 3.3 for clock setting, configuration and gating information.

Table 5- 2. NDFC Clock Sources

Clock Sources	Description
OSC24M	24 MHz Crystal
PLL_PERIPH0(1X)	Peripheral Clock, default value is 600 MHz
PLL_PERIPH1(1X)	Peripheral Clock, default value is 600 MHz
PLL_PERIPH0(2X)	Peripheral Clock, default value is 1.2 GHz
PLL_PERIPH1(2X)	Peripheral Clock, default value is 1.2 GHz

5.2.3.3. Timing Diagram

Typically, there are two kinds of serial access methods. One method is conventional method which fetching data at the rise edge of NDFC_RE# signal line. Another one is EDO type which fetching data at the next fall edge of NDFC_RE# signal line.

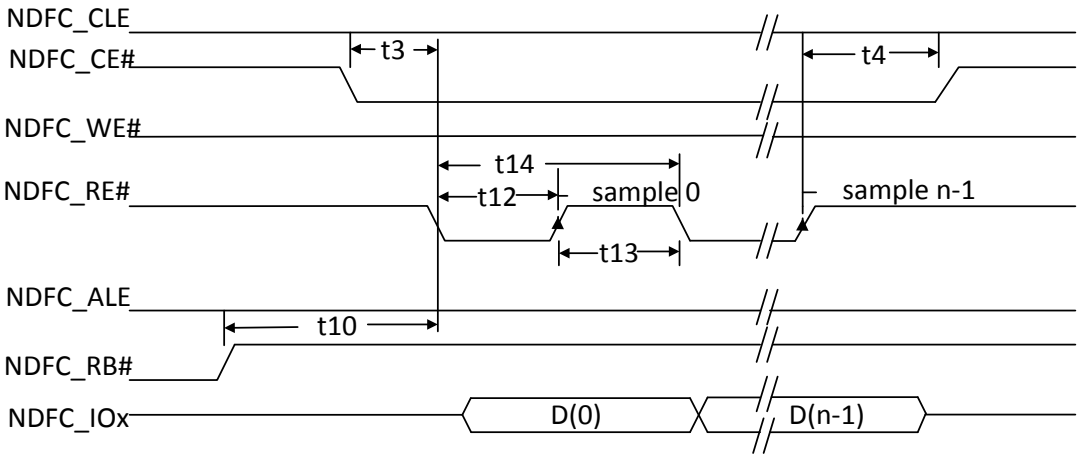

Figure 5- 2. Conventional Serial Access Cycle Diagram (SAM0)

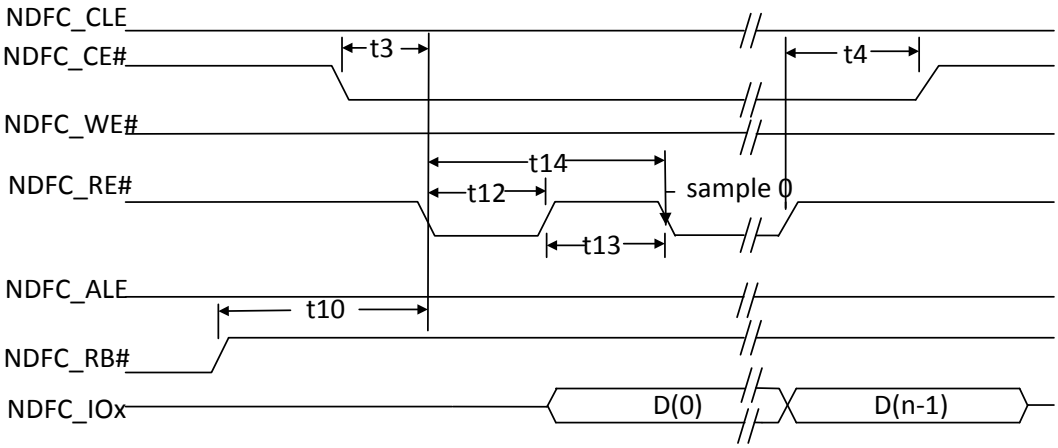

Figure 5- 3. EDO Type Serial Access after Read Cycle (SAM1)

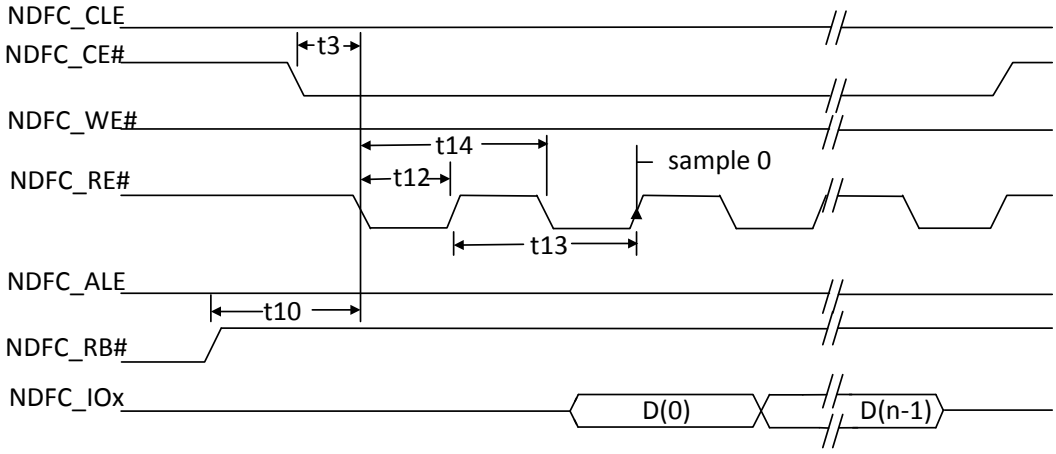

Figure 5- 4. Extending EDO Type Serial Access Mode (SAM2)

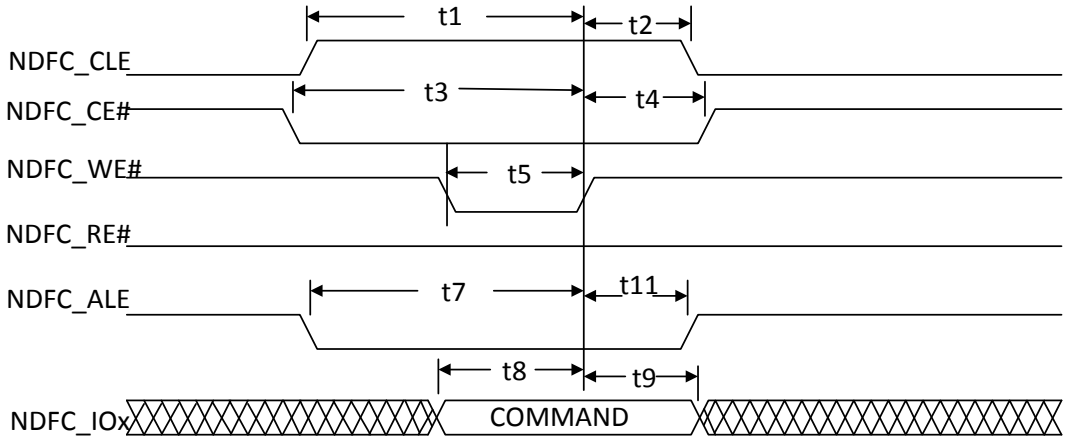

Figure 5- 5. Command Latch Cycle

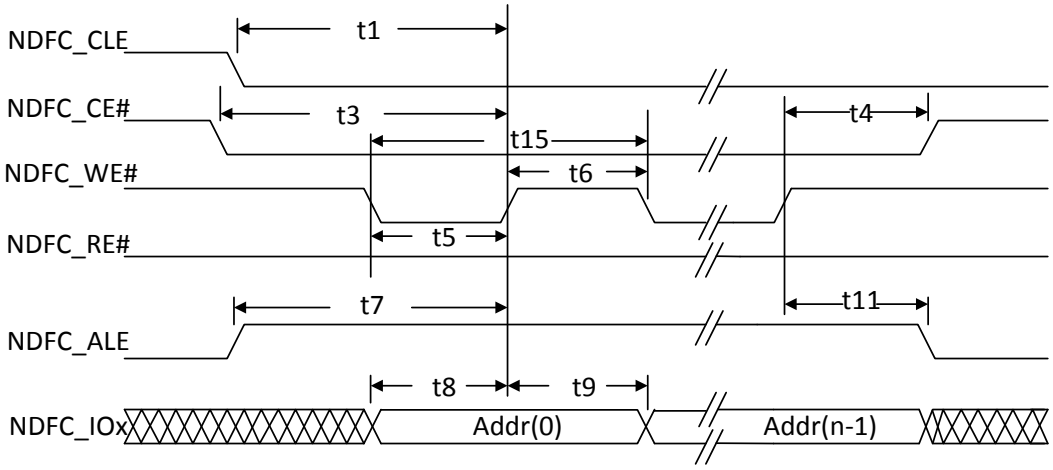

Figure 5- 6. Address Latch Cycle

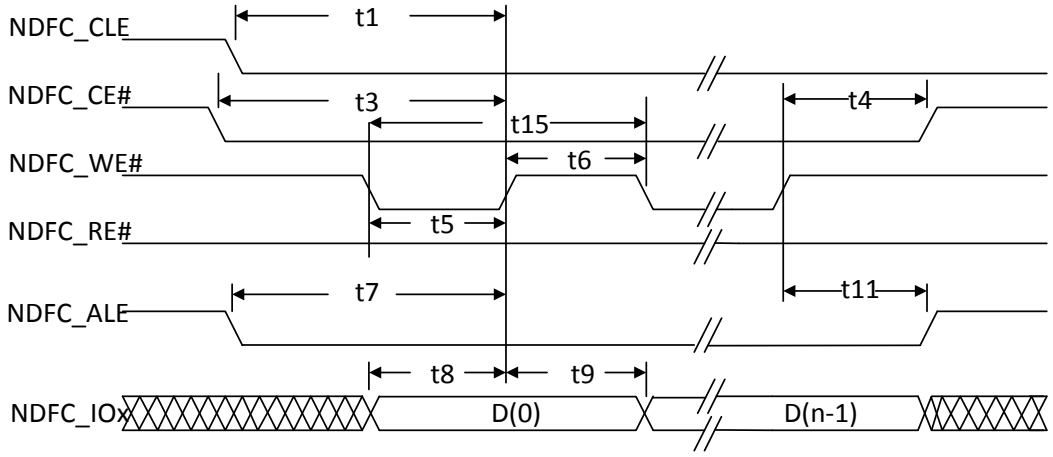

Figure 5- 7. Write Data to Flash Cycle

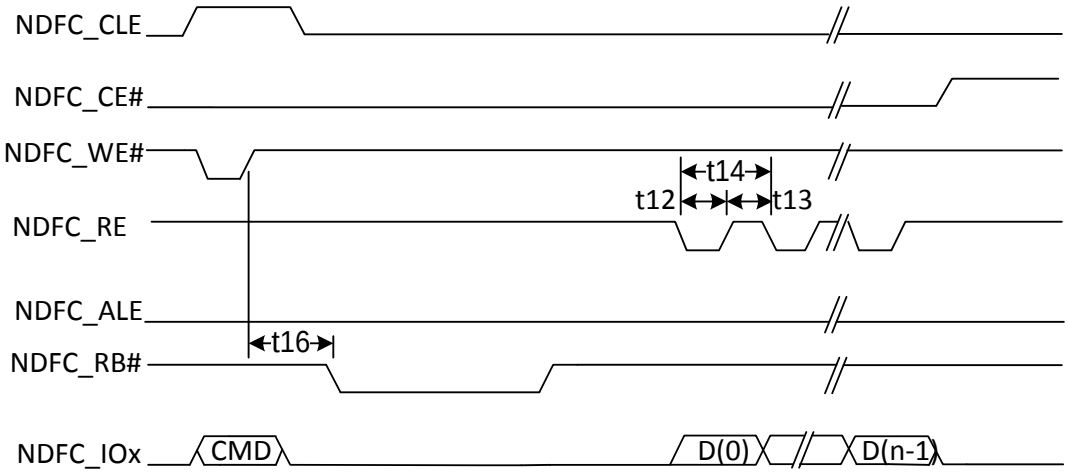

Figure 5- 8. Waiting R/B# Ready Diagram

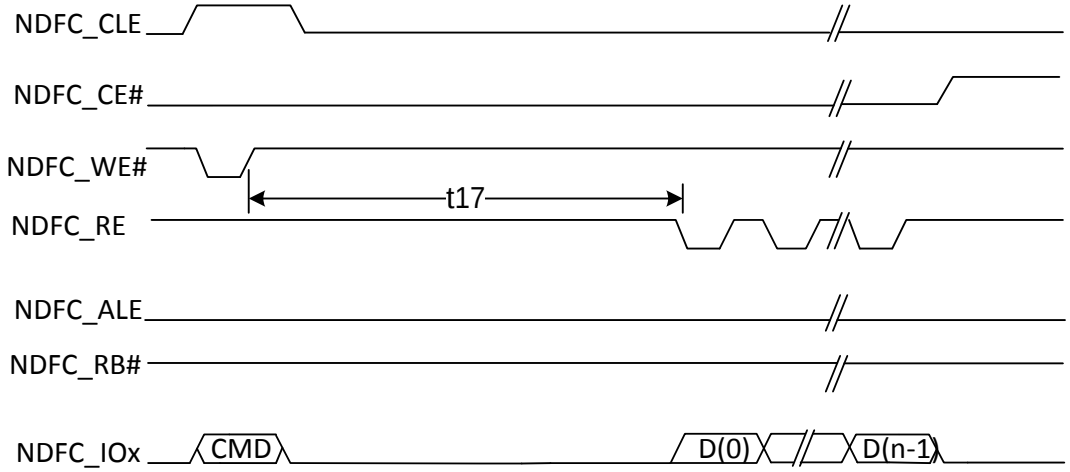

Figure 5- 9. WE# High to RE# Low Timing Diagram

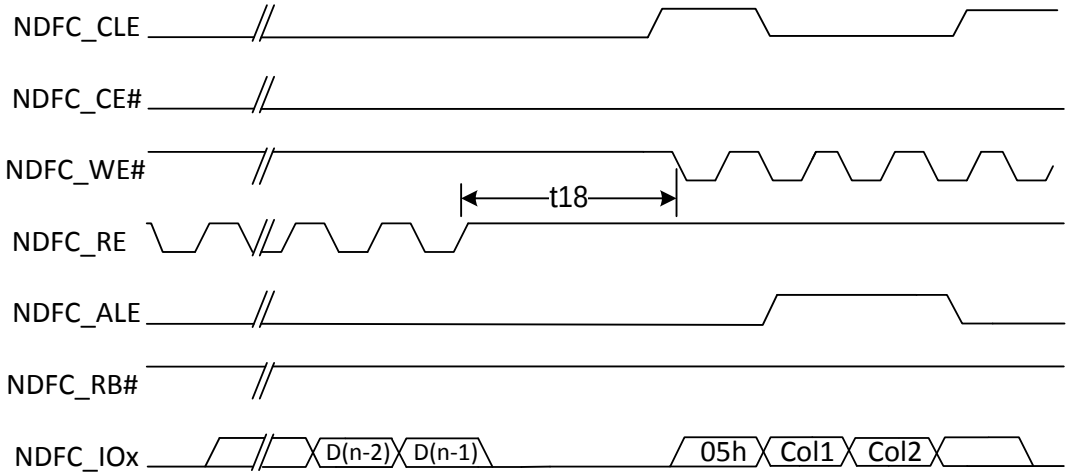

Figure 5- 10. RE# High to WE# Low Timing Diagram

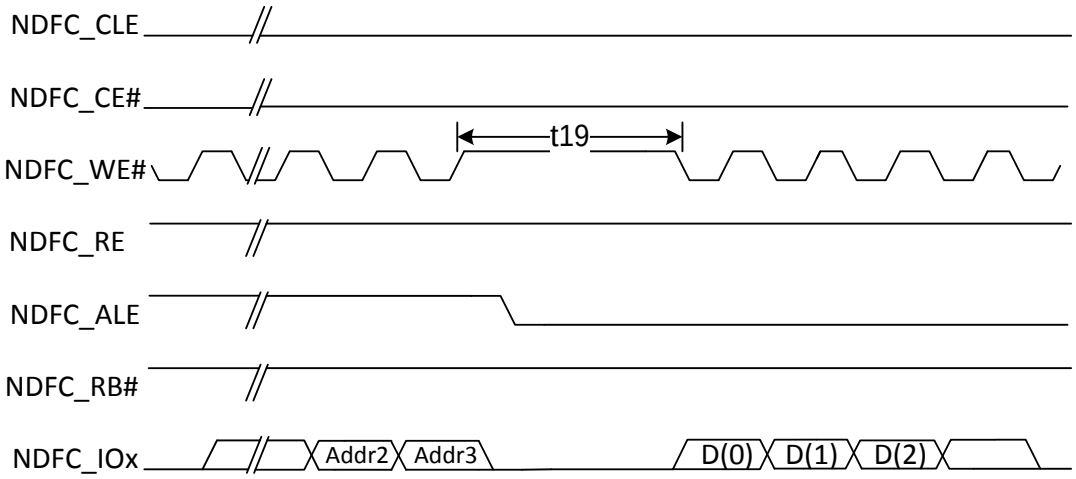

Figure 5- 11. Address to Data Loading Timing Diagram

Timing cycle list:

ID	Parameter	Timing(ns)	Notes
t1	NDFC_CLE setup time	2T	
t2	NDFC_CLE hold time	2T	
t3	NDFC_CE setup time	2T	
t4	NDFC_CE hold time	2T	
t5	NDFC_WE# pulse width	T(1)	
t6	NDFC_WE# hold time	T	
t7	NDFC_ALE setup time	2T	
t8	Data setup time	T	
t9	Data hold time	T	
t10	Ready to NDFC_RE# low	3T	
t11	NDFC_ALE hold time	2T	
t12	NDFC_RE# pulse width	T	
t13	NDFC_RE# hold time	T	
t14	Read cycle time	2T	
t15	Write cycle time	2T	
t16	NDFC_WE# high to R/B# busy	T_WB(2)	Specified by timing configure register (NDFC_TIMING_CFG)
t17	NDFC_WE# high to NDFC_RE# low	T_WHR(3)	Specified by timing configure register (NDFC_TIMING_CFG)
t18	NDFC_RE# high to NDFC_WE# low	T_RHW(4)	Specified by timing configure register (NDFC_TIMING_CFG)
t19	Address to Data Loading time	T_ADL(5)	Specified by timing configure register (NDFC_TIMING_CFG)

Note(1): T is the cycle of the internal clock.

Note(2),(3),(4),(5): These values are configurable in nand flash controller. The value of T_WB could be 14*2T/22*2T/30*2T/38*2T, the value of T_WHR could be 0*2T/6*2T/14*2T/22*2T, the value of T_RHW could be 4*2T/12*2T/20*2T/28*2T, the value of T_ADL could be 0*2T/6*2T/14*2T/22*2T.

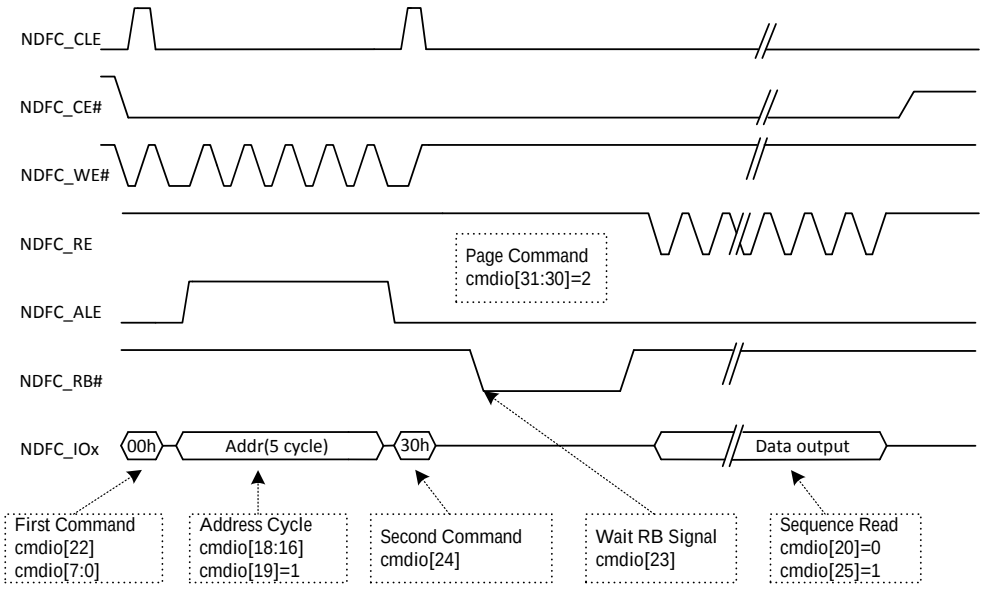

Figure 5- 12. Page Read Command Diagram

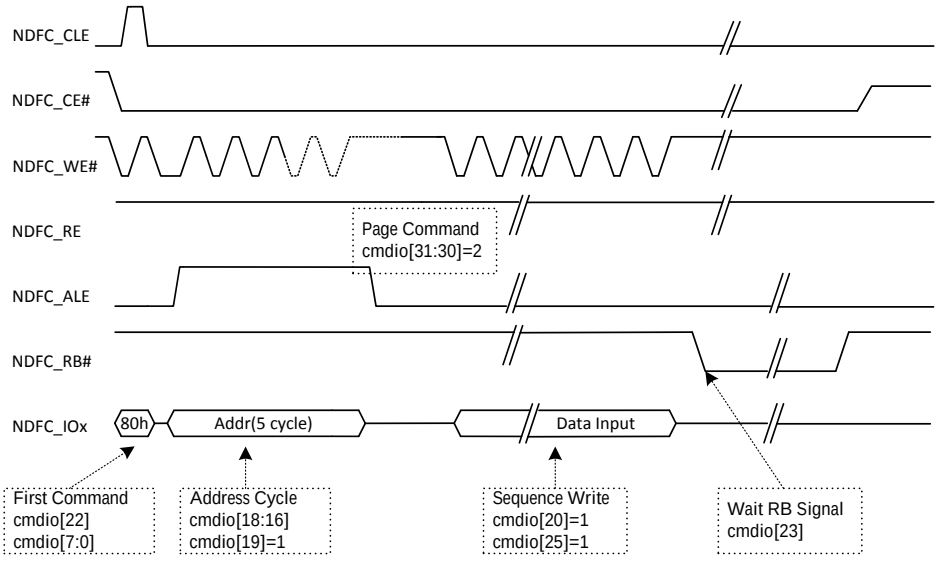

Figure 5- 13. Page Program Diagram

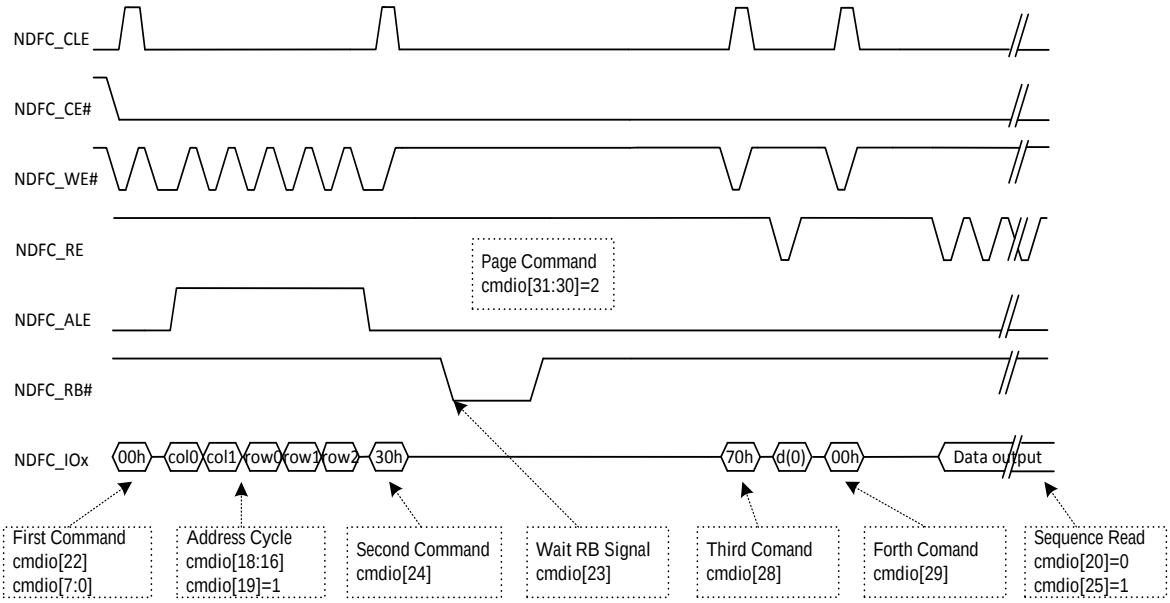

Figure 5- 14. EF-NAND Page Read Diagram

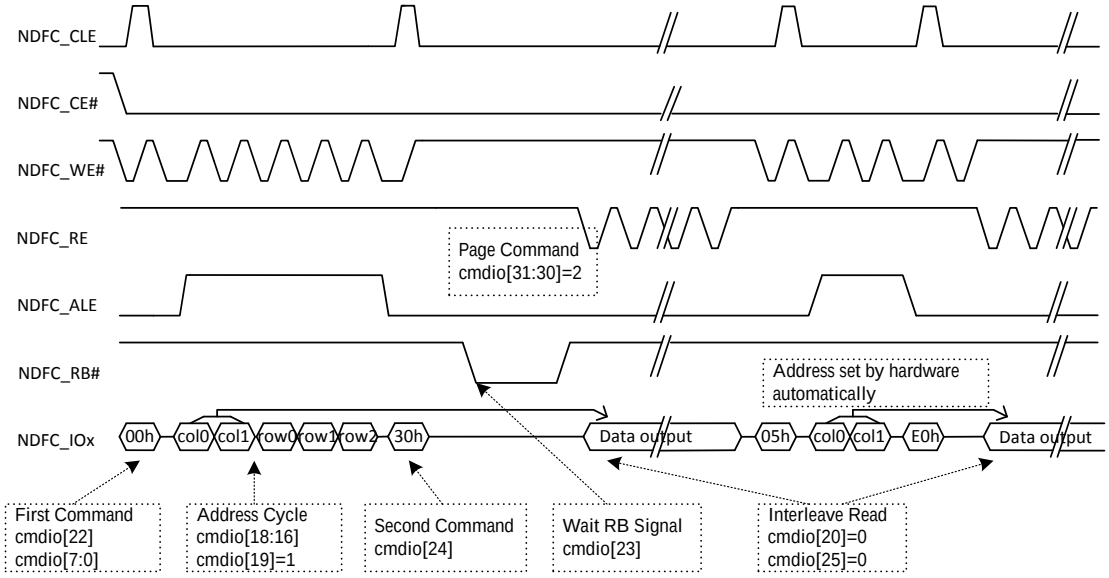

Figure 5- 15. Interleave Page Read Diagram

5.2.3.4. Internal DMA Controller Descriptors

5.2.3.4.1. Descriptor Structure

The internal DMA controller of the NDFC can transfer data between DMA FIFO in NDFC and DMA buffer in host memory using DMA descriptors. DMA descriptors in the host memory with chain structure is shown in Figure 5-16.

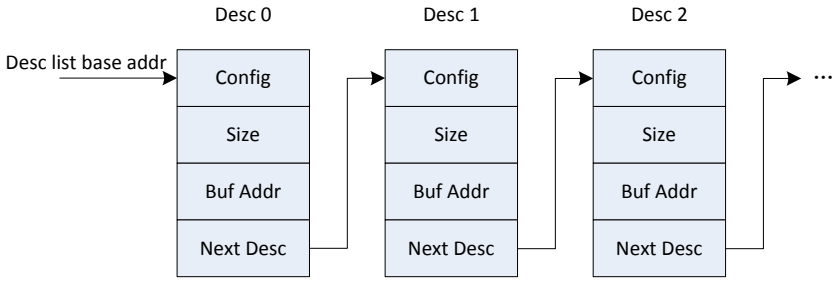

Figure 5- 16. Internal DMA Descriptor Chain Structure

The start address of DMA descriptor list must be word (32-bit) aligned, and will be configured to **NDFC DMA Descriptor List Base Address Register**. Each DMA descriptor consists of four words(32-bit).

5.2.3.4.2. Descriptor Definition

Config	
Bit	**Description**
31:4	/
3	FIRST_FLAG When set, this bit indicates that this descriptor contains the first buffer of data. Must be set to 1 in first descriptor.
2	LAST_FLAG When set, this bit indicates that the buffers pointed by this descriptor are the last data buffer.
1:0	/

Size	
Bit	**Description**
31:16	/
15:0	BUFF_SIZE These bits indicate the data buffer byte size, which must be a multiple of 8 bytes. If this filed is 0, the DMA ignores this buffer and proceeds to the next descriptor.

Buff Addr	
Bit	**Description**
31:0	BUFF_ADDR These bits indicate the physical address of DMA data buffer in host memory. The buffer address must be 4 bytes aligned.

Next Description	
Bit	**Description**
31:0	NEXT_DESC_ADDR These bits indicate the pointer to the physical host memory of the next descriptor is present.

5.2.3.5. NDFC Data Block Mask Register

ECC_DATA_BLOCK is written or read through the value of **NDFC Data Block Mask Register**. But in real application scenario, capacity could not waste, so writing operation usually does not use the function, only reading operation uses. In reading operation, we divides Sequence mode and Interleave mode through the store position of user_data.
Sequence mode: The user_data of every 1K main area data and ECC encoder data are next to main area data.
Interleave mode: All user_data and ECC encoder data are stored from page_size position.

When any **ECC_DATA_BLOCK** within page is read through batch command(**NDFC_CMD_TYPE** in 0x24 register is 0x10), the register is used differently for Sequence mode and Interleave mode.

Sequence mode can only support continue **ECC_DATA_BLOCK**, the register value can only be 0x1,0x3,0x7,etc. But Interleave mode has not limit.

Whether Sequence mode or Interleave mode, the first reading **ECC_DATA_BLOCK** is used to calculate corresponding column address, and column address is written to 0x14 and 0x18 register.

5.2.3.6. NDFC Enhanced Feature Register

The bit[24] and bit[23:16] of the register are used to judge whether free space need be padded random data except valid data when batch command function is used.

Take a SanDisk chip(SDTNQGAMA-008G) as an example:
Refer to the specification of the SanDisk chip, the page_size of the SanDisk chip is (16384+1280) bytes, but BCH level uses 40bit/1K, if user_data is 32 bytes, then the used space is 1152 bytes(14*40/8*16+32), the 128 bytes (1280-1152) space is not written. If there need be filled with 1 page, then the bit[24] of the register can be set to 1, and the bit[23:16] is written to 0x80, that the controller can automatically pad 128 bytes random data.

NOTE

Make sure that random function is enabled if there need be sent random data,that is, the NDFC_RANDOM_EN of 0x34 register is 0x1, or else the padding data is non-random, is all-0.

5.2.4. Programming Guidelines

5.2.4.1. Initializing Nand Flash

The NAND Flash is initialized as follows:

Step1: Read NDFC_ST[NDFC_RB_STATE0] to wait flash in the idle status.
Step2:Configure NDFC_CMD[NDFC_SEND_FIRST_CMD] to 1 to send the first command, configure NDFC_CMD[NDFC_WAIT_FLAG] to 1 to set wait RB; write 0xFF to NDFC_CMD[NDFC_CMD_LOW_BYTE] to send reset command.
Step3: Read NDFC_ST[NDFC_CMD_INT_FLAG] to wait *transfer command end interrupt flag* pending, after pending,

write 1 to clear the flag.

5.2.4.2. Erasing Nand Flash

The NAND Flash is erased as follows:

Step1: Read NDFC_ST[NDFC_RB_STATE0] to wait flash in the idle status.

Step2: Configure NDFC_CMD[NDFC_SEND_FIRST_CMD] to 1 to send the first command, configure NDFC_CMD[NDFC_WAIT_FLAG] to 1 to set wait RB;
Configure NDFC_CMD[NDFC_SEND_ADR] to 1 to enable transfer address, configure NDFC_CMD[NDFC_ADR_NUM] to set the number of address to be transferred;
Write the address of the block to be erased in NDFC_ADDR_LOW and NDFC_ADDR_HIGH;
Set NDFC_CMD[NDFC_CMD_LOW_BYTE] to 0x60 to send block erase command.

Step3: Read NDFC_ST[NDFC_CMD_INT_FLAG] to wait *transfer command end interrupt flag* pending, after pending, write 1 to clear the flag.

Step4: Read NDFC_ST[NDFC_RB_STATE0] to wait flash in the idle status.

Step5: Set NDFC_CMD[NDFC_WAIT_FLAG] to 1 to ensure wait RB, set NDFC_CMD[NDFC_SEND_FIRST_CMD] to 1 to send the first command; set NDFC_CMD[NDFC_CMD_LOW_BYTE] to 0xD0 to send erasing command.

Step6: Read NDFC_ST[NDFC_CMD_INT_FLAG] to wait *transfer command end interrupt flag* pending, after pending, write 1 to clear the flag.

Step7: Read flash state until flash is ready, configure NDFC_CNT[NDFC_DATA_CNT] to set 1-byte transfer data, set NDFC_CMD[NDFC_SEND_FIRST_CMD, NDFC_DATA_TRANS] to 0x3 to send the first command and transfer data. Set NDFC_CMD[NDFC_CMD_LOW_BYTE] to 0x70 to send read status command, read RAM0_BASE to wait ready status.

5.2.4.3. Writing Nand Flash

Step1: Erase the address of the block to be operated.

Step2: Read NDFC_ST[NDFC_RB_STATE0] to wait flash in the idle status.

Step3: Configure RAM0_BASE to write data to RAM0.

Step4: Configure NDFC_CNT[NDFC_DATA_CNT] to set transferred data;
Set NDFC_CMD[NDFC_SEND_FIRST_CMD] to 1 to send the first command, configure NDFC_CMD[NDFC_DATA_TRANS, NDFC_ACCESS_DIR] to 0x3 to set access direction as writing;
Set NDFC_CMD[NDFC_SEND_ADR] to 1 to enable transfer address, configure NDFC_CMD[NDFC_ADR_NUM] to set the number of the address to be transferred, write the address of the block to be operated in NDFC_ADDR_LOW and NDFC_ADDR_HIGH;
Set NDFC_CMD[NDFC_CMD_LOW_BYTE] to 0x80 to send page program command.

Step5: Read NDFC_ST[NDFC_RB_STATE0] to wait flash in the idle status.

Step6: Configure NDFC_CMD[NDFC_SEND_FIRST_CMD] to 1 to send the first command, configure NDFC_CMD[NDFC_WAIT_FLAG] to 1 to set wait RB; configure NDFC_CMD[NDFC_CMD_LOW_BYTE] to 0x10 to send end command.

Step7: Read NDFC_ST[NDFC_CMD_INT_FLAG] to wait *transfer command end interrupt flag* pending, after pending, write 1 to clear the flag.

5.2.4.4. Reading Nand Flash

Step1: Read NDFC_ST[NDFC_RB_STATE0] to wait flash in the idle status.

Step2: Configure NDFC_CNT[NDFC_DATA_CNT] to set transferred data;

Configure NDFC_CMD[NDFC_SEND_FIRST_CMD] to 1 to send the first command;

Configure NDFC_CMD[NDFC_ACCESS_DIR] to 0 to set access direction as reading;

Set NDFC_CMD[NDFC_SEND_ADR] to 1 to enable transfer address, configure NDFC_CMD[NDFC_ADR_NUM] to set the number of the address to be transferred, write the address of the block to be operated in NDFC_ADDR_LOW and NDFC_ADDR_HIGH;

Set NDFC_CMD[NDFC_CMD_LOW_BYTE] to 0x00 to send page read command.

Step3: Read NDFC_ST[NDFC_RB_STATE0] to wait flash in the idle status.

Step4: Configure NDFC_CMD[NDFC_SEND_FIRST_CMD] to 1 to send the first command, configure NDFC_CMD[NDFC_WAIT_FLAG] to 1 to set wait RB; configure NDFC_CMD[NDFC_CMD_LOW_BYTE] to 0x30 to send end command.

Step5: Read RAM0_BASE to get data from flash.

Step6: Read NDFC_ST[NDFC_CMD_INT_FLAG] to wait *transfer command end interrupt flag* pending, after pending, write 1 to clear the flag.

5.2.5. Register List

Module Name	Base Address
NDFC	0x04011000

Register Name	Offset	Description
NDFC_CTL	0x0000	NDFC Configure and Control Register
NDFC_ST	0x0004	NDFC Status Information Register
NDFC_INT	0x0008	NDFC Interrupt Control Register
NDFC_TIMING_CTL	0x000C	NDFC Timing Control Register
NDFC_TIMING_CFG	0x0010	NDFC Timing Configure Register
NDFC_ADDR_LOW	0x0014	NDFC Low Word Address Register
NDFC_ADDR_HIGH	0x0018	NDFC High Word Address Register
NDFC_DATA_BLOCK_MASK	0x001C	NDFC Data Block Mask Register
NDFC_CNT	0x0020	NDFC Data Counter Register
NDFC_CMD	0x0024	NDFC Commands IO Register
NDFC_RCMD_SET	0x0028	Read Command Set Register for Vendor's NAND Memory
NDFC_WCMD_SET	0x002C	Write Command Set Register for Vendor's NAND Memory
NDFC_ECC_CTL	0x0034	NDFC ECC Control Register
NDFC_ECC_ST	0x0038	NDFC ECC Status Register
NDFC_DATA_PAT_STA	0x003C	NDFC Data Pattern Status Register
NDFC_EFR	0x0040	NDFC Enhanced Feature Register

NDFC_RDATA_STA_CTL	0x0044	NDFC Read Data Status Control Register
NDFC_RDATA_STA_0	0x0048	NDFC Read Data Status Register 0
NDFC_RDATA_STA_1	0x004C	NDFC Read Data Status Register 1
NDFC_ERR_CNT_N	0x0050+0x04*N	NDFC Error Counter Register(N from 0 to 7)
NDFC_USER_DATA_LEN_N	0x0070+0x04*N	NDFC User Data Length Register(N from 0 to 3)
NDFC_USER_DATA_N	0x0080+0x04*N	NDFC User Data Field Register N (N from 0 to 31)
NDFC_EFNAND_STA	0x0110	NDFC EFNAND Status Register
NDFC_SPARE_AREA	0x0114	NDFC Spare Area Register
NDFC_PAT_ID	0x0118	NDFC Pattern ID Register
NDFC_DDR2_SPEC_CTL	0x011C	NDFC DDR2 Specific Control Register
NDFC_NDMA_MODE_CTL	0x0120	NDFC Normal DMA Mode Control Register
NDFC_MDMA_DLBA_REG	0x0200	NDFC MBUS DMA Descriptor List Base Address Register
NDFC_MDMA_STA	0x0204	NDFC MBUS DMA Interrupt Status Register
NDFC_DMA_INT_MASK	0x0208	NDFC MBUS DMA Interrupt Enable Register
NDFC_MDMA_CUR_DESC_ADDR	0x020C	NDFC MBUS DMA Current Descriptor Address Register
NDFC_MDMA_CUR_BUF_ADDR	0x0210	NDFC MBUS DMA Current Buffer Address Register
NDFC_DMA_CNT	0x0214	NDFC DMA Byte Counter Register
NDFC_IO_DATA	0x0300	NDFC Input/Output Data Register

5.2.6. Register Description

5.2.6.1. 0x0000 NDFC Control Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: NDFC_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28	R/W	0x0	NDFC_DDR_TYPE Type of DDR data interface This bit is valid when NF_TYPE is 0x2 or 0x3. 0: DDR 1: DDR2
27:24	R/W	0x0	NDFC_CE_SEL Chip Select for NAND Flash Chips 0000: NDFC Select Chip 0 0001: NDFC Select Chip 1 0010: NDFC Select Chip 2 0011: NDFC Select Chip 3 0100: NDFC Select Chip 4 0101: NDFC Select Chip 5 0110: NDFC Select Chip 6 0111: NDFC Select Chip 7 1000: NDFC Select Chip 8 1001: NDFC Select Chip 9

			1010: NDFC Select Chip 10 1011: NDFC Select Chip 11 1100: NDFC Select Chip 12 1101: NDFC Select Chip 13 1110: NDFC Select Chip 14 1111: NDFC Select Chip 15
23:22	/	/	/
21	R/W	0x0	NDFC_DDR_RM DDR Repeat Data Mode 0: Lower byte 1: Higher byte
20	R/W	0x0	NDFC_DDR_REN DDR Repeat Enable 0: Disable 1: Enable
19:18	R/W	0x0	NF_TYPE NAND Flash Type 00: Normal SDR NAND 01: Reserved 10: ONFI DDR NAND 11: Toggle DDR NAND
17	R/W	0x0	NDFC_CLE_POL NDFC Command Latch Enable (CLE) Signal Polarity Select 0: High active 1: Low active
16	R/W	0x0	NDFC_ALE_POL NDFC Address Latch Enable (ALE) Signal Polarity Select 0: High active 1: Low active
15	R/W	0x0	NDFC_DMA_TYPE 0: Dedicated DMA 1: Normal DMA
14	R/W	0x0	NDFC_RAM_METHOD Access internal RAM method 0: Access internal RAM by AHB method 1: Access internal RAM by DMA method
13:12	/	/	/
11:8	R/W	0x0	NDFC_PAGE_SIZE 000: 1KB 001: 2KB 010: 4KB 011: 8KB 100: 16KB 101: 32KB The page size is for main field data.

7	/	/	/
6	R/W	0x0	NDFC_CE_ACT Chip Select Signal CE# Control during NAND Operation 0: De-active Chip Select Signal NDFC_CE# during data loading, serial access and other no operation stage for power consumption. NDFC automatic controls Chip Select Signals. 1: Chip select signal NDFC_CE# is always active after NDFC is enabled
5	/	/	/
4:3	R/W	0x0	NDFC_RB_SEL NDFC External R/B Signal Select The value 0-3 selects the external R/B signal. The same R/B signal can be used for multiple chip select flash.
2	R/W	0x0	NDFC_BUS_WIDTH 0: 8-bit bus 1: 16-bit bus
1	R/W1C	0x0	NDFC_RESET NDFC Reset Write 1 to reset NDFC and clear to 0 after reset
0	R/W	0x0	NDFC_EN NDFC Enable Control 0: Disable NDFC 1: Enable NDFC

5.2.6.2. 0x0004 NDFC Status Register (Default Value: 0x0000_0F00)

Offset: 0x0004			**Register Name: NDFC_ST**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:14	/	/	/
13	R	0x0	NDFC_RDATA_STA_0 0: The number of bit 1 during current read operation is more than threshold value. 1: The number of bit 1 during current read operation is less than or equal to threshold value. This field only is valid when NDFC_RDATA_STA_EN is 1. The threshold value is configured in NDFC_RDATA_STA_TH.
12	R	0x0	NDFC_RDATA_STA_1 0: The number of bit 0 during current read operation is more than threshold value. 1: The number of bit 0 during current read operation is less than or equal to the threshold value. This field only is valid when NDFC_RDATA_STA_EN is 1. The threshold value is configured in NDFC_RDATA_STA_TH.
11	R	0x1	NDFC_RB_STATE3 NAND Flash R/B 3 Line State

			0: NAND Flash in BUSY State 1: NAND Flash in READY State
10	R	0x1	NDFC_RB_STATE2 NAND Flash R/B 2 Line State 0: NAND Flash in BUSY State 1: NAND Flash in READY State
9	R	0x1	NDFC_RB_STATE1 NAND Flash R/B 1 Line State 0: NAND Flash in BUSY State 1: NAND Flash in READY State
8	R	0x1	NDFC_RB_STATE0 NAND Flash R/B 0 Line State 0: NAND Flash in BUSY State 1: NAND Flash in READY State
7:5	/	/	/
4	R	0x0	NDFC_STA 0: NDFC FSM in IDLE state 1: NDFC FSM in BUSY state When NDFC_STA is 0, NDFC can accept new command and process command.
3	R	0x0	NDFC_CMD_FIFO_STATUS 0: Command FIFO not full and can receive new command 1: Full and waiting NDFC to process commands in FIFO Since there is only one 32-bit FIFO for command. When NDFC latches one command, command FIFO is free and can accept another new command.
2	R/W1C	0x0	NDFC_DMA_INT_FLAG When it is 1, it means that a pending DMA is completed. It will be cleared after writing 1 to this bit or it will be automatically cleared before FSM processing an new command.
1	R/W1C	0x0	NDFC_CMD_INT_FLAG When it is 1, it means that NDFC has finished one Normal Command Mode or one Batch Command Work Mode. It will be cleared after writing 1 to this bit or it will be automatically cleared before FSM processing an new command.
0	R/W1C	0x0	NDFC_RB_B2R When it is 1, it means that NDFC_R/B# signal is transferred from BUSY state to READY state. It will be cleared after writing 1 to this bit.

5.2.6.3. 0x0008 NDFC Interrupt and DMA Enable Register(Default Value: 0x0000_0000)

Offset: 0x0008			**Register Name: NDFC_INT**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x0	NDFC_DMA_INT_ENABLE Enable or disable interrupt when a pending DMA is completed.

1	R/W	0x0	NDFC_CMD_INT_ENABLE Enable or disable interrupt when NDFC has finished the procession of a single command in normal command work mode or one batch command work mode. 0: Disable 1: Enable
0	R/W	0x0	NDFC_B2R_INT_ENABLE Enable or disable interrupt when NDFC_RB# signal is transferring from BUSY state to READY state. 0: Disable 1: Enable

5.2.6.4. 0x000C NDFC Timing Control Register(Default Value: 0x0000_0000)

Offset: 0x000C			Register Name: NDFC_TIMING_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11:8	R/W	0x0	NDFC_READ_PIPE In SDR mode: 00: Normal 01: EDO 10: E-EDO Others: Reserved In DDR mode: 1~15 is valid.(These bits configure the number of clock when data is valid after RE#'s falling edge)
7:6	/	/	/
5:0	R/W	0x0	NDFC_DC_CTL NDFC Delay Chain Control. These bits are only valid in DDR data interface, and configure the relative phase between DQS and DQ[0…7] .

5.2.6.5. 0x0010 NDFC Timing Configure Register(Default Value: 0x0000_0095)

Offset: 0x0010			Register Name: NDFC_TIMING_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/
19:18	R/W	0x0	T_WC Write Cycle Time 00: 1*2T 01: 2*2T 10: 3*2T 11: 4*2T

17:16	R/W	0x0	T_CCS Change Column Setup Time 00: 12*2T 01: 20*2T 10: 28*2T 11: 60*2T
15:14	R/W	0x0	T_CLHZ CLE High to Output Hi-z 00: 2*2T 01: 8*2T 10: 16*2T 11: 31*2T
13:12	R/W	0x0	T_CS CE Setup Time 00: 2*2T 01: 8*2T 10: 16*2T 11: 31*2T
11	R/W	0x0	T_CDQSS DQS Setup Time for Data Input Start 0: 4*2T 1: 20*2T
10:8	R/W	0x0	T_CAD Command, Address, Data Delay 000: 2*2T 001: 6*2T 010: 10*2T 011: 14*2T 100: 22*2T 101: 30*2T 110/111: 62*2T
7:6	R/W	0x2	T_RHW Cycle Number from RE# High to WE# Low 00: 4*2T 01: 12*2T 10: 20*2T 11: 28*2T
5:4	R/W	0x1	T_WHR Cycle Number from WE# High to RE# Low 00: 0*2T 01: 6*2T 10: 14*2T 11: 22*2T
3:2	R/W	0x1	T_ADL Cycle Number from Address to Data Loading

			00: 0*2T 01: 6*2T 10: 14*2T 11: 22*2T
1:0	R/W	0x1	T_WB Cycle Number from WE# High to Busy 00:14*2T 01: 22*2T 10: 30*2T 11: 38*2T

5.2.6.6. 0x0014 NDFC Address Low Word Register(Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: NDFC_ADDR_LOW
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0x0	ADDR_DATA4 NAND Flash 4th Cycle Address Data
23:16	R/W	0x0	ADDR_DATA3 NAND Flash 3rd Cycle Address Data
15:8	R/W	0x0	ADDR_DATA2 NAND Flash 2nd Cycle Address Data
7:0	R/W	0x0	ADDR_DATA1 NAND Flash 1st Cycle Address Data

5.2.6.7. 0x0018 NDFC Address High Word Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: NDFC_ADDR_HIGH
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0x0	ADDR_DATA8 NAND Flash 8th Cycle Address Data
23:16	R/W	0x0	ADDR_DATA7 NAND Flash 7th Cycle Address Data
15:8	R/W	0x0	ADDR_DATA6 NAND Flash 6th Cycle Address Data
7:0	R/W	0x0	ADDR_DATA5 NAND Flash 5th Cycle Address Data

5.2.6.8. 0x001C NDFC Data Block Mask Register(Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: NDFC_DATA_BLOCK_MASK
Bit	**Read/Write**	**Default/Hex**	**Description**

31	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 31 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
30	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 30 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
29	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 29 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
28	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 28 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
27	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 27 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
26	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 26 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
25	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 25 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
24	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 24 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**).

			0: Disable 1: Enable 1 data block = 1024 bytes main field data.
23	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 23 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
22	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 22 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
21	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 21 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
20	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 20 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
19	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 19 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
18	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 18 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
17	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 17 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.

16	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 16 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
15	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 15 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
14	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 14 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
13	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 13 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
12	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 12 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
11	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 11 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
10	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 10 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
9	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 9 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**).

			0: Disable 1: Enable 1 data block = 1024 bytes main field data.
8	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 8 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
7	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 7 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
6	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 6 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
5	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 5 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
4	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 4 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
3	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 3 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
2	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 2 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.

1	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 1 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.
0	R/W	0x0	NDFC_DATA_BLOCK_MASK It is used to indicate the data block 0 should be written or read during batch command procession(NDFC_CMD_TYPE=0x10 in **NDFC_CMD**). 0: Disable 1: Enable 1 data block = 1024 bytes main field data.

5.2.6.9. 0x0020 NDFC Data Counter Register(Default Value: 0x0000_0000)

Offset: 0x0020			**Register Name: NDFC_CNT**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9:0	R/W	0x0	NDFC_DATA_CNT Transfer Data Byte Counter The length can be set from 1 byte to 1024 bytes. However, 1024 bytes is set when it is zero.

5.2.6.10. 0x0024 NDFC Command IO Register (Default Value: 0x0000_0000)

Offset: 0x0024			**Register Name: NDFC_CMD**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	R/W	0x0	NDFC_CMD_TYPE 00: Common command for normal operation 01: Special command for Flash spare field operation 10: Page command for batch process operation 11: Reserved
29	R/W	0x0	NDFC_SEND_FOURTH_CMD 0: Donot send fourth set command 1: Send it on the external memory's bus It is used for EF-NAND page read.
28	R/W	0x0	NDFC_SEND_THIRD_CMD 0: Donot send third set command 1: Send it on the external memory's bus It is used for EF-NAND page read.
27	R/W	0x0	NDFC_SEND_RANDOM_CMD2_CTL 0: Donot send random cmd2 (NDFC_RANDOM_CMD2) 1: Send random cmd2

			Note: It is only valid in batch cmd operation and writing operation.
26	R/W	0x0	NDFC_DATA_METHOD Data swap method when the internal RAM and system memory It is only active for common command and special command. 0: No action 1: DMA transfer automatically It only is active when NDFC_RAM_METHOD is 1. If this bit is set to 1, NDFC should setup DRQ to fetch data before output to Flash or NDFC should setup DRQ to send to system memory after fetching data from Flash. If this bit is set to 0, NDFC output the data in internal RAM or do nothing after fetching data from Flash.
25	R/W	0x0	NDFC_SEQ User data & BCH check word position. It only is active for Page Command, donot care about this bit for other two commands. 0: Interleave Method (on page spare area) 1: Sequence Method (following data block)
24	R/W	0x0	NDFC_SEND_SECOND_CMD 0: Donot send second set command 1: Send it on the external memory's bus
23	R/W	0x0	NDFC_WAIT_FLAG 0: NDFC can transfer data regardless of the internal NDFC_RB wire 1: NDFC can transfer data when the internal NDFC_RB wire is READY; otherwise it cannot when the internal NDFC_RB wire is BUSY
22	R/W	0x0	NDFC_SEND_FIRST_CMD 0: Donot send first set command 1: Send it on the external memory's bus
21	R/W	0x0	NDFC_DATA_TRANS 0: No data transfer on external memory bus 1: Data transfer and direction is decided by the field NDFC_ACCESS_DIR
20	R/W	0x0	NDFC_ACCESS_DIR 0: Read NAND Flash 1: Write NAND Flash
19	R/W	0x0	NDFC_SEND_ADR 0: Donot send ADDRESS 1: Send N cycles ADDRESS, the number N is specified by NDFC_ADR_NUM field
18:16	R/W	0x0	NDFC_ADR_NUM Address Cycles' Number 000: 1 cycle address field 001: 2 cycles address field 010: 3 cycles address field 011: 4 cycles address field 100: 5 cycles address field 101: 6 cycles address field 110: 7 cycles address field

			111: 8 cycles address field
15:10	/	/	/
9:8	R/W	0x0	NDFC_ADR_NUM_IN_PAGE_CMD The number of address cycles during page command. 00: 2 address cycles 11: 5 address cycles Others: reserved
7:0	R/W	0x0	NDFC_CMD_LOW_BYTE NDFC command low byte data This command will be sent to external Flash by NDFC.

5.2.6.11. 0x0028 NDFC Command Set Register 0(Default Value: 0x00E0_0530)

Offset: 0x0028			Register Name: NDFC_CMD_SET0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0x00	NDFC_RANDOM_CMD2 Used for Batch Operation
23:16	R/W	0xE0	NDFC_RANDOM_READ_CMD1 Used for Batch Read Operation
15:8	R/W	0x05	NDFC_RANDOM_READ_CMD0 Used for Batch Read Operation
7:0	R/W	0x30	NDFC_READ_CMD Used for Batch Read Operation

5.2.6.12. 0x002C NDFC Command Set Register 1(Default Value: 0x7000_8510)

Offset: 0x002C			Register Name: NDFC_CMD_SET1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0x70	NDFC_READ_CMD0 Used for EF-NAND Page Read Operation
23:16	R/W	0x00	NDFC_READ_CMD1 Used for EF-NAND Page Read Operation
15:8	R/W	0x85	NDFC_RANDOM_WRITE_CMD Used for Batch Write Operation
7:0	R/W	0x10	NDFC_PROGRAM_CMD Used for Batch Write Operation

5.2.6.13. 0x0034 NDFC ECC Control Register(Default Value: 0x4A80_0008)

Offset: 0x0034			Register Name: NDFC_ECC_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**

31	/	/	/
30:16	R/W	0x4a80	NDFC_RANDOM_SEED The seed value for randomize engine. It is only active when NDFC_RANDOM_EN is set to '1'.
15:8	R/W	0x0	NDFC_ECC_MODE 00000000: BCH-16 00000001: BCH-24 00000010: BCH-28 00000011: BCH-32 00000100: BCH-40 00000101: BCH-44 00000110: BCH-48 00000111: BCH-52 00001000: BCH-56 00001001: BCH-60 00001010: BCH-64 00001011: BCH-68 00001100: BCH-72 00001101: BCH-76 00001110: BCH-80 Others : Reserved
7	R/W	0x0	NDFC_RANDOM_SIZE 0: ECC block size 1: Page size
6	R/W	0x0	NDFC_RANDOM_DIRECTION 0: LSB first 1: MSB first
5	R/W	0x0	NDFC_RANDOM_EN 0: Disable Data Randomize 1: Enable Data Randomize
4	R/W	0x0	NDFC_ECC_EXCEPTION 0: Normal ECC 1: For ECC, there is an exception. If all data is 0xff or 0x00 for the block. When reading this page, ECC assumes that it is right. For this case, no error information is reported. **Note: It is only active when ECC is ON**
3	R/W	0x1	NDFC_ECC_PIPELINE Pipeline function enable or disable for batch command 0: Error Correction function no pipeline with next block operation 1: Error Correction pipeline
2:1	/	/	/
0	R/W	0x0	NDFC_ECC_EN 0: ECC is OFF 1: ECC is ON

5.2.6.14. 0x0038 NDFC ECC Status Register(Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: NDFC_ECC_ST
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 31 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[31] of this register is corresponding the 31th ECC data block. 1 ECC Data Block = 1024 bytes.
30	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 30 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[30] of this register is corresponding the 30th ECC data block. 1 ECC Data Block = 1024 bytes.
29	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 29 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[29] of this register is corresponding the 29th ECC data block. 1 ECC Data Block = 1024 bytes.
28	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 28 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[28] of this register is corresponding the 28th ECC data block. 1 ECC Data Block = 1024 bytes.
27	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 27 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[27] of this register is corresponding the 27th ECC data block. 1 ECC Data Block = 1024 bytes.
26	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 26 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[26] of this register is corresponding the 26th ECC data block. 1 ECC Data Block = 1024 bytes.
25	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 25 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[25] of this register is corresponding the 25th ECC data block. 1 ECC Data

			Block = 1024 bytes.
24	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 24 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[24] of this register is corresponding the 24th ECC data block. 1 ECC Data Block = 1024 bytes.
23	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 23 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[23] of this register is corresponding the 23th ECC data block. 1 ECC Data Block = 1024 bytes.
22	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 22 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[22] of this register is corresponding the 22th ECC data block. 1 ECC Data Block = 1024 bytes.
21	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 21 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[21] of this register is corresponding the 21th ECC data block. 1 ECC Data Block = 1024 bytes.
20	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 20 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[20] of this register is corresponding the 20th ECC data block. 1 ECC Data Block = 1024 bytes.
19	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 19 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[19] of this register is corresponding the 19th ECC data block. 1 ECC Data Block = 1024 bytes.
18	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 18 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[18] of this register is corresponding the 18th ECC data block. 1 ECC Data Block = 1024 bytes.
17	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 17

			0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[17] of this register is corresponding the 17th ECC data block. 1 ECC Data Block = 1024 bytes.
16	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 16 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[16] of this register is corresponding the 16th ECC data block. 1 ECC Data Block = 1024 bytes.
15	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 15 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[15] of this register is corresponding the 15th ECC data block. 1 ECC Data Block = 1024 bytes.
14	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 14 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[14] of this register is corresponding the 14th ECC data block. 1 ECC Data Block = 1024 bytes.
13	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 13 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[13] of this register is corresponding the 13th ECC data block. 1 ECC Data Block = 1024 bytes.
12	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 12 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[12] of this register is corresponding the 12th ECC data block. 1 ECC Data Block = 1024 bytes.
11	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 11 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[11] of this register is corresponding the 11th ECC data block. 1 ECC Data Block = 1024 bytes.
10	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 10 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[10] of this register is corresponding the 10th ECC data block. 1 ECC Data

			Block = 1024 bytes.
9	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 9 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[9] of this register is corresponding the 9th ECC data block. 1 ECC Data Block = 1024 bytes.
8	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 8 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[8] of this register is corresponding the 8th ECC data block. 1 ECC Data Block = 1024 bytes.
7	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 7 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[7] of this register is corresponding the 7th ECC data block. 1 ECC Data Block = 1024 bytes.
6	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 6 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[6] of this register is corresponding the 6th ECC data block. 1 ECC Data Block = 1024 bytes.
5	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 5 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[5] of this register is corresponding the 5th ECC data block. 1 ECC Data Block = 1024 bytes.
4	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 4 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[4] of this register is corresponding the 4th ECC data block. 1 ECC Data Block = 1024 bytes.
3	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 3 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[3] of this register is corresponding the 3rd ECC data block. 1 ECC Data Block = 1024 bytes.
2	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 2

			0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[2] of this register is corresponding the 2nd ECC data block. 1 ECC Data Block = 1024 bytes.
1	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 1 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[1] of this register is corresponding the 1st ECC data block. 1 ECC Data Block = 1024 bytes.
0	R	0x0	NDFC_ECC_ERR Error information bit of Data Block 0 0: ECC can correct these error bits or there is no error bit 1: Error bits number beyond of ECC correction capability and cannot correct them The bit[0] of this register is corresponding the 0 ECC data block. 1 ECC Data Block = 1024 bytes.

5.2.6.15. 0x003C NDFC Data Pattern Status Register(Default Value: 0x0000_0000)

Offset: 0x003C			**Register Name: NDFC_DATA_PAT_STA**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 31 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
30	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 30 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
29	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 29 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
28	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 28 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
27	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 27 when read from external NAND flash. 0: No found

			1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
26	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 26 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
25	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 25 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
24	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 24 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
23	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 23 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
22	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 22 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
21	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 21 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
20	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 20 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
19	R	0x0	Special pattern (all 0x00 or all 0xff) found Flag for Data Block 19 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
18	R	0x0	Special pattern (all 0x00 or all 0xff) Found flag for Data Block 18 when read from external NAND flash. 0: No found

			1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
17	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 17 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
16	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 16 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
15	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 15 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
14	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 14 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
13	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 13 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
12	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 12 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
11	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 11 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
10	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 10 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
9	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 9 when read from external NAND flash. 0: No found

			1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
8	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 8 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
7	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 7 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
6	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 6 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
5	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 5 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
4	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 4 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
3	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 3 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
2	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 2 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
1	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 1 when read from external NAND flash. 0: No found 1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.
0	R	0x0	Special pattern (all 0x00 or all 0xff) found flag for Data Block 0 when read from external NAND flash. 0: No found

			1: Special pattern is found The register of NDFC_PAT_ID would indicate which kind of pattern is found.

5.2.6.16. 0x0040 NDFC Enhanced Feature Register(Default Value: 0x0000_0000)

Offset: 0x0040			Register Name: NDFC_EFR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	DB_CNT_EN Dummy_Byte_Count_EN 0:Disable fill dummy byte 1:Enable fill dummy byte
23:16	R/W	0x0	DB_CNT Dummy_Byte_Count After PAGE CMD operation finishing sending out the main data , user data and ECC code, controller would send dummy byte to fill the unused space in one page. **Note: It is only valid in PAGE CMD operation(NDFC_CMD_TYPE=0x3), and this function is disabled when Dummy_Byte_Count_EN is 0.** **If the NDFC_RANDOM_EN = 0x0, the value of the dummy byte is 0, so in order to improve the stability, when using this function , it is better to set the NDFC_RANDOM_EN to 0x1.**
15:9	/	/	/
8	R/W	0x0	NDFC_WP_CTRL NAND Flash Write Protect Control Bit 0: Write Protect is active 1: Write Protect is not active When this bit is '0', WP signal line is low level and external NAND flash is on protected state.
7	/	/	/
6:0	R/W	0x0	NDFC_ECC_DEBUG For the purpose of debugging ECC engine, special error bits are inserted before writing external Flash Memory. 0: No error is inserted (ECC Normal Operation) n: N bits error are inserted

5.2.6.17. 0x0044 NDFC Read Data Status Control Register(Default Value: 0x0100_0000)

Offset: 0x0044			Register Name: NDFC_RDATA_STA_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x1	NDFC_RDATA_STA_EN

			0: Disable to count the number of bit 1 and bit 0 during current read operation 1: Enable to count the number of bit 1 and bit 0 during current read operation The number of bit 1 and bit 0 during current read operation can be used to check whether a page is blank or bad.
23:19	/	/	/
18:0	R/W	0x0	NDFC_RDATA_STA_TH The threshold value to generate data status If the number of bit 1 during current read operation is less than or equal to threshold value, the bit 13 of NDFC_ST register will be set. If the number of bit 0 during current read operation is less than or equal to threshold value, the bit 12 of NDFC_ST register will be set.

5.2.6.18. 0x0048 NDFC Read Data Status Register 0(Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: NDFC_RDATA_STA_0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	BIT_CNT_1 The number of input bit 1 during current command. It will be cleared automatically when next command is executed.

5.2.6.19. 0x004C NDFC Read Data Status Register 1(Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: NDFC_RDATA_STA_1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	BIT_CNT_0 The number of input bit 0 during current command. It will be cleared automatically when next command is executed.

5.2.6.20. 0x0050+N*0x04 NDFC Error Counter Register N(Default Value: 0x0000_0000)

Offset: 0x0050+N*0x04(N=0~7)			Register Name: NDFC_ERR_CNT_N
Bit	**Read/Write**	**Default/Hex**	**Description**
[8M+7:8M] (M=0~3)	R	0x0	ECC_COR_NUM ECC Corrected Bits Number for ECC Data Block[N*0x04+M] 00000000: No corrected bits 00000001: 1 corrected bit 00000010: 2 corrected bits ..... 01010000 : 80 corrected bits Others: Reserved **Note: 1 ECC Data Block =1024 bytes**

5.2.6.21. 0x0070+N*0x04 NDFC User Data Length Register N(Default Value: 0x0000_0000)

Offset: 0x0070+N*0x04(N=0~3)			Register Name: NDFC_USER_DATA_LEN_N
Bit	**Read/Write**	**Default/Hex**	**Description**
[4M+3 : 4M] (M=0~7)	R/W	0x0	It's used to indicate user data's length of ECC DATA BLOCK[0x08*N+M]. 0000 : no user data 0001 : 4 bytes user data 0010 : 8 bytes user data 0011 : 12 bytes user data 0100 : 16 bytes user data 0101 : 20 bytes user data 0110 : 24 bytes user data 0111 : 28 bytes user data 1000 : 32 bytes user data Other : reserved

5.2.6.22. 0x0080 + N*0x04 NDFC User Data Register N(Default Value: 0xFFFF_FFFF)

Offset: 0x0080 + N*0x04(N=0~31)			Register Name: NDFC_USER_DATA_N
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0xffffffff	USER_DATA All of the user data in one page is stored in NDFC_USER_DATA_N. The start register address of each ECC DATA BLOCK's user data is determined by its length configured in NDFC_USER_DATA_LEN_N. For example: ECC DATA BLOCK[0] user data len = 8 bytes, address = 0x80 ECC DATA BLOCK[1] user data len = 0 byte, ECC DATA BLOCK[2] user data len = 4 bytes, address = 0x80+8 ECC DATA BLOCK[3] user data len = 4 bytes, address = 0x80+8+4 ECC DATA BLOCK[4] user data len = 0 byte ECC DATA BLOCK[5] user data len = 16 bytes, address = 0x80+8+4+4 ECC DATA BLOCK[6] user data len = 0 byte ECC DATA BLOCK[7] user data len = 0 byte

5.2.6.23. 0x110 NDFC EFNAND Status Register(Default Value: 0x0000_0000)

Offset: 0x0110			Register Name: NDFC_EFNAND_STATUS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R	0x0	EF_NAND_STATUS The status value for EF-NAND page read operation

5.2.6.24. 0x0114 NDFC Spare Area Register(Default Value: 0x0000_0400)

Offset: 0x0114			Register Name: NDFC_SPARE_AREA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x400	NDFC_SPARE_ADR This value indicates the spare area first byte address for NDFC interleave page operation.

5.2.6.25. 0x0118 NDFC Pattern ID Register(Default Value: 0x0000_0000)

Offset: 0x0118			Register Name: NDFC_PAT_ID
Bit	**Read/Write**	**Default/Hex**	**Description**
n (n=0~31)	R	0x0	PAT_ID Special Pattern ID for ECC data block[n] 0: All 0x00 is found 1: All 0xFF is found

5.2.6.26. 0x011C NDFC DDR2 Specific Control Register(Default Value: 0x0000_0000)

Offset: 0x011C			Register Name: NDFC_DDR2_SPEC_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:12	R/W	0x0	DLEN_WR The number of latency DQS cycle for write 0000: No latency 0001: One latency DQS cycle 0010: Two latency DQS cycle 0011: Four latency DQS cycle
11:8	R/W	0x0	DLEN_RD The number of latency DQS cycle for read 0000: No latency 0001: One latency DQS cycle 0010: Two latency DQS cycle 0011: Four latency DQS cycle
7:3	/	/	/
2	R/W	0x0	EN_RE_C Enable the complementary RE# signal 0: Disable 1: Enable
1	R/W	0x0	EN_DQS_C Enable the complementary DQS signal

			0: Disable 1: Enable
0	/	/	/

5.2.6.27. 0x0120 NDFC Normal DMA Mode Control Register(Default Value: 0x0000_00E5)

Offset: 0x0120			Register Name: NDFC_NDMA_MODE_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:6	R/W	0x11	DMA_ACT_STA 00:dma_active is low 01:dma_active is high 10:dma_active is controlled by dma_request(DRQ) 11:dma_active is controlled by controller
5	R/W	0x1	DMA_ACK_EN 0: active fall do not care ack 1: active fall must after detect ack is high
4:0	R/W	0x05	DELAY_CYCLE The counts of hold cycles from DMA last signal high to dma_active high

5.2.6.28. 0x0200 NDFC MBUS DMA Descriptor List Base Address Register(Default Value: 0x0000_0000)

Offset: 0x0200			Register Name: NDFC_MDMA_DLBA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	NFC_MDMA_DESC_BASE_ADDR Start Address of Descriptor List Contains the base address of the First Descriptor. The LSB bits [1:0] are ignored and taken as all-zero by the DMA internally. Hence these LSB bits are read-only.

5.2.6.29. 0x0204 NDFC MBUS DMA Interrupt Status Register(Default Value: 0x0000_0000)

Offset: 0x0204			Register Name: NDFC_MDMA_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	NFC_MDMA_TRANS_FINISH_INT Transfer Finish Interrupt Indicates that data transmission is finished for a descriptor. Writing a '1' clears this bit. Bit 0: Corresponding DMA descriptor 0 Bit 1: Corresponding DMA descriptor 1 … Bit 31: Corresponding DMA descriptor 31

5.2.6.30. 0x0208 NDFC MBUS DMA Interrupt Enable Register(Default Value: 0x0000_0000)

Offset: 0x0208			Register Name: NDFC_DMA_INT_MASK
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	NFC_MDMA_TRANS_INT_ENB Transfer Interrupt Enable When set, Transmit Interrupt is enabled. When reset, Transmit Interrupt is disabled. Bit 0: Corresponding DMA descriptor 0 Bit 1: Corresponding DMA descriptor 1 … Bit 31: Corresponding DMA descriptor 31

5.2.6.31. 0x020C NDFC MBUS DMA Current Descriptor Address Register(Default Value: 0x0000_0000)

Offset: 0x020C			Register Name: NDFC_MDMA_CUR_DESC_ADDR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	CUR_DESC_ADDR Current Descriptor Address Pointer Cleared on reset. Pointer updated by DMA during operation. This register points to the start address of the current descriptor read by the DMA.

5.2.6.32. 0x0210 NDFC MBUS DMA Current Buffer Address Register(Default Value: 0x0000_0000)

Offset: 0x0210			Register Name: NDFC_MDMA_CUR_BUF_ADDR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	CUR_BUFF_ADDR Current Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation. This register points to the current Data Buffer Address being accessed by the DMA.

5.2.6.33. 0x0214 NDFC DMA Byte Counter Register(Default Value: 0x0000_0000)

Offset: 0x0214			Register Name: NDFC_DMA_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0	DMA_CNT DMA data counter, including MBUS DMA and Normal DMA

5.2.6.34. 0x0300 NDFC IO Data Register(Default Value: 0x0000_0000)

Offset: 0x0300			Register Name: NDFC_IO_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	NDFC_IO_DATA Read/Write data into internal RAM Access unit is 32-bit.

5.3. SD/MMC Host Controller(SMHC)

5.3.1. Overview

The SD-MMC Host Controller(SMHC) controls the read/write operations on the secure digital(SD) card and multimedia card(MMC), and supports various extended devices based on the secure digital input/output(SDIO) protocol. The H616 provides three SMHC interfacs for controlling the SD card,MMC and SDIO device.

The SMHC has the following features:

- Supports eMMC boot operation
- Supports command completion signal and interrupt to host processor and command completion signal disable feature
- SMHC0 supports SD (Version1.0 to 3.0), 4-bit bus width
  - SDR mode 50 MHz@3.3V IO pad
  - DDR mode 50 MHz@3.3V IO pad
  - SDR mode 150 MHz@1.8V IO pad
- SMHC1 supports SDIO(Version1.1 to 3.0), 4-bit bus width
  - SDR mode 50 MHz@3.3V IO pad
  - DDR mode 50 MHz@3.3V IO pad
  - SDR mode 150 MHz@1.8V IO pad
- SMHC2 supports MMC(Version3.3 to 5.0), 8-bit bus width
  - SDR mode 50 MHz@3.3V IO pad
  - DDR mode 50 MHz@3.3V IO pad
  - SDR mode 150 MHz@1.8V IO pad
  - DDR mode 100 MHz@1.8V IO pad
- Hardware CRC generation and error detection
- Programmable baud rate
- Host pull-up control
- Supports SDIO interrupt in 1-bit and 4-bit modes
- Block size of 1 to 65535 bytes
- Descriptor-based internal DMA controller
- Internal 1KB FIFO for data transfer
- SMHC0 realizes the conversion from 3.3 V to 1.8 V, no independent LDO power is required

5.3.2. Block Diagram

Figure 5-1 shows a block diagram of the SMHC.

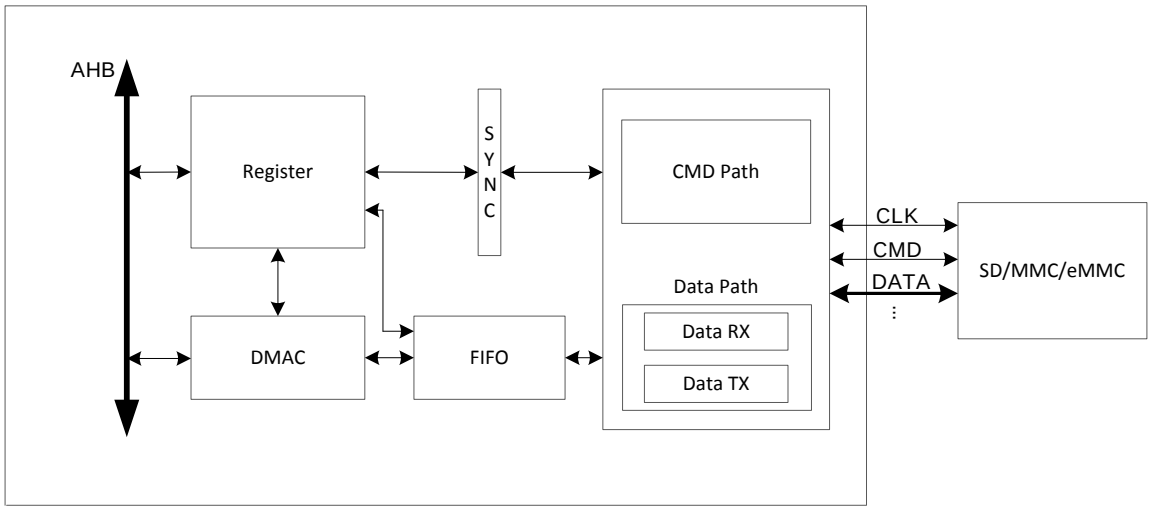

Figure 5- 17. SMHC Block Diagram

5.3.3. Operations and Functional Descriptions

5.3.3.1. External Signals

Table 5-1 describes the external signals of SMHC.

Table 5- 3. SMHC External Signals

Port Name	Width	Type	Description
SDC0_CLK	1	O	Clock output for SD/TF card
SDC0_CMD	1	I/O,OD	CMD line for SD/TF card
SDC0_D[i] (i=0~3)	4	I/O	Data line for SD/TF card
SDC1_CLK	1	O	Clock output for SDIO Wi-Fi
SDC1_CMD	1	I/O,OD	CMD line for SDIO Wi-Fi
SDC1_D[i] (i=0~3)	4	I/O	Data line for SDIO Wi-Fi
SDC2_CLK	1	O	Clock output for MMC
SDC2_CMD	1	I/O,OD	CMD line for MMC
SDC2_D[i] (i=0~7)	8	I/O	Data line for MMC
SDC2_RST	1	O	Reset signal for MMC
SDC2_DS	1	I	Data Strobe for MMC

5.3.3.2. Clock Sources

Each SMHC gets three different clocks. User can select one of them to make SMHC clock source. Table 5-2 describes the clock sources of SMHC. Users can see CCU in chapter 3.3 for clock setting, configuration and gating information.

Table 5- 4. SMHC Clock Sources

Clock Sources	Description
OSC24M	24 MHz Crystal
PLL_PERI0(2X)	Peripheral Clock, the default value is 1.2 GHz
PLL_PERI1(2X)	Peripheral Clock, the default value is 1.2 GHz

5.3.3.3. Timing Diagram

Please refer to relative specifications:

- Physical Layer Specification Ver3.00 Final
- SDIO Specification Ver2.00
- Multimedia Cards (MMC – version 4.2)
- JEDEC Standard – JESD84-44, Embedded Multimedia Card(eMMC) Card Product Standard
- JEDEC Standard – JESD84-B45, Embedded Multimedia Card(eMMC) Electrical Standard(4.5 Device)
- JEDEC Standard – JESD84-B50, Embedded Multimedia Card (eMMC) Electrical Standard(5.0)

5.3.3.4. Internal DMA Controller Description

SMHC has an internal DMA controller (IDMAC) to transfer data between host memory and SMHC port. With a descriptor, IDMAC can efficiently move data from source to destination by automatically loading next DMA transfer arguments, which need less CPU intervention. Before transfer data in IDMAC, host driver should construct a descriptor list, configure arguments of every DMA transfer, then launch the descriptor and start the DMA. IDMAC has an interrupt controller, when enabled, it can interrupt the HOST CPU in situations such as data transmission completed or some errors happened.

5.3.3.4.1. IDMAC Descriptor Structure

The IDMAC uses a descriptor with a chain structure, and each descriptor points to a unique buffer and the next descriptor.

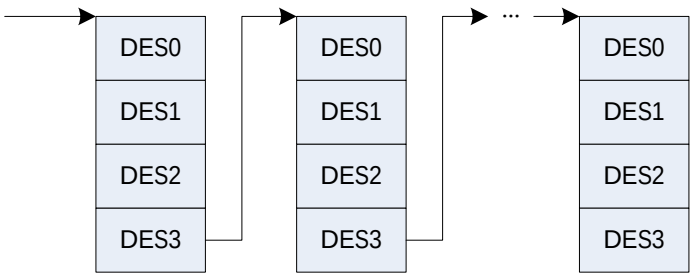

Figure 5- 18. IDMAC Descriptor Structure Diagram

This figure illustrates the internal formats of a descriptor. The descriptor address must be aligned to the bus width used for 32-bit buses. Each descriptor contains 16 bytes of control and status information.

DES0 is a notation used to denote the [31:0] bits, DES1 to denote [63:32] bits, DES2 to denote [95:64] bits, and DES3 to denote [127:96] bits in a descriptor.

5.3.3.4.2. DES0 Definition

Bits	Name	Descriptor
31	HOLD	DES_OWN_FLAG When set, this bit indicates that the descriptor is owned by the IDMAC. When this bit is reset, it indicates that the descriptor is owned by the host. This bit is cleared when transfer is over.
30	ERROR	ERR_FLAG When some error happened in transfer, this bit will be set.
29:5	/	/
4	Chain Flag	CHAIM_MOD When set, this bit indicates that the second address in descriptor is the next descriptor address. Must be set 1.
3	First DES Flag	FIRST_FLAG When set, this bit indicates that this descriptor contains the first buffer of data. Must be set to 1 in first DES.
2	Last DES Flag	LAST_FLAG When set, this bit indicates that the buffers pointed to by this descriptor are the last data buffer
1	Disable Interrupt on completion	CUR_TXRX_OVER_INT_DIS When set, this bit will prevent the setting of the TX/RX interrupt bit of the IDMAC status register for data that ends in the buffer pointed to by this descriptor
0	/	/

5.3.3.4.3. DES1 Definition

For SMHC0/SMCH1:

Bits	Name	Descriptor
31:16	/	/
15:0	Buffer size	BUFF_SIZE These bits indicate the data buffer byte size, which must be a multiple of 4 bytes. If this filed is 0, the DMA ignores this buffer and proceeds to the next descriptor.

For SMHC2:

Bits	Name	Descriptor
31:13	/	/

12:0	Buffer size	BUFF_SIZE These bits indicate the data buffer byte size, which must be a multiple of 4 bytes. If this filed is 0, the DMA ignores this buffer and proceeds to the next descriptor.

5.3.3.4.4. DES2 Definition

Bits	Name	Descriptor
31:0	Buffer address pointer	BUFF_ADDR These bits indicate the physical address of data buffer. For SMHC0,SMHC1, the field is a word address. For SMHC2, the field is a byte address.

5.3.3.4.5. DES3 Definition

Bits	Name	Descriptor
31:0	Next descriptor address	NEXT_DESP_ADDR These bits indicate the pointer to the physical memory where the next descriptor is present. For SMHC0,SMHC1, the field is a word address. For SMHC2, the field is a byte address.

5.3.3.5. Calibrate Delay Chain

The sample clock delay chain and data strobe delay chain(this chain is only in SMHC2) are used to generate delay to make proper timing between internal card clock/data strobe and data signals. Each delay chain is made up with 64 delay cells. The delay time of one delay cell can be estimated through delay chain calibration.

The steps to calibrate delay chain are as follows.

Step1: Enable SMHC. In order to calibrate delay chain by operation registers in SMHC, SMHC must be enabled through **SMHC Bus Gating Reset Register** and **SMHC0/1/2 Clock Register**.

Step2: Configure a proper clock for SMHC. Calibration delay chain is based on the clock for SMHC from Clock Control Unit(CCU). Calibration delay chain is an internal function in SMHC and does not need device. So, it is unnecessary to enable clock signal for device. The recommended clock frequency is 200 MHz.

Step3: Set proper initial delay value. Writing 0xA0 to **delay control register** enables **Delay Software Enable** (bit[7]) and sets initial delay value 0x20 to **Delay chain**(bit[5:0]). Then write 0x0 to **delay control register** to clear the value.

Step4: Write 0x8000 to **delay control register** to start calibrate delay chain.

Step5: Wait until the flag(bit14 in **delay control register**) of calibration done is set. The number of delay cells is shown at bit[13:8] in **delay control register**. The delay time generated by these delay cells is equal to the cycle of SMHC's clock nearly. This value is the result of calibration.

Step6: Calculate the delay time of one delay cell according to the cycle of SMHC's clock and the result of calibration.

NOTE

In the above descriptions, **delay control register** contains **SMHC Sample Delay Control Register** and **SMHC Data Strobe Delay Control Register**. **Delay Software Enable** contains **Sample Delay Software Enable** and **Data Strobe Delay Software Enable**. **Delay chain** contains **Sample Delay Software** and **Data Strobe Delay Software**.

5.3.4. Programming Guidelines

5.3.4.1. Initialization

Before data and command are exchanged between a card and the SMHC, the SMHC need to be initialized. The SMHC is initialized as follows.

Step1: Configure GPIO register as SMHC function by Port Controller module; reset clock by writing 1 to SMHC_BGR_REG[SMHCx_RST], open clock gating by writing 1 to SMHC_BGR_REG[SMHCx_GATING]; select clock sources and set division factor by configuring the SMHCx_CLK_REG(x=0,1,2) register.

Step2: Configure SMHC_CTRL to reset FIFO and controller, enable total interrupt; configure SMHC_INTMASK to 0xFFCE to enable normal interrupt and error abnormal interrupt, and register interrupt function.

Step3: Configure SMHC_CLKDIV to open clock for device; configure SMHC_CMD as change clock command(for example 0x80202000); send update clock command to deliver clock to device.

Step4: Configure SMHC_CMD to normal command, configure SMHC_CMDARG to set command parameter, configure SMHC_CMD to set response type, etc, then command can send. According to initial process in the protocol, you can finish SMHC initializing by sending corresponding command one by one.

5.3.4.2. Writing a Single Data Block

To Write a single data block, perform the following steps:

Step1: Write 0x1 to SMHC_CTRL[DMA_RST] to reset internal DMA controller; write 0x82 to SMHC_IDMAC to enable IDMAC interrupt, configure AHB master burst transfers; configure SMHC_IDIE to enable transfer interrupt, receive interrupt, and abnormal interrupt.

Step2: Configure SMHC_FIFOTH to determine burst size, TX/RX trigger level. For example, if SMHC_FIFOTH is configured as 0x300F00F0, which indicates that Burst size is 16, TX_TL is 15, RX_TL is 240. Configure SMHC_DLBA to determine the start address of DMA descriptor.

Step3: If writing 1 data block to the sector 1, then SMHC_BYCNT[BYTE_CNT] need be set to 0x200, the descriptor is set based on data size; set the data sector address of CMD24(Single Data Block Write) to 0x1, write 0x80002758 to SMHC_CMD, send CMD24 command to write data to device.

Step4: Check whether SMHC_RINTSTS[CC] is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.

Step5: Check whether SMHC_IDST_REG[TX_INT] is 1. If yes, writing DMA data transfer is complete, then write 0x337 to SMHC_IDST_REG to clear interrupt flag; if no, continue to wait until timeout, then exit process.

Step6: Check whether SMHC_RINTSTS[DTC] is 1. If yes, data transfer is complete and CMD24 writing operation is complete. If no, that is, abnormity exists. Read SMHC_RINTSTS,SMHC_STATUS to query existing abnormity.

Step7: Send CMD13 command to query whether device writing operation is complete and whether return to idle status. For example, device RCA is 0x1234, first set SMHC_CMDARG to 0x12340000, write 0x8000014D to SMHC_CMD, go to step4 to ensure command transfer completed, then check whether the highest bit of SMHC_RESP0(CMD13 response) is 1. If yes, device is in Idle status, then the next command can be sent. If no, device is in busy status, then continue to send CMD13 to wait device in the idle status until timeout exit.

5.3.4.3. Reading a Single Data Block

To read a single data block, perform the following steps:

Step1: Write 0x1 to SMHC_CTRL[DMA_RST] to reset internal DMA controller; write SMHC_IDMAC to 0x82 to enable IDMAC interrupt and configure AHB master burst transfers; configure SMHC_IDIE to enable transfer interrupt,receive interrupt, and abnormal interrupt.

Step2: Configure SMHC_FIFOTH to determine burst size, TX/RX trigger level. For example, if SMHC_FIFOTH is configured as 0x300F00F0, which indicates that Burst size is 16, TX_TL is 15,RX_TL is 240. Configure SMHC_DLBA to determine the start address of DMA descriptor.

Step3: If reading 1 data block from the sector 1, then SMHC_BYCNT[BYTE_CNT] need be set to 0x200, the descriptor is set based on data size; set the data sector address of CMD17 command(Single Data Block Read) to 0x1, write 0x80002351 to SMHC_CMD, send CMD17 command to read data from device to DRAM/SRAM.

Step4: Check whether SMHC_RINTSTS[CC] is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.

Step5: Check whether SMHC_IDST_REG[RX_INT] is 1. If yes, writing DMA data transfer is complete, then write 0x337 to SMHC_IDST_REG to clear interrupt flag; if no, continue to wait until timeout, then exit process.

Step6: Check whether SMHC_RINTSTS[DTC] is 1. If yes, data transfer is complete and CMD17 reading operation is complete. If no, that is, abnormity exists. Read SMHC_RINTSTS,SMHC_STATUS to query existing abnormity.

5.3.4.4. Writing Open-ended Multiple Data Blocks(CMD25+Auto CMD12)

To write open-ended multiple data blocks, perform the following steps:

Step1: Write 0x1 to SMHC_CTRL[DMA_RST] to reset internal DMA controller; write SMHC_IDMAC to 0x82 to enable IDMAC interrupt and configure AHB master burst transfers; configure SMHC_IDIE to enable transfer interrupt,receive interrupt, and abnormal interrupt.

Step2: Configure SMHC_FIFOTH to determine burst size, TX/RX trigger level. For example, if SMHC_FIFOTH is configured as 0x300F00F0, which indicates that Burst size is 16, TX_TL is 15, RX_TL is 240. Configure SMHC_DLBA to determine the start address of DMA descriptor.

Step3: If writing 3 data blocks to the sector 0, then SMHC_BYCNT[BYTE_CNT] need be set to 0x600, the descriptor is set based on data size; set the data sector address of CMD25 command(Multiple Data Blocks Write) to 0x0, write 0x80003759 to SMHC_CMD, send CMD25 command to write data to device, when data transfer is complete, CMD12 will be sent automatically .

Step4: Check whether SMHC_RINTSTS[CC] is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.

Step5: Check whether SMHC_IDST_REG[TX_INT] is 1. If yes, writing DMA data transfer is complete, then write 0x337 to SMHC_IDST_REG to clear interrupt flag; if no, continue to wait until timeout, then exit process.

Step6: Check whether SMHC_RINTSTS[ACD] and SMHC_RINTSTS[DTC] are all 1. If yes, data transfer is complete, CMD12 transfer is complete and CMD25 writing operation is complete. If no, that is, abnormity exists. Read SMHC_RINTSTS,SMHC_STATUS to query existing abnormity.

Step7: Send CMD13 command to query whether device writing operation is complete and whether return to idle status. For example, device RCA is 0x1234,first set SMHC_CMDARG to 0x12340000, write 0x8000014D to SMHC_CMD, go to step4 to ensure command transfer completed, then check whether the highest bit of SMHC_RESP0(CMD13 response) is 1. If yes, device is in Idle status,then the next command can be sent. If no, device is in busy status, then continue to send CMD13 to wait device in the idle status until timeout exit.

5.3.4.5. Reading Open-ended Multiple Data Blocks(CMD18+Auto CMD12)

To read open-ended multiple data blocks, perform the following steps:

Step1: Write 0x1 to SMHC_CTRL[DMA_RST] to reset internal DMA controller; write SMHC_IDMAC to 0x82 to enable IDMAC interrupt and configure AHB master burst transfers; configure SMHC_IDIE to enable transfer interrupt,receive interrupt, and abnormal interrupt.

Step2: Configure SMHC_FIFOTH to determine burst size, TX/RX trigger level. For example, if SMHC_FIFOTH is configured as 0x300F00F0, which indicates that Burst size is 16, TX_TL is 15, RX_TL is 240. Configure SMHC_DLBA to determine the start address of DMA descriptor.

Step3: If reading 3 data blocks from the sector 0, then SMHC_BYCNT[BYTE_CNT] need be set to 0x600, the descriptor is set based on data size; set the data sector address of CMD18 command(Multiple Data Blocks Read) to 0x0, write 0x80003352 to SMHC_CMD, send CMD18 command to read data to device, when data transfer is complete, CMD12 will be sent automatically.

Step4: Check whether SMHC_RINTSTS[CC] is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.

Step5: Check whether SMHC_IDST_REG[RX_INT] is 1. If yes, writing DMA data transfer is complete, then write 0x337 to SMHC_IDST_REG to clear interrupt flag; if no, continue to wait until timeout, then exit process.

Step6: Check whether SMHC_RINTSTS[ACD] and SMHC_RINTSTS[DTC] are all 1. If yes, data transfer is complete, CMD12 transfer is complete and CMD18 reading operation is complete. If no, that is, abnormity exists. Read SMHC_RINTSTS,SMHC_STATUS to query existing abnormity.

5.3.4.6. Writing Pre-defined Multiple Data Blocks(CMD23+CMD25)

To write pre-defined multiple data blocks, perform the following steps:

Step1: Write 0x1 to SMHC_CTRL[DMA_RST] to reset internal DMA controller; write SMHC_IDMAC to 0x82 to enable IDMAC interrupt and configure AHB master burst transfers; configure SMHC_IDIE to enable transfer interrupt,receive interrupt, and abnormal interrupt.

Step2: Configure SMHC_FIFOTH to determine burst size, TX/RX trigger level. For example, if SMHC_FIFOTH is configured as 0x300F00F0, which indicates that Burst size is 16, TX_TL is 15,RX_TL is 240. Configure SMHC_DLBA to determine the start address of DMA descriptor.

Step3: If writing 3 data blocks, setting SMHC_CMDARG to 0x3 to ensure the block number to be operated, writing 0x80000157 to SMHC_CMD to send CMD23 command. Check whether SMHC_RINTSTS[CC] is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.

Step4: SMHC_BYCNT[BYTE_CNT] need be set to 0x600, the descriptor is set based on data size; set the data sector address of CMD25 command(Multiple Data Blocks Write) to 0x0, write 0x80002759 to SMHC_CMD, send CMD25 command to write data to device.

Step5: Check whether SMHC_RINTSTS[CC] is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.

Step6: Check whether SMHC_IDST_REG[TX_INT] is 1. If yes, writing DMA data transfer is complete, then write 0x337 to SMHC_IDST_REG to clear interrupt flag; if no, continue to wait until timeout, then exit process.

Step7: Check whether SMHC_RINTSTS[DTC] is 1. If yes, data transfer is complete and CMD25 writing operation is complete. If no, that is, abnormity exists. Read SMHC_RINTSTS,SMHC_STATUS to query existing abnormity.

Step8: Send CMD13 command to query whether device writing operation is complete and whether return to idle status. For example, device RCA is 0x1234,first set SMHC_CMDARG to 0x12340000, write 0x8000014D to SMHC_CMD,go to step4 to ensure command transfer completed, then check whether the highest bit of SMHC_RESP0(CMD13 response) is 1. If yes, device is in Idle status,then the next command can be sent. If no, device is in busy status, then continue to send CMD13 to wait device in the idle status until timeout exit.

5.3.4.7. Reading Pre-defined Multiple Data Blocks(CMD23+CMD18)

To read pre-defined multiple data blocks, perform the following steps:

Step1: Write 0x1 to SMHC_CTRL[DMA_RST] to reset internal DMA controller; write SMHC_IDMAC to 0x82 to enable IDMAC interrupt and configure AHB master burst transfers; configure SMHC_IDIE to enable transfer interrupt,receive interrupt, and abnormal interrupt.

Step2: Configure SMHC_FIFOTH to determine burst size, TX/RX trigger level. For example, if SMHC_FIFOTH is configured as 0x300F00F0, which indicates that Burst size is 16, TX_TL is 15,RX_TL is 240. Configure SMHC_DLBA to determine the start address of DMA descriptor.

Step3: If reading 3 data blocks, setting SMHC_CMDARG to 0x3 to ensure the block number to be operated, writing 0x80000157 to SMHC_CMD to send CMD23 command. Check whether SMHC_RINTSTS[CC] is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.

Step4: SMHC_BYCNT[BYTE_CNT] need be set to 0x600, the descriptor is set based on data size; set the data sector address of CMD18(Multiple Data Blocks Read) to 0x0, write 0x80002352 to SMHC_CMD, send CMD18 command to read data from device to DRAM/SRAM.

Step5: Check whether SMHC_RINTSTS[CC] is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.

Step6: Check whether SMHC_IDST_REG[TX_INT] is 1. If yes, writing DMA data transfer is complete, then write 0x337 to SMHC_IDST_REG to clear interrupt flag; if no, continue to wait until timeout, then exit process.

Step7: Check whether SMHC_RINTSTS[DTC] is 1. If yes, data transfer is complete and CMD18 writing operation is complete. If no, that is, abnormity exists. Read SMHC_RINTSTS,SMHC_STATUS to query existing abnormity.

5.3.5. Register List

Module Name	Base Address
SMHC0	0x04020000
SMHC1	0x04021000
SMHC2	0x04022000

Register Name	Offset	Description
SMHC_CTRL	0x0000	Control Register
SMHC_CLKDIV	0x0004	Clock Control Register
SMHC_TMOUT	0x0008	Time Out Register
SMHC_CTYPE	0x000C	Bus Width Register
SMHC_BLKSIZ	0x0010	Block Size Register
SMHC_BYTCNT	0x0014	Byte Count Register
SMHC_CMD	0x0018	Command Register
SMHC_CMDARG	0x001C	Command Argument Register
SMHC_RESP0	0x0020	Response 0 Register
SMHC_RESP1	0x0024	Response 1 Register
SMHC_RESP2	0x0028	Response 2 Register
SMHC_RESP3	0x002C	Response 3 Register
SMHC_INTMASK	0x0030	Interrupt Mask Register
SMHC_MINTSTS	0x0034	Masked Interrupt Status Register
SMHC_RINTSTS	0x0038	Raw Interrupt Status Register
SMHC_STATUS	0x003C	Status Register
SMHC_FIFOTH	0x0040	FIFO Water Level Register
SMHC_FUNS	0x0044	FIFO Function Select Register
SMHC_TCBCNT	0x0048	Transferred Byte Count between Controller and Card
SMHC_TBBCNT	0x004C	Transferred Byte Count between Host Memory and Internal FIFO
SMHC_DBGC	0x0050	Current Debug Control Register
SMHC_CSDC	0x0054	CRC Status Detect Control Register (Only for SMHC2)
SMHC_A12A	0x0058	Auto Command 12 Argument Register
SMHC_NTSR	0x005C	SD New Timing Set Register (Only for SMHC0, SMHC1)
SMHC_HWRST	0x0078	Hardware Reset Register
SMHC_IDMAC	0x0080	IDMAC Control Register
SMHC_DLBA	0x0084	Descriptor List Base Address Register
SMHC_IDST	0x0088	IDMAC Status Register
SMHC_IDIE	0x008C	IDMAC Interrupt Enable Register
SMHC_THLD	0x0100	Card Threshold Control Register
SMHC_SFC	0x0104	Sample FIFO Control Register (Only for SMHC2)

SMHC_A23A	0x0108	Auto Command 23 Argument Register (Only for SMHC2)
EMMC_DDR_SBIT_DET	0x010C	eMMC4.5 DDR Start Bit Detection Control Register
SMHC_RES_CRC	0x0110	Response CRC from Device (Only for SMHC0, SMHC1)
SMHC_D7_CRC	0x0114	CRC in Data7 from Device (Only for SMHC0, SMHC1)
SMHC_D6_CRC	0x0118	CRC in Data6 from Device (Only for SMHC0, SMHC1)
SMHC_D5_CRC	0x011C	CRC in Data5 from Device (Only for SMHC0, SMHC1)
SMHC_D4_CRC	0x0120	CRC in Data4 from Device (Only for SMHC0, SMHC1)
SMHC_D3_CRC	0x0124	CRC in Data3 from Device (Only for SMHC0, SMHC1)
SMHC_D2_CRC	0x0128	CRC in Data2 from Device (Only for SMHC0, SMHC1)
SMHC_D1_CRC	0x012C	CRC in Data1 from Device (Only for SMHC0, SMHC1)
SMHC_D0_CRC	0x0130	CRC in Data0 from Device (Only for SMHC0, SMHC1)
SMHC_CRC_STA	0x0134	Write CRC Status Register (Only for SMHC0, SMHC1)
SMHC_EXT_CMD	0x0138	Extended Command Register (Only for SMHC2)
SMHC_EXT_RESP	0x013C	Extended Response Register (Only for SMHC2)
SMHC_DRV_DL	0x0140	Drive Delay Control Register
SMHC_SMAP_DL	0x0144	Sample Delay Control Register
SMHC_DS_DL	0x0148	Data Strobe Delay Control Register (Only for SMHC2)
SMHC_FIFO	0x0200	Read/Write FIFO

5.3.6. Register Description

5.3.6.1. 0x0000 SMHC Global Control Register(Default Value: 0x0000_0100)

Offset: 0x0000			**Register Name: SMHC_CTRL**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	FIFO_AC_MOD FIFO Access Mode 0: DMA bus 1: AHB bus
30:13	/	/	/
12	R/W	0x0	TIME_UNIT_CMD Time unit for command line Time unit is used to calculate command line time out value defined in RTO_LMT. 0: 1 card clock period 1: 256 card clock period
11	R/W	0x0	TIME_UNIT_DAT Time unit for data line Time unit is used to calculate data line time out value defined in DTO_LMT. 0: 1 card clock period 1: 256 card clock period
10	R/W	0x0	DDR_MOD_SEL

			DDR Mode Select Although eMMC's HS400 speed mode is 8-bit DDR, this filed should be cleared when HS400_MD_EN is set. 0: SDR mode 1: DDR mode
9	/	/	/
8	R/W	0x1	CD_DBC_ENB Card Detect (Data[3] status) De-bounce Enable 0: Disable de-bounce 1: Enable de-bounce
7:6	/	/	/
5	R/W	0x0	DMA_ENB DMA Global Enable 0: Disable DMA to transfer data, using AHB bus 1: Enable DMA to transfer data
4	R/W	0x0	INT_ENB Global Interrupt Enable 0: Disable interrupts 1: Enable interrupts
3	/	/	/
2	R/W	0x0	DMA_RST DMA Reset
1	R/W	0x0	FIFO_RST FIFO Reset 0: No change 1: Reset FIFO This bit is auto-cleared after completion of reset operation.
0	R/W	0x0	SOFT_RST Software Reset 0: No change 1: Reset SD/MMC controller This bit is auto-cleared after completion of reset operation.

5.3.6.2. 0x0004 SMHC Clock Control Register(Default Value: 0x0000_0000)

Offset: 0x0004			**Register Name: SMHC_CLKDIV**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MASK_DATA0 0: Do not mask data0 when update clock 1: Mask data0 when update clock
30:18	/	/	/
17	R/W	0x0	CCLK_CTRL Card Clock Output Control

			0: Card clock always on 1: Turn off card clock when FSM is in IDLE state
16	R/W	0x0	CCLK_ENB Card Clock Enable 0: Card Clock off 1: Card Clock on
15:8	/	/	/
7:0	R/W	0x0	CCLK_DIV Card Clock Divider n: Source clock is divided by 2*n.(n=0~255) when HS400_MD_EN is set, this field must be cleared.

5.3.6.3. 0x0008 SMHC Timeout Register(Default Value:0xFFFF_FF40)

Offset: 0x0008			Register Name: SMHC_TMOUT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	R/W	0xffffff	DTO_LMT Data Timeout Limit This field can set time of the Host wait for the data from the Device. Ensure to communicate with the Device, this field must be set to maximum that greater than the time N_{AC}. About the N_{AC}, the explanation is as follows: When Host read data,data transmission from the Device starts after the access time delay N_{AC} beginning from the end bit of the read command(ACMD51,CMD8,CMD17,CMD18). When Host read multiple block(CMD18), a next block's data transmission from the Device starts after the access time delay N_{AC} beginning from the end bit of the previous block. When Host write data, the value is no effect.
7:0	R/W	0x40	RTO_LMT Response Timeout Limit

5.3.6.4. 0x000C SMHC Bus Width Register(Default Value:0x0000_0000)

Offset: 0x000C			Register Name: SMHC_CTYPE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1:0	R/W	0x0	CARD_WID Card Width 00: 1-bit width 01: 4-bit width 1x: 8-bit width

5.3.6.5. 0x0010 SMHC Block Size Register(Default Value:0x0000_0200)

Offset: 0x0010			Register Name: SMHC_BLKSIZ
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x200	BLK_SZ Block Size

5.3.6.6. 0x0014 SMHC Byte Count Register(Default Value:0x0000_0200)

Offset: 0x0014			Register Name: SMHC_BYTCNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x200	BYTE_CNT Byte counter Number of bytes to be transferred. It must be integer multiple of Block Size(BLK_SZ) for block transfers.

5.3.6.7. 0x0018 SMHC Command Register(Default Value:0x0000_0000)

Offset: 0x0018			Register Name: SMHC_CMD
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CMD_LOAD Start Command This bit is auto cleared when current command is sent. If there is no any response error happened, a command complete interrupt bit (CMD_OVER) will be set in interrupt register. You should not write any other command before this bit is cleared.
30:29	/	/	/
28	R/W	0x0	VOL_SW Voltage Switch 0: normal command 1: Voltage switch command, set for CMD11 only
27	R/W	0x0	BOOT_ABT Boot Abort Setting this bit will terminate the boot operation.
26	R/W	0x0	EXP_BOOT_ACK Expect Boot Acknowledge When software sets this bit along in mandatory boot operation, the controller expects a boot acknowledge start pattern of 0-1-0 from the selected card.
25:24	R/W	0x0	BOOT_MOD Boot Mode

			00: Normal command 01: Mandatory Boot operation 10: Alternate Boot operation 11: Reserved
23:22	/	/	/
21	R/W	0x0	PRG_CLK Change Clock 0: Normal command 1: Change Card Clock When this bit is set, controller will change clock domain and clock output. No command will be sent.
20:16	/	/	/
15	R/W	0x0	SEND_INIT_SEQ Send Initialization 0: Normal command sending 1: Send initialization sequence before sending this command.
14	R/W	0x0	STOP_ABT_CMD Stop Abort Command 0: Normal command sending 1: Send *Stop* or *Abort* command to stop current data transfer in progress.(CMD12, CMD52 for writing “I/O Abort” in SDIO CCCR)
13	R/W	0x0	WAIT_PRE_OVER Wait Data Transfer Over 0: Send command at once, do not care of data transferring 1: Wait for data transfer completion before sending current command
12	R/W	0x0	STOP_CMD_FLAG Send Stop CMD Automatically (CMD12) 0: Do not send stop command at end of data transfer 1: Send stop command automatically at end of data transfer If set, the **SMHC_RESP1** will record the response of auto CMD12.
11	R/W	0x0	TRANS_MODE Transfer Mode 0: Block data transfer command 1: Stream data transfer command
10	R/W	0x0	TRANS_DIR Transfer Direction 0: Read operation 1: Write operation
9	R/W	0x0	DATA_TRANS Data Transfer 0: Without data transfer 1: With data transfer
8	R/W	0x0	CHK_RESP_CRC Check Response CRC 0: Do not check response CRC

			1: Check response CRC
7	R/W	0x0	LONG_RESP Response Type 0:Short Response (48 bits) 1:Long Response (136 bits)
6	R/W	0x0	RESP_RCV Response Receive 0: Command without response 1: Command with response
5:0	R/W	0x0	CMD_IDX CMD Index Command index value

5.3.6.8. 0x001C SMHC Command Argument Register(Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: SMHC_CMDARG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	CMD_ARG Command argument

5.3.6.9. 0x0020 SMHC Response 0 Register(Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: SMHC_RESP0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	CMD_RESP0 Response 0 Bit[31:0] of response

5.3.6.10. 0x0024 SMHC Response 1 Register(Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: SMHC_RESP1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	CMD_RESP1 Response 1 Bit[63:31] of response

5.3.6.11. 0x0028 SMHC Response 2 Register(Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: SMHC_RESP2
Bit	**Read/Write**	**Default/Hex**	**Description**

31:0	R	0x0	CMD_RESP2 Response 2 Bit[95:64] of response

5.3.6.12. 0x002C SMHC Response 3 Register(Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: SMHC_RESP3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	CMD_RESP3 Response 3 Bit[127:96] of response

5.3.6.13. 0x0030 SMHC Interrupt Mask Register(Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: SMHC_INTMASK
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CARD_REMOVAL_INT_EN Card Removed Interrupt Enable
30	R/W	0x0	CARD_INSERT_INT_EN Card Inserted Interrupt Enable
29:17	/	/	/
16	R/W	0x0	SDIO_INT_EN SDIO Interrupt Enable
15	R/W	0x0	DEE_INT_EN Data End-bit Error Interrupt Enable
14	R/W	0x0	ACD_INT_EN Auto Command Done Interrupt Enable
13	R/W	0x0	DSE_BC_INT_EN Data Start Error Interrupt Enable
12	R/W	0x0	CB_IW_INT_EN Command Busy and Illegal Write Interrupt Enable
11	R/W	0x0	FU_FO_INT_EN FIFO Underrun/Overflow Interrupt Enable
10	R/W	0x0	DSTO_VSD_INT_EN Data Starvation Timeout/V1.8 Switch Done Interrupt Enable
9	R/W	0x0	DTO_BDS_INT_EN Data Timeout/Boot Data Start Interrupt Enable
8	R/W	0x0	RTO_BACK_INT_EN Response Timeout/Boot ACK Received Interrupt Enable
7	R/W	0x0	DCE_INT_EN Data CRC Error Interrupt Enable
6	R/W	0x0	RCE_INT_EN

			Response CRC Error Interrupt Enable
5	R/W	0x0	DRR_INT_EN Data Receive Request Interrupt Enable
4	R/W	0x0	DTR_INT_EN Data Transmit Request Interrupt Enable
3	R/W	0x0	DTC_INT_EN Data Transfer Complete Interrupt Enable
2	R/W	0x0	CC_INT_EN Command Complete Interrupt Enable
1	R/W	0x0	RE_INT_EN Response Error Interrupt Enable
0	/	/	/

5.3.6.14. 0x0034 SMHC Masked Interrupt Status Register(Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: SMHC_MINTSTS
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	M_CARD_REMOVAL_INT Card Removed
30	R/W	0x0	M_CARD_INSERT Card Inserted
29:17	/	/	/
16	R/W	0x0	M_SDIO_INT SDIO Interrupt
15	R/W	0x0	M_DEE_INT Data End-bit Error When set during receiving data, it means that host controller does not receive valid data end bit. When set during transmitting data, it means that host controller does not receive CRC status taken or received CRC status taken is negative.
14	R/W	0x0	M_ACD_INT Auto Command Done When set, it means auto stop command(CMD12) completed.
13	R/W	0x0	M_DSE_BC_INT Data Start Error When set during receiving data, it means that host controller found an error start bit. When set during transmitting data, it means that busy signal is cleared.
12	R/W	0x0	M_CB_IW_INT Command Busy and Illegal Write
11	R/W	0x0	M_FU_FO_INT FIFO Underrun/Overflow
10	R/W	0x0	M_DSTO_VSD_INT

			Data Starvation Timeout/V1.8 Switch Done
9	R/W	0x0	M_DTO_BDS_INT Data Timeout/Boot Data Start
8	R/W	0x0	M_RTO_BACK_INT Response Timeout/Boot ACK Received
7	R/W	0x0	M_DCE_INT Data CRC Error When set during receiving data, it means that the received data have data CRC error. When set during transmitting data, it means that the received CRC status taken is negative.
6	R/W	0x0	M_RCE_INT Response CRC Error
5	R/W	0x0	M_DRR_INT Data Receive Request When set, it means that there are enough data in FIFO during receiving data.
4	R/W	0x0	M_DTR_INT Data Transmit Request When set, it means that there are enough space in FIFO during transmitting data.
3	R/W	0x0	M_DTC_INT Data Transfer Complete
2	R/W	0x0	M_CC_INT Command Complete
1	R/W	0x0	M_RE_INT Response Error When set, Transmit Bit error or End Bit error or CMD Index error may occurs.
0	/	/	/

5.3.6.15. 0x0038 SMHC Raw Interrupt Status Register(Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: SMHC_RINTSTS
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W1C	0x0	CARD_REMOVAL Card Removed This is write-1-to-clear bits.
30	R/W1C	0x0	CARD_INSERT Card Inserted This is write-1-to-clear bits.
29:17	/	/	/
16	R/W1C	0x0	SDIOI_INT

			SDIO Interrupt This is write-1-to-clear bits.
15	R/W1C	0x0	DEE Data End-bit Error When set during receiving data, it means that host controller does not receive valid data end bit. When set during transmitting data, it means that host controller does not receive CRC status taken. This is write-1-to-clear bits.
14	R/W1C	0x0	ACD Auto Command Done When set, it means auto stop command(CMD12) completed. This is write-1-to-clear bits.
13	R/W1C	0x0	DSE_BC Data Start Error When set during receiving data, it means that host controller found a error start bit. It is valid at 4-bit or 8-bit bus mode. When it set, host finds start bit at data0, but does not find start bit at some or all of the other data lines. When set during transmitting data, it means that busy signal is cleared. This is write-1-to-clear bits.
12	R/W1C	0x0	CB_IW Command Busy and Illegal Write This is write-1-to-clear bits.
11	R/W1C	0x0	FU_FO FIFO Underrun/Overflow This is write-1-to-clear bits.
10	R/W1C	0x0	DSTO_VSD Data Starvation Timeout/V1.8 Switch Done This is write-1-to-clear bits.
9	R/W1C	0x0	DTO_BDS Data Timeout/Boot Data Start When set during receiving data, it means host does not find start bit on data0. This is write-1-to-clear bits.
8	R/W1C	0x0	RTO_BACK Response Timeout/Boot ACK Received This is write-1-to-clear bits.
7	R/W1C	0x0	DCE Data CRC Error When set during receiving data, it means that the received data have data CRC error. When set during transmitting data, it means that the received CRC status taken is negative. This is write-1-to-clear bits.

6	R/W1C	0x0	RCE Response CRC Error This is write-1-to-clear bits.
5	R/W1C	0x0	DRR Data Receive Request When set, it means that there are enough data in FIFO during receiving data. This is write-1-to-clear bits.
4	R/W1C	0x0	DTR Data Transmit Request When set, it means that there are enough space in FIFO during transmitting data. This is write-1-to-clear bits.
3	R/W1C	0x0	DTC Data Transfer Complete When set, it means that current command completes even through error occurs. This is write-1-to-clear bits.
2	R/W1C	0x0	CC Command Complete When set, it means that current command completes even through error occurs. This is write-1-to-clear bits.
1	R/W1C	0x0	RE Response Error When set, Transmit Bit error or End Bit error or CMD Index error may occur. This is write-1-to-clear bits.
0	/	/	/

5.3.6.16. 0x003C SMHC Status Register(Default Value: 0x0000_0006)

Offset: 0x003C			**Register Name: SMHC_STATUS**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x0	DMA_REQ DMA Request DMA request signal state
30:26	/	/	/
25:17	R	0x0	FIFO_LEVEL FIFO Level Number of filled locations in FIFO
16:11	R	0x0	RESP_IDX Response Index Index of previous response, including any auto-stop sent by controller

10	R	0x0	FSM_BUSY Data FSM Busy Data transmit or receive state-machine is busy
9	R	0x0	CARD_BUSY Card Data Busy Inverted version of DATA[0] 0: card data not busy 1: card data busy
8	R	0x0	CARD_PRESENT Data[3] Status Level of DATA[3], checks whether card is present 0: card not present 1: card present
7:4	R	0x0	FSM_STA Command FSM States 0000: Idle 0001: Send init sequence 0010: TX CMD start bit 0011: TX CMD TX bit 0100: TX CMD index + argument 0101: TX CMD CRC7 0110: TX CMD end bit 0111: RX response start bit 1000: RX response IRQ response 1001: RX response TX bit 1010: RX response CMD index 1011: RX response data 1100: RX response CRC7 1101: RX response end bit 1110: CMD path wait NCC 1111: Wait; CMD-to-response turnaround
3	R	0x0	FIFO_FULL FIFO Full 1: FIFO full 0: FIFO not full
2	R	0x1	FIFO_EMPTY FIFO Empty 1: FIFO Empty 0: FIFO not Empty
1	R	0x1	FIFO_TX_LEVEL FIFO TX Water Level Flag 0: FIFO didn’t reach transmit trigger level 1: FIFO reached transmit trigger level
0	R	0x0	FIFO_RX_LEVEL FIFO RX Water Level Flag

			0: FIFO didn't reach receive trigger level 1: FIFO reached receive trigger level

5.3.6.17. 0x0040 SMHC FIFO Water Level Register(Default Value: 0x000F_0000)

Offset: 0x0040			Register Name: SMHC_FIFOTH
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30:28	R/W	0x0	BSIZE_OF_TRANS Burst Size of Multiple Transaction 000: 1 transfers 001: 4 010: 8 011: 16 Others: Reserved It should be programmed same as DMA controller multiple transaction size. The units for transfers are the DWORD. A single transfer would be signaled based on this value. Value should be sub-multiple of (RX_TL + 1) and (FIFO_DEPTH - TX_TL) FIFO_DEPTH = 256, FIFO_SIZE = 256 * 32 = 1K Recommended: MSize = 16, TX_TL = 240, RX_TL = 15 (for SMHC2) MSize = 8, TX_TL = 248, RX_TL = 7 (for SMHC0,SMHC1)
27:24	/	/	/
23:16	R/W	0xF	RX_TL RX Trigger Level 0x0~0xFE: RX Trigger Level is 0~254 0xFF: Reserved FIFO threshold when FIFO request host to receive data from FIFO. When FIFO data level is greater than this value, DMA is request is raised if DMA enabled, or RX interrupt bit is set if interrupt enabled. At the end of packet, if the last transfer is less than this level, the value is ignored and relative request will be raised as usual. Recommended: 15 (means greater than 15, for SMHC2) 7 (means greater than 7, for SMHC0,SMHC1)
15:8	/	/	/
7:0	R/W	0x0	TX_TL TX Trigger Level 0x1~0xFF: TX Trigger Level is 1~255 0x0: No trigger FIFO threshold when FIFO requests host to transmit data to FIFO. When

			FIFO data level is less than or equal to this value, DMA TX request is raised if DMA enabled, or TX request interrupt bit is set if interrupt enabled. At the end of packet, if the last transfer is less than this level, the value is ignored and relative request will be raised as usual. Recommended: 240(means less than or equal to 240, for SMHC2) 248(means less than or equal to 248, for SMHC0,SMHC1)

5.3.6.18. 0x0044 SMHC Function Select Register(Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: SMHC_FUNS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x0	ABT_RDATA Abort Read Data 0: Ignored 1: After suspend command is issued during read-transfer, software polls card to find when suspend happened. Once suspend occurs, software sets bit to reset data state-machine, which is waiting for next block of data. Used in SDIO card suspends sequence. This bit is auto-cleared once controller reset to idle state.
1	R/W	0x0	READ_WAIT Read Wait 0: Clear SDIO read wait 1: Assert SDIO read wait
0	R/W	0x0	HOST_SEND_MMC_IRQRESQ Host Send MMC IRQ Response 0: Ignored 1: Send auto IRQ response When host is waiting MMC card interrupt response, setting this bit will make controller cancel wait state and return to idle state, at which time, controller will receive IRQ response sent by itself. This bit is auto-cleared after response is sent.

5.3.6.19. 0x0048 SMHC Transferred Byte Count Register 0 (Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: SMHC_TBC0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	TBC0 Transferred Count 0 Number of bytes transferred between card and internal FIFO.

			The register should be accessed in full to avoid read-coherency problems and read only after data transfer completes.

5.3.6.20. 0x004C SMHC Transferred Byte Count Register 1 (Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: SMHC_TBC1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	TBC1 Transferred Count 1 Number of bytes transferred between Host/DMA memory and internal FIFO. The register should be accessed in full to avoid read-coherency problems and read only after data transfer completes.

5.3.6.21. 0x0054 SMHC CRC Status Detect Control Register(Default Value: 0x0000_0003)

Offset: 0x0054			Register Name: SMHC_CSDC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3:0	R/W	0x3	CRC_DET_PARA 110: HS400 speed mode 011: Other speed mode

NOTE

The register is only for SMHC2.

5.3.6.22. 0x0058 SMHC Auto Command 12 Argument Register (Default Value: 0x0000_FFFF)

Offset: 0x0058			Register Name: SMHC_A12A
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0xffff	SD_A12A Auto CMD12 Argument SD_A12A set the argument of command 12 automatically send by controller.

5.3.6.23. 0x005C SMHC New Timing Set Register (Default Value: 0x8171_0000)

Offset: 0x005C			Register Name: SMHC_NTSR
Bit	**Read/Write**	**Default/Hex**	**Description**

31	R/W	0x1	MODE_SELEC 0: Old mode of Sample/Output Timing 1: New mode of Sample/Output Timing
30:28	/	/	/
27	R/W	0x0	DAT0_BYPASS Select data0 input asyn or bypass sample logic, it is used to check card busy or not. 0: Enable data0 bypass 1: Disable data0 bypass
26:25	/	/	/
24	R/W	0x1	CMD_DAT_RX_PHASE_CLR Clear command line's and data lines' input phase during update clock operation. 0: Disable 1: Enable
23	/	/	/
22	R/W	0x1	DAT_CRC_STATUS_RX_PHASE_CLR Clear data lines' input phase before receive CRC status. 0: Disable 1: Enable
21	R/W	0x1	DAT_TRANS_RX_PHASE_CLR Clear data lines' input phase before transfer data. 0: Disable 1: Enable
20	R/W	0x1	DAT_RECV_RX_PHASE_CLR Clear data lines' input phase before receive data. 0: Disable 1: Enable
19:17	/	/	/
16	R/W	0x1	CMD_SEND_RX_PHASE_CLR Clear command rx phase before send command. 0: Disable 1: Enable
15:10	/	/	/
9:8	R/W	0x0	DAT_SAMPLE_TIMING_PHASE 00: Sample timing phase offset 90^0 01: Sample timing phase offset 180^0 10: Sample timing phase offset 270^0 11: Ignore
7:6	/	/	/
5:4	R/W	0x0	CMD_SAMPLE_TIMING_PHASE 00: Sample timing phase offset 90^0 01: Sample timing phase offset 180^0 10: Sample timing phase offset 270^0

			11: Ignore
3:0	/	/	/

NOTE

This register is valid for SMHC0,SMHC1.

5.3.6.24. 0x0078 SMHC Hardware Reset Register (Default Value: 0x0000_0001)

Offset: 0x0078			Register Name: SMHC_HWRST
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x1	HW_RST 1: Active mode 0: Reset These bits cause the cards to enter pre-idle state, which requires them to be re-initialized.

5.3.6.25. 0x0080 SMHC IDMAC Control Register (Default Value: 0x0000_0000)

Offset: 0x0080			Register Name: SMHC_IDMAC
Bit	**Read/Write**	**Default/Hex**	**Description**
31	W	0x0	DES_LOAD_CTRL When IDMAC fetches a descriptor, if the valid bit of a descriptor is not set, IDMAC FSM will go to the suspend state. Setting this bit will make IDMAC refetch descriptor again and do the transfer normally.
30:11	/	/	/
10:8	R	0x0	Reserved
7	R/W	0x0	IDMAC_ENB IDMAC Enable When set, the IDMAC is enabled.
6:2	R/W	0x0	Reserved
1	R/W	0x0	FIX_BUST_CTRL Fixed Burst Controls whether the AHB Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 during start of normal burst transfers. When reset, the AHB will use SINGLE and INCR burst transfer operations.
0	R/W	0x0	IDMAC_RST DMA Reset When set, the DMA Controller resets all its internal registers. SWR is read/write. It is automatically cleared after 1 clock cycle.

5.3.6.26. 0x0084 SMHC Descriptor List Base Address Register (Default Value: 0x0000_0000)

Offset: 0x0084			Register Name: SMHC_DLBA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	DES_BASE_ADDR Start of Descriptor List Contains the base address of the First Descriptor. For SMHC0, SMHC1, it is a word address. For SMHC2, it is a byte address.

5.3.6.27. 0x0088 SMHC IDMAC Status Register (Default Value: 0x0000_0000)

Offset: 0x0088			Register Name: SMHC_IDST_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16:13	R	0x0	Reserved
12:10	R	0x0	IDMAC_ERR_STA Error Bits Indicates the type of error that caused a Bus Error. Valid only with Fatal Bus Error bit (IDSTS[2]) set. This field does not generate an interrupt. 001: Host Abort received during transmission 010: Host Abort received during reception Others: Reserved The bit is read-only.
9	R/W1C	0x0	ABN_INT_SUM(AIS) Abnormal Interrupt Summary Logical OR of the following: IDSTS[2]: Fatal Bus Interrupt IDSTS[4]: Descriptor Unavailable bit Interrupt IDSTS[5]: Card Error Summary Interrupt Only unmasked bits affect this bit. This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared. Writing a 1 clears this bit.
8	R/W1C	0x0	NOR_INT_SUM(NIS) Normal Interrupt Summary Logical OR of the following: IDSTS[0]: Transmit Interrupt IDSTS[1]: Receive Interrupt Only unmasked bits affect this bit. This is a sticky bit and must be cleared each time a corresponding bit that causes NIS to be set is cleared. Writing a 1 clears this bit.
7:6	/	/	/
5	R/W1C	0x0	ERR_FLAG_SUM

			Card Error Summary Indicates the status of the transaction to/from the card; also present in RINTSTS. Indicates the logical OR of the following bits: EBE: End Bit Error RTO: Response Timeout/Boot ACK Timeout RCRC: Response CRC SBE: Start Bit Error DRTO: Data Read Timeout/BDS timeout DCRC: Data CRC for Receive RE: Response Error Writing a 1 clears this bit.
4	R/W1C	0x0	DES_UNAVL_INT Descriptor Unavailable Interrupt This bit is set when the descriptor is unavailable due to OWN bit = 0 (DES0[31] =0). Writing a 1 clears this bit.
3	/	/	/
2	R/W1C	0x0	FATAL_BERR_INT Fatal Bus Error Interrupt Indicates that a Bus Error occurred (IDSTS[12:10]). When this bit is set, the DMA disables all its bus accesses. Writing a 1 clears this bit.
1	R/W1C	0x0	RX_INT Receive Interrupt Indicates the completion of data reception for a descriptor. Writing a 1 clears this bit.
0	R/W1C	0x0	TX_INT Transmit Interrupt Indicates that data transmission is finished for a descriptor. Writing a '1' clears this bit.

5.3.6.28. 0x008C SMHC IDMAC Interrupt Enable Register (Default Value: 0x0000_0000)

Offset: 0x008C			**Register Name: SMHC_IDIE_REG**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9:8	R/W	0x0	Reserved
7:6	/	/	/
5	R/W	0x0	ERR_SUM_INT_ENB Card Error Summary Interrupt Enable. When setting, it enables the Card Interrupt Summary.
4	R/W	0x0	DES_UNAVL_INT_ENB Descriptor Unavailable Interrupt. When setting along with Abnormal Interrupt Summary Enable, the

			Descriptor Unavailable interrupt is enabled.
3	/	/	/
2	R/W	0x0	FERR_INT_ENB Fatal Bus Error Enable When setting with Abnormal Interrupt Summary Enable, the Fatal Bus Error Interrupt is enabled. When reset, Fatal Bus Error Enable Interrupt is disabled.
1	R/W	0x0	RX_INT_ENB Receive Interrupt Enable. When setting with Normal Interrupt Summary Enable, Receive Interrupt is enabled. When reset, Receive Interrupt is disabled.
0	R/W	0x0	TX_INT_ENB Transmit Interrupt Enable. When setting with Normal Interrupt Summary Enable, Transmit Interrupt is enabled. When reset, Transmit Interrupt is disabled.

5.3.6.29. 0x0100 SMHC Card Threshold Control Register (Default Value: 0x0000_0000)

Offset: 0x0100			Register Name: SMHC_THLD
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0x0	CARD_RD_THLD Card Read Threshold Size
15:3	/	/	/
2	R/W	0x0	CARD_WR_THLD_ENB (only for SMHC2) Card Write Threshold Enable(HS400) 0: Card write threshold disabled 1: Card write threshold enabled Host controller initiates write transfer only if card threshold amount of data is available in transmit FIFO
1	R/W	0x0	BCIG (only for SMHC2) Busy Clear Interrupt Generation 0: Busy clear interrupt disabled 1: Busy clear interrupt enabled The application can disable this feature if it does not want to wait for a Busy Clear Interrupt.
0	R/W	0x0	CARD_RD_THLD_ENB Card Read Threshold Enable 0: Card read threshold disabled 1: Card read threshold enabled Host controller initiates Read Transfer only if CARD_RD_THLD amount of space is available in receive FIFO.

5.3.6.30. 0x0104 SMHC Sample FIFO Control Register (Default Value: 0x0000_0006)

Offset: 0x0104			Register Name: SMHC_SFC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4:1	R/W	0x3	STOP_CLK_CTRL Stop Clock Control When receiving data, if CARD_RD_THLD_ENB is set and CARD_RD_THLD is set same with BLK_SZ, the device clock may stop at block gap during data receiving. This field is used to control the position of stopping clock. The value can be changed between 0x0 and 0xF, but actually the available value and the position of stopping clock must be decided by the actual situation. The value increases one in this field is linked to one cycle(two cycles in DDR mode) that the position of stopping clock moved up.
0	R/W	0x0	BYPASS_EN Bypass enable When set, sample FIFO will be bypassed.

NOTE

The register is for SMHC2.

5.3.6.31. 0x0108 SMHC Auto Command 23 Argument Register (Default Value: 0x0000_0000)

Offset: 0x0108			Register Name: SMHC_A23A
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	A23A Auto CMD23 Argument The argument of command 23 is automatically sent by controller with this field.

NOTE

The register is for SMHC2.

5.3.6.32. 0x010C SMHC eMMC4.5 DDR Start Bit Detection Control Register (Default Value: 0x0000_0000)

Offset: 0x010C			Register Name: EMMC_DDR_SBIT_DET
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	HS400_MD_EN(for SMHC2 only) HS400 Mode Enable 0: Disable 1: Enable

			It is required to set this bit to '1' before initiating any data transfer CMD in HS400 mode.
30:1	/	/	/
0	R/W	0x0	HALF_START_BIT Control for start bit detection mechanism inside mstorage based on duration of start bit. For eMMC 4.5, start bit can be: 0: Full cycle 1: Less than one full cycle Set HALF_START_BIT=1 for eMMC 4.5 and above; set to 0 for SD applications.

5.3.6.33. 0x0110 SMHC Response CRC Register (Default Value: 0x0000_0000)

Offset: 0x0110			Register Name: SMHC_RESP_CRC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6:0	R	0x0	RESP_CRC Response CRC Response CRC from device.

NOTE

This register is valid for SMHC0, SMHC1.

5.3.6.34. 0x0114 SMHC Data7 CRC Register (Default Value: 0x0000_0000)

Offset: 0x0114			Register Name: SMHC_DAT7_CRC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DAT7_CRC Data[7] CRC CRC in data[7] from device. In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data. In 4 bits DDR mode, it is not used. In SDR mode, the higher 16 bits indicate the CRC of all data.

NOTE

This register is valid for SMHC0, SMHC1.

5.3.6.35. 0x0118 SMHC Data6 CRC Register (Default Value: 0x0000_0000)

Offset: 0x0118	Register Name: SMHC_DAT6_CRC

Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	DAT6_CRC Data[6] CRC CRC in data[6] from device. In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data. In 4 bits DDR mode, it is not used. In SDR mode, the higher 16 bits indicate the CRC of all data.

NOTE

This register is valid for SMHC0, SMHC1.

5.3.6.36. 0x011C SMHC Data5 CRC Register (Default Value: 0x0000_0000)

Offset: 0x011C			Register Name: SMHC_DAT5_CRC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DAT5_CRC Data[5] CRC CRC in data[5] from device. In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data. In 4 bits DDR mode, it is not used. In SDR mode, the higher 16 bits indicate the CRC of all data.

NOTE

This register is valid for SMHC0, SMHC1.

5.3.6.37. 0x0120 SMHC Data4 CRC Register (Default Value: 0x0000_0000)

Offset: 0x0120			Register Name: SMHC_DAT4_CRC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DAT4_CRC Data[4] CRC CRC in data[4] from device. In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data. In 4 bits DDR mode, the higher 16 bits indicate the CRC of odd data, and the lower 16 bits indicate the CRC of even data. In SDR mode, the higher 16 bits indicate the CRC of all data.

NOTE

This register is valid for SMHC0, SMHC1.

5.3.6.38. 0x0124 SMHC Data3 CRC Register (Default Value: 0x0000_0000)

Offset: 0x0124			Register Name: SMHC_DAT3_CRC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DAT3_CRC Data[3] CRC CRC in data[3] from device. In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data. In 4 bits DDR mode, the higher 16 bits indicate the CRC of odd data, and the lower 16 bits indicate the CRC of even data. In SDR mode, the higher 16 bits indicate the CRC of all data.

NOTE

This register is valid for SMHC0, SMHC1.

5.3.6.39. 0x0128 SMHC Data2 CRC Register (Default Value: 0x0000_0000)

Offset: 0x0128			Register Name: SMHC_DAT2_CRC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DAT2_CRC Data[2] CRC CRC in data[2] from device. In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data. In 4 bits DDR mode, the higher 16 bits indicate the CRC of odd data, and the lower 16 bits indicate the CRC of even data. In SDR mode, the higher 16 bits indicate the CRC of all data.

NOTE

This register is valid for SMHC0, SMHC1.

5.3.6.40. 0x012C SMHC Data1 CRC Register (Default Value: 0x0000_0000)

Offset: 0x012C			Register Name: SMHC_DAT1_CRC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DAT1_CRC Data[1] CRC CRC in data[1] from device. In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data. In 4 bits DDR mode, the higher 16 bits indicate the CRC of odd data, and the lower 16 bits indicate the CRC of even data.

			In SDR mode, the higher 16 bits indicate the CRC of all data.

NOTE

This register is valid for SMHC0, SMHC1.

5.3.6.41. 0x0130 SMHC Data0 CRC Register (Default Value: 0x0000_0000)

Offset: 0x0130			Register Name: SMHC_DAT0_CRC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DAT0_CRC Data[0] CRC CRC in data[0] from device. In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data. In 4 bits DDR mode, the higher 16 bits indicate the CRC of odd data, and the lower 16 bits indicate the CRC of even data. In SDR mode, the higher 16 bits indicate the CRC of all data.

NOTE

This register is valid for SMHC0, SMHC1.

5.3.6.42. 0x0134 SMHC CRC Status Register (Default Value: 0x0000_0000)

Offset: 0x0134			Register Name: SMHC_CRC_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2:0	R	0x0	CRC_STA CRC Status CRC status from device in write operation Positive CRC status token: 3’b010 Negative CRC status token: 3’b101

NOTE

This register is valid for SMHC0, SMHC1.

5.3.6.43. 0x0138 SMHC Extended Command Register (Default Value: 0x0000_0000)

Offset: 0x0138			Register Name: SMHC_EXT_CMD
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	AUTO_CMD23_EN

			Send CMD23 Automatically When setting this bit, send CMD23 automatically before send command specified in SMHC_CMD register. When SOFT_RST is set, this field will be cleared.

NOTE

This register is valid for SMHC2.

5.3.6.44. 0x013C SMHC Extended Response Register (Default Value: 0x0000_0000)

Offset: 0x013C			Register Name: SMHC_EXT_RESP
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	SMHC_EXT_RESP When AUTO_CMD23_EN is set, this register stores the response of CMD23.

NOTE

This register is valid for SMHC2.

5.3.6.45. 0x0140 SMHC Drive Delay Control Register (Default Value: 0x0001_0000)

Offset: 0x0140			Register Name: SMHC_DRV_DL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R/W	0x0	DAT_DRV_PH_SEL Data Drive Phase Select For SMHC0,SMHC1: 0: Data drive phase offset is 90^0 at SDR mode, 45^0 at DDR mode 1: Data drive phase offset is 180^0 at SDR mode, 90^0 at DDR mode For SMHC2: 0: Data drive phase offset is 90^0 at SDR mode, 45^0 at DDR8 mode, 90^0 at DDR4/HS400 mode 1: Data drive phase offset is 180^0 at SDR mode, 90^0 at DDR8 mode, 0^0 at DDR4/HS400 mode
16	R/W	0x1	CMD_DRV_PH_SEL Command Drive Phase Select For SMHC0,SMHC1: 0: Command drive phase offset is 90^0 at SDR mode, 45^0 at DDR mode 1: Command drive phase offset is 180^0 at SDR mode, 90^0 at DDR mode For SMHC2: 0: Command drive phase offset is 90^0 at SDR mode, 45^0 at DDR8 mode, 90^0 at DDR4/HS400 mode

			1: Command drive phase offset is 180^0 at SDR mode, 90^0 at DDR8 mode, 180^0 at DDR4/HS400 mode
15:0	/	/	/

5.3.6.46. 0x0144 SMHC Sample Delay Control Register (Default Value: 0x0000_2000)

Offset: 0x0144			Register Name: SMHC_SAMP_DL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15	R/W	0x0	SAMP_DL_CAL_START Sample Delay Calibration Start When set, start sample delay chain calibration.
14	R	0x0	SAMP_DL_CAL_DONE Sample Delay Calibration Done When set, it means that sample delay chain calibration is done and the result of calibration is shown in SAMP_DL.
13:8	R	0x20	SAMP_DL Sample Delay It indicates the number of delay cells corresponding to current card clock. The delay time generated by these delay cells is equal to the cycle of card clock nearly. Generally, it is necessary to do drive delay calibration when card clock is changed. This bit is valid only when SAMP_DL_CAL_DONE is set.
7	R/W	0x0	SAMP_DL_SW_EN Sample Delay Software Enable When set, enable sample delay specified at SAMP_DL_SW
6	/	/	/
5:0	R/W	0x0	SAMP_DL_SW Sample Delay Software The relative delay between clock line and command line, data lines. It can be determined according to the value of SAMP_DL, the cycle of card clock and device's input timing requirement.

5.3.6.47. 0x0148 SMHC Data Strobe Delay Control Register(Default Value: 0x0000_2000)

Offset: 0x0148			Register Name: SMHC_DS_DL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15	R/W	0x0	DS_DL_CAL_START Data Strobe Delay Calibration Start When set, start sample delay chain calibration.

14	R	0x0	DS_DL_CAL_DONE Data Strobe Delay Calibration Done When set, it means that sample delay chain calibration is done and the result of calibration is shown in DS_DL.
13:8	R	0x20	DS_DL Data Strobe Delay It indicates the number of delay cells corresponding to current card clock. The delay time generated by these delay cells is equal to the cycle of SMHC's clock nearly. This bit is valid only when SAMP_DL_CAL_DONE is set.
7	R/W	0x0	DS_DL_SW_EN Sample Delay Software Enable
6	/	/	/
5:0	R/W	0x0	DS_DL_SW Data Strobe Delay Software

5.3.6.48. 0x0200 SMHC FIFO Register (Default Value: 0x0000_0000)

Offset: 0x0200			**Register Name: SMHC_FIFO**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	TX/RX_FIFO Data FIFO

Contents

Figures

Tables

Chapter 6 EMAC

6.1. Overview

The Ethernet Medium Access Controller (EMAC) enables a host to transmit and receive data over Ethernet in compliance with the IEEE 802.3-2002 standard. It supports 10/100/1000 Mbps external PHY with RMII/RGMII interface in both full and half duplex mode. The internal DMA is designed for packet-oriented data transfers based on a linked list of descriptors. 4 KB TXFIFO and 16 KB RXFIFO are provided to keep continuous transmission and reception. Flow Control, CRC Pad & Stripping, and address filtering are also supported in this module.

The EMAC has the following features:

- Two EMAC interfaces
  - EMAC0: 10/100/1000 Mbps Ethernet port with RGMII and RMII interfaces, for connecting the external Ethernet PHY
  - EMAC1: 10/100 Mbps Ethernet port with RMII interface, and it emdedded with 100M EPHY
  - EMAC1 has no external pins
  - EMAC0 and EMAC1 can use at the same time
- Compliant with IEEE 802.3-2002 standard
- Supports MDIO
- Supports both full-duplex and half-duplex operation
- 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

6.2. Block Diagram

The block diagram of EMAC is shown below.

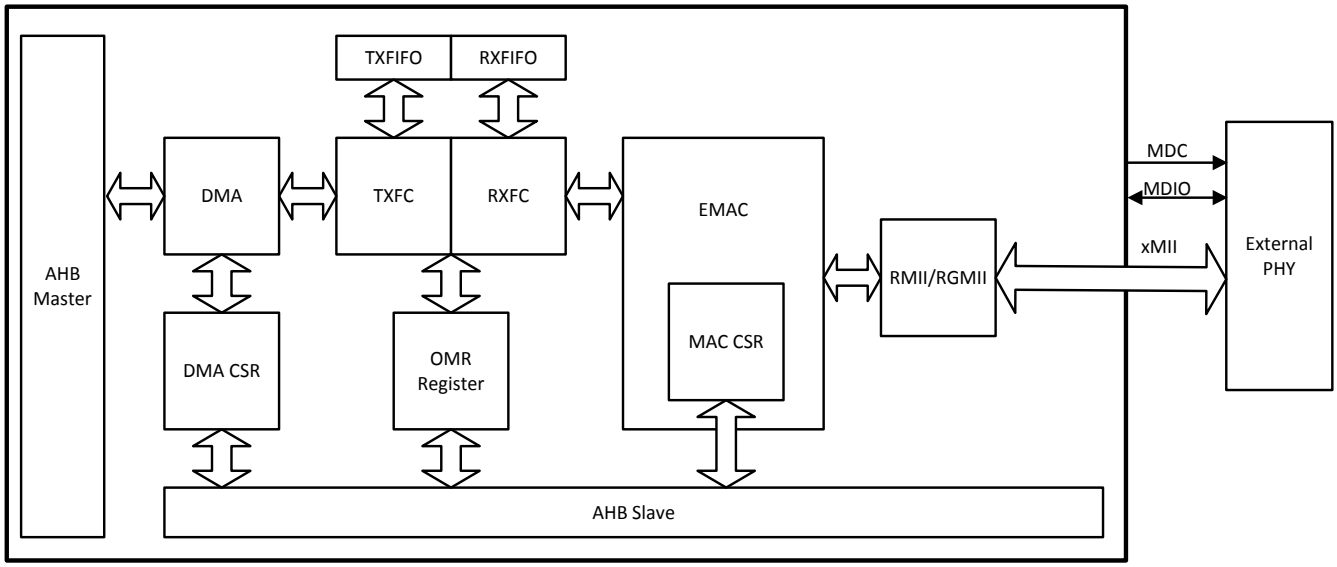

Figure 6- 1. EMAC Block Diagram

6.3. Operations and Functional Descriptions

6.3.1. External Signals

Table 6-1 describes the pin mapping of EMAC.

Table 6- 1. EMAC Pin Mapping

Pin Name	RGMII	RMII
RGMII_RXD3/RMII_NULL	RXD3	/
RGMII_RXD2/RMII_NULL	RXD2	/
RGMII_RXD1/RMII_RXD1	RXD1	RXD1
RGMII_RXD0/RMII_RXD0	RXD0	RXD0
RGMII_RXCK/RMII_NULL	RXCK	/
RGMII_RXCTL/RMII_CRS_DV	RXCTL	CRS_DV
RGMII_NULL/RMII_RXER	/	RXER
RGMII_TXD3/RMII_NULL	TXD3	/
RGMII_TXD2/RMII_NULL	TXD2	/
RGMII_TXD1/RMII_TXD1	TXD1	TXD1
RGMII_TXD0/RMII_TXD0	TXD0	TXD0
RGMII_TXCK/RMII_TXCK	TXCK	TXCK
RGMII_TXCTL/RMII_TXEN	TXCTL	TXEN
RGMII_CLKIN/RMII_NULL	CLKIN	/
MDC	MDC	MDC
MDIO	MDIO	MDIO

EPHY_25M	EPHY_25M	EPHY_25M

Table 6-2 describes the pin list of RGMII.

Table 6- 2. EMAC RGMII Pin List

Pin Name	Description	Type
RGMII_TXD[3:0]	EMAC RGMII Transmit Data	O
RGMII_TXCTL	EMAC RGMII Transmit Control	O
RGMII_TXCK	EMAC RGMII Transmit Clock	O
RGMII_RXD[3:0]	EMAC RGMII Receive Data	I
RGMII_RXCTL	EMAC RGMII Receive Control	I
RGMII_RXCK	EMAC RGMII Receive Clock	I
RGMII_CKIN	EMAC RGMII 125M Reference Clock Input	I
MDC	EMAC Management Data Clock	O
MDIO	EMAC Management Data Input Output	I/O
EPHY_25M	25 MHz Output for EMAC PHY	O

Table 6-3 describes the pin list of RMII.

Table 6- 3. EMAC RMII Pin List

Pin Name	Description	Type
RMII_TXD[1:0]	EMAC RMII Transmit Data	O
RMII_TXEN	EMAC RMII Transmit Enable	O
RMII_TXCK	EMAC RMII Transmit Clock	I
RMII_RXD[1:0]	EMAC RMII Receive Data	I
RMII_CRS_DV	EMAC RMII Receive Data Valid	I
RMII_RXER	EMAC RMII Receive Error	I
MDC	EMAC Management Data Clock	O
MDIO	EMAC Management Data Input Output	I/O
EPHY_25M	25 MHz Output for EMAC PHY	O

Table 6-4 describes the pin list of internal Ethernet PHY.

Table 6- 4. Ethernet PHY Pin list

Pin Name	Description	Type
EPHY_RTX	EPHY External Resistance to Ground	AI
EPHY_RXN	EPHY Transceiver Negative Output/Input	AI/O
EPHY_RXP	EPHY Transceiver Positive Output/Input	AI/O
EPHY_TXN	EPHY Transceiver Negative Output/Input	AI/O
EPHY_TXP	EPHY Transceiver Positive Output/Input	AI/O
VCC_EPHY	EPHY Power Supply	P

6.3.2. Clock Sources

Table 6-5 describes the clock of EMAC.

Table 6- 5. EMAC Clock Characteristics

Clock Name	Description	Type
RGMII_TXCK/ RMII_TXCK	In RGMII mode, output 2.5 MHz/25 MHz/125 MHz. In RMII mode, input 5 MHz/50 MHz.	O/I
RGMII_RXCK	In RGMII mode, input 2.5 MHz/25 MHz/125 MHz. In RMII mode, no input.	I
RGMII_CLKIN	In RGMII mode, input 125 MHz Reference Clock In RMII mode, no clock.	I

6.3.3. Typical Application

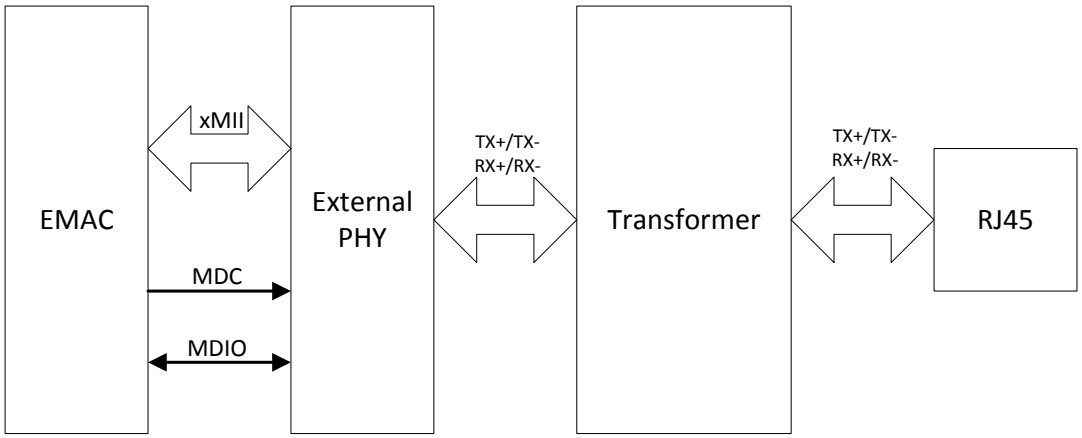

Figure 6- 2. EMAC Typical Application

6.3.4. EMAC RX/TX Descriptor

The internal DMA of EMAC transfers data between host memory and internal RX/TX FIFO with a linked list of descriptors. Each descriptor is consisted of four words, and contains some necessary information to transfer TX and RX frames. The descriptor list structure is shown in Figure 6-3. The address of each descriptor must be 32-bit aligned.

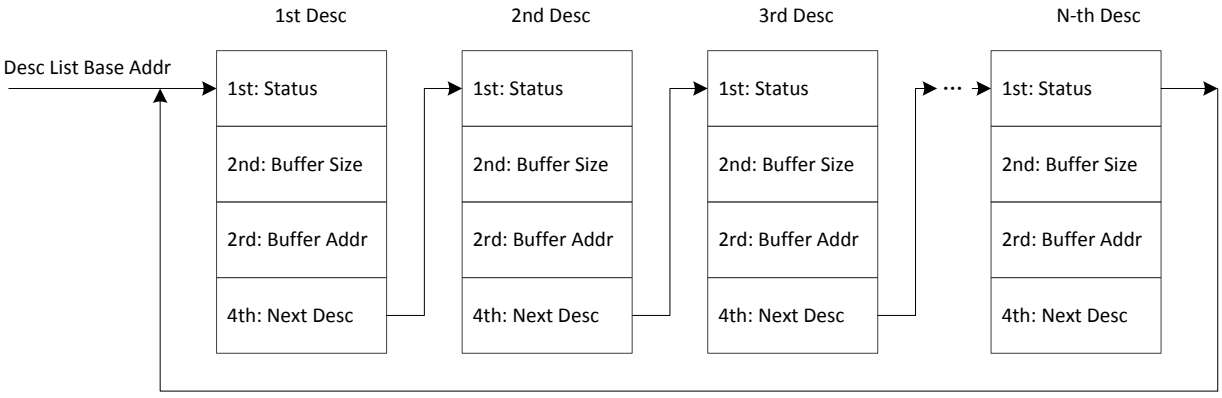

Figure 6- 3. EMAC RX/TX Descriptor List

6.3.5. Transmit Descriptor

6.3.5.1. 1st Word of Transmit Descriptor

Bits	Description
31	TX_DESC_CTL When set, current descriptor can be used by DMA. This bit is cleared by DMA when the whole frame is transmitted or all data in current descriptor's buffer are transmitted.
30:17	Reserved
16	TX_HEADER_ERR When set, the checksum of transmitted frame's header is wrong.
15	Reserved
14	TX_LENGHT_ERR When set, the length of transmitted frame is wrong.
13	Reserved
12	TX_PAYLOAD_ERR When set, the checksum of transmitted frame's payload is wrong.
11	Reserved
10	TX_CRS_ERR When set, carrier is lost during transmission.
9	TX_COL_ERR_0 When set, the frame is aborted because of collision after contention period.
8	TX_COL_ERR_1 When set, the frame is aborted because of too many collisions.
7	Reserved
6:3	TX_COL_CNT The number of collisions before transmission.
2	TX_DEFER_ERR When set, the frame is aborted because of too much deferral.
1	TX_UNDERFLOW_ERR When set, the frame is aborted because of TX FIFO underflow error.

0	TX_DEFER When set in Half-Duplex mode, the EMAC defers the frame transmission.

6.3.5.2. 2nd Word of Transmit Descriptor

Bits	Description
31	TX_INT_CTL When set and the current frame have been transmitted, the TX_INT in Interrupt Status Register will be set.
30	LAST_DESC When set, current descriptor is the last one for current frame.
29	FIR_DESC When set, current descriptor is the first one for current frame.
28:27	CHECKSUM_CTL These bits control to insert checksums in transmit frame.
26	CRC_CTL When set, CRC field is not transmitted.
25:11	Reserved
10:0	BUF_SIZE The size of buffer specified by current descriptor.

6.3.5.3. 3rd Word of Transmit Descriptor

Bits	Description
31:0	BUF_ADDR The address of buffer specified by current descriptor.

6.3.5.4. 4th Word of Transmit Descriptor

Bits	Description
31:0	NEXT_DESC_ADDR The address of next descriptor. It must be 32-bit aligned.

6.3.6. Receive Descriptor

6.3.6.1. 1st Word of Receive Descriptor

Bits	Description
31	RX_DESC_CTL When setting the bit, current descriptor can be used by DMA. This bit is cleared by DMA when complete frame is received or current descriptor's buffer is full.

30	RX_DAF_FAIL When setting the bit, current frame does not pass DA filter.
29:16	RX_FRM_LEN When LAST_DESC is not set and no error bit is set, this field is the length of received data for current frame. When LAST_DESC is set, RX_OVERFLOW_ERR and RX_NO_ENOUGH_BUF_ERR are not set, this field is the length of receive frame.
15	Reserved
14	RX_NO_ENOUGH_BUF_ERR When setting the bit, current frame is clipped because of no enough buffer.
13	RX_SAF_FAIL When setting the bit, current fame does not pass SA filter.
12	Reserved
11	RX_OVERFLOW_ERR When set, a buffer overflow error occurred and current frame is wrong.
10	Reserved
9	FIR_DESC When setting the bit, current descriptor is the first descriptor for current frame.
8	LAST_DESC When setting the bit, current descriptor is the last descriptor for current frame.
7	RX_HEADER_ERR When setting the bit, the checksum of frame's header is wrong.
6	RX_COL_ERR When setting the bit, there is a late collision during reception in half-duplex mode.
5	Reserved
4	RX_LENGTH_ERR When setting the bit, the length of current frame is wrong.
3	RX_PHY_ERR When setting the bit, the receive error signal from PHY is asserted during reception.
2	Reserved
1	RX_CRC_ERR When setting the bit, the CRC filed of received frame is wrong.
0	RX_PAYLOAD_ERR When setting the bit, the checksum or length of received frame's payload is wrong.

6.3.6.2. 2nd Word of Receive Descriptor

Bits	Description
31	RX_INT_CTL When setting the bit, and a frame has been received, the RX_INT will not be set.
30:11	Reserved
10:0	BUF_SIZE The size of buffer specified by current descriptor.

6.3.6.3. 3rd Word of Receive Descriptor

Bits	Description
31:0	BUF_ADDR The address of buffer specified by current descriptor.

6.3.6.4. 4th Word of Receive Descriptor

Bits	Description
31:0	NEXT_DESC_ADDR The address of next descriptor. This field must be 32-bit aligned.

6.4. Register List

Module Name	Base Address
EMAC0	0x05020000
EMAC1(connected to the internal Ethernet PHY)	0x05030000

Register Name	Offset	Description
EMAC_BASIC_CTL0	0x0000	EMAC Basic Control Register0
EMAC_BASIC_CTL1	0x0004	EMAC Basic Control Register1
EMAC_INT_STA	0x0008	EMAC Interrupt Status Register
EMAC_INT_EN	0x000C	EMAC Interrupt Enable Register
EMAC_TX_CTL0	0x0010	EMAC Transmit Control Register0
EMAC_TX_CTL1	0x0014	EMAC Transmit Control Register1
EMAC_TX_FLOW_CTL	0x001C	EMAC Transmit Flow Control Register
EMAC_TX_DMA_DESC_LIST	0x0020	EMAC Transmit Descriptor List Address Register
EMAC_RX_CTL0	0x0024	EMAC Receive Control Register0
EMAC_RX_CTL1	0x0028	EMAC Receive Control Register1
EMAC_RX_DMA_DESC_LIST	0x0034	EMAC Receive Descriptor List Address Register
EMAC_RX_FRM_FLT	0x0038	EMAC Receive Frame Filter Register
EMAC_RX_HASH0	0x0040	EMAC Hash Table Register0
EMAC_RX_HASH1	0x0044	EMAC Hash Table Register1
EMAC_MII_CMD	0x0048	EMAC Management Interface Command Register
EMAC_MII_DATA	0x004C	EMAC Management Interface Data Register
EMAC_ADDR_HIGH0	0x0050	EMAC MAC Address High Register0
EMAC_ADDR_LOW0	0x0054	EMAC MAC Address Low Register0
EMAC_ADDR_HIGHx	0x0050+0x08*N(N=1~7)	EMAC MAC Address High RegisterN(N:1~7)
EMAC_ADDR_LOWx	0x0054+0x08*N(N=1~7)	EMAC MAC Address Low RegisterN(N:1~7)

EMAC_TX_DMA_STA	0x00B0	EMAC Transmit DMA Status Register
EMAC_TX_CUR_DESC	0x00B4	EMAC Current Transmit Descriptor Register
EMAC_TX_CUR_BUF	0x00B8	EMAC Current Transmit Buffer Address Register
EMAC_RX_DMA_STA	0x00C0	EMAC Receive DMA Status Register
EMAC_RX_CUR_DESC	0x00C4	EMAC Current Receive Descriptor Register
EMAC_RX_CUR_BUF	0x00C8	EMAC Current Receive Buffer Address Register
EMAC_RGMII_STA	0x00D0	EMAC RGMII Status Register

6.5. Register Description

6.5.1. 0x0000 EMAC Basic Control Register0 (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: EMAC_BASIC_CTL0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3:2	R/W	0x0	SPEED 00: 1000 Mbit/s 01: Reserved 10: 10 Mbit/s 11: 100 Mbit/s
1	R/W	0x0	LOOPBACK 0: Disable 1: Enable
0	R/W	0x0	DUPLEX 0: Half-duplex 1: Full-duplex

6.5.2. 0x0004 EMAC Basic Control Register1 (Default Value: 0x0800_0000)

Offset: 0x0004			Register Name: EMAC_BASIC_CTL1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:24	R/W	0x8	BURST_LEN The burst length of RX and TX DMA transfer.
23:2	/	/	/
1	R/W	0x0	RX_TX_PRI RX TX DMA priority 0: Same priority 1: RX priority over TX
0	R/W	0x0	SOFT_RST Soft Reset all Registers and Logic

			0: No valid 1: Reset All clock inputs must be valid before soft rest. This bit is cleared internally when the reset operation is completed fully. Before write any register, this bit should read a 0.

6.5.3. 0x0008 EMAC Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: EMAC_INT_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W1C	0x0	RGMII_LINK_STA_P RMII Link Status Changed Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it.
15:14	/	/	/
13	R/W1C	0x0	RX_EARLY_P RX DMA Filled First data Buffer of the Receive Frame Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it.
12	R/W1C	0x0	RX_OVERFLOW_P RX FIFO Overflow Error Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it.
11	R/W1C	0x0	RX_TIMEOUT_P RX Timeout Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it. When this bit asserted, the length of receive frame is greater than 2048 bytes(10240 when JUMBO_FRM_EN is set)
10	R/W1C	0x0	RX_DMA_STOPPED_P When this bit asserted, the RX DMA FSM is stopped.
9	R/W1C	0x0	RX_BUF_UA _P RX Buffer UA Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it. When this asserted, the RX DMA cannot acquire next RX descriptor and RX DMA FSM is suspended. The ownership of next RX descriptor should be changed to RX DMA. The RX DMA FSM will resume when writing to RX_DMA_START bit or next receive frame is coming.
8	R/W1C	0x0	RX_P

			Frame RX Completed Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it. When this bit is asserted, a frame reception is completed. The RX DMA FSM remains in the running state.
7:6	/	/	/
5	R/W1C	0x0	TX_EARLY_P Frame is transmitted to FIFO totally Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it.
4	R/W1C	0x0	TX_UNDERFLOW_P TX FIFO Underflow Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it.
3	R/W1C	0x0	TX_TIMEOUT_P Transmitter Timeout Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it.
2	R/W1C	0x0	TX_BUF_UA_P TX Buffer UA Interrupt Pending 0: No Pending 1: Pending When this asserted, the TX DMA can not acquire next TX descriptor and TX DMA FSM is suspended. The ownership of next TX descriptor should be changed to TX DMA. The TX DMA FSM will resume when writing to TX_DMA_START bit.
1	R/W1C	0x0	TX_DMA_STOPPED_P Transmission DMA Stopped Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it.
0	R/W1C	0x0	TX_P Frame Transmission Interrupt Pending 0: No Pending 1: Pending Write '1' to clear it.

6.5.4. 0x000C EMAC Interrupt Enable Register (Default Value: 0x0000_0000)

Offset: 0x000C			Register Name: EMAC_INT_EN
Bit	**Read/Write**	**Default/Hex**	**Description**

31:14	/	/	/
13	R/W	0x0	RX_EARLY_INT_EN Early Receive Interrupt 0: Disable 1: Enable
12	R/W	0x0	RX_OVERFLOW_INT_EN Receive Overflow Interrupt 0: Disable 1: Enable
11	R/W	0x0	RX_TIMEOUT_INT_EN Receive Timeout Interrupt 0: Disable 1: Enable
10	R/W	0x0	RX_DMA_STOPPED_INT_EN Receive DMA FSM Stopped Interrupt 0: Disable 1: Enable
9	R/W	0x0	RX_BUF_UA_INT_EN Receive Buffer Unavailable Interrupt 0: Disable 1: Enable
8	R/W	0x0	RX_INT_EN Receive Interrupt 0: Disable 1: Enable
7:6	/	/	/
5	R/W	0x0	TX_EARLY_INT_EN Early Transmit Interrupt 0: Disable 1: Enable
4	R/W	0x0	TX_UNDERFLOW_INT_EN Transmit Underflow Interrupt 0: Disable 1: Enable
3	R/W	0x0	TX_TIMEOUT_INT_EN Transmit Timeout Interrupt 0: Disable 1: Enable
2	R/W	0x0	TX_BUF_UA_INT_EN Transmit Buffer Available Interrupt 0: Disable 1: Enable
1	R/W	0x0	TX_DMA_STOPPED_INT_EN Transmit DMA FSM Stopped Interrupt 0: Disable

			1: Enable
0	R/W	0x0	TX_INT_EN Transmit Interrupt 0: Disable 1: Enable

6.5.5. 0x0010 EMAC Transmit Control Register0 (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: EMAC_TX_CTL0
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	TX_EN Enable Transmitter 0: Disable 1: Enable When disable, transmit will continue until current transmit finishes.
30	R/W	0x0	TX_FRM_LEN_CTL Frame Transmit Length Control 0: Up to 2,048 bytes (JUMBO_FRM_EN==0) Up to 10,240 bytes (JUMBO_FRM_EN==1) 1: Up to 16,384 bytes Any bytes after that is cut off.
29:0	/	/	/

6.5.6. 0x0014 EMAC Transmit Control Register1 (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: EMAC_TX_CTL1
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	TX_DMA_START Transmit DMA FSM Start 0: No valid 1: Start It is cleared internally and always read a 0
30	R/W	0x0	TX_DMA_EN 0: Stop TX DMA after the completion of current frame transmission. 1: Start and run TX DMA.
29:11	/	/	/
10:8	R/W	0x0	TX_TH Threshold value of TX DMA FIFO When TX_MD is 0, transmission starts when the size of frame in TX DMA FIFO is greater than the threshold. In addition, full frames with a length less than the threshold are transferred automatically. 000: 64

			001: 128 010: 192 011: 256 Others: Reserved
7:2	/	/	/
1	R/W	0x0	TX_MD Transmission Mode 0: TX start after TX DMA FIFO bytes is greater than TX_TH 1: TX start after TX DMA FIFO located a full frame
0	R/W	0x0	FLUSH_TX_FIFO Flush the data in the TX FIFO 0: Enable 1: Disable

6.5.7. 0x001C EMAC Transmit Flow Control Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: EMAC_TX_FLOW_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	TX_FLOW_CTL_STA This bit indicates a pause frame transmission is in progress. When the configuration of flow control is ready, set this bit to transmit a pause frame in full-duplex mode or activate the backpressure function. After completion of transmission, this bit will be cleared automatically. Before write register TX_FLOW_CTRL, this bit must be read as 0.
30:22	/	/	/
21:20	R/W	0x0	TX_PAUSE_FRM_SLOT The threshold of the pause timer at which the input flow control signal is checked for automatic retransmission of pause frame. The threshold values should be always less than the PAUSE_TIME
19:4	R/W	0x0	PAUSE_TIME The pause time field in the transmitted control frame.
3:2	/	/	/
1	R/W	0x0	ZQP_FRM_EN 0: Disable 1: Enable When set, enable the functionality to generate Zero-Quanta Pause control frame.
0	R/W	0x0	TX_FLOW_CTL_EN TX Flow Control Enable 0: Disable 1: Enable When set, enable flow control operation to transmit pause frames in full-duplex mode, or enable the back-pressure operation in half-duplex mode.

6.5.8. 0x0020 EMAC Transmit DMA Descriptor List Address Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: EMAC_TX_DMA_LIST
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	TX_DESC_LIST The base address of transmit descriptor list. It must be 32-bit aligned.

6.5.9. 0x0024 EMAC Receive Control Register0 (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: EMAC_RX_CTL0
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	RX_EN Enable Receiver 0: Disable receiver after current reception 1: Enable
30	R/W	0x0	RX_FRM_LEN_CTL Frame Receive Length Control 0: Up to 2,048 bytes (JUMBO_FRM_EN==0) Up to 10,240 bytes (JUMBO_FRM_EN==1) 1: Up to 16,384 bytes Any bytes after that is cut off
29	R/W	0x0	JUMBO_FRM_EN Jumbo Frame Enable 0: Disable 1: Enable Jumbo frames of 9,018 bytes without reporting a giant
28	R/W	0x0	STRIP_FCS When set, strip the Pad/FCS field on received frames only when the length's field value is less than or equal to 1,500 bytes.
27	R/W	0x0	CHECK_CRC Check CRC Enable 0: Disable 1: Calculate CRC and check the IPv4 Header Checksum.
26:18	/	/	/
17	R/W	0x0	RX_PAUSE_FRM_MD 0: Only detect multicast pause frame specified in the 802.3x standard. 1: In addition to detect multicast pause frame specified in the 802.3x standard, also detect unicast pause frame with address specified in MAC Address 0 High Register and MAC address 0 Low Register.
16	R/W	0x0	RX_FLOW_CTL_EN When set, enable the functionality that decode the received pause frame and disable its transmitter for a specified time by pause frame.

15:0	/	/	/

6.5.10. 0x0028 EMAC Receive Control Register1 (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: EMAC_RX_CTL1
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	RX_DMA_START When set, the RX DMA will not work. It is cleared internally and always read a 0.
30	R/W	0x0	RX_DMA_EN Receive DMA Enable 0: Stop RX DMA after finish receiving current frame 1: Start and run RX DMA
29:25	/	/	/
24	R/W	0x0	RX_FIFO_FLOW_CTL Receive FIFO Flow Control Enable 0: Disable 1: Enable,base on RX_FLOW_CTL_TH_DEACT and RX_FLOW_CTL_TH_ACT
23:22	R/W	0x0	RX_FLOW_CTL_TH_DEACT Threshold for Deactivating Flow Control 00: Full minus 1 KB 01: Full minus 2 KB 10: Full minus 3 KB 11: Full minus 4 KB Valid in both half-duplex mode and full-duplex mode.
21:20	R/W	0x0	RX_FLOW_CTL_TH_ACT Threshold for Activating Flow Control 00: Full minus 1 KB 01: Full minus 2 KB 10: Full minus 3 KB 11: Full minus 4 KB Valid in both half-duplex mode and full-duplex mode.
19:6	/	/	/
5:4	R/W	0x0	RX_TH Threshold for RX DMA FIFO Start 00: 64 01: 32 10: 96 11: 128 **Note: Only valid when RX_MD == 0, full frames with a length less than the threshold are transferred automatically.**
3	R/W	0x0	RX_ERR_FRM 0: RX DMA drops frames with error 1: RX DMA forwards frames with error

2	R/W	0x0	RX_RUNT_FRM When setting, forward undersized frames with no error and length less than 64 bytes
1	R/W	0x0	RX_MD Receive Mode 0: RX start read after RX DMA FIFO bytes is greater than RX_TH 1: RX start read after RX DMA FIFO located a full frame
0	R/W	0x0	FLUSH_RX_FRM Flush Receive Frames 0: Enable when receive descriptors/buffers is unavailable 1: Disable

6.5.11. 0x0034 EMAC Receive DMA Descriptor List Address Register (Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: EMAC_RX_DMA_LIST
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	RX_DESC_LIST The base address of receive descriptor list. It must be 32-bit aligned.

6.5.12. 0x0038 EMAC Receive Frame Filter Register (Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: EMAC_RX_FRM_FLT
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	DIS_ADDR_FILTER Disable Address Filter 0: Enable 1: Disable
30:18	/	/	/
17	R/W	0x0	DIS_BROADCAST Disable Receive Broadcast Frames 0: Receive 1: Drop
16	R/W	0x0	RX_ALL_MULTICAST Receive All Multicast Frames Filter 0: Filter according to HASH_MULTICAST 1: Receive All
15:14	/	/	/
13:12	R/W	0x0	CTL_FRM_FILTER Receive Control Frames Filter 00: Drop all control frames 01: Drop all control frames 10: Receive all control frames

			11: Receive all control frames when pass the address filter
11:10	/	/	/
9	R/W	0x0	HASH_MULTICAST Filter Multicast Frames Set 0: by comparing the DA field in DA MAC address registers 1: according to the hash table
8	R/W	0x0	HASH_UNICAST Filter Unicast Frames Set 0: by comparing the DA field in DA MAC address registers 1: according to the hash table
7	/	/	/
6	R/W	0x0	SA_FILTER_EN Receive SA Filter Enable 0: Receive frames and update the result of SA filter 1: Update the result of SA filter. In addition, if the SA field of received frame does not match the values in SA MAC address registers, drop this frame.
5	R/W	0x0	SA_INV_FILTER Receive SA Invert Filter Set 0: Pass Frames whose SA field matches SA MAC address registers 1: Pass Frames whose SA field not matches SA MAC address registers
4	R/W	0x0	DA_INV_FILTER 0: Normal filtering of frames is performed 1: Filter both unicast and multicast frames by comparing DA field in inverse filtering mode
3:2	/	/	/
1	R/W	0x0	FLT_MD 0: If the HASH_MULTICAST or HASH_UNICAST is set, the frame is passed only when it matches the Hash filter 1: Receive the frame when it passes the address register filter or the hash filter(set by HASH_MULTICAST or HASH_UNICAST)
0	R/W	0x0	RX_ALL Receive All Frame Enable 0: Receive the frames that pass the SA/DA address filter 1: Receive all frames and update the result of address filter(pass or fail) in the receive status word

6.5.13. 0x0040 EMAC Receive Hash Table Register0 (Default Value: 0x0000_0000)

Offset: 0x0040			**Register Name: EMAC_RX_HASH0**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	HASH_TAB0 The upper 32 bits of Hash table for receive frame filter.

6.5.14. 0x0044 EMAC Receive Hash Table Register1 (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: EMAC_RX_HASH1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	HASH_TAB1 The lower 32 bits of Hash table for receive frame filter.

6.5.15. 0x0048 EMAC MII Command Register (Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: EMAC_MII_CMD
Bit	**Read/Write**	**Default/Hex**	**Description**
31:23	/	/	/
22:20	R/W	0x0	MDC_DIV_RATIO_M MDC Clock Divide Ratio 000: 16 001: 32 010: 64 011: 128 Others: Reserved **Note: MDC Clock is divided from AHB clock.**
19:17	/	/	/
16:12	R/W	0x0	PHY_ADDR PHY Address
11:9	/	/	/
8:4	R/W	0x0	PHY_REG_ADDR PHY Register Address
3:2	/	/	/
1	R/W	0x0	MII_WR MII Write and Read 0: Read 1: Write
0	R/W	0x0	MII_BUSY 0: Write no valid, read 0 indicates finish in read or write operation 1: Write start read or write operation, read 1 indicates busy.

6.5.16. 0x004C EMAC MII Data Register (Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: EMAC_MII_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0	MII_DATA Write to or read from the register in the selected PHY.

6.5.17. 0x0050 EMAC MAC Address High Register0 (Default Value: 0x0000_FFFF)

Offset: 0x0050			Register Name: EMAC_ADDR_HIGH0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0xFFFF	MAC_ADDR_HIGH0 The upper 16 bits of the 1st MAC address.

6.5.18. 0x0054 EMAC MAC Address Low Register0 (Default Value: 0xFFFF_FFFF)

Offset: 0x0054			Register Name: EMAC_ADDR_LOW0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0xFFFFFFFF	MAC_ADDR_LOW0 The lower 32 bits of 1st MAC address.

6.5.19. 0x0050+0x08*N EMAC MAC Address High Register N (Default Value: 0x0000_0000)

Offset: 0x0050+0x08*N (N=1~7)			Register Name: EMAC_ADDR_HIGHN
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MAC_ADDR_CTL MAC Address Valid 0: Not valid 1: Valid
30	R/W	0x0	MAC_ADDR_TYPE MAC Address Type 0: Used to compare with the destination address of the received frame 1: Used to compare with the source address of the received frame
29:24	R/W	0x0	MAC_ADDR_BYTE_CTL MAC Address Byte Control Mask The lower bit of mask controls the lower byte of MAC address. When the bit of mask is 1, do not compare the corresponding byte.
23:16	/	/	/
15:0	R/W	0x0	MAC_ADDR_HIGH The upper 16bits of the MAC address.

6.5.20. 0x0054+0x08*N EMAC MAC Address Low Register N (Default Value: 0x0000_0000)

Offset: 0x0054+0x08*N (N=1~7)			Register Name: EMAC_ADDR_LOWN
Bit	**Read/Write**	**Default/Hex**	**Description**

31:0	R/W	0x0	MAC_ADDR_LOWN The lower 32bits of MAC address N (N: 1~7).

6.5.21. 0x00B0 EMAC Transmit DMA Status Register (Default Value: 0x0000_0000)

Offset: 0x00B0			Register Name: EMAC_TX_DMA_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2:0	R	0x0	TX_DMA_STA The State of Transmit DMA FSM 000: STOP, when reset or disable TX DMA 001: RUN_FETCH_DESC, fetching TX DMA descriptor 010: RUN_WAIT_STA, waiting for the status of TX frame 011: RUN_TRANS_DATA, passing frame from host memory to TX DMA FIFO 100: Reserved 101: Reserved 111: RUN_CLOSE_DESC, closing TX descriptor 110: SUSPEND, TX descriptor unavailable or TX DMA FIFO underflow

6.5.22. 0x00B4 EMAC Transmit DMA Current Descriptor Register (Default Value: 0x0000_0000)

Offset: 0x00B4			Register Name: EMAC_TX_DMA_CUR_DESC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	The address of current transmit descriptor.

6.5.23. 0x00B8 EMAC Transmit DMA Current Buffer Address Register (Default Value: 0x0000_0000)

Offset: 0x00B8			Register Name: EMAC_TX_DMA_CUR_BUF
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	The address of current transmit DMA buffer.

6.5.24. 0x00C0 EMAC Receive DMA Status Register (Default Value: 0x0000_0000)

Offset: 0x00C0			Register Name: EMAC_RX_DMA_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2:0	R	0x0	RX_DMA_STA The State of RX DMA FSM 000: STOP, when reset or disable RX DMA 001: RUN_FETCH_DESC, fetching RX DMA descriptor

			010: Reserved 011: RUN_WAIT_FRM, waiting for frame 100: SUSPEND, RX descriptor unavailable 101: RUN_CLOSE_DESC, closing RX descriptor 110: Reserved 111: RUN_TRANS_DATA, passing frame from host memory to RX DMA FIFO;

6.5.25. 0x00C4 EMAC Receive DMA Current Descriptor Register (Default Value: 0x0000_0000)

Offset: 0x00C4			Register Name: EMAC_RX_DMA_CUR_DESC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	The address of current receive descriptor

6.5.26. 0x00C8 EMAC Receive DMA Current Buffer Address Register (Default Value: 0x0000_0000)

Offset: 0x00C8			Register Name: EMAC_RX_DMA_CUR_BUF
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	The address of current receive DMA buffer

6.5.27. 0x00D0 EMAC RGMII Status Register (Default Value: 0x0000_0000)

Offset: 0x00D0			Register Name: EMAC_RGMII_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3	R	0x0	RGMII_LINK The link status of RGMII interface 0: down 1: up
2:1	R	0x0	RGMII_LINK_SPD The link speed of RGMII interface 00: 2.5 MHz 01: 25 MHz 10: 125 MHz 11: Reserved
0	R	0x0	RGMII_LINK_MD The link mode of RGMII interface 0: Half-Duplex 1: Full-Duplex

Contents

Figures

Tables

Chapter 7 Video Output Interfaces

7.1. TCON_TV

7.1.1. Overview

The TCON_TV(Timing Controller_TV) is a module that processes video signals received from systems using a complicated arithmetic and then generates control signals and transmits them to the HDMI or TVE.

The TCON_TV includes the following features:

- Supports 10-bit pixel depth YUV422/YUV420, HV format output up to 4K@60Hz
- Supports 8-bit pixel depth YUV444, HV format Output up to 4K@60Hz

7.1.2. Block Diagram

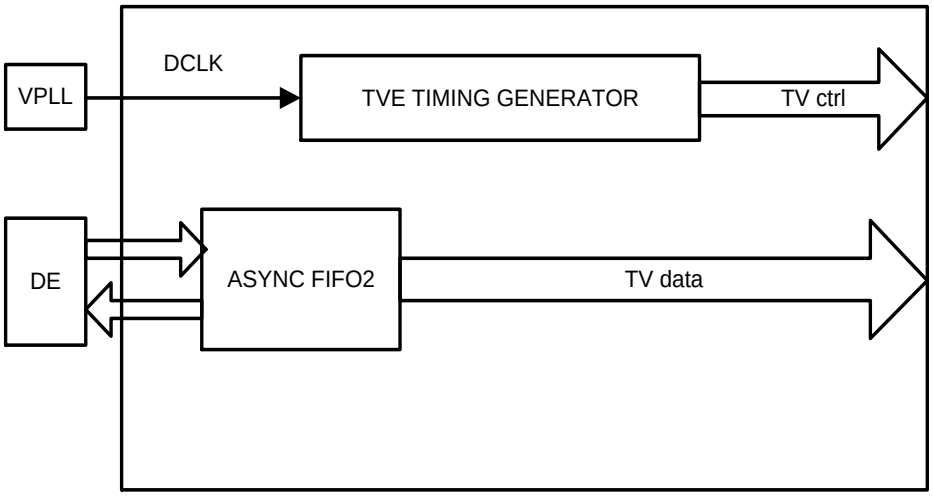

Figure 7- 1. TCON_TV Block Diagram

7.1.3. Operations and Functional Descriptions

7.1.3.1. Panel Interface

HV I/F is also known as Sync + DE mode, which is used to transfer signal to HDMI I/F. Its signals are defined as:

Table 7- 1. HV Panel Signals

Signal	Description	Type
Vsync	Vertical sync, indicates one new frame	O
Hsync	Horizontal sync, indicate one new scan line	O
DCLK	Dot clock, pixel data are sync by this clock	O
LDE	LCD data enable	O
LD[29..0]	30-bit RGB/YUV output from input FIFO for panel	O

HV control signals are active low.

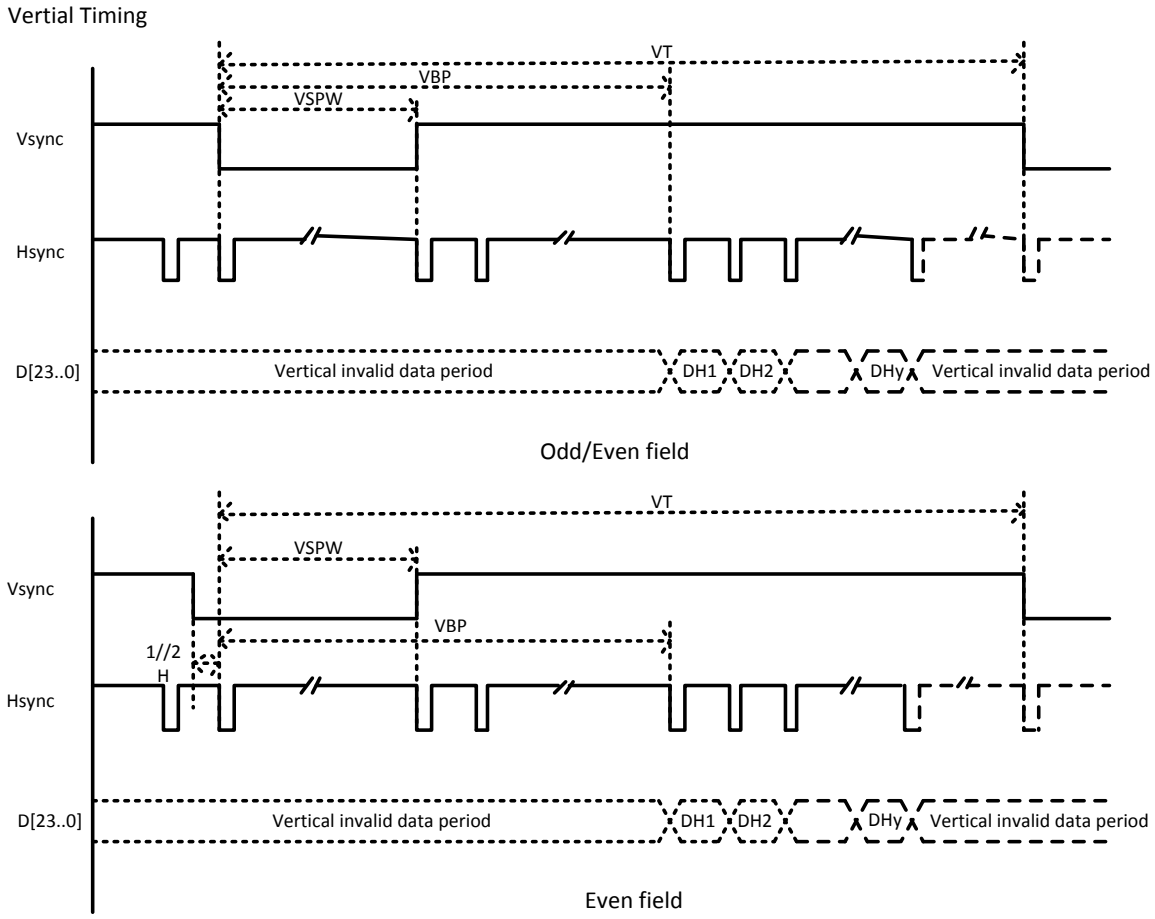

Figure 7- 2. HV Interface Vertical Timing

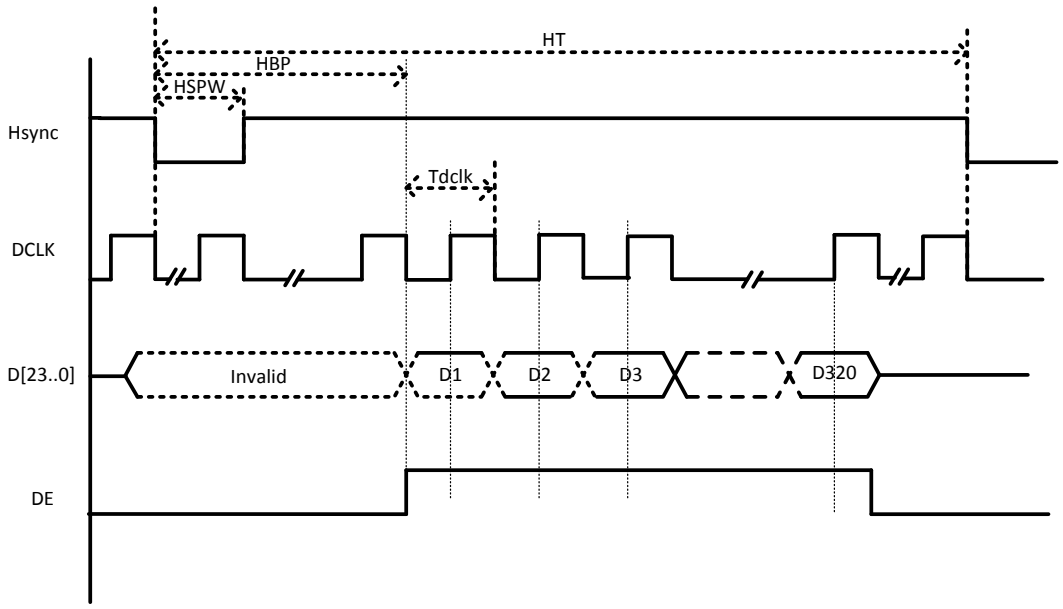

Figure 7- 3. HV Interface Horizontal Timing

7.1.3.2. Clock Sources

The following table describes the clock sources of TCON_TV. Table 7-2 describes the clock sources of TCON_TV.

Table 7- 2. TCON_TV Clock Sources

Clock Sources	Description
PLL_VIDEO0(1X)	Video PLL Clock,default value is 297 MHz
PLL_VIDEO0(4X)	Video PLL Clock,default value is 1188 MHz
PLL_VIDEO1(1X)	Video PLL Clock,default value is 297 MHz
PLL_VIDEO1(4X)	Video PLL Clock,default value is 1188 MHz

The clock system of TCON_TV is as follows.

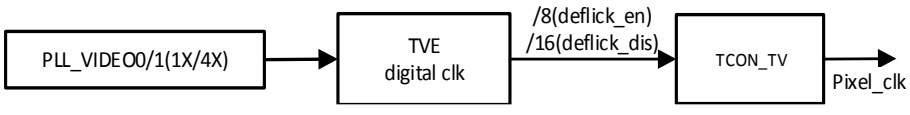

Figure 7- 4. TCON_TV Clock System

TCON_TV_CLK is produced by internal frequency division. If deflick function is enabled, TCON_TV_CLK is one-eighth of TVE_CLK; if deflick function is disabled, TCON_TV_CLK is one-sixteenth of TVE_CLK.

7.1.3.3. CEU Module

This module enhances color data from DE .

R' = Rr*R + Rg*G + Rb*B

G’ = Gr*R + Gg*G + Gb*B

B’ = Br*R + Bg*G + Bb*B

NOTE

Rr, Rg, Rb, ,Gr, Gg, Gb, Br, Bg, Bb bool 0,1

R, G, B u10 [0-1023]

R’ have the range of [Rmin ,Rmax]

G’ have the range of [Rmin ,Rmax]

B’ have the range of [Rmin ,Rmax]

7.1.4. Programming Guidelines

7.1.4.1. TCON_TV Configuration Process

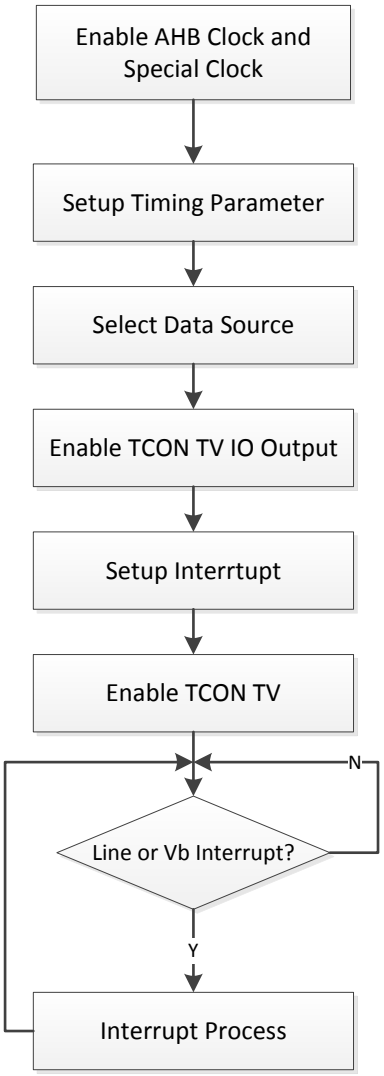

Figure 7- 5. TCON TV Initial Process

Step1: Set special clock of CCU ,and dessert TCON TV related AHB clock gating and AHB reset .

Step2: Set timing parameter register of TCON TV, set corresponding resolution and standards followed, such as EIA or VESA.There are 8 parameters, including X,HT,HBP,HSPW,Y,VT,VBP,VSPW. Note that for the controller,HBP includes HSPW width, VBP includes VSPW width, this is different with standard HBP and VBP.Note that for conversion.
Step3: Select TCON TV data sources. For the selecting of TCON TV data sources, it is decided by two setting. The first setting is the bit1(TV_SRC_SEL_GOBAL) of **TV_CTL_REG**, if setting to 1,then blue data is output; if setting to 0, then data source is decided by **TV0_SRC_CTL_REG**. According to needs, set up TV_SRC_SEL, select the required data sources.
Step4: The register offset of **TCON TV IO Output Function Setting** is 0x8C, writing 0 to the register open output function.
Step5: Set and open interrupt. When using line interrupt, firstly the TV_LINE_INT_NUM bit of **TV_GINT1** need be set, secondly line interrupt is enabled, that is , the bit 28(TV_LINE_INT_EN) of **TV_GINT0** is set to 1.
Step6: Start TCON TV.

7.1.4.2. 3D Mode Notes

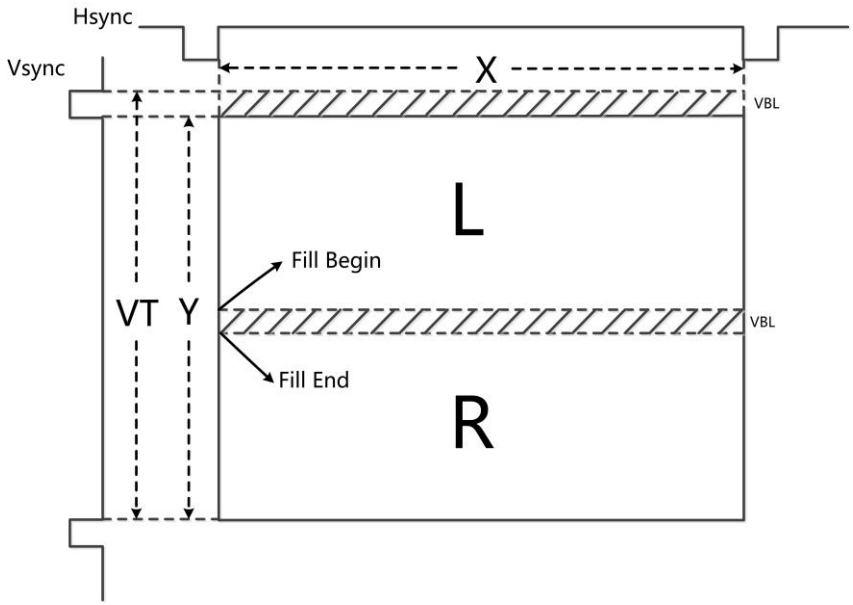

Figure 7- 6. TCON TV 3D Mode Diagram

As shown in the above figure, VT= $VBL_L + Y_L + VBL_R + Y_R$, Y = $Y_L + VBL_R + Y_R$. But note that VT in this picture is the actual VT, is the half of VT in register.
In 3D mode, the 2 frames is synthesized into 1 frame to send data, so the effective data area will contain a blank area, this blank need be filled, and generally filled 0. The rest is to confirm the beginning and the end line of padding, the formula is as follows:
L_{begin} = VT/2 + 1, L_{end}=VT/2+(VT-Y)/2
Lastly, write L_{begin} to the bit[23:12] of **TV_FILL_BEGIN_REG0**(0x304), write L_{end} to the bit[23:12] of **TV_FILL_END_RGB0**(0x308), write 0 to **TV_FILL_DATA_REG0**(0x30C).

7.1.5. Register List

Module Name	Base Address
TCON_TV0	0x06515000
TCON_TV1	0x06516000

Register Name	Offset	Description
TV_GCTL_REG	0x0000	TV Global Control Register
TV_GINT0_REG	0x0004	TV Global Interrupt Register0
TV_GINT1_REG	0x0008	TV Global Interrupt Register1
TV_SRC_CTL_REG	0x0040	TV Source Control Register
TV_IO_POL_REG	0x0088	TV IO Polarity Register
TV_IO_TRI_REG	0x008C	TV IO Control Register
TV_CTL_REG	0x0090	TV Control Register
TV_BASIC0_REG	0x0094	TV Basic Timing Register0
TV_BASIC1_REG	0x0098	TV Basic Timing Register1
TV_BASIC2_REG	0x009C	TV Basic Timing Register2
TV_BASIC3_REG	0x00A0	TV Basic Timing Register3
TV_BASIC4_REG	0x00A4	TV Basic Timing Register4
TV_BASIC5_REG	0x00A8	TV Basic Timing Register5
TV_ECC_FIFO_REG	0x00F8	TV ECC FIFO Register
TV_DEBUG_REG	0x00FC	TV Debug Register
TV_CEU_CTL_REG	0x0100	TV CEU Control Register
TV_CEU_COEF_MUL_REG	0x0110+N*0x04	TV CEU Coefficient MUL Register(N=0,1,2,4,5,6,8,9,10)
TV_CEU_COEF_RANG_REG	0x0140+N*0x04	TV CEU Coefficient Range Register(N=0,1,2)
TV_SAFE_PERIOD_REG	0x01F0	TV Safe Period Register
TV_FILL_CTL_REG	0x0300	TV Fill Data Control Register
TV_FILL_BEGIN_REG	0x0304+N*0x0C(N=0,1,2)	TV Fill Data Begin Register
TV_FILL_END_REG	0x0308+N*0x0C(N=0,1,2)	TV Fill Data End Register
TV_FILL_DATA_REG	0x030C+N*0x0C(N=0,1,2)	TV Fill Data Value Register
TV_DATA_IO_POL0_REG	0x0330	TV Data IO Polarity0 Register
TV_DATA_IO_POL1_REG	0x0334	TV Data IO Polarity1 Register
TV_DATA_IO_TRI0_REG	0x0338	TV Data IO Trigger0 Register
TV_DATA_IO_TRI1_REG	0x033C	TV Data IO Trigger1 Register
TV_PIXELDEPTH_MODE_REG	0x0340	TV Pixel-depth Mode Register

7.1.6. Registers Description

7.1.6.1. 0x0000 TV Global Control Register(Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: TV_GCTL_REG
Bit	Read/Write	Default/Hex	Description

31	R/W	0x0	TV_EN 0: Disable 1: Enable When it is disabled, the module will be reset to idle state.
30:0	/	/	/

7.1.6.2. 0x0004 TV Global Interrupt Register0(Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: TV_GINT0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R/W	0x0	TV_VB_INT_EN 0: Disable 1: Enable Enable the Vb interrupt.
29	/	/	/
28	R/W	0x0	TV_LINE_INT_EN 0: Disable 1: Enable Enable the line interrupt.
27:15	/	/	/
14	R/W	0x0	TV_VB_INT_FLAG Assert during vertical no-display period every frame. Write 0 to clear it.
13	/	/	/
12	R/W	0x0	TV_LINT_INT_FLAG Trigger when SY1 match the current TV scan line Write 0 to clear it.
11:0	/	/	/

7.1.6.3. 0x0008 TV Global Interrupt Register1(Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: TV_GINT1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11:0	R/W	0x0	TV_LINE_INT_NUM Scan line for TV line trigger(including inactive lines) Setting it for the specified line of trigger 1. **Note: SY1 is writable only when LINE_TRG1 is disabled.**

7.1.6.4. 0x0040 TV Source Control Register(Default Value: 0x0000_0000)

Offset: 0x0040			Register Name: TV_SRC_CTL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2:0	R/W	0x0	TV_SRC_SEL 000: DE 001: Color Check 010: Grayscale Check 011: Black by White Check 100: Reserved 101: Reserved 111: Gridding Check

7.1.6.5. 0x0088 TV IO Polarity Register(Default Value: 0x0000_0000)

Offset: 0x0088			Register Name: TV_IO_POL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27	R/W	0x0	IO3_INV 0: Not invert 1: Invert Enable invert function of IO3.
26	R/W	0x0	IO2_ INV 0: Not invert 1: Invert Enable invert function of IO2.
25	R/W	0x0	IO1_INV 0: Not invert 1: Invert Enable invert function of IO1.
24	R/W	0x0	IO0_INV 0: Not invert 1: Invert Enable invert function of IO0.
23:0	/	/	/

7.1.6.6. 0x008C TV IO Control Register(Default Value: 0x0F00_0000)

Offset: 0x008C			Register Name: TV_IO_TRI_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/

27	R/W	0x1	IO3_OUTPUT_TRI_EN 1: Disable 0: Enable Enable the output of IO3.
26	R/W	0x1	IO2_OUTPUT_TRI_EN 1: Disable 0: Enable Enable the output of IO2.
25	R/W	0x1	IO1_OUTPUR_TRI_EN 1: Disable 0: Enable Enable the output of IO1.
24	R/W	0x1	IO0_OUTPUT_TRI_EN 1: Disable 0: Enable Enable the output of IO0.
23:0	/	/	/

7.1.6.7. 0x0090 TV Control Register(Default Value: 0x0000_0000)

Offset: 0x0090			**Register Name: TV_CTL_REG**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	TV_EN 0: Disable 1: Enable
30:9	/	/	/
8:4	R/W	0x0	START_DELAY This is for DE0 and DE1.
3:2	/	/	/
1	R/W	0x0	TV_SRC_SEL_GOBAL 0: reserved 1: BLUE data **Note: The priority of this bit is higher than TV_SRC_SEL(bit[2:0]) in TV_SRC_CTL_REG.**
0	/	/	/

7.1.6.8. 0x0094 TV Basic Timing Register0(Default Value: 0x0000_0000)

Offset: 0x0094			**Register Name: TV_BASIC0_REG**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0x0	XI

			source width is X+1
15:12	/	/	/
11:0	R/W	0x0	YI source height is Y+1

7.1.6.9. 0x0098 TV Basic Timing Register1(Default Value: 0x0000_0000)

Offset: 0x0098			Register Name: TV_BASIC1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0x0	LS_XO Width is LS_XO+1
15:12	/	/	/
11:0	R/W	0x0	LS_YO Width is LS_YO+1 **Note: LS_YO = TV_YI**

7.1.6.10. 0x009C TV Basic Timing Register2(Default Value: 0x0000_0000)

Offset: 0x009C			Register Name: TV_BASIC2_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0x0	TV_XO Width is TV_XO+1
15:12	/	/	/
11:0	R/W	0x0	TV_YO Height is TV_YO+1

7.1.6.11. 0x00A0 TV Basic Timing Register3(Default Value: 0x0000_0000)

Offset: 0x00A0			Register Name: TV_BASIC3_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28:16	R/W	0x0	HT Horizontal total time T_{hcycle} = (HT+1) * T_{hdclk}
15:12	/	/	/
11:0	R/W	0x0	HBP Horizontal back porch T_{hbp} = (HBP +1) * T_{hdclk}

7.1.6.12. 0x00A4 TV Basic Timing Register4(Default Value: 0x0000_0000)

Offset: 0x00A4			Register Name: TV_BASIC4_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28:16	R/W	0x0	VT Vertical total time (in HD line) T_{vt} = VT/2 * T_h
15:12	/	/	/
11:0	R/W	0x0	VBP Vertical back porch (in HD line) T_{vbp} = (VBP +1) * T_h

7.1.6.13. 0x00A8 TV Basic Timing Register5(Default Value: 0x0000_0000)

Offset: 0x00A8			Register Name: TV_BASIC5_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	R/W	0x0	HSPW Horizontal Sync Pulse Width (in dclk) T_{hspw} = (HSPW+1) * T_{dclk} HT> (HSPW+1)
15:10	/	/	/
9:0	R/W	0x0	VSPW Vertical Sync Pulse Width (in lines) T_{vspw} = (VSPW+1) * T_h VT/2 > (VSPW+1)

7.1.6.14. 0x00F8 TV ECC FIFO Register(Default Value: 0x0000_0000)

Offset: 0x00F8			Register Name: TV_ECC_FIFO_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	ECC_FIFO_BIST_EN 0: Disable 1: Enable Enable ECC FIFO BIST test function.
30	R	0x0	ECC_FIFO_ERR_FLAG Indicates the error information in ECC FIFO.
29:24	/	/	/
23:16	R	0x0	ECC_FIFO_ERR_BITS Indicates the error information in ECC FIFO.
15:9	/	/	/

8	R/W	0x0	ECC_FIFO_BLANK_EN 0: Disable ECC function in blanking 1: Enable ECC function in blanking ECC function is tent to trigger in blanking area at HV mode, set '0' when in HV mode.
7:4	/	/	/
3	R/W	0x0	ECC_FIFO_SETTING 0:Enable 1:Disable Enable ECC FIFO function.
2:0	/	/	/

7.1.6.15. 0x00FC TV Debug Register(Default Value: 0x0000_0000)

Offset: 0x00FC			Register Name: TV_DEBUG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R	0x0	TV_FIFO_UNDERFLOW 0: Not underflow 1: Underflow The flag shows whether the FIFOs in underflow status.
29	/	/	/
28	R	0x0	TV_FIELD_POL 0: Second field 1: First field The flag indicates the current field polarity.
27:12	/	/	/
13	R/W	0x0	ECC_FIFO_BYPASS 0: Used 1: Bypass Setup that whether to bypass ECC FIFO.
12	/	/	/
11:0	R	0x0	TV_CURRENT_LINE Current scan line.

7.1.6.16. 0x0100 TV CEU Control Register(Default Value: 0x0000_0000)

Offset: 0x0100			Register Name: TV_CEU_CTL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CEU_EN 0: Bypass 1: Enable

			Enable CEU function.
30:0	/	/	/

7.1.6.17. 0x0110+N*0x04 TV CEU Coefficient Register0(Default Value: 0x0000_0000)

Offset: 0x0110+N*0x04 (N=0,1,2,4,5,6,8,9,10)			Register Name: TV_CEU_COEF_MUL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8	R/W	0x0	CEU_COEF_MUL_VALUE CEU_Coef_Mul_Value only can be 0 or 1. REG Map: N=0: Rr N=1: Rg N=2: Rb N=4: Gr N=5: Gg N=6: Gb N=8: Br N=9: Bg N=10: Bb

7.1.6.18. 0x0140+N*0x04 TV CEU Coefficient Register(Default Value: 0x0000_0000)

Offset: 0x0140+N*0x04 (N=0,1,2)			Register Name: TV_CEU_COEF_RANG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	R/W	0x0	CEU_COEF_RANGE_MIN Unsigned 10-bit value, range of [0,1023]
15:10	/	/	/
9:0	R/W	0x0	CEU_COEF_RANGE_MAX Unsigned 10-bit value, range of [0,1023]

7.1.6.19. 0x01F0 TV Safe Period Register(Default Value: 0x0000_0000)

Offset: 0x01F0			Register Name: TV_SAFE_PERIOD_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28:16	R/W	0x0	SAFE_PERIOD_FIFO_NUM When the data length in line buffer is more than SAFE_PERIOD_FIFO_NUM,LCD controller will allow dram controller to stop

			working to change frequency.
15:4	R/W	0x0	SAFE_PERIOD_LINE Set a fixed line and during the line time,LCD controller allow dram controller to change frequency.The fixed line should be set in the blanking area.
3	/	/	/
2:0	R/W	0x0	SAFE_PERIOD_MODE Select the save mode 000: unsafe 001: safe 010: safe at FIFO_CURR_NUM > SAFE_PERIOD_FIFO_NUM 011: safe at 2 and safe at sync active 100: safe at line

7.1.6.20. 0x0300 TV Fill Data Control Register(Default Value: 0x0000_0000)

Offset: 0x0300			Register Name: TV_FILL_CTL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	TV_FILL_EN 0: Bypass 1: Enable Enable the fill data function in blanking area. This is only used in HDMI 3D mode.
30:0	/	/	/

7.1.6.21. 0x0304+N*0x0C TV Fill Data Begin Register(Default Value: 0x0000_0000)

Offset: 0x0304+N*0x0C (N=0,1,2)			Register Name: TV_FILL_BEGIN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:0	R/W	0x0	FILL_BEGIN Set the begin point of the fill data area.

7.1.6.22. 0x0308+N*0x0C TV Fill Data End Register(Default Value: 0x0000_0000)

Offset: 0x0308+N*0x0C (N=0,1,2)			Register Name: TV_FILL_END_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:0	R/W	0x0	FILL_END Set the end point of the fill data area.

7.1.6.23. 0x030C+N*0x0C TV Fill Data Value Register(Default Value: 0x0000_0000)

Offset: 0x030C+N*0x0C (N=0,1,2)			Register Name: TV_FILL_DATA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:0	R/W	0x0	FILL_VALUE Set the data value which is used to fill into the blanking area between two frames in HDMI 3D mode.

7.1.6.24. 0x0330 TV Data IO Polarity0 Register(Default Value: 0x0000_0000)

Offset: 0x0330			Register Name: TV_DATA_IO_POL0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	R/W	0x0	R/Cb Channel Data_Inv [bit9:0] 0: normal polarity 1: invert the specify output
15:10	/	/	/
9:0	R/W	0x0	G/Y Channel Data_Inv [bit9:0] 0: normal polarity 1: invert the specify output

7.1.6.25. 0x0334 TV Data IO Polarity1 Register(Default Value: 0x0000_0000)

Offset: 0x0334			Register Name: TV_DATA_IO_POL1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	R/W	0x0	B/Cr Channel Data_Inv 0: normal polarity 1: invert the specify output
15:0	/	/	/

7.1.6.26. 0x0338 TV Data IO Trigger0 Register(Default Value: 0x03FF_03FF)

Offset: 0x0338			Register Name: TV_DATA_IO_TRI0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	R/W	0x3ff	R/Cb Channel Data_Output_Tri_En 1: disable 0: enable Only higher 6-bit is valid.

15:10	/	/	/
9:0	R/W	0x3ff	G/Y Channel Data_Output_Tri_En 1: disable 0: enable Only higher 6-bit is valid.

7.1.6.27. 0x033C TV Data IO Trigger1 Register(Default Value: 0x03FF_0000)

Offset: 0x033C			Register Name: TV_DATA_IO_TRI1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	R/W	0x3ff	B/Cr Channel Data_Output_Tri_En 1: disable 0: enable Only higher 6-bit is valid.
15:0	/	/	/

7.1.6.28. 0x0340 TV Pixel-depth Mode Register(Default Value: 0x0000_0000)

Offset: 0x0340			Register Name: TV_PIXELDEPTH_MODE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0	Colorbar Pixeldepth mode(The bit is valid only when Colorbar output) 0: 8-bit mode When data source is colorbar, the 8-bit mode of colorbar pattern is transferred. 1: 10-bit mode When data source is colorbar, the 10-bit mode of colorbar pattern is transferred.

7.2. TVE

7.2.1. Overview

The TV Encoder(TVE) module is a highly programmable digital video encoder supporting worldwide video standards Composite Video Broadcast Signal (CVBS).

The TVE includes the following features:

- 1 channel CVBS, PAL-D and NTSC-M supported
- Plug status auto detecting
- 10 bits DAC output

7.2.2. Block Diagram

Figure 7-7 shows a block diagram of the TVE.

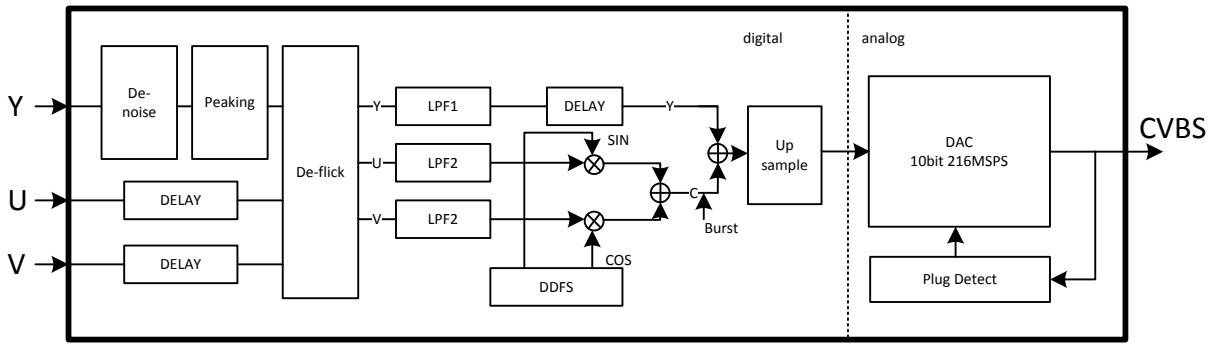

Figure 7- 7. TVE Block Diagram

7.2.3. Operations and Functional Descriptions

7.2.3.1. External Signals

Table 7-3 describes the external signals of TVE.

Table 7- 3. TVE External Signals

Pin Name	Function Description	Type
TV_VCC	TV DAC power	P
TV_OUT	TV CVBS output	AO

7.2.3.2. Clock Sources

The TVE module requires one clock with 50% duty. Digital circuit and Analog circuit work by this clock. Mode and Clock frequency is shown below.

Table 7- 4. TVE Clock Sources

Mode	TVE Clock Frequency
NTSC	216 MHz
PAL	216 MHz

7.2.4. Programming Guidelines

(1) Operate TVE module by the following step, the process is shown in Figure 7-8.
Step1: Set CCU clock source for TVE, and release AHB bus, and module reset.
Step2: Initial DAC amplitude value from efuse calibration value which has burned.
Step3: Enable plug-in detect function, and detect plug-in status every 200ms.
Step4: When plug-in has detected, configure TVE module to output mode setting by application.

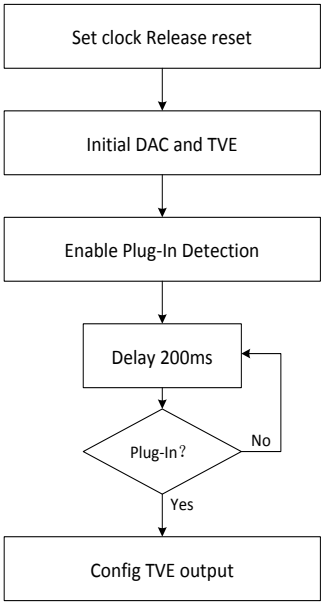

Figure 7- 8. Operate TVE Process

(2) Auto Detect Function

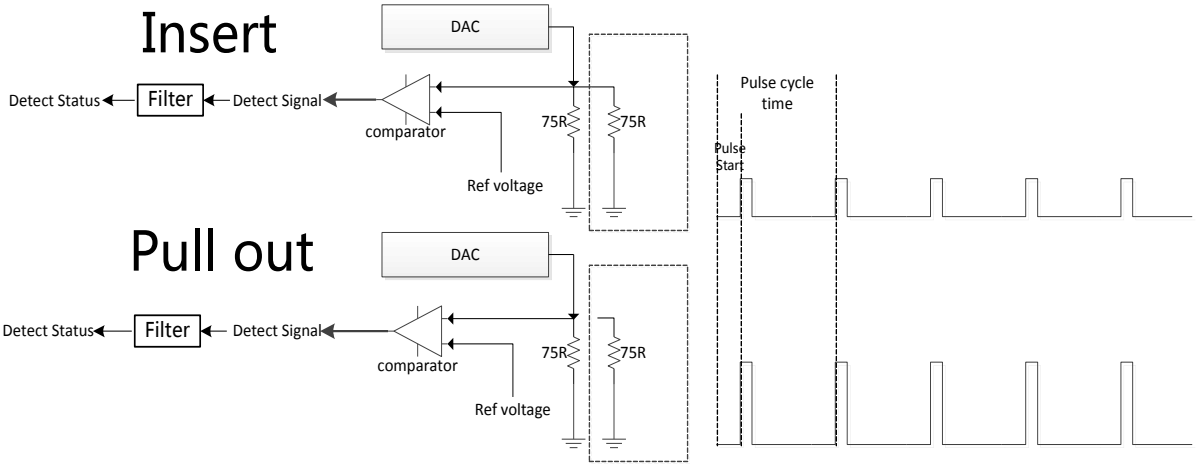

Figure 7- 9. Auto Detect Function

DAC outputs constant current, when insert, external load is 37.5Ω; when pull out, external load is 75Ω. The method that comparator judges pin level can detect plug action.

Because plug action may exist jitter, then there need be a filter to filter jitter, the debounce time of filter is set through the bit[3:0] of TV Encoder Auto Detection de-bounce Setting Register.

The pulse cycle time can be set through the bit[30:16] of TV Encoder Auto Detect Configuration Register1, the pulse start time can be set through the bit[14:0] of TV Encoder Auto Detect Configuration Register1.The clock sources of the two time are 32KHz clock.

Pulse width is cycle time of 4 clock sources.

Pulse amplitude can be set through the bit[9:0] of TV Encoder Auto Detect Configuration Register0.

(3) DAC Calibration

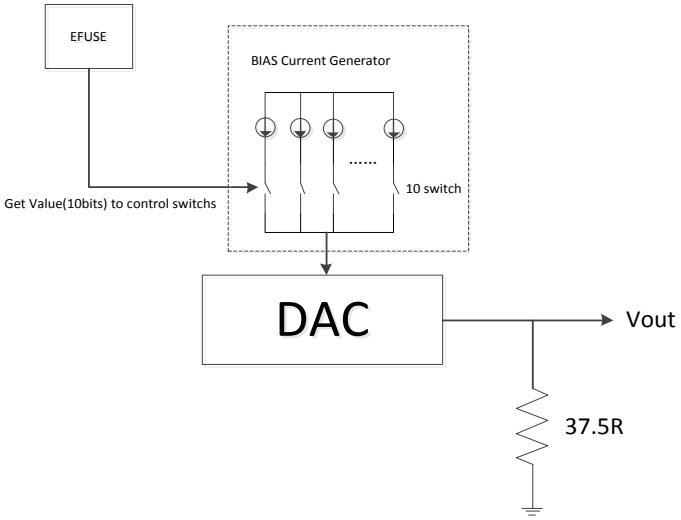

Figure 7- 10. DAC Calibration

After FT, 10-bit calibration value is burned into efuse. Every time software can read the 10-bit calibration value from

efuse, to control BIAS current and BIAS current switch, then a specific BIAS current is generated to calibrate maximum output voltage of DAC.

7.2.5. Register List

Module Name	Base Address
TVE_TOP	0x06520000
TVE	0x06524000

Register Name	Offset	Description
TVE_TOP		
TVE_DAC_MAP	0x0020	TV Encoder DAC MAP Register
TVE_DAC_STATUS	0x0024	TV Encoder DAC STAUTS Register
TVE_DAC_CFG0	0x0028	TV Encoder DAC CFG0 Register
TVE_DAC_CFG1	0x002C	TV Encoder DAC CFG1 Register
TVE_DAC_CFG2	0x0030	TV Encoder DAC CFG2 Register
TVE_DAC_CFG3	0x0034	TV Encoder DAC CFG2 Register
TVE_DAC_TEST	0x00F0	TV Encoder DAC TEST Register
TVE		
TVE_000_REG	0x0000	TV Encoder Clock Gating Register
TVE_004_REG	0x0004	TV Encoder Configuration Register
TVE_008_REG	0x0008	TV Encoder DAC Register1
TVE_00C_REG	0x000C	TV Encoder Notch and DAC Delay Register
TVE_010_REG	0x0010	TV Encoder Chroma Frequency Register
TVE_014_REG	0x0014	TV Encoder Front/Back Porch Register
TVE_018_REG	0x0018	TV Encoder HD Mode VSYNC Register
TVE_01C_REG	0x001C	TV Encoder Line Number Register
TVE_020_REG	0x0020	TV Encoder Level Register
TVE_024_REG	0x0024	TV Encoder DAC Register2
TVE_030_REG	0x0030	TV Encoder Auto Detection Enable Register
TVE_034_REG	0x0034	TV Encoder Auto Detection Interrupt Status Register
TVE_038_REG	0x0038	TV Encoder Auto Detection Status Register
TVE_03C_REG	0x003C	TV Encoder Auto Detection De-bounce Setting Register
TVE_0F8_REG	0x00F8	TV Encoder Auto Detect Configuration Register0
TVE_0FC_REG	0x00FC	TV Encoder Auto Detect Configuration Register1
TVE_100_REG	0x0100	TV Encoder Color Burst Phase Reset Configuration Register
TVE_104_REG	0x0104	TV Encoder VSYNC Number Register
TVE_108_REG	0x0108	TV Encoder Notch Filter Frequency Register
TVE_10C_REG	0x010C	TV Encoder Cb/Cr Level/Gain Register
TVE_110_REG	0x0110	TV Encoder Tint and Color Burst Phase Register
TVE_114_REG	0x0114	TV Encoder Burst Width Register
TVE_118_REG	0x0118	TV Encoder Cb/Cr Gain Register

Register Name	Offset	Description
TVE_11C_REG	0x011C	TV Encoder Sync and VBI Level Register
TVE_120_REG	0x0120	TV Encoder White Level Register
TVE_124_REG	0x0124	TV Encoder Video Active Line Register
TVE_128_REG	0x0128	TV Encoder Video Chroma BW and CompGain Register
TVE_12C_REG	0x012C	TV Encoder Register
TVE_130_REG	0x0130	TV Encoder Re-sync Parameters Register
TVE_134_REG	0x0134	TV Encoder Slave Parameter Register
TVE_138_REG	0x0138	TV Encoder Configuration Register0
TVE_13C_REG	0x013C	TV Encoder Configuration Register1
TVE_380_REG	0x0380	TV Encoder Low Pass Control Register
TVE_384_REG	0x0384	TV Encoder Low Pass Filter Control Register
TVE_388_REG	0x0388	TV Encoder Low Pass Gain Register
TVE_38C_REG	0x038C	TV Encoder Low Pass Gain Control Register
TVE_390_REG	0x0390	TV Encoder Low Pass Shoot Control Register
TVE_394_REG	0x0394	TV Encoder Low Pass Coring Register
TVE_3A0_REG	0x03A0	TV Encoder Noise Reduction Register

7.2.6. Register Description

7.2.6.1. 0x0020 TV Encoder DAC MAP Register(Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: TVE_DAC_MAP
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6:4	R/W	0x0	DAC_MAP 000: OUT0 Others: Reserved
3:2	/	/	/
1:0	R/W	0x0	DAC_SEL 00: Reserved 01: TVE0 10: Reserved 11: Reserved

7.2.6.2. 0x0024 TV Encoder DAC Status Register(Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: TVE_DAC_STATUS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1:0	R/W	0x0	DAC_STATUS 00: Unconnected

Offset: 0x0024			Register Name: TVE_DAC_STATUS
Bit	Read/Write	Default/Hex	Description
			01: Connected 11: Short to ground 10: Reserved

7.2.6.3. 0x0028 TV Encoder DAC Configuration0 Register(Default Value: 0x8000_4200)

Offset: 0x0028			Register Name: TVE_DAC_CFG0
Bit	Read/Write	Default/Hex	Description
31	R/W	0x1	DAC_CLOCK_INVERT 0: Not invert 1: Invert
30:26	/	/	/
25:16	R/W	0x0	CALI_IN
15:12	R/W	0x4	LOW_BIAS 500uA to 4mA
11:10	/	/	/
9	R/W	0x1	BIAS_EXT_SEL 0: disable 1: enable (A_SEL_BIAS_ADDA)
8	R/W	0x0	BIAS_INT_SEL 0: disable 1: enable (A_SEL_BIAS_RES)
7:5	/	/	/
4	R/W	0x0	BIAS_REF_INT_EN 0: disable 1: enable (A_EN_RESREF)
3:1	/	/	/
0	R/W	0x0	DAC_EN 0: Disable 1: Enable

7.2.6.4. 0x002C TV Encoder DAC Configuration1 Register(Default Value: 0x0000_023A)

Offset: 0x002C			Register Name: TVE_DAC_CFG1
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9	R/W	0x1	REF_EXT_SEL 0: disable

Offset: 0x002C			Register Name: TVE_DAC_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
			1: enable (A_SEL_DETREF_LDO)
8	R/W	0x0	REF_INT_SEL 0: disable 1: enable (A_SEL_DETREF_RES)
7:6	/	/	/
5:4	R/W	0x3	REF2_SEL 00: 0.25V 01: 0.30V 10: 0.35V 11: 0.40V (a_refslct2<1:0>)
3:0	R/W	0xA	REF1_SEL 0000: 0.50V 0001: 0.55V 0010: 0.60V 0011: 0.65V 0100: 0.70V 0101: 0.75V 0110: 0.80V 0111: 0.85V 1000: 0.90V 1001: 0.95V 1010: 1.00V 1011: 1.05V 1100: 1.10V 1101: 1.15V 1110: 1.20V 1111: 1.25V (a_refslct1<3:0>) The reference voltage is used for hot plug detect function.

7.2.6.5. 0x0030 TV Encoder DAC Configuration2 Register(Default Value: 0x0000_0010)

Offset: 0x0030			Register Name: TVE_DAC_CFG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
12:8	R/W	0x0	AB (I config output current for different peak voltage)
7:6	R/W	0x0	S2S1
5:0	R/W	0x10	R_SET

7.2.6.6. 0x0034 TV Encoder DAC Configuration3 Register(Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: TVE_DAC_CFG3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	R/W	0x0	FORCE_DATA_SET Force DAC input data
15:1	/	/	/
0	R/W	0x0	FORCE_DATA_EN 0:DAC input data from TVE 1: DAC input data from FORCE_DATA_SET

7.2.6.7. 0x00F0 TV Encoder DAC Test Register(Default Value: 0x0000_0000)

Offset: 0x00F0			Register Name: TVE_DAC_TEST
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	R/W	0x0	DAC_TEST_LENGTH DAC TEST DATA LENGTH
15:6	/	/	/
5:4	R/W	0x0	DAC_TEST_SEL 00: DAC0 Others:Reserved
3:1	/	/	/
0	R/W	0x0	DAC_TEST_ENABLE 0: Reserved 1: Repeat DAC data from DAC sram

7.2.6.8. 0x0000 TV Encoder Clock Gating Register(Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: TVE_000_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CLOCK_GATE_DIS 0: Enable 1: Disable
30:29	/	/	/
28	R/W	0x0	BIST_EN 0: Normal mode 1: Bist mode
27:23	/	/	/

Offset: 0x0000			Register Name: TVE_000_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
22	R/W	0x0	upsample for YPbPr 0: 1x 1: 2x
21:20	R/W	0x0	upsample for CVBS Out up sample 00: 27 MHz 01: 54 MHz 10: 108 MHz 11: 216 MHz
19:1	/	/	/
0	R/W	0x0	TVE_EN 0: Disable 1: Enable Video Encoder enable, default disable, write 1 to take it out of the reset state

7.2.6.9. 0x0004 TV Encoder Configuration Register(Default Value: 0x0001_0000)

Offset: 0x0004			Register Name: TVE_004_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29	R/W	0x0	BYPASS_TV 0: Disable 1: Enable
28:27	R/W	0x0	DAC_SRC_SEL 00: TV Encoder 01: LCD controller, override all other TV encoder setting, the DAC clock can from LCD controller. 10: DAC test mode,DAC using DAC clock 11: DAC test mode, DAC using AHB clock
26	R/W	0x0	DAC_CONTROL_LOGIC_CLOCK_SEL 0: Using 27 MHz clock or 74.25 MHz clock depend on CCU setting 1: Using 54 MHz clock or 148.5 MHz clock depend on CCU setting
25	R/W	0x0	CORE_DATAPATH_LOGIC_CLOCK_SEL 0: Using 27 MHz clock or 74.25 MHz clock depend on CCU setting 1: Using 54 MHz clock or 148.5 MHz clock depend on CCU setting
24	R/W	0x0	CORE_CONTROL_LOGIC_CLOCK_SEL 0: Using 27 MHz clock or 74.25 MHz clock depend on CCU setting 1: Using 54 MHz clock or 148.5 MHz clock depend on CCU setting
23:21	/	/	/
20	R/W	0x0	CB_CR_SEQ_FOR_422_MODE 0: Cb first 1: Cr first

Offset: 0x0004			Register Name: TVE_004_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
19	R/W	0x0	INPUT_CHROMA_DATA_SAMPLING_RATE_SEL 0: 4:4:4 1: 4:2:2
18	R/W	0x0	YUV_RGB_OUTPUT_EN 0: CVBS 1: Reserved
17	R/W	0x0	YC_EN S-port Video enable Selection. 0: Y/C is disable 1: Reserved This bit selects whether the S-port(Y/C) video output is enabled or disabled.
16	R/W	0x1	CVBS_EN Composite video enables selection 0: Composite video is disabled, Only Y/C is enabled 1: Composite video is enabled., CVBS and Y/C are enabled This bit selects whether the composite video output (CVBS) is enabled or disabled.
15:10	/	/	/
9	R/W	0x0	COLOR_BAR_TYPE 0: 75/7.5/75/7.5 (NTSC), 100/0/75/0(PAL) 1: 100/7.5/100/7.5(NTSC), 100/0/100/0(PAL)
8	R/W	0x0	COLOR_BAR_MODE Standard Color bar input selection 0: The Video Encoder input is coming from the Display Engineer 1: The Video Encoder input is coming from an internal standard color bar generator. This bit selects whether the Video Encoder video data input is replaced by an internal standard color bar generator or not.
7:5	/	/	/
4	R/W	0x0	MODE_1080I_1250LINE_SEL 0: 1125 Line mode 1: 1250 Line mode
3:0	R/W	0x0	TVMODE_SELECT 0000: NTSC 0001: PAL 0010: Reserved 0011: Reserved 01xx: Reserved 100x: Reserved 101x: Reserved 110x: Reserved 111x: Reserved **Note: Changing this register value will cause some relative register setting to**

Offset: 0x0004			Register Name: TVE_004_REG
Bit	Read/Write	Default/Hex	Description
			relative value.

7.2.6.10. 0x0008 TV Encoder DAC Register1(Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: TVE_008_REG
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:4	R/W	0x0	DAC0_SRC_SEL 000: Composite Others: Reserved
3:0	/	/	/

7.2.6.11. 0x000C TV Encoder Notch and DAC Delay Register(Default Value: 0x0201_4924)

Offset: 0x000C			Register Name: TVE_00C_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	CHROMA_FILTER_ACTIVE_VALID 0: Disable 1: Enable
30	R/W	0x0	LUMA_FILTER_LTI_ENABLE 0: Disable Luma filter lti 1: Enable Luma filter lti
27:25	R/W	0x1	Y_DELAY_BEFORE_DITHER
24	R/W	0x0	HD_MODE_CB_FILTER_BYPASS 0: Bypass Enable 1: Bypass Disable
23	R/W	0x0	HD_MODE_CR_FILTER_BYPASS 0: Bypass Enable 1: Bypass Disable
22	R/W	0x0	CHROMA_FILTER_1_444_EN 0: Chroma Filter 1 444 Disable 1: Chroma Filter 1 444 Enable
21	R/W	0x0	CHROMA_HD_MODE_FILTER_EN 0: Chroma HD Filter Disable 1: Chroma HD Filter Enable
20	R/W	0x0	CHROMA_FILTER_STAGE_1_BYPASS 0: Chroma Filter Stage 1 Enable 1: Chroma Filter Stage 1 bypass
19	R/W	0x0	CHROMA_FILTER_STAGE_2_BYPASS 0: Chroma Filter Stage 2 Enable 1: Chroma Filter Stage 2 bypass

Offset: 0x000C			Register Name: TVE_00C_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
18	R/W	0x0	CHROMA_FILTER_STAGE_3_BYPASS 0: Chroma Filter Stage 3 Enable 1: Chroma Filter Stage 3 bypass
17	R/W	0x0	LUMA_FILTER_BYPASS 0: Luma Filter Enable 1: Luma Filter bypass
16	R/W	0x1	NOTCH_EN 0: The luma notch filter is bypassed 1: The luma notch filter is operating Luma notch filter on/off selection **Note: This bit selects if the luma notch filter is operating or bypassed.**
15:12	R/W	0x4	C_DELAY_BEFORE_DITHER
11:0	R/W	0x924	Reserved

7.2.6.12. 0x0010 TV Encoder Chroma Frequency Register(Default Value: 0x21F0_7C1F)

Offset: 0x0010			Register Name: TVE_010_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x21f07c1f	CHROMA_FREQ Specify the ratio between the color burst frequency. 32 bits unsigned fraction. Default value is h21f07c1f, which is compatible with NTSC specs. 3.5795455 MHz (X'21F07C1F'): NTSC-M, NTSC-J 4.43361875 MHz(X'2A098ACB'): PAL-B, D, G, H,I, N 3.582056 MHz (X'21F69446'): PAL-N(Argentina) 3.579611 MHz (X'21E6EFE3'): PAL-M

7.2.6.13. 0x0014 TV Encoder Front/Back Porch Register(Default Value: 0x0076_0020)

Offset: 0x0014			Register Name: TVE_014_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24:16	R/W	0x76	BACK_PORCH Specify the width of the back porch in encoder clock cycles. Min value is (*burst_width+breeze_way*+17). 8 bits unsigned integer. 720p mode, is 260 1080i/p mode, is 192
15:12	/	/	/
11:0	R/W	0x20	FRONT_PORCH Must be even. Specify the width of the front porch in encoder clock cycles. 6 bits unsigned even integer. Allowed range is 10 to 62.

Offset: 0x0014			Register Name: TVE_014_REG
Bit	Read/Write	Default/Hex	Description
			In 1080i mode the value is 44.

7.2.6.14. 0x0018 TV Encoder HD Mode VSYNC Register(Default Value: 0x0000_0016)

Offset: 0x0018			Register Name: TVE_018_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0x0	BROAD_PLUS_CYCLE_NUMBER_IN_HD_MODE_VSYNC
15:12	/	/	/
11:0	R/W	0x16	FRONT_PORCH_LIKE_IN_HD_MODE_VSYNC

7.2.6.15. 0x001C TV Encoder Line Number Register(Default Value: 0x0016_020D)

Offset: 0x001C			Register Name: TVE_01C_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:16	R/W	0x16	FIRST_VIDEO_LINE Specify the index of the first line in a field/frame to have active video. 8 bits unsigned integer. For interlaced video: When VSync5=B'0', FirstVideoLine is restricted to be greater than 7. When VSync5=B'1', FirstVideoLine is restricted to be greater than 9.
15:11	/	/	/
10:0	R/W	0x20D	NUM_LINES Specify the total number of lines in a video frame. 11 bits unsigned integer. Allowed range is 0 to 2048. For interlaced video: When NTSC, and FirstVideoLine is greater than 20, then NumLines is restricted to be greater than 2*(FirstVideoLine+18). When NTSC, and FirstVideoLine is not greater than 20, then NumLines is restricted to be greater than 77. When PAL, and FirstVideoLine is greater than 22, then NumLines is restricted to be greater than 2*(FirstVideoLine+18). When PAL, and FirstVideoLine is not greater than 22, then NumLines is restricted to be greater than 81. If NumLines is even, then it is restricted to be divisible by 4. If NumLines is odd, then it is restricted to be divisible by 4 with a remainder of 1.

7.2.6.16. 0x0020 TV Encoder Level Register(Default Value: 0x00F0_011A)

Offset: 0x0020			Register Name: TVE_020_REG
Bit	Read/Write	Default/Hex	Description

Offset: 0x0020			Register Name: TVE_020_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	R/W	0xf0	BLANK_LEVEL Specify the blank level setting for active lines. This is 10 bits unsigned integer. Allowed range is from 0 to 1023.
15:10	/	/	/
9:0	R/W	0x11a	BLACK_LEVEL Specify the black level setting. This is 10 bits unsigned integer. Allowed range is from 240 to 1023.

7.2.6.17. 0x0030 TV Encoder Auto Detection Enable Register(Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: TVE_030_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	DAC_AUTO_DETECT_MODE_SEL 0: Old Mode 1: New Mode
30:17	/	/	/
16	R/W	0x0	DAC0_AUTO_DETECT_INTERRUPT_EN
15:1	/	/	/
0	R/W	0x0	DAC0_AUTO_DETECT_ENABLE

7.2.6.18. 0x0034 TV Encoder Auto Detection Interrupt Status Register(Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: TVE_034_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W1C	0x0	DAC0_AUTO_DETECT_INTERRUPT_ACTIVE_FLAG Write 1 to inactive DAC0 auto detection interrupt

7.2.6.19. 0x0038 TV Encoder Auto Detection Status Register(Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: TVE_038_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1:0	R	0x0	DAC0_STATUS 00: Unconnected 01: Connected 11: Short to ground 10: Reserved

7.2.6.20. 0x003C TV Encoder Auto Detection Debounce Setting Register(Default Value: 0x0000_0000)

Offset: 0x003C			Register Name: TVE_03C_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	R/W	0x0	DAC_TEST_REGISTER DAC test register.
15:4	/	/	/
3:0	R/W	0x0	DAC0_DE_BOUNCE_TIMES The de_bounce time for hot plug detect function.

7.2.6.21. 0x00F8 TV Encoder Auto Detection Configuration Register0(Default Value: 0x0000_0000)

Offset: 0x00F8			Register Name: TVE_0F8_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9:0	R/W	0x0	DETECT_PULSE_VALUE Use for DAC data input at auto detect pluse. Set the pulse amplitude.

7.2.6.22. 0x00FC TV Encoder Auto Detection Configuration Register1(Default Value: 0x0000_0000)

Offset: 0x00FC			Register Name: TVE_0FC_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30:16	R/W	0x0	DETECT_PULSE_PERIODS Use 32K clock
15	/	/	/
14:0	R/W	0x0	DETECT_PULSE_START Detect signal start time

7.2.6.23. 0x0100 TV Encoder Color Burst Phase Reset Configuration Register (Default Value: 0x0000_0001)

Offset: 0x0100			Register Name: TVE_100_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1:0	R/W	0x1	COLOR_PHASE_RESET Color burst phase period selection These bits select the number of fields or lines after which the color burst phase is reset to its initial value as specified by the ChromaPhase parameter, This parameter is application only for interlaced video. 00: 8 field

Offset: 0x0100			Register Name: TVE_100_REG
Bit	Read/Write	Default/Hex	Description
			01: 4 field 10: 2 lines 11: only once

7.2.6.24. 0x0104 TV Encoder VSYNC Number Register (Default Value: 0x0000_0000)

Offset: 0x0104			Register Name: TVE_104_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	VSync5 Number of equalization pulse selection This bit selects whether the number of equalization pulses is 5 or 6. This parameter is applicable only for interlaced video. 0: 5 equalization pulse(default) 1: 6 equalization pulses

7.2.6.25. 0x0108 TV Encoder Notch Filter Frequency Register (Default Value: 0x0000_0002)

Offset: 0x0108			Register Name: TVE_108_REG
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2:0	R/W	0x2	NOTCH_FREQ Luma notch filter center frequency selection These bits select the luma notch filter (which is a band-reject filter) center frequency. In two of the selections, the filter width affects also the selection of the center frequency. 000: 1.1875 001: 1.1406 010: 1.0938 when notch_wide value is B‘1’ (this selection is proper for CCIR-NTSC), or 1.0000 when notch_wide value is B‘0’ 011: 0.9922. This selection is proper for NTSC with square pixels 100: 0.9531. This selection is proper for PAL with square pixel 101: 0.8359 when notch_wide value is B‘1’ (this selection is proper for CCIR-PAL), or 0.7734 when notch_wide value is B‘0’ 110: 0.7813 111: 0.7188

7.2.6.26. 0x010C TV Encoder Cb/Cr Level/Gain Register (Default Value: 0x0000_004F)

Offset: 0x010C	Register Name: TVE_10C_REG

Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:8	R/W	0x0	CR_BURST_LEVEL Specify the amplitude of the Cr burst. 8 bit 2's complement integer. Allowed range is from (-127) to 127.
7:0	R/W	0x4f	CB_BURST_LEVEL Specify the amplitude of the Cb burst. 8 bit 2's complement integer. Allowed range is from (-127) to 127.

7.2.6.27. 0x0110 TV Encoder Tint and Color Burst Phase Register (Default Value: 0x0000_0000)

Offset: 0x0110			Register Name: TVE_110_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:16	R/W	0x0	TINT Specify the tint adjustment of the chroma signal for CVBS and Y/C outputs. The adjustment is effected by setting the sub-carrier phase to the value of this parameter. 8.8 bit unsigned fraction. Units are cycles of the color burst frequency.
15:8	/	/	/
7:0	R/W	0x0	CHROMA_PHASE Specify the color burst initial phase (*ChromaPhase)*. 8.8 bit unsigned fraction. Units are cycles of the color burst frequency. The color burst is set to this phase at the first *HSYNC* and then reset to the same value at further *HSYNC*s as specified by the *CPhaseRset* bits of the *EncConfig5* parameter (see above)

7.2.6.28. 0x0114 TV Encoder Burst Width Register (Default Value: 0x0016_447E)

Offset: 0x0114			Register Name: TVE_114_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0x0	BACK_PORCH Breezeway like in HD mode VSync 720p mode, is 220 2080i/p mode is 88(default)
23	/	/	/
22:16	R/W	0x16	BREEZEWAY Must be even. Specify the width of the breezeway in encoder clock cycles. 5 bit unsigned integer. Allowed range is 0 to 31. In 1080i mode, is 44 In 1080p mode, is 44 In 720p mode,is 40

Offset: 0x0114			Register Name: TVE_114_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
15	/	/	/
14:8	R/W	0x44	BURST_WIDTH Specify the width of the color frequency burst in encoder clock cycles. 7 bit unsigned integer. Allowed range is 0 to 127. In hd mode, ignored
7:0	R/W	0x7e	HSYNC_WIDTH Specify the width of the horizontal sync pulse in encoder clock cycles. Min value is 16. Max value is (*FrontPorch + ActiveLine - BackPorch*). Default value is 126. The sum of *HSyncSize* and *BackPorch* is restricted to be divisible by 4. In 720p mode, is 40 In 1080i/p mode, is 44

7.2.6.29. 0x0118 TV Encoder Cb/Cr Gain Register (Default Value: 0x0000_A0A0)

Offset: 0x0118			Register Name: TVE_118_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:8	R/W	0xa0	CR_GAIN Specify the Cr color gain. 8-bit unsigned fraction.
7:0	R/W	0xa0	CB_GAIN Specify the Cb color gain. 8-bit unsigned fraction.

7.2.6.30. 0x011C TV Encoder Sync and VBI Level Register (Default Value: 0x0010_00F0)

Offset: 0x011C			Register Name: TVE_11C_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	R/W	0x10	SYNC_LEVEL Specify the sync pulse level setting. 8-bit unsigned integer. Allowed range is from 0 to *ABlankLevel*-1 or *VBlankLevel*-1 (whichever is smaller).
15:10	/	/	/
9:0	R/W	0xf0	VBLANK_LEVEL Specify the blank level setting for non active lines. 10-bit unsigned integer. Allow range is from 0 to 1023.

7.2.6.31. 0x0120 TV Encoder White Level Register (Default Value: 0x01E8_0320)

Offset: 0x0120			Register Name: TVE_120_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/

Offset: 0x0120			Register Name: TVE_120_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
25:16	R/W	0x1e8	HD_SYNC_BREEZEWAY_LEVEL Specify the breezeway level setting. 10-bit unsigned integer. Allowed range is from 0 to 1023.
15:10	/	/	/
9:0	R/W	0x320	WHITE_LEVEL Specify the white level setting. 10-bit unsigned integer. Allowed range is from black_level+1 or vbi_blank_level +1 (whichever is greater) to 1023.

7.2.6.32. 0x0124 TV Encoder Video Active Line Register (Default Value: 0x0000_05A0)

Offset: 0x0124			Register Name: TVE_124_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11:0	R/W	0x5A0	ACTIVE_LINE Specify the width of the video line in encoder clock cycles. 12-bit unsigned multiple of 4 integer. Allowed range is from 0 to 4092.

7.2.6.33. 0x0128 TV Encoder Video Chroma BW and CompGain Register (Default Value: 0x0000_0000)

Offset: 0x0128			Register Name: TVE_128_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17:16	R/W	0x0	CHROMA_BW Chroma filter bandwidth selection This bit specifies whether the bandwidth of the chroma filter is: 00: Narrow width 0.6 MHz 01: Wide width 1.2 MHz 10: Extra width 1.8 MHz 11: Ultra width 2.5 MHz Default is 0.6 MHz(value 0)
15:2	/	/	/
1:0	R/W	0x0	COMP_CH_GAIN Chroma gain selection for the composite video signal. These bits specify the gain of the chroma signal for composing with the luma signal to generate the composite video signal: 00: 100% 01: 25% 10: 50% 11: 75%

7.2.6.34. 0x012C TV Encoder Register (Default Value: 0x0000_0101)

Offset: 0x012C			Register Name: TVE_12C_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8	R/W	0x1	NOTCH_WIDTH Luma notch filter width selection This bit selects the luma notch filter (which is a band-reject filter) width. 0: Narrow 1: Wide
7:1	/	/	/
0	R/W	0x1	COMP_YUV_EN This bit selects if the components video output are the RGB components or the YUV components. 0: The three component outputs are the RGB components. 1: The three component outputs are the YUV components, (i.e. the color conversion unit is by-passed)

7.2.6.35. 0x0130 TV Encoder Re-sync Parameters Register (Default Value: 0x0010_0001)

Offset: 0x0130			Register Name: TVE_130_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	RE_SYNC_FIELD Re-sync field
30	R/W	0x0	RE_SYNC_DIS 0: Re-Sync Enable 1: Re-Sync Disable
29:27	/	/	/
26:16	R/W	0x10	RE_SYNC_LINE_NUM Re-sync line number from TCON
15:11	/	/	/
10:0	R/W	0x1	RE_SYNC_PIXEL_NUM Re-sync line pixel from TCON

7.2.6.36. 0x0134 TV Encoder Slave Parameter Register (Default Value: 0x0000_0000)

Offset: 0x0134			Register Name: TVE_134_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8	R/W	0x0	SLAVE_THRESH Horizontal line adjustment threshold selection This bit selects whether the number of lines after which the Video Encoder starts the horizontal line length adjustment is slave mode is 0 or 30.

Offset: 0x0134			Register Name: TVE_134_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			0: Number of lines is 0 1: Number of lines is 30
7:1	/	/	/
0	R/W	0x0	SLAVE_MODE Slave mode selection This bit selects whether the Video Encoder is sync slave, partial slave or sync master. It should be set to B'0'. 0: The Video Encoder is not a full sync slave (i.e. it is a partial sync slave or a sync master) 1: Reserved

7.2.6.37. 0x0138 TV Encoder Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0138			Register Name: TVE_138_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8	R/W	0x0	INVERT_TOP Field parity input signal (top_field) polarity selection. This bit selects whether the top field is indicated by a high level of the field parity signal or by the low level. The bit is applicable both when the Video Encoder is the sync master and when the Video Encoder is the sync slave. 0: Top field is indicated by low level 1: Top field is indicated by high level
7:1	/	/	/
0	R/W	0x0	UV_ORDER This bit selects if the sample order at the chroma input to the Video Encoder is Cb first (i.e. Cb 0 Cr 0 Cb 1 Cr 1) or Cr first (i.e. Cr 0 Cb 0 Cr 1 Cb 1). 0: The chroma sample input order is Cb first 1: The chroma sample input order is Cr first

7.2.6.38. 0x013C TV Encoder Configuration Register (Default Value: 0x0000_0001)

Offset: 0x013C			Register Name: TVE_13C_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:27	/	/	/
26:24	R/W	0x0	RGB_SYNC R, G and B signals sync embedding selection. These bits specify whether the sync signal is added to each of the R, G and B components (b'1') or not (b'0'). The bit[26] specify if the R signal has embedded syncs, the bit[25] specify if the G signal has embedded syncs and the bit[24] specify if the B signal has embedded syncs. When comp_yuv is

Offset: 0x013C			Register Name: TVE_13C_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
			equal to b‘1’, these bits are N.A. and should be set to b‘000’. When the value is different from b‘000’, **RGB_SETUP** should be set to b‘1’.
23:17	/	/	/
16	R/W	0x0	RGB_SETUP “Set-up” enable for RGB outputs. This bit specifies if the “set-up” implied value (black_level – blank_level) specified for the CVBS signal is used also for the RGB signals. 0: The “set-up” is not used, or i.e. comp_yuv is equal to b‘1’. 1: The implied “set-up” is used for the RGB signals
15:1	/	/	/
0	R/W	0x1	BYPASS_YCLAMP Y input clamping selection This bit selects whether the Video Encoder Y input is clamped to 64 to 940 or not. When not clamped the expected range is 0 to 1023. The U and V inputs are always clamped to the range 64 to 960. 0: The Video Encoder Y input is clamped 1: The Video Encoder Y input is not clamped

7.2.6.39. 0x0380 TV Encoder Low Pass Control Register(Default Value: 0x0000_0000)

Offset: 0x0380			Register Name: TVE_380_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:14	/	/	/
13:10	R/W	0x0	USER_DEFLICKER_COEF up : coef/32 Center :1-coef/16 Down :coef/32
9	R/W	0x0	FIX_COEF_DEFLICKER 0: Auto deflicker 1: User deflicker
8	R/W	0x0	ENABLE_DEFLICKER 0: Disable deflicker 1: Enable deflicker
7:1	/	/	/
0	R/W	0x0	EN LP function enable 0: Disable 1: Enable

7.2.6.40. 0x0384 TV Encoder Low Pass Filter Control Register(Default Value: 0x0000_0000)

Offset: 0x0384			Register Name: TVE_384_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21:16	R/W	0x0	HP_RATIO Default high-pass filter ratio In two complement,the range is from -31 to 31.
15:14	/	/	/
13:8	R/W	0x0	BP0_RATIO Default band-pass filter0 ratio In two complement,the range is from -31 to 31.
7:6	/	/	/
5:0	R/W	0x0	BP1_RATIO Default band-pass filter1 ratio In two complement,the range is from -31 to 31.

7.2.6.41. 0x0388 TV Encoder Low Pass Gain Register(Default Value: 0x0000_0000)

Offset: 0x0388			Register Name: TVE_388_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	GAIN Peaking gain setting.

7.2.6.42. 0x038C TV Encoder Low Pass Gain Control Register(Default Value: 0x0000_0000)

Offset: 0x038C			Register Name: TVE_38C_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:16	R/W	0x0	DIF_UP Gain control: limitation threshold.
15:8	/	/	/
4:0	R/W	0x0	BETA Gain control: large gain limitation.

7.2.6.43. 0x0390 TV Encoder Low Pass Shoot Control Register(Default Value: 0x0000_0000)

Offset: 0x0390			Register Name: TVE_390_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:6	/	/	/

Offset: 0x0390			Register Name: TVE_390_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
5:0	R/W	0x0	NEG_GAIN Undershoot gain control.

7.2.6.44. 0x0394 TV Encoder Low Pass Coring Register(Default Value: 0x0000_0000)

Offset: 0x0394			Register Name: TVE_394_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	CORTHR Coring threshold.

7.2.6.45. 0x03A0 TV Encoder Noise Reduction Register(Default Value: 0x0000_0000)

Offset: 0x03A0			Register Name: TVE_3A0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:16	R/W	0x0	T_Value
15:1	/	/	/
0	R/W	0x0	EN

7.3. HDMI

7.3.1. Overview

- Compatible with HDCP 2.2 and HDCP 1.4
- Supports DDC and SCDC
- Integrated CEC hardware engine
- Video support:
  - 2D Video: 4K/1080P/1080I/720P/576P/480P/576I/480I, up to 4K@60fps
  - 3D Video: 4K/1080P/720P/576P/480P, up to 4K@30fps
  - Supports RGB/YUV444/YUV422/YUV420 output
  - Color depth: 8/10-bit
  - HDR10: compliant with CTA-861.3 and SMPTE ST 2048
- Audio support:
  - Uncompressed audio formats: IEC60985 L-PCM audio samples, up to 192 kHz
  - Compressed audio formats: IEC61937 compressed audio, up to 1536 kHz

Contents

Figures

Tables

Chapter 8 Audio

8.1. Audio HUB

8.1.1. Overview

The Audio HUB(AHUB) defines an audio subsystem to support various types of audio protocols and function modules. To provide a flexible audio streaming environment, it is essential to implement a versatile audio fabric to connect audio modules independently and simultaneously.

The Audio HUB is a crossbar switch matrix connecting various audio modules such as I2S/PCM, Digital Audio MIXER(DAM), etc. Audio HUB is attached to the APB bus and is programmable through the bus.

Features:

- Concurrent switching between audio clients
  - The audio client are I2S/PCM, DAM and APBIF
  - A TX client can talk to multiple RX clients simultaneously
  - A RX client can only talk to one TX clients
- Scalable MxN crossbar switch, where
  - M is the number of TX clients
  - N is the number of RX clients
- Supports three 64x32bit TX streams FIFO and three 128x32bit RX streams FIFO for APB DMA operations
- Supports 2 DAM, and 1 I2S/PCM for HDMI

8.1.2. Block Diagram

Figure 8-1 shows the block diagram of the Audio HUB.

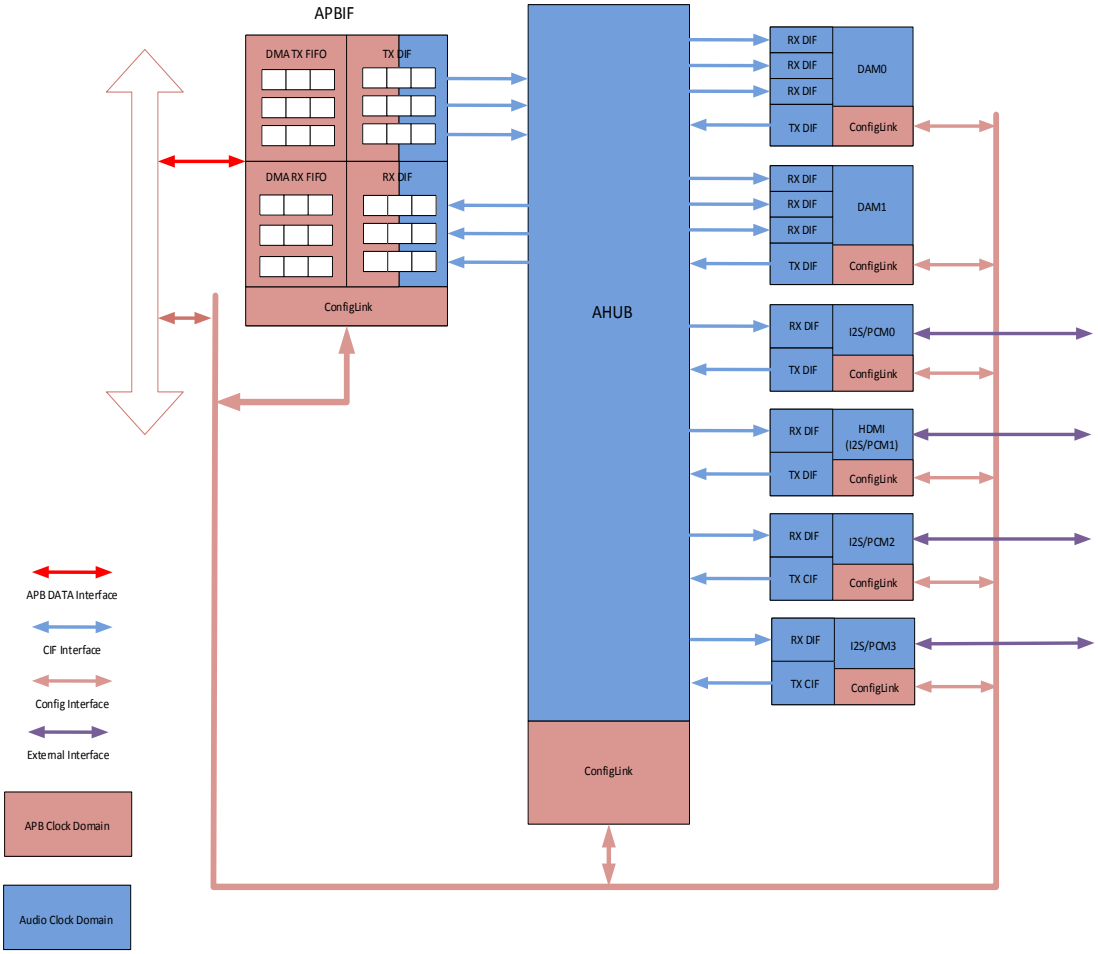

Figure 8- 1. Audio HUB Block Diagram

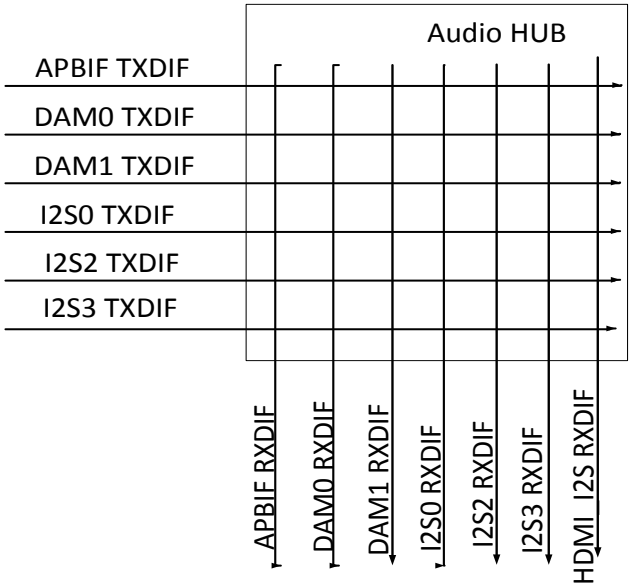

Figure 8- 2. Audio HUB Crossbar Switch and Clients

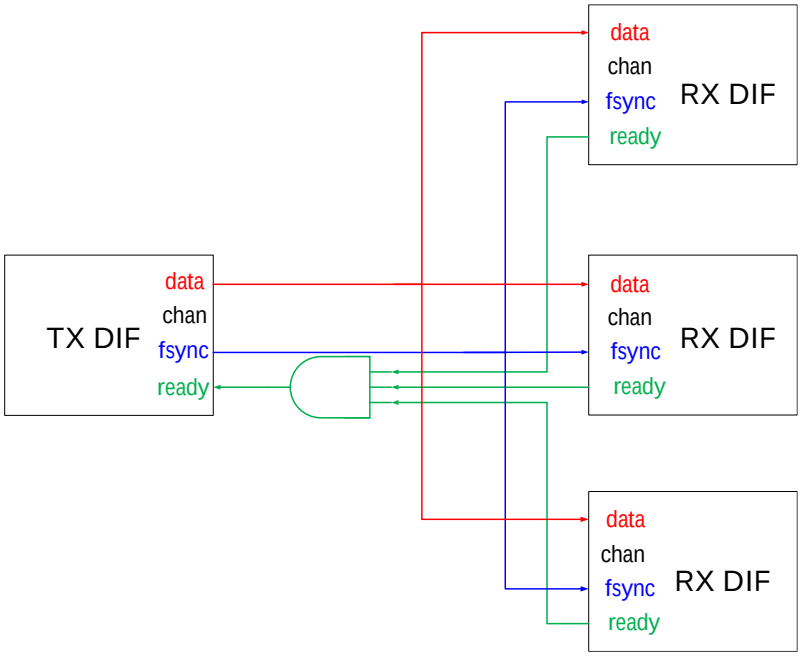

Figure 8- 3. Signal Exchange between TX and RX Clients

An AHUB session has four signals to transmit audio data from a TX client to RX client. The fsync signal is asserted when a new frame starts. A frame consists of samples from multiple channels, the chan signal is asserted when a new channel starts. When a RX client asserts the ready signal, it should be ready to receive data.

8.1.3. Operations and Functional Descriptions

8.1.3.1. External Signals

The following table describes the external signals of Audio HUB. In the Audio HUB controller, the I2S/PCM contacts external signals. BCLK and LRCK are bidirectional I/O. When I2S/PCM is configured as Master device, BCLK and LRCK is output pin; when I2S/PCM is configure as slave device, BCLK and LRCK is input pin. MCLK is output pin for external device. SDO is always the serial data output pin, and SDI is the serial data input. For information about General Purpose I/O port, see Port Controller.

Table 8- 1. Audio HUB External Signals

Signal	Description	Type
H_I2S0_MCLK	Audio HUB I2S/PCM 0 Master Clock	O
H_I2S0_BCLK	Audio HUB I2S/PCM 0 Sample Rate Serial Clock	I/O
H_I2S0_LRCK	Audio HUB I2S/PCM 0 Sample Rate Left and Right Channel Select Clock/Sync	I/O
H_I2S0_DIN0	Audio HUB I2S/PCM 0 Serial Data Input0	I
H_I2S0_DIN1	Audio HUB I2S/PCM 0 Serial Data Input1	I
H_I2S0_DOUT0	Audio HUB I2S/PCM 0 Serial Data Output0	O
H_I2S0_DOUT1	Audio HUB I2S/PCM 0 Serial Data Output1	O
H_I2S2_MCLK	Audio HUB I2S/PCM 2 Master Clock	O
H_I2S2_BCLK	Audio HUB I2S/PCM 2 Sample Rate Serial Clock	I/O
H_I2S2_LRCK	Audio HUB I2S/PCM 2 Sample Rate Left and Right Channel Select Clock/Sync	I/O

H_I2S2_DIN0	Audio HUB I2S/PCM 2 Serial Data Input0	I
H_I2S2_DIN1	Audio HUB I2S/PCM 2 Serial Data Input1	I
H_I2S2_DOUT0	Audio HUB I2S/PCM 2 Serial Data Output0	O
H_I2S2_DOUT1	Audio HUB I2S/PCM 2 Serial Data Output1	O
H_I2S3_MCLK	Audio HUB I2S/PCM 3 Master Clock	O
H_I2S3_BCLK	Audio HUB I2S/PCM 3 Sample Rate Serial Clock	I/O
H_I2S3_LRCK	Audio HUB I2S/PCM 3 Sample Rate Left and Right Channel Select Clock/Sync	I/O
H_I2S3_DIN0	Audio HUB I2S/PCM 3 Serial Data Input0	I
H_I2S3_DIN1	Audio HUB I2S/PCM 3 Serial Data Input1	I
H_I2S3_DOUT0	Audio HUB I2S/PCM 3 Serial Data Output0	O
H_I2S3_DOUT1	Audio HUB I2S/PCM 3 Serial Data Output1	O

8.1.3.2. Clock Sources

Audio HUB System controller uses the APB CLK and AUDIO_PLL. The APB CLK is the system clock and the Audio PLL is the protocol clock. Table 8-2 describes the clock sources for Audio HUB system. Users can see Clock Controller Unit(CCU) for clock setting, configuration and gating information.

Table 8- 2. Audio HUB Clock Sources

Clock Sources	Description
APB CLK	from the System CLK
AUDIO_PLL	24.576 MHz or 22.5792 MHz generated by AUDIO_PLL to produce 48 kHz or 44.1 kHz serial frequency.

8.1.3.3. I2S/PCM Transmit Format

The Audio HUB consists of three I2S/PCM, one I2S/PCM for HDMI, and two DAM(Digital Audio MIXER). The I2S/PCM supports standard I2S mode, Left-justified I2S mode, Right-justified I2S mode, PCM mode and TDM mode. Software can select one of them in which the I2S/PCM works by setting the I2S/PCM Control Register. From Figure 8-4 to Figure 8-8 describe the waveforms for LRCK, BCLK and DOUT, DIN.

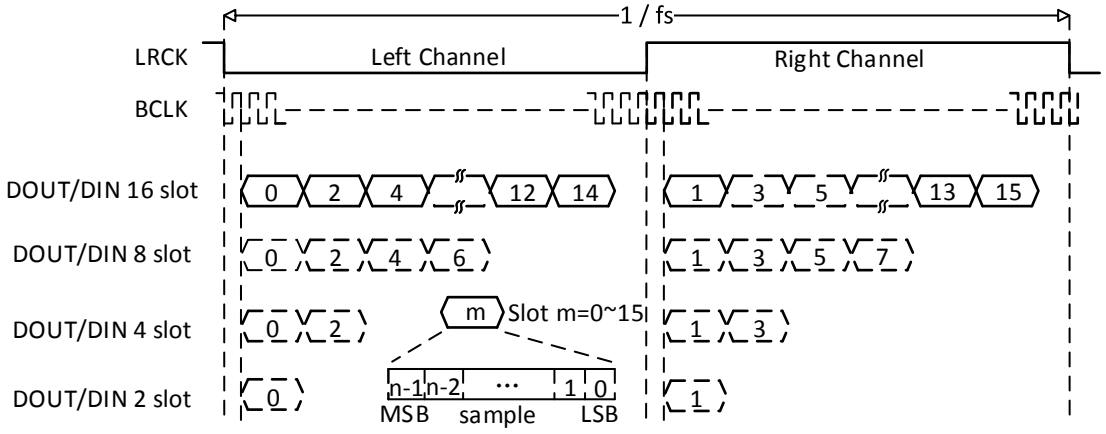

Figure 8- 4. Timing Diagram for Standard I2S/TDM-I2S Mode

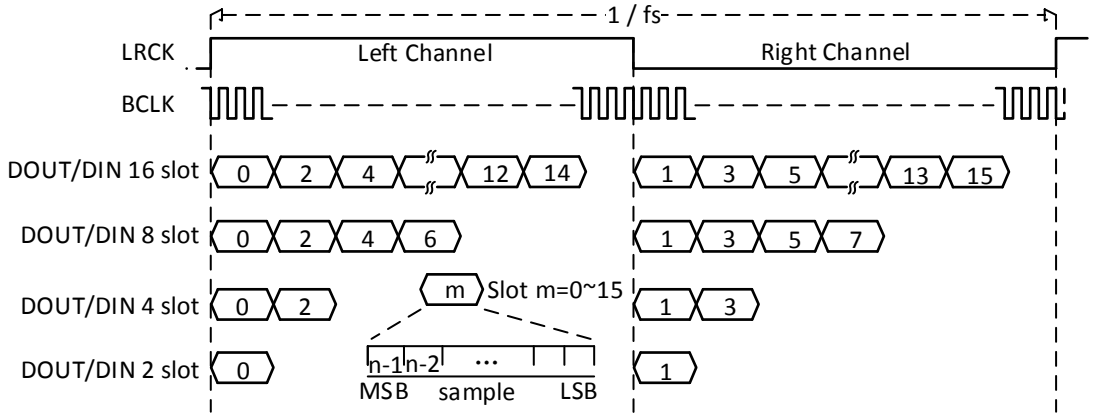

Figure 8- 5. Timing Diagram for Left-justified/TDM-Left Mode

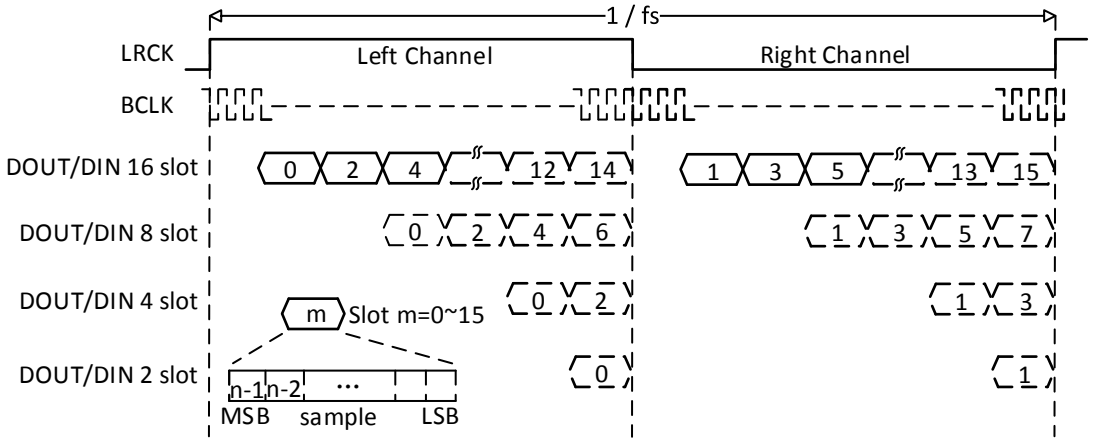

Figure 8- 6. Timing Diagram for Right-justified/TDM-Right Mode

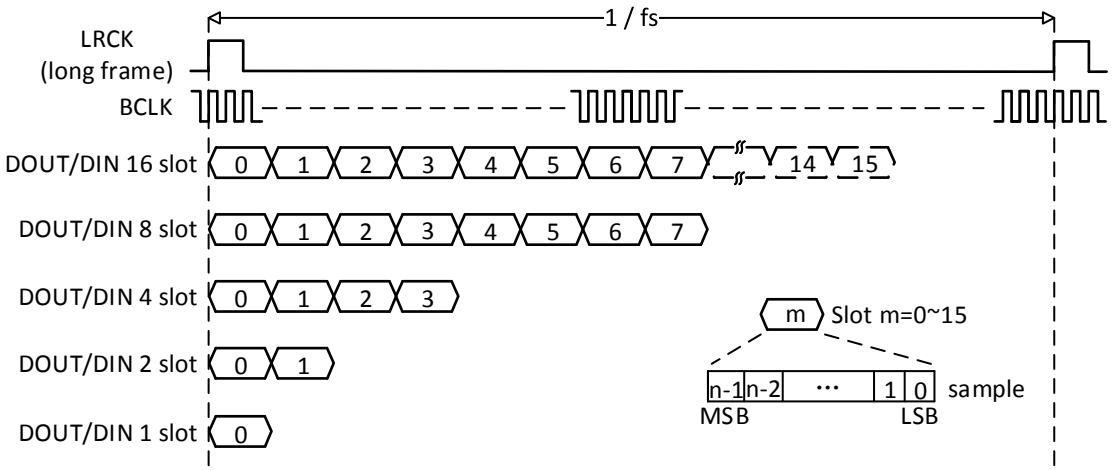

Figure 8- 7. Timing Diagram for PCM Mode (long frame)

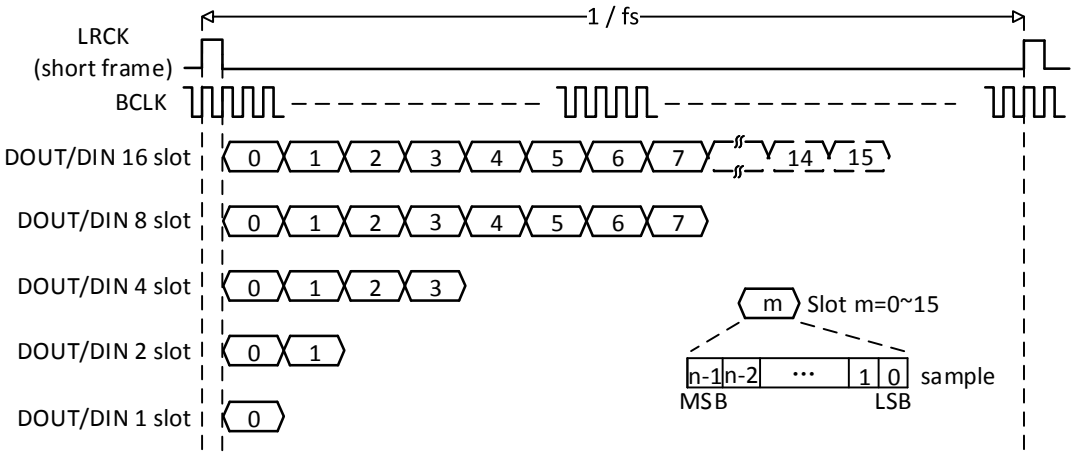

Figure 8- 8. Timing Diagram for PCM Mode (short frame)

8.1.4. Operation Modes

The software operation of the AHUB has eight steps: system setup, TXDIF Initial and Enable, RXDIF Initial and Enable, I2SnInitial and Enable, DAM Initial and Enable, DMA setup, AHUB disable and Check Record_buffer. Eight steps are described in detail in the following sections.

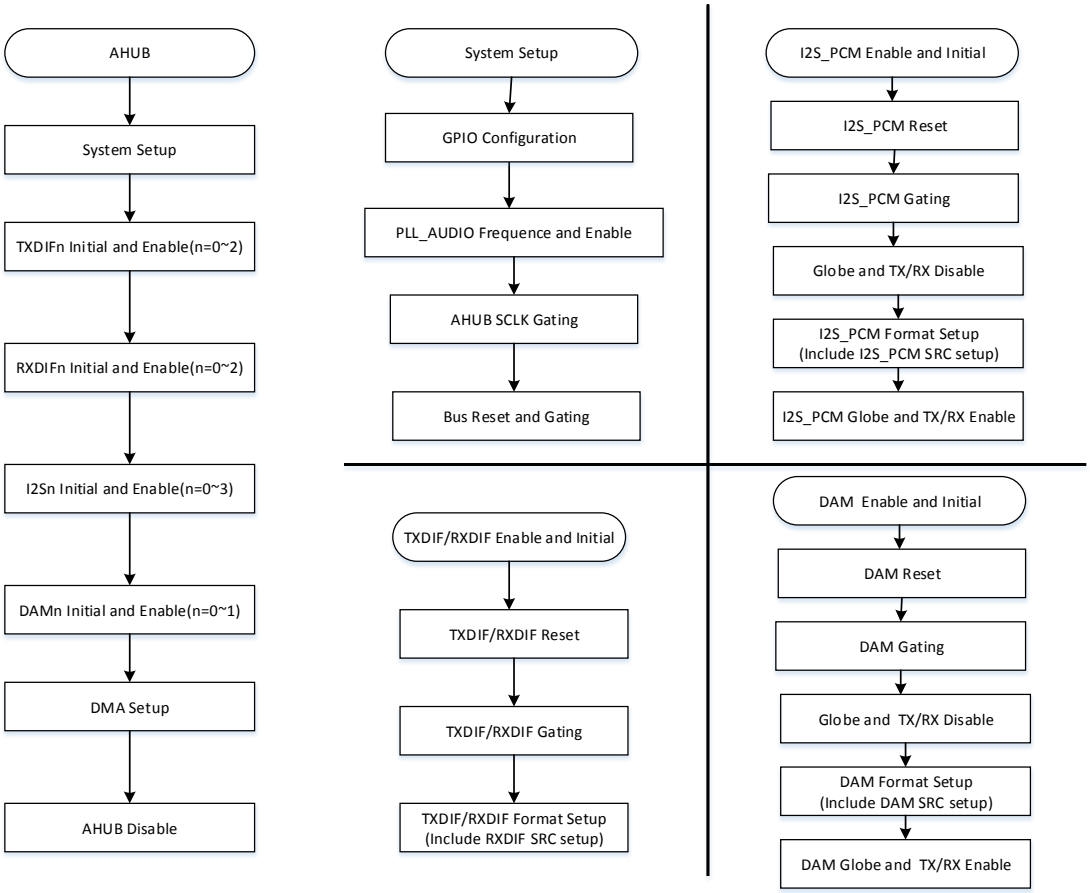

Figure 8- 9. AUDIO HUB Operation Flow

8.1.4.1. System Setup

In the system setup, the first step is properly programming the GPIO.

Follow the clock source for Audio HUB. Choose 24.576 MHz or 22.5792 MHz. At first, set up the frequency of **PLL_AUDIO** in the **PLL_AUDIO_CTRL_REG**, and disable the **PLL_ENABLE** bit of **PLL_AUDIO_CTRL_REG**. Then, enable the **PLL_ENABLE** bit of **PLL_AUDIO_CTRL_REG**, clear the **AUDIO_HUB Clock Register** and the **AUDI0_HUB Bus Gating Reset Register**. Then open the **Audio HUB SCLK_GATING** by writing 1 to **AUDIO_HUB Clock Register**[31], and open the Bus reset and gating by writing 1 to **AUDI0_HUB Bus Gating Reset Register**[0]/[16].

8.1.4.2. TXDIF/RXDIF Initialization

Firstly, Reset and open the gating clock of the TXDIFn(n=0~2) by writing 1 to the **AHUB Reset**[31:29] and **AHUB Gating**[31:29]. When the TXDIF is used, the corresponding bit will be set. Secondly, set up the format of the TXDIF, including TX_width, chan_num, txim and txtl. You can setup the format by writing value to **TXn_Control** and **TXn FIFO Control**.

RXDIF initialization is similar to TXDIF. Firstly, reset and open the gating clock of the RXDIFn(n=0~2) by writing 1 to the **AHUB Reset**[27:25] and **AHUB Gating**[27:25]. When the TXDIF is used, the corresponding bit will be set. Secondly, set up the format of the RXDIF, including RX_width, chan_num, rxom , rxtl and rx_src. You can setup the format by writing value to **RXn_Control** and **RXn FIFO Control**. And setup the rx_src by writing value to **RXn Contact Select Register**. When the TXDIF contact to this RXDIF, the corresponding bit will be set.

8.1.4.3. I2S Initialization and DAM Initialization

Firstly, reset and open the gating clock of the I2Sn(n=0~3) by writing 1 to the **AHUB Reset**[23:20] and **AHUB Gating**[23:20]. When the I2S is used, the corresponding bit will be set. Secondly, you should close the **globe enable bit**(I2Sn_CTRL[0]), disable **TX and RX bit**(I2Sn_CTRL[2:1]). Thirdly, you can setup the I2S/PCM of mater and slave. And choose the contact object to setup **I2Sn_RXDIF_CONT**. The configuration can be referred to the protocol of I2S/PCM. Thirdly, you can set up the translation mode, the sample resolution, the wide of slot, the channel slot number and so on. And then, setup the **globe enable**, **TX enable** and **RX enable**.

DAM initialization is similar to I2S. Firstly, reset and open the gating clock of the DAMn(n=0~1) by writing 1 to the **AHUB Reset**[15:14] and **AHUB Gating**[15:14]. When the DAM is used, the corresponding bit will be set. Secondly, you can setup the DAM of **RXn_chan_num**(n=0~2) and **TX_chan_num**. And choose the contact object to setup **DAM_RXDIFn_SRC**(n=0~2). Thirdly, you can set up the RXn(n=0~2) and the TX channel in the DAM, and the channel volume. Please refer to the specification for more details.

8.1.4.4. DMA Setup

The Audio HUB supports two methods to transfer the data. The most common way is DMA, the setup of DMA can be found in the DMA specification. In this module, you just enable the DRQ and open the streaming start.

8.1.4.5. AHUB Disable

At last, you must disable the Audio HUB by writing 0x0 to the **AHUB_RST Register**.

8.1.5. Typical Application

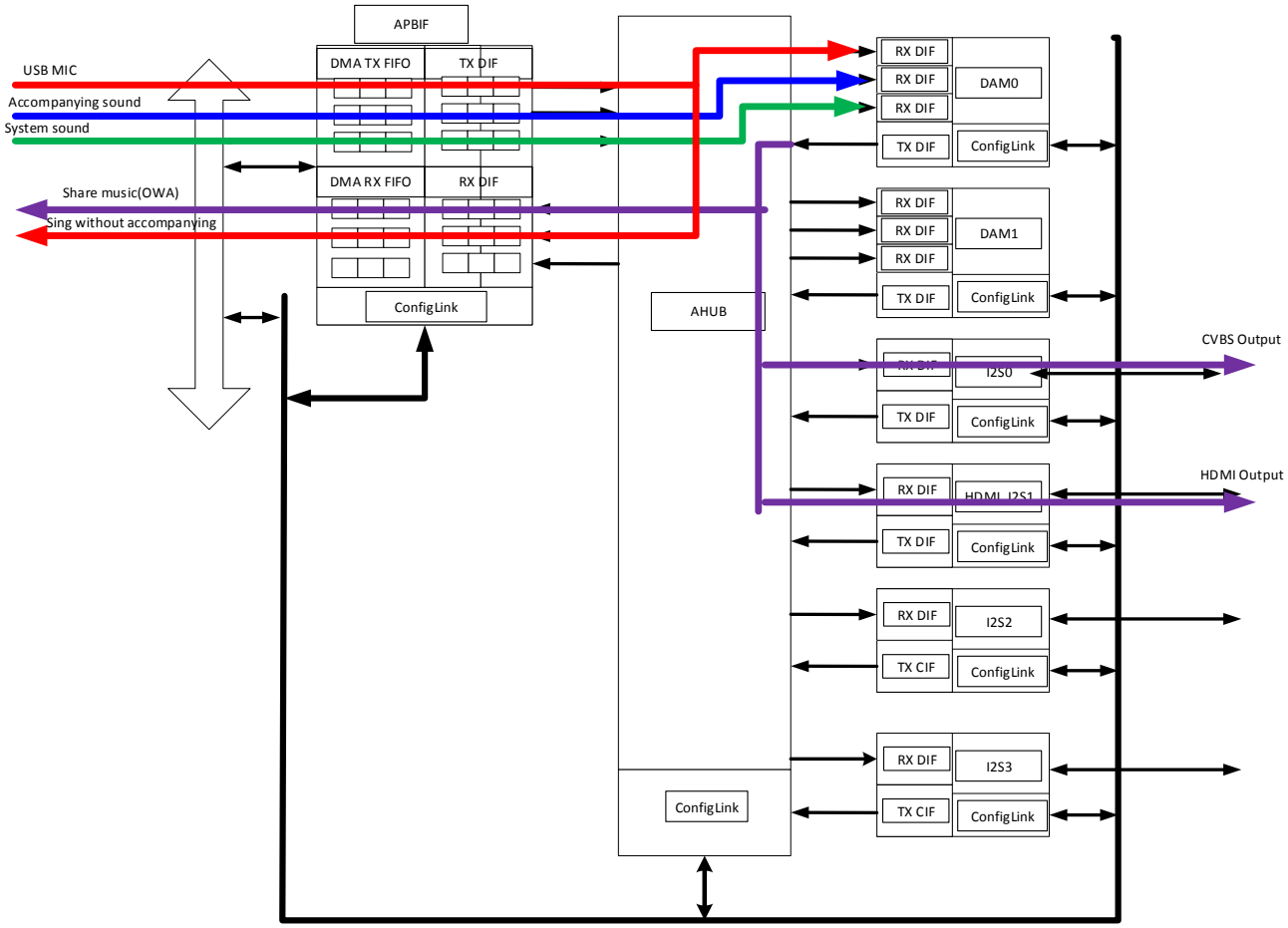

Figure 8- 10.USB MIC Karaoke Date Streaming

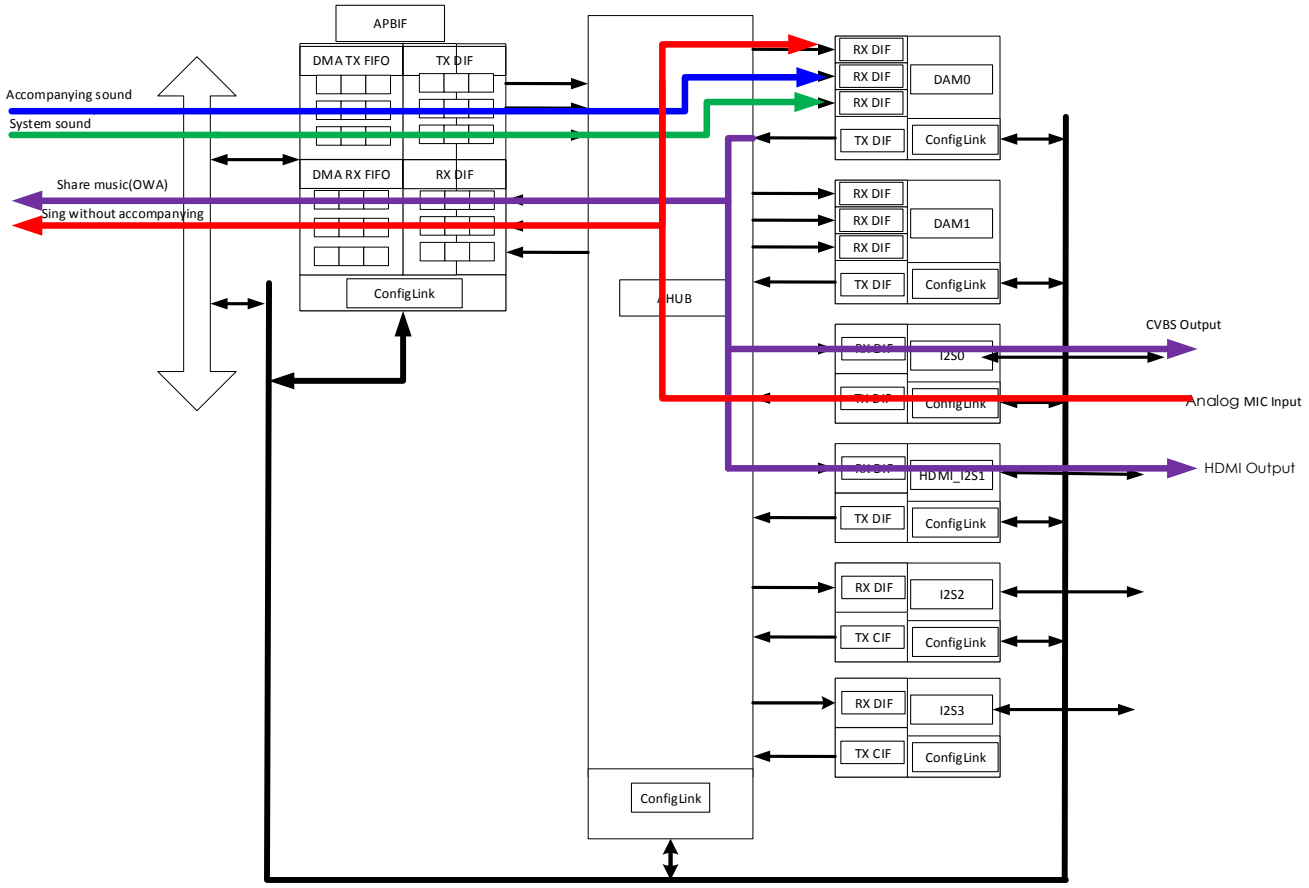

Figure 8- 11. Analog MIC Karaoke Date Streaming

8.1.6. Register List

Module Name	Base Address
AHUB	0x05097000

Register Name	Offset	Description
AHUB_RST	0x0008	AHUB Reset
AHUB_GAT	0x000C	AHUB Gating
APBIF_TXn_CTRL	0x0010+n*0x0030(n=0~2)	APBIF TXn Control
APBIF_TXnIRQ_CTRL	0x0014+n*0x0030(n=0~2)	APBIF TXn DMA & Interrupt Control
APBIF_TXnIRQ_STS	0x0018+n*0x0030(n=0~2)	AHUB APBIF TXn DMA & Interrupt Status
APBIF_TXnFIFO_CTRL	0x0020+n*0x0030(n=0~2)	AHUB APBIF TXn FIFO Control
APBIF_TXnFIFO_STS	0x0024+n*0x0030(n=0~2)	APBIF TXn FIFO Status
APBIF_TXnFIFO	0x0030+n*0x0030(n=0~2)	APBIF TXn FIFO
APBIF_TXnFIFO_CNT	0x0034+n*0x0030(n=0~2)	APBIF TXn FIFO Counter
APBIF_RXn_CTRL	0x0100+n*0x0030(n=0~2)	APBIF RXn Control
APBIF_RXnIRQ_CTRL	0x0104+n*0x0030(n=0~2)	APBIF RXn DMA & Interrupt Control

APBIF_RXnIRQ_STS	0x0108+n*0x0030(n=0~2)	APBIF RXn DMA & Interrupt Status
APBIF_RXnFIFO_CTRL	0x0110+n*0x0030(n=0~2)	APBIF RXn FIFO Control
APBIF_RXnFIFO_STS	0x0114+n*0x0030(n=0~2)	APBIF RX0 FIFO Status
APBIF_RXn_CONT	0x0118+n*0x0030(n=0~2)	APBIF RXn Contact Select
APBIF_RXnFIFO	0x0120+n*0x0030(n=0~2)	APBIF RXn FIFO
APBIF_RXnFIFO_CNT	0x0124+n*0x0030(n=0~2)	APBIF RXn FIFO Counter
I2Sn_CTRL	0x0200+n*0x0100(n=0~3)	I2Sn Control
I2Sn_FMT0	0x0204+n*0x0100(n=0~3)	I2Sn Format 0
I2Sn_FMT1	0x0208+n*0x0100(n=0~3)	I2Sn Format 1
I2Sn_CLKD	0x020C+n*0x0100(n=0~3)	I2Sn Clock Divide
I2Sn_RXDIF_CONT	0x0220+n*0x0100(n=0~3)	I2Sn RXDIF Contact Select
I2Sn_CHCFG	0x0224+n*0x0100(n=0~3)	I2Sn Channel Configuration
I2Sn_IRQ_CTRL	0x0228+n*0x0100(n = 0~3)	I2Sn DMA & Interrupt Control
I2Sn_IRQ_STS	0x022C+n*0x0100(n = 0~3)	I2Sn DMA & Interrupt Status
I2Sn_SDOUTm_SLOTCTR	0x0230+n*0x0100+m*0x0010 (n=0~3)(m=0~3)	I2Sn Output Slot Control
I2Sn_SDOUTmCHMAP0	0x0234+n*0x0100+m*0x0010 (n=0~3)(m=0~3)	I2Sn SDOUTm Channel Mapping 0
I2Sn_SDOUTmCHMAP1	0x0238+n*0x0100+m*0x0010 (n=0~3)(m=0~3)	I2Sn SDOUTm Channel Mapping 1
I2Sn_SDIN_SLOTCTR	0x0270+n*0x0100(n=0~3)	I2Sn Input Slot Control
I2Sn_SDINCHMAP0	0x0274+n*0x0100(n=0~3)	I2Sn SDIN Channel Mapping 0
I2Sn_SDINCHMAP1	0x0278+n*0x0100(n=0~3)	I2Sn SDIN Channel Mapping 1
I2Sn_SDINCHMAP2	0x0274+n*0x0100(n=0~3)	I2Sn SDIN Channel Mapping 2
I2Sn_SDINCHMAP3	0x0278+n*0x0100(n=0~3)	I2Sn SDIN Channel Mapping 3
DAMn_CTRL	0x0A00+n*0x0080(n=0,1)	DAM Control
DAMn_RX0_SRC	0x0A10+n*0x0080(n=0,1)	DAM RXDIF0 Source Select
DAMn_RX1_SRC	0x0A14+n*0x0080(n=0,1)	DAM RXDIF1 Source Select
DAMn_RX2_SRC	0x0A18+n*0x0080(n=0,1)	DAM RXDIF2 Source Select
DAMn_MIX_CTRL0	0x0A30+n*0x0080(n=0,1)	DAM MIX Control 0
DAMn_MIX_CTRL1	0x0A34+n*0x0080(n=0,1)	DAM MIX Control 1
DAMn_MIX_CTRL2	0x0A38+n*0x0080(n=0,1)	DAM MIX Control 2
DAMn_MIX_CTRL3	0x0A3C+n*0x0080(n=0,1)	DAM MIX Control 3
DAMn_MIX_CTRL4	0x0A40+n*0x0080(n=0,1)	DAM MIX Control 4
DAMn_MIX_CTRL5	0x0A44+n*0x0080(n=0,1)	DAM MIX Control 5
DAMn_MIX_CTRL6	0x0A48+n*0x0080(n=0,1)	DAM MIX Control 6
DAMn_MIX_CTRL7	0x0A4C+n*0x0080(n=0,1)	DAM MIX Control 7
DAMn_GAIN_CTRL0	0x0A50+n*0x0080(n=0,1)	DAM GAIN Control 0
DAMn_GAIN_CTRL1	0x0A54+n*0x0080(n=0,1)	DAM GAIN Control 1
DAMn_GAIN_CTRL2	0x0A58+n*0x0080(n=0,1)	DAM GAIN Control 2
DAMn_GAIN_CTRL3	0x0A5C+n*0x0080(n=0,1)	DAM GAIN Control 3
DAMn_GAIN_CTRL4	0x0A60+n*0x0080(n=0,1)	DAM GAIN Control 4
DAMn_GAIN_CTRL5	0x0A64+n*0x0080(n=0,1)	DAM GAIN Control 5
DAMn_GAIN_CTRL6	0x0A68+n*0x0080(n=0,1)	DAM GAIN Control 6
DAMn_GAIN_CTRL7	0x0A6C+n*0x0080(n=0,1)	DAM GAIN Control 7

8.1.7. Register Description

8.1.7.1. 0x0008 AHUB Reset Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: AHUB_ RST
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	APBIF_TXDIF0_RST 0: Assert 1: De-assert
30	R/W	0x0	APBIF_TXDIF1_RST 0: Assert 1: De-assert
29	R/W	0x0	APBIF_TXDIF2_RST 0: Assert 1: De-assert
28	/	/	/
27	R/W	0x0	APBIF_RXDIF0_RST 0: Assert 1: De-assert
26	R/W	0x0	APBIF_RXDIF1_RST 0: Assert 1: De-assert
25	R/W	0x0	APBIF_RXDIF2_RST 0: Assert 1: De-assert
24	/	/	/
23	R/W	0x0	I2S0_RST 0: Assert 1: De-assert
22	R/W	0x0	I2S1_RST 0: Assert 1: De-assert
21	R/W	0x0	I2S2_RST 0: Assert 1: De-assert
20	R/W	0x0	I2S3_RST 0: Assert 1: De-assert
19:16	/	/	/
15	R/W	0x0	DAM0_RST 0: Assert 1: De-assert

14	R/W	0x0	DAM1_RST 0: Assert 1: De-assert
13:0	/	/	/

8.1.7.2. 0x000C AHUB Clock Gating Register (Default Value: 0x0000_0000)

Offset: 0x000C			Register Name: AHUB_GAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	APBIF_TXDIF0_GAT 0: Clock is OFF 1: Clock is ON
30	R/W	0x0	APBIF_TXDIF1_GAT 0: Clock is OFF 1: Clock is ON
29	R/W	0x0	APBIF_TXDIF2_GAT 0: Clock is OFF 1: Clock is ON
28	/	/	/
27	R/W	0x0	APBIF_RXDIF0_GAT 0: Clock is OFF 1: Clock is ON
26	R/W	0x0	APBIF_RXDIF1_GAT 0: Clock is OFF 1: Clock is ON
25	R/W	0x0	APBIF_RXDIF2_GAT 0: Clock is OFF 1: Clock is ON
24	/	/	/
23	R/W	0x0	I2S0_GAT 0: Clock is OFF 1: Clock is ON
22	R/W	0x0	I2S1_GAT 0: Clock is OFF 1: Clock is ON
21	R/W	0x0	I2S2_GAT 0: Clock is OFF 1: Clock is ON
20	R/W	0x0	I2S3_GAT 0: Clock is OFF 1: Clock is ON
19:16	/	/	/
15	R/W	0x0	DAM0_GAT

			0: Clock is OFF 1: Clock is ON
14	R/W	0x0	DAM1_GAT 0: Clock is OFF 1: Clock is ON
13:0	/	/	/

8.1.7.3. 0x0010+n*0x0030 AHUB APBIF TXn Control Register (Default Value: 0x0000_0100)

Offset: 0x0010+n*0x0030(n= 0~2)			Register Name: APBIF_TXn_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:19	/	/	/
18:16	R/W	0x0	TXn_WS TX Width Select 000:Reserved 001:8-bit 010:12-bit 011:16-bit 100:20-bit 101:24-bit 110:28-bit 111:32-bit
15:12	/	/	/
11:8	R/W	0x1	TXn_CHAN_NUM TX Channel Number which between CPU/DMA and FIFO 0000: 1 Channel 0001: 2 Channel ... 1110: 15 Channel 1111: 16 Channel
7:5	/	/	/
4	R/W	0x0	TXn_START APBIF TX Streaming Start
3:0	/	/	/

8.1.7.4. 0x0014+n*0x0030 AHUB APBIF TXn DMA & Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0014+n*0x0030(n = 0~2)			Register Name: APBIF_TXnIRQ_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3	R/W	0x0	TXn_DRQ TX FIFO Empty DRQ Enable 0: Disable

			1: Enable
2	/	/	/
1	R/W	0x0	TXnOI_EN TX FIFO Overrun Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	TXnEI_EN TX FIFO Empty Interrupt Enable 0: Disable 1: Enable

8.1.7.5. 0x0018+n*0x0030 AHUB APBIF TXn DMA & Interrupt Status Register (Default Value: 0x0000_0001)

Offset: 0x0018+n*0x0030(n = 0~2)			Register Name: APBIF_TXnIRQ_STS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1	R/W1C	0x0	TXnO_INT TXFIFO Overrun Pending Interrupt 0: No Pending Interrupt 1: TXFIFO Overrun Pending Interrupt Write '1' to clear this interrupt.
0	R/W1C	0x1	TXnE_INT TXFIFO Empty pending Interrupt 0: No pending IRQ 1: TXFIFO Empty Pending Interrupt When Data in TXFIFO are Less than TX Trigger Level Write '1' to clear this Interrupt or automatic clear if Interrupt condition fails.

8.1.7.6. 0x0020+n*0x0030 AHUB APBIF TXn FIFO Control Register (Default Value: 0x0000_0200)

Offset: 0x0020+n*0x0030(n = 0~2)			Register Name: APBIF_TXnFIFO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
12	R/WAC	0x0	FTXn Write '1' to flush TX FIFO, self clear to '0'.
11:10	/	/	/
9:4	R/W	0x20	TXnTL TX FIFO Empty Trigger Level Interrupt and DMA request trigger level for TXFIFO normal condition Trigger Level = TXnTL
3:1	/	/	/

0	R/W	0x0	TXnIM TX FIFO Input Mode(Mode 0, 1) 0: Valid data at the MSB of TXFIFO register 1: Valid data at the LSB of TXFIFO register Example for 20-bit transmitted audio sample: Mode0: FIFO_I[31:0]={APB_WDATA[31:12], 12'h0} Mode1: FIFO_I[31:0]={APB_WDATA[19:0], 12'h0}

8.1.7.7. 0x0024+n*0x0030 AHUB APBIF TXn FIFO Status Register (Default Value: 0x0000_0140)

Offset: 0x0024+n*0x0030(n = 0~2)			Register Name: APBIF_TXnFIFO_STS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8	R	0x1	TXnE TXFIFO Empty 0: No Room for New Sample in TXFIFO 1: More than One Room for New Sample in TXFIFO (>= 1 Word)
7	/	/	/
6:0	R	0x40	TXnE_CNT TXFIFO Empty Space Word Counter

8.1.7.8. 0x0030+n*0x0030 AHUB APBIF TXn FIFO Register (Default Value: 0x0000_0000)

Offset: 0x0030+n*0x0030(n = 0~2)			Register Name: APBIF_TXnFIFO
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	W	0x0	TXn_DATA TX Sample Transmitting left, right channel sample data should be written this register one by one. The left channel sample data is first and then the right channel sample.

8.1.7.9. 0x0034+n*0x0030 AHUB APBIF TXn FIFO Counter Register (Default Value: 0x0000_0000)

Offset: 0x0034+n*0x0030(n = 0~2)			Register Name: APBIF_TXnFIFO_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	TXn_CNT TX Sample Counter The audio sample number of sending into TXFIFO. When one sample is put into TXFIFO by DMA or by host IO, the TX sample counter register increases by one. The TX sample counter register can be set to any initial value at any time. After been updated by the initial value, the counter register should counter on base of this initial value.

8.1.7.10. 0x0100+n*0x0030 AHUB APBIF RXn Control Register (Default Value: 0x0000_0100)

Offset: 0x0100+n*0x0030(n = 0~2)			Register Name: APBIF_RXn_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:19	/	/	/
18:16	R/W	0x0	RXn_WS RX Width Select 000:Reserved 001:8-bit 010:12-bit 011:16-bit 100:20-bit 101:24-bit 110:28-bit 111:32-bit
15:12	/	/	/
11:8	R/W	0x1	RXn_CHAN_NUM TX Channel Number which between CPU/DMA and FIFO 0: 1 Channel 1: 2 Channel ... 14: 15 Channel 15: 16 Channel
7:5	/	/	/
4	R/W	0x0	RXn_START APBIF RX Streaming Start
3:0	/	/	/

8.1.7.11. 0x0104+n*0x0030 AHUB APBIF RXn DMA & Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0104+n*0x0030(n = 0~2)			Register Name: APBIF_RXnIRQ_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3	R/W	0x0	RXn_DRQ RX FIFO Data Available DRQ Enable 0: Disable 1: Enable When set to '1', RXFIFO DMA Request line is asserted if Data is available in RXFIFO.
2	R/W	0x0	RXnUI_EN RX FIFO Underrun Interrupt Enable 0: Disable

			1: Enable
1	/	/	/
0	R/W	0x0	RXnAI_EN RX FIFO Data Available Interrupt Enable 0: Disable 1: Enable

8.1.7.12. 0x0108+n*0x0030 AHUB APBIF RXn DMA & Interrupt Status Register (Default Value: 0x0000_0001)

Offset: 0x0108+n*0x0030(n = 0~2)			Register Name: APBIF_RXnIRQ_STS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W1C	0x0	RXnU_INT RX FIFO Underrun Pending Interrupt 0: No Pending Interrupt 1: RXFIFO Underrun Pending Interrupt Write '1' to clear this interrupt.
1	/	/	/
0	R/W1C	0x0	RXnA_INT RX FIFO Data Available Pending Interrupt 0: No Pending IRQ 1: TX FIFO Data Available Pending Interrupt When Data in RX FIFO are more than RX Trigger Level Write '1' to clear this Interrupt or Automatic clear if Interrupt condition fails

8.1.7.13. 0x0110+n*0x0030 AHUB APBIF RXn FIFO Control Register (Default Value: 0x0000_0400)

Offset: 0x0110+n*0x0030(n = 0~2)			Register Name: APBIF_RXnFIFO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
12	R/W1C	0x0	FRXn Write '1' to flush RX FIFO, self clear to '0'.
11	/	/	/
10:4	R/W	0x40	RXnTL RX FIFO Empty Trigger Level Interrupt and DMA request trigger level for RXFIFO normal condition Trigger Level = RXnTL
3:2	/	/	/
1:0	R/W	0x0	RXnOM RX FIFO Output Mode(Mode 0, 1,2,3) 00: Expanding '0' at LSB of RX FIFO register

			01: Expanding received sample sign bit at MSB of RX FIFO register. 10: Truncating received samples at high half-word of RX FIFO register and low half-word of RX FIFO register is filled by '0'. 11: Truncating received samples at low half-word of RX FIFO register and high half-word of RX FIFO register is expanded by its sign bit. Example for 20-bit received audio sample: Mode 0: APB_RDATA[31:0] = {FIFO_O[31:12], 12'h0}. Mode 1: APB_RDATA[31:0] = {12{FIFO_O[31]}, FIFO_O[31:12]}. Mode 2: APB_RDATA[31:0] = {FIFO_O[31:16], 16'h0}. Mode 3: APB_RDATA[31:0] = {16{FIFO_O[31]}, FIFO_O[31:16]}.

8.1.7.14. 0x0114+n*0x0030 AHUB APBIF RXn FIFO Status Register (Default Value: 0x0000_0000)

Offset: 0x0114+n*0x0030(n = 0~ 2)			Register Name: APBIF_RXnFIFO_STS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8	R	0x0	RXnA RX FIFO Available 0: No available data in RX FIFO 1: More than One Sample in RX FIFO (>= 1 Word)
7:0	R	0x0	RXnA_CNT RX FIFO Available Sample Word Counter

8.1.7.15. 0x0118+n*0x0030 AHUB APBIF RXn Contact Select Register (Default Value: 0x0000_0000)

Offset: 0x0118+n*0x0030(n = 0~ 2)			Register Name: APBIF_RXn_CONT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	R/W	0x0	RXn_CONTACT_RXDIF Bit[31]:APBIF_TXDIF0 Bit[30]:APBIF_TXDIF1 Bit[29]:APBIF_TXDIF2 Bit[28]:Reserved Bit[27]:I2S0_TXDIF Bit[26]:I2S1_TXDIF(HDMI) Bit[25]:I2S2_TXDIF Bit[24]:Reserved Bit[23]:I2S3_TXDIF Bit[22~20]:Reserved Bit[19]:DAM0_TXDIF Bit[18~16]:Reserved Bit[15]:DAM1_TXDIF Bit[14~0]:Reserved When the TXDIF Contact to this RXDIF, the corresponding bit will be set.

11:0	/	/	/

8.1.7.16. 0x0120+n*0x0030 AHUB APBIF RXn FIFO Register (Default Value: 0x0000_0000)

Offset: 0x0120+n*0x0030(n = 0~2)			Register Name: APBIF_RXnFIFO
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	RXn_DATA RX Sample Host can get one sample by reading this register. The left channel sample data is first and then the right channel sample.

8.1.7.17. 0x0124+n*0x0030 AHUB APBIF RXn FIFO Counter Register (Default Value: 0x0000_0000)

Offset: 0x0124+n*0x0030(n = 0~2)			Register Name: APBIF_RXnFIFO_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	RXn_CNT RX Sample Counter The audio sample number of writing into RXFIFO. When one sample is write into RXFIFO by function module, the RX sample counter register increases by one. The RX sample counter register can be set to any initial value at any time. After been updated by the initial value, the counter register should counter on base of this initial value.

8.1.7.18. 0x0200+n*0x0100 AHUB I2Sn Control Register (Default Value: 0x0004_0000)

Offset: 0x0200+n*0x0100(n = 0~3)			Register Name: I2Sn_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23	R/W	0x0	LOOPBACK3 Loop back test 0: Normal mode 1: Loopback test When set ‘1’, connecting the SDO0 with the SDI3
22	R/W	0x0	LOOPBACK2 Loop back test 0: Normal mode 1: Loopback test When set ‘1’, connecting the SDO0 with the SDI2
21	R/W	0x0	LOOPBACK1 Loop back test 0: Normal mode 1: Loopback test

			When set '1', connecting the SDO0 with the SDI1
20	R/W	0x0	LOOPBACK0 Loop back test 0: Normal mode 1: Loopback test When set '1', connecting the SDO0 with the SDI0
19	/	/	/
18	R/W	0x1	BCLK/LRCK Direction 0:Input 1:Output
17:16	/	/	/
15	R/W	0x0	SDI3_EN 0:Disable, Hi-Z state 1:Enable
14	R/W	0x0	SDI2_EN 0:Disable, Hi-Z state 1:Enable
13	R/W	0x0	SDI1_EN 0:Disable, Hi-Z state 1:Enable
12	R/W	0x0	SDI0_EN 0:Disable, Hi-Z state 1:Enable
11	R/W	0x0	SDO3_EN 0:Disable, Hi-Z state 1:Enable
10	R/W	0x0	SDO2_EN 0:Disable, Hi-Z state 1:Enable
9	R/W	0x0	SDO1_EN 0:Disable, Hi-Z state 1:Enable
8	R/W	0x0	SDO0_EN 0:Disable, Hi-Z state 1:Enable
7	/	/	/
6	R/W	0x0	OUT Mute 0: Normal transfer 1: Force DOUT to output 0
5:4	R/W	0x0	MODE_SEL Mode Selection 00: PCM mode(offset 0: DSP_B; offset 1: DSP_A) 01: Left mode(offset 0: L-J Mode; offset 1: I2S mode) 10: Right-Justified mode 11: Reserved

3	R/W	0x0	LOOPBACK Loop back test 0: Normal mode 1: Loopback test When set '1', connecting the SDO0 with the SDI
2	R/W	0x0	TXEN Transmitter Block Enable 0: Disable 1: Enable
1	R/W	0x0	RXEN Receiver Block Enable 0: Disable 1: Enable
0	R/W	0x0	GEN Globe Enable A disable on this bit overrides any other block or channel enables. 0: Disable 1: Enable

8.1.7.19. 0x0204+n*0x0100 AHUB I2Sn Format Register 0(Default Value: 0x0000_0033)

Offset: 0x0204+n*0x0100(n = 0~3)			**Register Name: I2Sn_FMT0**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R/W	0x0	LRCK_WIDTH (Only apply in PCM mode) LRCK width 0: LRCK = 1 BCLK width (short frame) 1: LRCK = 2 BCLK width (long frame)
29:20	/	/	/
19	R/W	0x0	LRCK_POLARITY When apply in I2S/Left-Justified/Right-Justified mode: 0: Left channel when LRCK is low 1: Left channel when LRCK is high When apply in PCM mode: 0: PCM LRCK asserted at the negative edge 1: PCM LRCK asserted at the positive edge
18	/	/	/
17:8	R/W	0x0	LRCK_PERIOD It is used to program the number of BCLKs per channel of sample frame. This value is interpreted as follow: PCM Mode: Number of BCLKs within (Left + Right) channel width. I2S/Left-Justified/Right-Justified Mode: Number of BCLKs within each individual channel width(Left or Right) For example:

			N = 7 : 8 BCLKs width ... N = 1023 : 1024 BCLKs width
7	R/W	0x0	BCLK_POLARITY 0: Normal mode, negative edge drive and positive edge sample 1: Invert mode, positive edge drive and negative edge sample
6:4	R/W	0x3	SR Sample Resolution 000: Reserved 001: 8-bit 010: 12-bit 011: 16-bit 100: 20-bit 101: 24-bit 110: 28-bit 111: 32-bit
3	R/W	0x0	EDGE_TRANSFER 0: SDO drive data and SDI sample data at the different BCLK edge 1: SDO drive data and SDI sample data at the sample BCLK edge BCLK_PLARITY = 0, use negative edge BCLK_PLARITY = 1, use positive edge
2:0	R/W	0x3	SW Slot Width Select 000: Reserved 001: 8-bit 010: 12-bit 011: 16-bit 100: 20-bit 101: 24-bit 110: 28-bit 111: 32-bit

8.1.7.20. 0x0208+n*0x0100 AHUB I2Sn Format Register 1(Default Value: 0x0000_0030)

Offset: 0x0208+n*0x0100(n = 0~3)			Register Name: I2Sn_FMT1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	RX MLS MSB/LSB First Select 0: MSB First 1: LSB First
6	R/W	0x0	TX MLS MSB/LSB First Select 0: MSB First

			1: LSB First
5:4	R/W	0x3	SEXT Sign Extend in slot [sample resolution < width] 00: Zeros or audio gain padding at LSB position 01: Sign extension at MSB position 10: Reserved 11: Transfer 0 after each sample in each slot
3:2	R/W	0x0	RX_PDM PCM Data Mode 00: Linear PCM 01: reserved 10: 8-bit u-law 11: 8-bit A-law
1:0	R/W	0x0	TX_PDM PCM Data Mode 00: Linear PCM 01: reserved 10: 8-bit u-law 11: 8-bit A-law

8.1.7.21. 0x020C+n*0x0100 AHUB I2Sn Clock Divide Register (Default Value: 0x0000_0000)

Offset: 0x020C+n*0x0100(n = 0~3)			Register Name: I2Sn_CLKD
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8	R/W	0x0	MCLKO_EN 0: Disable MCLK Output 1: Enable MCLK Output
7:4	R/W	0x0	BCLKDIV BCLK Divide Ratio from PLL_Audio 0000: reserved 0001: Divide by 1 0010: Divide by 2 0011: Divide by 4 0100: Divide by 6 0101: Divide by 8 0110: Divide by 12 0111: Divide by 16 1000: Divide by 24 1001: Divide by 32 1010: Divide by 48 1011: Divide by 64 1100: Divide by 96 1101: Divide by 128

			1110: Divide by 176 1111: Divide by 192
3:0	R/W	0x0	MCLKDIV MCLK Divide Ratio from PLL_Audio 0000: reserved 0001: Divide by 1 0010: Divide by 2 0011: Divide by 4 0100: Divide by 6 0101: Divide by 8 0110: Divide by 12 0111: Divide by 16 1000: Divide by 24 1001: Divide by 32 1010: Divide by 48 1011: Divide by 64 1100: Divide by 96 1101: Divide by 128 1110: Divide by 176 1111: Divide by 192

8.1.7.22. 0x0220+n*0x0100 AHUB I2Sn RXDIF Contact Select Register (Default Value: 0x0000_0000)

Offset: 0x0220+n*0x0100(n = 0~3)			Register Name: I2Sn_RXDIF_CONT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	R/W	0x0	RXn_CONTACT_RXDIF Bit[31]:APBIF_TXDIF0 Bit[30]:APBIF_TXDIF1 Bit[29]:APBIF_TXDIF2 Bit[28]:Reserved Bit[27]:I2S0_TXDIF Bit[26]:I2S1_TXDIF(HDMI) Bit[25]:I2S2_TXDIF Bit[24]:Reserved Bit[23]:I2S3_TXDIF Bit[22]:Reserved Bit[21]:Reserved Bit[20]:Reserved Bit[19]:DAM0_TXDIF Bit[18]:Reserved Bit[17]:Reserved Bit[16]:Reserved Bit[15]:DAM1_TXDIF Bit[14]:Reserved

			Bit[13]:Reserved Bit[12]:Reserved When the TXDIF Contact to this RXDIF, the corresponding bit will be set.
11:0	/	/	/

8.1.7.23. 0x0224+n*0x0100 AHUB I2Sn Channel Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0224+n*0x0100(n = 0~3)			Register Name: I2Sn_CHCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9	R/W	0x0	TX_SLOT_HIZ 0:Normal Mode for the Last Half Cycle of BCLK in the Slot 1:Turn to Hi-Z State for the Last Half Cycle of BCLK in the Slot
8	R/W	0x0	TX_STATE 0: Transfer Level 0 When Not Transferring Slot 1:Turn to Hi-Z State(TDM) When Not Transferring Slot
7:4	R/W	0x0	RX_CHAN_NUM RX Channel/Slot Number which between AHUB and I2Sn 0000: 1 channel or slot … 0111: 8 channels or slots 1000: 9 channels or slots … 1111: 16 channels or slots
3:0	R/W	0x0	TX_CHAN_NUM TX Channel/Slot Number which between AHUB and I2Sn 0000: 1 channel or slot … 0111: 8 channels or slots 1000: 9 channels or slots … 1111: 16 channels or slots

8.1.7.24. 0x0228+n*0x0100 AHUB I2Sn DMA & Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0228+n*0x0100(n = 0~3)			Register Name: I2Sn_IRQ_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1	R/W	0x0	RXnOI_EN RX FIFO Overrun Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	TXnUI_EN

			TX FIFO Underrun Interrupt Enable 0: Disable 1: Enable

8.1.7.25. 0x022C+n*0x0100 AHUB I2Sn DMA & Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x022C+n*0x0100(n = 0~3)			Register Name: I2Sn_IRQ_STS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1	R/W1C	0x0	RXnO_INT RX FIFO Overrun Pending Interrupt 0: No Pending Interrupt 1: RXFIFO Overrun Pending Interrupt Write '1' to clear this interrupt
0	R/W1C	0x0	TXnU_INT TX FIFO Underrun Pending Interrupt 0: No Pending Interrupt 1: TXFIFO Underrun Pending Interrupt Write '1' to clear this interrupt

8.1.7.26. 0x0230+n*0x0100+m*0x0010 AHUB I2Sn Output SLOT Control Register (Default Value: 0x0000_0000)

Offset: 0x0230+n*0x0100+m*0x0010 (n=0~3)(m=0~3)			Register Name: I2Sn_SDOUTm_SLOTCTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21:20	R/W	0x0	SDOUTm_OFFSET SDOUT offset tune, SDOUT data offset to LRCK 0: no offset n: data is offset by n BCLKs to LRCK
19:16	R/W	0x0	SDOUTm_SLOT_NUM SDOUT slot number select for each output 0000: 1 Slots ... 0111: 8 Slots 1000: 9 Slots ... 1111: 16 Slots
15:0	R/W	0x0	SDOUTm_SLOT_EN SDOUT slot enable, bit[15:0] refer to slot[15:0]. When one or more slot(s) is(are) disable, the affected slot(s) is(are) set to disable state 0: Disable 1: Enable

8.1.7.27. 0x0234+n*0x0100+m*0x0010 AHUB SDOUTm Channel Mapping Register 0(Default Value: 0x7654_3210)

Offset: 0x0234+n*0x0100+m*0x0010 (n=0~3)(m=0~3)			Register Name: I2Sn_SDOUTmCHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x7	SDOUTm_SLOT7_MAP SDOUT Slot7 Mapping 0000: 1st channel data ... 1111: 16th channel data
27:24	R/W	0x6	SDOUTm_SLOT6_MAP SDOUT Slot6 Mapping 0000: 1st channel data ... 1111: 16th channel data
23:20	R/W	0x5	SDOUTm_SLOT5_MAP SDOUT Slot5 Mapping 0000: 1st channel data ... 1111: 16th channel data
19:16	R/W	0x4	SDOUTm_SLOT4_MAP SDOUT Slot4 Mapping 0000: 1st channel data ... 1111: 16th channel data
15:12	R/W	0x3	SDOUTm_SLOT3_MAP SDOUT Slot3 Mapping 0000: 1st channel data ... 1111: 16th channel data
11:8	R/W	0x2	SDOUTm_SLOT2_MAP SDOUT Slot2 Mapping 0000: 1st channel data ... 1111: 16th channel data
7:4	R/W	0x1	SDOUTm_SLOT1_MAP SDOUT Slot1 Mapping 0000: 1st channel data ... 1111: 16th channel data
3:0	R/W	0x0	SDOUTm_SLOT0_MAP SDOUT Slot0 Mapping 0000: 1st channel data

			... 1111: 16th channel data

8.1.7.28. 0x0238+n*0x0100+m*0x0010 AHUB SDOUTm Channel Mapping Register 1(Default Value: 0xFEDC_BA98)

Offset: 0x0238+n*0x0100+m*0x0010 (n=0~3)(m=0~3)			Register Name: I2Sn_SDOUTmCHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0xF	SDOUTm_SLOT15_MAP SDOUT Slot15 Mapping 0000: 1st channel data ... 1111: 16th channel data
27:24	R/W	0xE	SDOUTm_SLOT14_MAP SDOUT Slot14 Mapping 0000: 1st channel data ... 1111: 16th channel data
23:20	R/W	0xD	SDOUTm_SLOT13_MAP SDOUT Slot13 Mapping 0000: 1st channel data ... 1111: 16th channel data
19:16	R/W	0xC	SDOUTm_SLOT12_MAP SDOUT Slot12 Mapping 0000: 1st channel data ... 1111: 16th channel data
15:12	R/W	0xB	SDOUTm_SLOT11_MAP SDOUT Slot11 Mapping 0000: 1st channel data ... 1111: 16th channel data
11:8	R/W	0xA	SDOUTm_SLOT10_MAP SDOUT Slot10 Mapping 0000: 1st channel data ... 1111: 16th channel data
7:4	R/W	0x9	SDOUTm_SLOT9_MAP SDOUT Slot9 Mapping 0000: 1st channel data ... 1111: 16th channel data
3:0	R/W	0x8	SDOUTm_SLOT8_MAP

			SDOUT Slot8 Mapping 0000: 1st channel data ... 1111: 16th channel data

8.1.7.29. 0x0270+n*0x0100 AHUB I2Sn Input Slot Control Register (Default Value: 0x0000_0000)

Offset: 0x0270+n*0x0100(n=0~3)			Register Name: I2Sn_SDIN_SLOTCTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21:20	R/W	0x0	SDIN_OFFSET SDIN offset tune, SDIN data offset to LRCK 0: no offset n: data is offset by n BCLKs to LRCK
19:16	R/W	0x0	SDIN_SLOT_NUM SDIN Slot number Select for each output 0000: 1 Slots ... 0111: 8 Slots 1000: 9 Slots ... 1111: 16 Slots
15:0	/	/	/

8.1.7.30. 0x0274+n*0x0100 AHUB SDIN Channel Mapping Register 0(Default Value: 0x0302_0100)

Offset: 0x0274+n*0x0100(n=0~3)			Register Name: I2Sn_SDINCHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:28	R/W	0x0	RXFIFO_Sample3_Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
27:24	R/W	0x3	RXFIFO_Sample3_MAP RXFIFO Sample3 Mapping 0: 1st channel data ... 15: 16th channel data
23:22	/	/	/
21:20	R/W	0x0	RXFIFO_Sample2_Select 00: SDI0 01: SDI1

			10: SDI2 11: SDI3
19:16	R/W	0x2	RXFIFO_Sample2_MAP RXFIFO Sample2 Mapping 0: 1st channel data ... 15: 16th channel data
15:14	/	/	/
13:12	R/W	0x0	RXFIFO_Sample1_Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
11:8	R/W	0x1	RXFIFO_Sample1_MAP RXFIFO Sample1 Mapping 0: 1st channel data ... 15: 16th channel data
7:6	/	/	/
5:4	R/W	0x0	RXFIFO_Sample0_Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
3:0	R/W	0x0	RXFIFO_Sample0_MAP RXFIFO Sample0 Mapping 0: 1st channel data ... 15: 16th channel data

8.1.7.31. 0x0278+n*0x0100 AHUB SDIN Channel Mapping Register 1(Default Value: 0x0706_0504)

Offset: 0x0278+n*0x0100(n=0~3)			Register Name: I2Sn_SDINCHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:28	R/W	0x0	RXFIFO_Sample7_Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
27:24	R/W	0x7	RXFIFO_Sample7_MAP RXFIFO Sample7 Mapping 0: 1st channel data ...

			15: 16th channel data
23:22	/	/	/
21:20	R/W	0x0	RXFIFO_Sample6_Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
19:16	R/W	0x6	RXFIFO_Sample6_MAP RXFIFO Sample6 Mapping 0: 1st channel data ... 15: 16th channel data
15:14	/	/	/
13:12	R/W	0x0	RXFIFO_Sample5_Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
11:8	R/W	0x5	RXFIFO_Sample5_MAP RXFIFO Sample5 Mapping 0: 1st channel data ... 15: 16th channel data
7:6	/	/	/
5:4	R/W	0x0	RXFIFO_Sample4_Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
3:0	R/W	0x4	RXFIFO_Sample4_MAP RXFIFO Sample4 Mapping 0: 1st channel data ... 15: 16th channel data

8.1.7.32. 0x027C+n*0x0100 AHUB SDIN Channel Mapping Register 2(Default Value: 0x0B0A_0908)

Offset: 0x027C+n*0x0100(n=0~3)			Register Name: I2Sn_SDINCHMAP2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:28	R/W	0x0	RXFIFO_Sample11_Select 00: SDI0 01: SDI1 10: SDI2

			11: SDI3
27:24	R/W	0xB	RXFIFO_Sample11_MAP RXFIFO Sample11 Mapping 0: 1st channel data ... 15: 16th channel data
23:22	/	/	/
21:20	R/W	0x0	RXFIFO_Sample10_Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
19:16	R/W	0xA	RXFIFO_Sample10_MAP RXFIFO Sample10 Mapping 0: 1st channel data ... 15: 16th channel data
15:14	/	/	/
13:12	R/W	0x0	RXFIFO_Sample9_Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
11:8	R/W	0x9	RXFIFO_Sample9_MAP RXFIFO Sample9 Mapping 0: 1st channel data ... 15: 16th channel data
7:6	/	/	/
5:4	R/W	0x0	RXFIFO_Sample8_Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
3:0	R/W	0x8	RXFIFO_Sample8_MAP RXFIFO Sample8 Mapping 0: 1st channel data ... 15: 16th channel data

8.1.7.33. 0x0280+n*0x0100 AHUB SDIN Channel Mapping Register 3(Default Value: 0x0F0E_0D0C)

Offset: 0x0280+n*0x0100(n=0~3)			Register Name: I2Sn_SDINCHMAP3
Bit	**Read/Write**	**Default/Hex**	**Description**

31:30	/	/	/
29:28	R/W	0x0	RXFIFO_Sample15_Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
27:24	R/W	0xF	RXFIFO_Sample15_MAP RXFIFO Sample15 Mapping 0: 1st channel data ... 15: 16th channel data
23:22	/	/	/
21:20	R/W	0x0	RXFIFO_Sample14_Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
19:16	R/W	0xE	RXFIFO_Sample14_MAP RXFIFO Sample14 Mapping 0: 1st channel data ... 15: 16th channel data
15:14	/	/	/
13:12	R/W	0x0	RXFIFO_Sample13_Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
11:8	R/W	0xD	RXFIFO_Sample13_MAP RXFIFO Sample13 Mapping 0: 1st channel data ... 15: 16th channel data
7:6	/	/	/
5:4	R/W	0x0	RXFIFO_Sample12_Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
3:0	R/W	0xC	RXFIFO_Sample12_MAP RXFIFO Sample12 Mapping 0: 1st channel data ... 15: 16th channel data

8.1.7.34. 0x0A00 + n*0x0080 AHUB DAM Control Register (Default Value: 0x0000_0000)

Offset: 0x0A00 + n*0x0080(n=0,1)			Register Name: DAM_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0x0	RX2_NUM RX2 Channel Num
23:20	R/W	0x0	RX1_NUM RX1 Channel Num
19:16	R/W	0x0	RX0_NUM RX0 Channel Num
15:12	/	/	/
11:8	R/W	0x0	TX Channel Num
7	/	/	/
6	R/W	0x0	RX2EN Receiver Enable 0: Disable 1: Enable
5	R/W	0x0	RX1EN Receiver Enable 0: Disable 1: Enable
4	R/W	0x0	RX0EN Receiver Enable 0: Disable 1: Enable
3:1	/	/	/
0	R/W	0x0	TXEN Transmitter Enable 0: Disable 1: Enable

8.1.7.35. 0x0A10 + n*0x0080 AHUB DAM RXDIF0 Source Select (Default Value: 0x0000_0000)

Offset: 0x0A10 + n*0x0080(n=0,1)			Register Name: DAM_RX0_SRC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	R/W	0x0	RXn_CONTACT_RXDIF Bit[31]:APBIF_TXDIF0 Bit[30]:APBIF_TXDIF1 Bit[29]:APBIF_TXDIF2 Bit[28]:Reserved

			Bit[27]:I2S0_TXDIF Bit[26]:I2S1_TXDIF(HDMI) Bit[25]:I2S2_TXDIF Bit[24]:Reserved Bit[23]:I2S3_TXDIF Bit[22]:Reserved Bit[21]:Reserved Bit[20]:Reserved Bit[19]:DAM0_TXDIF0 Bit[18]:Reserved Bit[17]:Reserved Bit[16]:Reserved Bit[15]:DAM1_TXDIF0 Bit[14]:Reserved Bit[13]:Reserved Bit[12]:Reserved When the TXDIF Contact to this RXDIF, the corresponding bit will be set.
11:0	/	/	/

8.1.7.36. 0x0A14 + n*0x0080 AHUB DAM RXDIF1 Source Select (Default Value: 0x0000_0000)

Offset: 0x0A14 + n*0x0080(n=0,1)			Register Name: DAM_RX1_SRC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	R/W	0x0	RXn_CONTACT_RXDIF Bit[31]:APBIF_TXDIF0 Bit[30]:APBIF_TXDIF1 Bit[29]:APBIF_TXDIF2 Bit[28]:Reserved Bit[27]:I2S0_TXDIF Bit[26]:I2S1_TXDIF(HDMI) Bit[25]:I2S2_TXDIF Bit[24]:Reserved Bit[23]:I2S3_TXDIF Bit[22]:Reserved Bit[21]:Reserved Bit[20]:Reserved Bit[19]:DAM0_TXDIF0 Bit[18]:Reserved Bit[17]:Reserved Bit[16]:Reserved Bit[15]:DAM1_TXDIF0 Bit[14]:Reserved Bit[13]:Reserved Bit[12]:Reserved

			When the TXDIF Contact to this RXDIF, the corresponding bit will be set.
11:0	/	/	/

8.1.7.37. 0x0A18 + n*0x0080 AHUB DAM RXDIF2 Source Select (Default Value: 0x0000_0000)

Offset: 0x0A18 + n*0x0080(n=0,1)			Register Name: DAM_RX2_SRC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	R/W	0x0	RXn_CONTACT_RXDIF Bit[31]:APBIF_TXDIF0 Bit[30]:APBIF_TXDIF1 Bit[29]:APBIF_TXDIF2 Bit[28]:Reserved Bit[27]:I2S0_TXDIF Bit[26]:I2S1_TXDIF(HDMI) Bit[25]:I2S2_TXDIF Bit[24]:Reserved Bit[23]:I2S3_TXDIF Bit[22]:Reserved Bit[21]:Reserved Bit[20]:Reserved Bit[19]:DAM0_TXDIF0 Bit[18]:Reserved Bit[17]:Reserved Bit[16]:Reserved Bit[15]:DAM1_TXDIF0 Bit[14]:Reserved Bit[13]:Reserved Bit[12]:Reserved When the TXDIF Contact to this RXDIF, the corresponding bit will be set.
11:0	/	/	/

8.1.7.38. 0x0A30 + n*0x0080 AHUB DAM MIX Control 0(Default Value: 0x0111_0000)

Offset: 0x0A30 + n*0x0080(n=0,1)			Register Name: DAM_MIX_CTRL0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0x1	TXCH1_MIX_RXCH2 RX2 Channel NUM to TXDIF Channel 1 N: TXDIF Channel 1 MIX RX2 Channel N
23:20	R/W	0x1	TXCH1_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel 1 N: TXDIF Channel 1 MIX RX1 Channel N
19:16	R/W	0x1	TXCH1_MIX_RXCH0

			RX0 Channel NUM to TXDIF Channel 1 N: TXDIF Channel 1 MIX RX0 Channel N
15:12	/	/	
11:8	R/W	0x0	TXCH0_MIX_RXCH2 RX2 Channel NUM to TXDIF Channel 0 N: TXDIF Channel 0 MIX RX2 Channel N
7:4	R/W	0x0	TXCH0_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel 0 N: TXDIF Channel 0 MIX RX1 Channel N
3:0	R/W	0x0	TXCH0_MIX_RXCH0 RX0 Channel NUM to TXDIF Channel 0 N: TXDIF Channel 0 MIX RX0 Channel N

8.1.7.39. 0x0A34 + n*0x0080 AHUB DAM MIX Control 1(Default Value: 0x0333_0222)

Offset: 0x0A34 + n*0x0080(n=0,1)			Register Name: DAM_MIX_CTRL1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0x3	TXCH3_MIX_RXCH2 RX2 Channel NUM to TXDIF Channel 3 N: TXDIF Channel 3 MIX RX2 Channel N
23:20	R/W	0x3	TXCH3_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel 3 N: TXDIF Channel 3 MIX RX1 Channel N
19:16	R/W	0x3	TXCH3_MIX_RXCH0 RX0 Channel NUM to TXDIF Channel 3 N: TXDIF Channel 3 MIX RX0 Channel N
15:12	/	/	/
11:8	R/W	0x2	TXCH2_MIX_RXCH2 RX2 Channel NUM to TXDIF Channel 2 N: TXDIF Channel 2 MIX RX2 Channel N
7:4	R/W	0x2	TXCH2_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel 2 N: TXDIF Channel 2 MIX RX1 Channel N
3:0	R/W	0x2	TXCH2_MIX_RXCH0 RX0 Channel NUM to TXDIF Channel 2 N: TXDIF Channel 2 MIX RX0 Channel N

8.1.7.40. 0x0A38 + n*0x0080 AHUB DAM MIX Control 2(Default Value: 0x0555_0444)

Offset: 0x0A38 + n*0x0080(n=0,1)			Register Name: DAM_MIX_CTRL2
Bit	**Read/Write**	**Default/Hex**	**Description**

31:28	/	/	/
27:24	R/W	0x5	TXCH5_MIX_RXCH2 RX2 Channel NUM to TXDIF Channel 5 N: TXDIF Channel 5 MIX RX2 Channel N
23:20	R/W	0x5	TXCH5_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel 5 N: TXDIF Channel 5 MIX RX1 Channel N
19:16	R/W	0x5	TXCH5_MIX_RXCH0 RX0 Channel NUM to TXDIF Channel 5 N: TXDIF Channel 5 MIX RX0 Channel N
15:12	/	/	/
11:8	R/W	0x4	TXCH4_MIX_RXCH2 RX2 Channel NUM to TXDIF Channel 4 N: TXDIF Channel 4 MIX RX2 Channel N
7:4	R/W	0x4	TXCH4_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel 4 N: TXDIF Channel 2 MIX RX1 Channel N
3:0	R/W	0x4	TXCH4_MIX_RXCH0 RX0 Channel NUM to TXDIF Channel 4 N: TXDIF Channel 4 MIX RX0 Channel N

8.1.7.41. 0x0A3C + n*0x0080 AHUB DAM MIX Control 3(Default Value: 0x0777_0666)

Offset: 0x0A3C + n*0x0080(n=0,1)			**Register Name: DAM_MIX_CTRL3**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0x7	TXCH7_MIX_RXCH2 RX2 Channel NUM to TXDIF Channel 7 N: TXDIF Channel 7 MIX RX2 Channel N
23:20	R/W	0x7	TXCH7_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel 7 N: TXDIF Channel 7 MIX RX1 Channel N
19:16	R/W	0x7	TXCH7_MIX_RXCH0 RX0 Channel NUM to TXDIF Channel 7 N: TXDIF Channel 7 MIX RX0 Channel N
15:12	/	/	/
11:8	R/W	0x6	TXCH6_MIX_RXCH2 RX2 Channel NUM to TXDIF Channel 6 N: TXDIF Channel 6 MIX RX2 Channel N
7:4	R/W	0x6	TXCH6_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel 6 N: TXDIF Channel 6 MIX RX1 Channel N
3:0	R/W	0x6	TXCH6_MIX_RXCH0

			RX0 Channel NUM to TXDIF Channel 6 N: TXDIF Channel 6 MIX RX0 Channel N

8.1.7.42. 0x0A40 + n*0x0080 AHUB DAM MIX Control 4(Default Value: 0x0999_0888)

Offset: 0x0A40 + n*0x0080(n=0,1)			Register Name: DAM_MIX_CTRL4
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0x9	TXCH9_MIX_RXCH2 RX2 Channel NUM to TXDIF Channel 9 N: TXDIF Channel 9 MIX RX2 Channel N
23:20	R/W	0x9	TXCH9_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel 9 N: TXDIF Channel 9 MIX RX1 Channel N
19:16	R/W	0x9	TXCH9_MIX_RXCH0 RX0 Channel NUM to TXDIF Channel 9 N: TXDIF Channel 9 MIX RX0 Channel N
15:12	/	/	/
11:8	R/W	0x8	TXCH8_MIX_RXCH2 RX2 Channel NUM to TXDIF Channel 8 N: TXDIF Channel 8 MIX RX2 Channel N
7:4	R/W	0x8	TXCH8_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel 8 N: TXDIF Channel 8 MIX RX1 Channel N
3:0	R/W	0x8	TXCH8_MIX_RXCH0 RX0 Channel NUM to TXDIF Channel 8 N: TXDIF Channel 8 MIX RX0 Channel N

8.1.7.43. 0x0A44 + n*0x0080 AHUB DAM MIX Control 5(Default Value: 0x0BBB_0AAA)

Offset: 0x0A44 + n*0x0080(n=0,1)			Register Name: DAM_MIX_CTRL5
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0xB	TXCHB_MIX_RXCH2 RX2 Channel NUM to TXDIF Channel B N: TXDIF Channel B MIX RX2 Channel N
23:20	R/W	0xB	TXCHB_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel B N: TXDIF Channel B MIX RX1 Channel N
19:16	R/W	0xB	TXCHB_MIX_RXCH0 RX0 Channel NUM to TXDIF Channel B N: TXDIF Channel B MIX RX0 Channel N

15:12	/	/	/
11:8	R/W	0xA	TXCHA_MIX_RXCH2 RX2 Channel NUM to TXDIF Channel A N: TXDIF Channel A MIX RX2 Channel N
7:4	R/W	0xA	TXCHA_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel A N: TXDIF Channel A MIX RX1 Channel N
3:0	R/W	0xA	TXCHA_MIX_RXCH0 RX0 Channel NUM to TXDIF Channel A N: TXDIF Channel A MIX RX0 Channel N

8.1.7.44. 0x0A48 + n*0x0080 AHUB DAM MIX Control 6(Default Value: 0x0DDD_0CCC)

Offset: 0x0A48 + n*0x0080(n=0,1)			Register Name: DAM_MIX_CTRL6
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0xD	TXCHD_MIX_RXCH2 RX2 Channel NUM to TXDIF Channel D N: TXDIF Channel D MIX RX2 Channel N
23:20	R/W	0xD	TXCHD_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel D N: TXDIF Channel D MIX RX1 Channel N
19:16	R/W	0xD	TXCHD_MIX_RXCH0 RX0 Channel NUM to TXDIF Channel D N: TXDIF Channel D MIX RX0 Channel N
15:12	/	/	/
11:8	R/W	0xC	TXCHC_MIX_RXCH2 RX2 Channel NUM to TXDIF Channel C N: TXDIF Channel C MIX RX2 Channel N
7:4	R/W	0xC	TXCHC_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel C N: TXDIF Channel C MIX RX1 Channel N
3:0	R/W	0xC	TXCHC_MIX_RXCH0 RX0 Channel NUM to TXDIF Channel C N: TXDIF Channel C MIX RX0 Channel N

8.1.7.45. 0x0A4C + n*0x0080 AHUB DAM MIX Control 7(Default Value: 0x0FFF_0EEE)

Offset: 0x0A4C + n*0x0080(n=0,1)			Register Name: DAM_MIX_CTRL7
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0xF	TXCHF_MIX_RXCH2

			RX2 Channel NUM to TXDIF Channel F N: TXDIF Channel F MIX RX2 Channel N
23:20	R/W	0xF	TXCHF_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel F N: TXDIF Channel F MIX RX1 Channel N
19:16	R/W	0xF	TXCHF_MIX_RXCH0 RX0 Channel NUM to TXDIF Channel F N: TXDIF Channel F MIX RX0 Channel N
15:12	/	/	/
11:8	R/W	0xE	TXCHE_MIX_RXCH2 RX2 Channel NUM to TXDIF Channel E N: TXDIF Channel E MIX RX2 Channel N
7:4	R/W	0xE	TXCHE_MIX_RXCH1 RX1 Channel NUM to TXDIF Channel E N: TXDIF Channel E MIX RX1 Channel N
3:0	R/W	0xE	TXCHE_MIX_RXCH0 RX0 Channel NUM to TXDIF Channel E N: TXDIF Channel E MIX RX0 Channel N

8.1.7.46. 0x0A50 + n*0x0080 AHUB DAM Volume Control 0(Default Value: 0x0111_0111)

Offset: 0x0A50 + n*0x0080(n=0,1)			Register Name: DAM_VOL_CTRL0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0x1	TXCH1_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel 1 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
23:20	R/W	0x1	TXCH1_GAIN_RXCH1 RX1 Channel NUM to TXDIF Channel 1 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
19:16	R/W	0x1	TXCH1_GAIN_RXCH0 RX0 Channel NUM to TXDIF Channel 1 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB

			Others: Reserved
15:12	/	/	/
11:8	R/W	0x1	TXCH0_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel 0 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
7:4	R/W	0x1	TXCH0_GAIN_RXCH1 RX1 Channel NUM to TXDIF Channel 0 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
3:0	R/W	0x1	TXCH0_GAIN_RXCH0 RX0 Channel NUM to TXDIF Channel 0 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved

8.1.7.47. 0x0A54 + n*0x0080 AHUB DAM Volume Control 1(Default Value: 0x0111_0111)

Offset: 0x0A54 + n*0x0080(n=0,1)			Register Name: DAM_VOL_CTRL1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0x1	TXCH3_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel 3 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
23:20	R/W	0x1	TXCH3_GAIN_RXCH1 RX1 Channel NUM to TXDIF Channel 3 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
19:16	R/W	0x1	TXCH3_GAIN_RXCH0

			RX0 Channel NUM to TXDIF Channel 3 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100: -12dB Others: Reserved
15:12	/	/	/
11:8	R/W	0x1	TXCH2_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel 2 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
7:4	R/W	0x1	TXCH2_GAIN_RXCH1 RX1 Channel NUM to TXDIF Channel 2 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
3:0	R/W	0x1	TXCH2_GAIN_RXCH0 RX0 Channel NUM to TXDIF Channel 2 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved

8.1.7.48. 0x0A58 + n*0x0080 AHUB DAM Volume Control 2(Default Value: 0x0111_0111)

Offset: 0x0A58 + n*0x0080(n=0,1)			**Register Name: DAM_VOL_CTRL2**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0x1	TXCH5_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel 5 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
23:20	R/W	0x1	TXCH5_GAIN_RXCH1 RX1 Channel NUM to TXDIF Channel 5 Gain 0000: Mute

			0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
19:16	R/W	0x1	TXCH5_GAIN_RXCH0 RX0 Channel NUM to TXDIF Channel 5 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
15:12	/	/	/
11:8	R/W	0x1	TXCH4_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel 4 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
7:4	R/W	0x1	TXCH4_GAIN_RXCH1 RX1 Channel NUM to TXDIF Channel 4 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
3:0	R/W	0x1	TXCH4_GAIN_RXCH0 RX0 Channel NUM to TXDIF Channel 4 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved

8.1.7.49. 0x0A5C + n*0x0080 AHUB DAM Volume Control 3(Default Value: 0x0111_0111)

Offset: 0x0A5C + n*0x0080(n=0,1)			Register Name: DAM_VOL_CTRL3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0x1	TXCH7_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel 7 Gain 0000: Mute 0001: 0dB 0010: -6dB

			0100:-12dB Others: Reserved
23:20	R/W	0x1	TXCH7_GAIN_RXCH1 RX1 Channel NUM to TXDIF Channel 7 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
19:16	R/W	0x1	TXCH7_GAIN_RXCH0 RX0 Channel NUM to TXDIF Channel 7 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
15:12	/	/	/
11:8	R/W	0x1	TXCH6_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel 6 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
7:4	R/W	0x1	TXCH6_GAIN_RXCH1 RX1 Channel NUM to TXDIF Channel 6 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
3:0	R/W	0x1	TXCH6_GAIN_RXCH0 RX0 Channel NUM to TXDIF Channel 6 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved

8.1.7.50. 0x0A60 + n*0x0080 AHUB DAM Volume Control 4(Default Value: 0x0111_0111)

Offset: 0x0A60 + n*0x0080(n=0,1)			Register Name: DAM_VOL_CTRL4
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/

27:24	R/W	0x1	TXCH9_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel 9 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
23:20	R/W	0x1	TXCH9_GAIN_RXCH1 RX1 Channel NUM to TXDIF Channel 9 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
19:16	R/W	0x1	TXCH9_GAIN_RXCH0 RX0 Channel NUM to TXDIF Channel 9 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
15:12	/	/	/
11:8	R/W	0x1	TXCH8_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel 8 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
7:4	R/W	0x1	TXCH8_GAIN_RXCH1 RX1 Channel NUM to TXDIF Channel 8 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
3:0	R/W	0x1	TXCH8_GAIN_RXCH0 RX0 Channel NUM to TXDIF Channel 8 Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved

8.1.7.51. 0x0A64 + n*0x0080 AHUB DAM Volume Control 5(Default Value: 0x0111_0111)

Offset: 0x0A64 + n*0x0080(n=0,1)			Register Name: DAM_VOL_CTRL5
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0x1	TXCHB_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel B Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
23:20	R/W	0x1	TXCHB_GAIN_RXCH1 RX1 Channel NUM to TXDIF Channel B Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
19:16	R/W	0x1	TXCHB_GAIN_RXCH0 RX0 Channel NUM to TXDIF Channel B Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
15:12	/	/	/
11:8	R/W	0x1	TXCHA_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel A Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
7:4	R/W	0x1	TXCHA_GAIN_RXCH1 RX1 Channel NUM to TXDIF Channel A Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
3:0	R/W	0x1	TXCHA_GAIN_RXCH0 RX0 Channel NUM to TXDIF Channel A Gain 0000: Mute 0001: 0dB

			0010: -6dB 0100:-12dB Others: Reserved

8.1.7.52. 0x0A68 + n*0x0080 AHUB DAM Volume Control 6(Default Value: 0x0111_0111)

Offset: 0x0A68 + n*0x0080(n=0,1)			Register Name: DAM_VOL_CTRL6
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0x1	TXCHD_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel D Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
23:20	R/W	0x1	TXCHD_GAIN_RXCH1 RX1 Channel NUM to TXDIF Channel D Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
19:16	R/W	0x1	TXCHD_GAIN_RXCH0 RX0 Channel NUM to TXDIF Channel D Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
15:12	/	/	/
11:8	R/W	0x1	TXCHC_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel C Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
7:4	R/W	0x1	TXCHC_GAIN_RXCH1 RX1 Channel NUM to TXDIF Channel C Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB

			Others: Reserved
3:0	R/W	0x1	TXCHC_GAIN_RXCH0 RX0 Channel NUM to TXDIF Channel C Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved

8.1.7.53. 0x0A6C + n*0x0080 AHUB DAM Volume Control 7(Default Value: 0x0111_0111)

Offset: 0x0A6C + n*0x0080(n=0,1)			Register Name: DAM_VOL_CTRL7
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0x1	TXCHF_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel F Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
23:20	R/W	0x1	TXCHF_GAIN_RXCH1 RX1 Channel NUM to TXDIF Channel F Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
19:16	R/W	0x1	TXCHF_GAIN_RXCH0 RX0 Channel NUM to TXDIF Channel F Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
15:12	/	/	/
11:8	R/W	0x1	TXCHE_GAIN_RXCH2 RX2 Channel NUM to TXDIF Channel E Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
7:4	R/W	0x1	TXCHE_GAIN_RXCH1

			RX1 Channel NUM to TXDIF Channel E Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved
3:0	R/W	0x1	TXCHE_GAIN_RXCH0 RX0 Channel NUM to TXDIF Channel E Gain 0000: Mute 0001: 0dB 0010: -6dB 0100:-12dB Others: Reserved

8.2. DMIC

8.2.1. Overview

The DMIC controller supports one 8-channels digital microphone interface, the DMIC controller can output 128fs or 64fs (fs= ADC sample rate).

The DMIC controller includes the following features:

- Supports up to 8 channels
- Sample rate from 8 kHz to 48 kHz

8.2.2. Block Diagram

Figure 8-12 shows a block diagram of the DMIC.

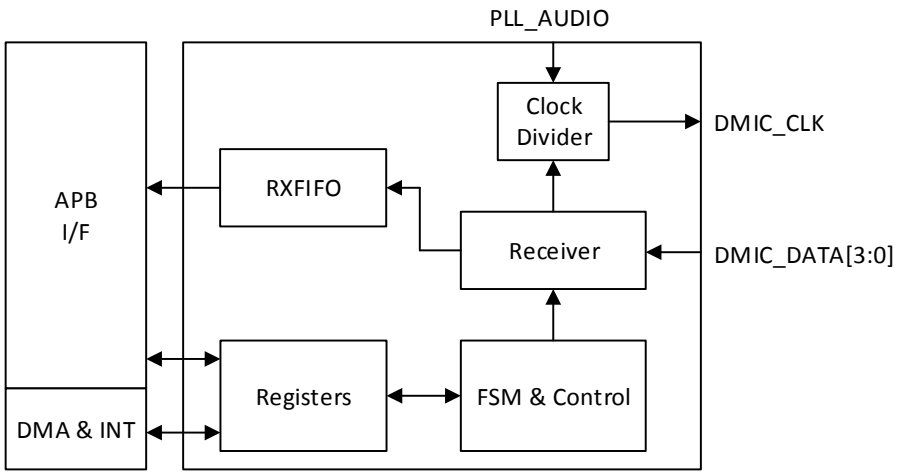

Figure 8- 12. DMIC Block Diagram

8.2.3. Operations and Functional Descriptions

8.2.3.1. External Signals

Table 8-3 describes the external signals of DMIC.

Table 8- 3. DMIC External Signals

Signal	Description	Type
DMIC_CLK	Digital Microphone Clock Output	O
DMIC_DATA0	Digital Microphone Data Input	I

DMIC_DATA1	Digital Microphone Data Input	I
DMIC_DATA2	Digital Microphone Data Input	I
DMIC_DATA3	Digital Microphone Data Input	I

8.2.3.2. Clock Sources

Table 8-4 describes the clock source for DMIC. Users can see **Clock Controller Unit(CCU)** for clock setting, configuration and gating information.

Table 8- 4. DMIC Clock Sources

Clock Sources	Description
PLL_AUDIO	24.576 MHz or 22.5792 MHz generated by PLL_AUDIO to produce 48 kHz or 44.1 kHz serial frequency.

8.2.3.3. Operation Mode

The software operation of the DMIC is divided into five steps: system setup, DMIC initialization, channel setup, DMA setup and Enable/Disable module. Five steps are described in detail in the following sections.

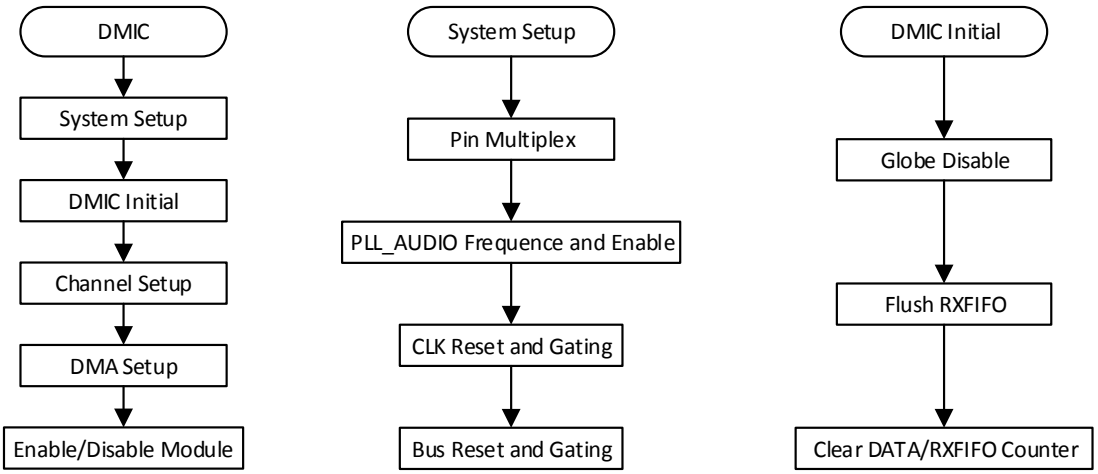

Figure 8- 13. DMIC Operation Mode

8.2.3.3.1. System Setup and DMIC Initialization

The first step in the system setup is properly programming the GPIO. Because the DMIC port is a multiplex pin. You can find the function in the pin multiplex specification.

The clock source for the DMIC should be followed. At first you must disable the PLL_AUDIO through the PLL_ENABLE bit of **PLL_AUDIO_CTRL_REG** in the CCU. The second step, you must set up the frequency of the PLL_AUDIO in the

PLL_AUDIO_CTRL_REG. Then enable PLL_AUDIO. After that, you must open the DMIC gating through the **DMIC_CLK_REG** when you checkout that the LOCK bit of **PLL_AUDIO_CTRL_REG** becomes 1. At last, you must reset and open the DMIC bus gating in the **DMIC_BGR_REG**.

After the system setup, the register of DMIC can be setup. At first, you should initialize the DMIC. You should close the **globe enable bit**(DMIC_EN[8]) , **data channel enable bit**(DMIC_EN[7:0]) by writing 0 to it. After that, you must flush the RXFIFO by writing 1 to **DMIC_RXFIFO_CTR**[31]. At last, you can clear the Data/RXFIFO counter by writing 1 to **DMIC_RXFIFO_STA, DMIC_CNT**.

8.2.3.3.2. Channel Setup and DMA Setup

You can set up the sample rate, the sample resolution, the over sample rate, the channel number, the RXFIFO output mode and the RXFIFO trigger level and so on. The setup of register can be found in the specification.

The DMIC supports two methods to transfer the data. The most common way is DMA, the setup of DMA can be found in the DMA specification. In this module, you just enable the DRQ.

8.2.3.3.3. Enable and Disable DMIC

To enable the function, you can enable **data channel enable bit** (DMIC_EN[7:0]) by writing 1 to it. After that, you must enable DMIC by writing 1 to the **Globe Enable bit** (DMIC_EN[8]). Write 0 to **Globe Enable bit** to disable DMIC.

8.2.4. Register List

Module Name	Base Address
DMIC	0x05095000

Register Name	Offset	Description
DMIC_EN	0x0000	DMIC Enable Control Register
DMIC_SR	0x0004	DMIC Sample Rate Register
DMIC_CTR	0x0008	DMIC Control Register
DMIC_DATA	0x0010	DMIC Data Register
DMIC_INTC	0x0014	MIC Interrupt Control Register
DMIC_INTS	0x0018	DMIC Interrupt Status Register
DMIC_RXFIFO_CTR	0x001C	DMIC RXFIFO Control Register
DMIC_RXFIFO_STA	0x0020	DMIC RXFIFO Status Register
DMIC_CH_NUM	0x0024	DMIC Channel Numbers Register
DMIC_CH_MAP	0x0028	DMIC Channel Mapping Register
DMIC_CNT	0x002C	DMIC Counter Register
DATA0_DATA1_VOL_CTR	0x0030	Data0 and Data1 Volume Control Register

DATA2_DATA3_VOL_CTR	0x0034	Data2 And Data3 Volume Control Register
HPF_EN_CTR	0x0038	High Pass Filter Enable Control Register
HPF_COEF_REG	0x003C	High Pass Filter Coef Register
HPF_GAIN_REG	0x0040	High Pass Filter Gain Register

8.2.5. Register Description

8.2.5.1. 0x0000 DMIC Enable Control Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: DMIC_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8	R/W	0x0	GLOBE_EN DMIC Globe Enable 0: Disable 1: Enable
7	R/W	0x0	DATA3_CHR_EN DATA3 Right Channel Enable 0: Disable 1: Enable
6	R/W	0x0	DATA3_CHL_EN DATA3 Left Channel Enable 0: Disable 1: Enable
5	R/W	0x0	DATA2_CHR_EN DATA2 Right Channel Enable 0: Disable 1: Enable
4	R/W	0x0	DATA2_CHL_EN DATA2 Left Channel Enable 0: Disable 1: Enable
3	R/W	0x0	DATA1_CHR_EN DATA1 Right Channel Enable 0: Disable 1: Enable
2	R/W	0x0	DATA1_CHL_EN DATA1 Left Channel Enable 0: Disable 1: Enable
1	R/W	0x0	DATA0_CHR_EN DATA0 Right Channel Enable 0: Disable

			1: Enable
0	R/W	0x0	DATA0_CHL_EN DATA0 Left Channel Enable 0: Disable 1: Enable

8.2.5.2. 0x0004 DMIC Sample Rate Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: DMIC_SR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2:0	R/W	0x0	DMIC_SR Sample Rate of DMIC 000: 48 kHz 010: 24 kHz 100: 12 kHz 110: Reserved 001: 32 kHz 011: 16 kHz 101: 8 kHz 111: Reserved 44.1 kHz/22.05 kHz/11.025 kHz can be supported by Audio PLL Configure Bit.

8.2.5.3. 0x0008 DMIC Control Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: DMIC_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:9	R/W	0x0	DMICFDT DMIC RXFIFO Delay Time for Writing Data after GLOBE_EN 00: 5ms 01: 10ms 10: 20ms 11: 30ms
8	R/W	0x0	DMICDFEN DMIC RXFIFO Delay Function for Writing Data after GLOBE_EN 0: Disable 1: Enable
7	R/W	0x0	DATA3 Left Data and Right Data Swap Enable 0: Disable 1: Enable
6	R/W	0x0	DATA2 Left Data and Right Data Swap Enable 0: Disable

			1: Enable
5	R/W	0x0	DATA1 Left Data and Right Data Swap Enable 0: Disable 1: Enable
4	R/W	0x0	DATA0 Left Data and Right Data Swap Enable 0: Disable 1: Enable
3:1	/	/	/
0	R/W	0x0	DMIC Oversample Rate 0: 128 (Support 8 kHz ~ 24 kHz) 1: 64 (Support 16 kHz ~ 48 kHz)

8.2.5.4. 0x0010 DMIC DATA Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: DMIC_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DMIC_DATA

8.2.5.5. 0x0014 DMIC Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: DMIC_INTC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x0	RXFIFO_DRQ_EN DMIC RXFIFO Data Available DRQ Enable 0: Disable 1: Enable
1	R/W	0x0	RXFIFO_OVERRUN_IRQ_EN DMIC RXFIFO Overrun IRQ Enable 0: Disable 1: Enable
0	R/W	0x0	DATA_IRQ_EN DMIC RXFIFO Data Available IRQ Enable 0: Disable 1: Enable

8.2.5.6. 0x0018 DMIC Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: DMIC_INTS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/

1	R/W1C	0x0	RXFIFO_OVERRUN_IRQ_PENDING DMIC RXFIFO Overrun Pending Interrupt 0: No pending IRQ 1: RXFIFO overrun pending IRQ Writing '1' to clear this interrupt or automatically clear if interrupt condition fails.
0	R/ W1C	0x0	RXFIFO_DATA_IRQ_PENDING DMIC RXFIFO Data Available Pending Interrupt 0: No pending IRQ 1: Data available pending IRQ Writing '1' to clear this interrupt or automatically clear if interrupt condition fails.

8.2.5.7. 0x001C DMIC RXFIFO Control Register (Default Value: 0x0000_0040)

Offset: 0x001C			**Register Name: DMIC_FIFO_CTR**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W1C	0x0	DMIC_RXFIFO_FLUSH DMIC RXFIFO Flush Writing '1' to flush RXFIFO, self clear to '0'
30:10	/	/	/
9	R/W	0x0	RXFIFO_MODE RXFIFO Output Mode (Mode 0, 1) 0: Expanding '0' at LSB of RXFIFO register 1: Expanding received sample sign bit at MSB of RXFIFO register For 24-bit received audio sample: Mode 0: RXDATA[31:0] = {RXFIFO_O[23:0], 8'h0} Mode 1: RXDATA[31:0] = {8{RXFIFO_O[23]}, RXFIFO_O[23:0]} For 16-bit received audio sample: Mode 0: RXDATA[31:0] = {RXFIFO_O[23:8], 16'h0} Mode 1: RXDATA[31:0] = {16{RXFIFO_O[23]}, RXFIFO_O[23:8]}
8	R/W	0x0	Sample_Resolution 0: 16-bit 1: 24- bit
7:0	R/W	0x40	RXFIFO_TRG_LEVEL RXFIFO Trigger Level (TRLV[7:0]) Interrupt and DMA request trigger level for DMIC RXFIFO normal condition IRQ/DRQ Generated when WLEVEL > TRLV[7:0]) WLEVEL represents the number of valid samples in the DMIC RXFIFO

8.2.5.8. 0x0020 DMIC RXFIFO Status Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: DMIC_RXFIFO_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	DMIC_DATA_CNT DMIC RXFIFO Available Sample Word Counter

8.2.5.9. 0x0024 DMIC Channel Numbers Register (Default Value: 0x0000_0001)

Offset: 0x0024			Register Name: DMIC_CH_NUM
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2:0	R/W	0x1	DMIC_CH_NUM DMIC enable channel numbers are (N+1).

8.2.5.10. 0x0028 DMIC Channel Mapping Register (Default Value: 0x7654_3210)

Offset: 0x0028			Register Name: DMIC_CH_MAP
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x7	DMIC_CH7_MAP DMIC Channel 7 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
27:24	R/W	0x6	DMIC_CH6_MAP DMIC Channel 6 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
23:20	R/W	0x5	DMIC_CH5_MAP DMIC Channel 5 Mapping 0000: DATA0 Left Channel

			0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
19:16	R/W	0x4	DMIC_CH4_MAP DMIC Channel 4 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
15:12	R/W	0x3	DMIC_CH3_MAP DMIC Channel 3 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
11:8	R/W	0x2	DMIC_CH2_MAP DMIC Channel 2 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
7:4	R/W	0x1	DMIC_CH1_MAP DMIC Channel 1 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel

			0110: DATA3 Left Channel 0111: DATA3 Right Channel
3:0	R/W	0x0	DMIC_CH0_MAP DMIC Channel0 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel

8.2.5.11. 0x002C DMIC Counter Register (Default Value: 0x0000_0000)

Offset: 0x002C			**Register Name: DMIC_CNT**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	DMIC_CNT RX Sample Counter The audio sample number of writing into RXFIFO. When one sample is written by Digital Audio Engine, the RX sample counter register increases by one. The RX sample counter register can be set to any initial valve at any time. After been updated by the initial value, the counter register should count on base of this initial value. **Note: It is used for Audio/ Video Synchronization.**

8.2.5.12. 0x0030 DATA0 and DATA1 Volume Control Register (Default Value: 0xA0A0_A0A0)

Offset: 0x0030			**Register Name: DATA0_DATA1_VOL_CTR**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0xA0	DATA1L_VOL (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB …….. 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ……. 0xFF: 71.25 dB
23:16	R/W	0xA0	DATA1R_VOL (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute

			0x01: -119.25 dB …….. 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB …….. 0xFF: 71.25 dB
15:8	R/W	0xA0	DATA0L_VOL (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB …….. 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB …….. 0xFF: 71.25 dB
7:0	R/W	0xA0	DATA0R_VOL (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB …….. 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB …….. 0xFF: 71.25 dB

8.2.5.13. 0x0034 DATA2 and DATA3 Volume Control Register (Default Value: 0xA0A0_A0A0)

Offset: 0x0034			Register Name: DATA2_DATA3_VOL_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0xA0	DATA3L_VOL (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB …….. 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB …….. 0xFF: 71.25 dB
23:16	R/W	0xA0	DATA3R_VOL (-119.25 dB to 71.25 dB, 0.75 dB/Step)

			0x00: Mute 0x01: -119.25 dB …….. 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ……. 0xFF: 71.25 dB
15:8	R/W	0xA0	DATA2L_VOL (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB …….. 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ……. 0xFF: 71.25 dB
7:0	R/W	0xA0	DATA2R_VOL (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB …….. 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB …….. 0xFF: 71.25 dB

8.2.5.14. 0x0038 High Pass Filter Enable Control Register (Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: HPF_EN_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	HPF_DATA3_CHR_EN High Pass Filter DATA3 Right Channel Enable 0: Disable 1: Enable
6	R/W	0x0	HPF_DATA3_CHL_EN High Pass Filter DATA3 Left Channel Enable 0: Disable 1: Enable
5	R/W	0x0	HPF_DATA2_CHR_EN High Pass Filter DATA2 Right Channel Enable

			0: Disable 1: Enable
4	R/W	0x0	HPF_DATA2_CHL_EN High Pass Filter DATA2 Left Channel Enable 0: Disable 1: Enable
3	R/W	0x0	HPF_DATA1_CHR_EN High Pass Filter DATA1 Right Channel Enable 0: Disable 1: Enable
2	R/W	0x0	HPF_DATA1_CHL_EN High Pass Filter DATA1 Left Channel Enable 0: Disable 1: Enable
1	R/W	0x0	HPF_DATA0_CHR_EN High Pass Filter DATA0 Right Channel Enable 0: Disable 1: Enable
0	R/W	0x0	HPF_DATA0_CHL_EN High Pass Filter DATA0 Left Channel Enable 0: Disable 1: Enable

8.2.5.15. 0x003C High Pass Filter Coef Register (Default Value: 0x00FF_AA45)

Offset: 0x003C			Register Name: HPF_COEF_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x00FFAA45	High Pass Filter Coefficient

8.2.5.16. 0x0040 High Pass Filter Gain Register (Default Value: 0x00FF_D522)

Offset: 0x0040			Register Name: HPF_GAIN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x00FFD522	High Pass Filter Gain

8.3. OWA

8.3.1. Overview

The One Wire Audio(OWA) provides a serial bus interface for audio data between system. This interface is widely used for consumer audio.

The OWA controller includes the following features:

- IEC-60958 transmitter functionality
- Compliance with S/PDIF Interface
- Supports channel status insertion for the transmitter
- Hardware parity generation on the transmitter
- One 128×24bits TXFIFO for audio data transfer
- Programmable FIFO thresholds
- Interrupt and DMA support
- Supports 16-bit, 20-bit, 24-bit data formats

8.3.2. Block Diagram

The block diagram of the OWA is shown as follows.

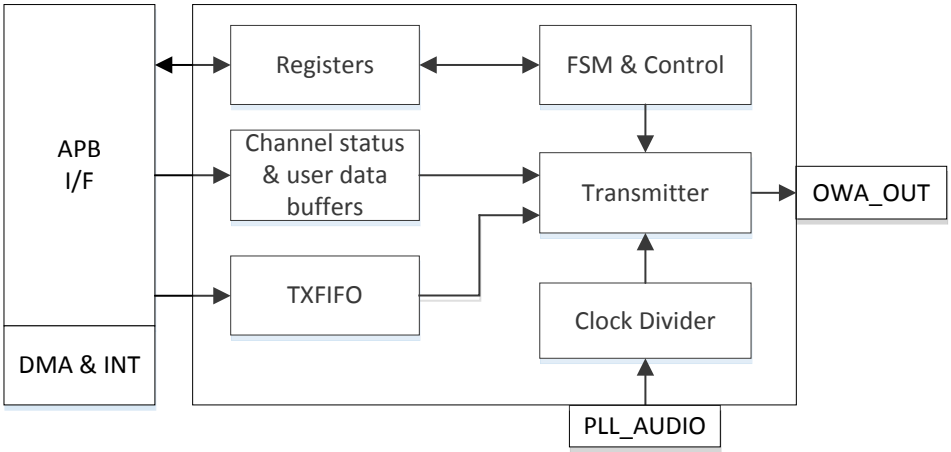

Figure 8- 14. OWA Block Diagram

8.3.3. Operations and Functional Descriptions

8.3.3.1. External Signals

OWA is a Biphase-Mark Encoding Digital Audio Transfer protocol. In this protocol, the clock signal and data signal are

transfer in the same line. Table 8-5 describes the external signals of OWA. OWA_OUT is output pin for output clock and DATA.

Table 8- 5. OWA External Signals

Signal Name	Description	Type
OWA_OUT	OWA Output	O
OWA_MCLK	OWA Master Clock	O

8.3.3.2. Clock Sources

Table 8-6 describes the clock sources for OWA. Users can see **Chapter 3.3.CCU** for clock setting, configuration and gating information.

Table 8- 6. OWA Clock Sources

Clock Name	Description
PLL_AUDIO	24.576 MHz or 22.5792 MHz generated by PLL_AUDIO to produce 48 kHz or 44.1 kHz serial frequency

8.3.3.3. Biphase-Mark Code (BMC)

In OWA format, the digital signal is coded using the biphase-mark code (BMC). The clock, frame, and data are embedded in only one signal—the data pin. In the BMC system, each data bit is encoded into two logical states (00, 01, 10, or 11) at the pin. Figure 8-15 and Table 8-7 show how data is encoded to the BMC format.

As shown in Figure 8-15, the frequency of the clock is twice the data bit rate. In addition, the clock is always programmed to 128xfs, where fs is the sample rate. The device receiving in OWA format can recover the clock and frame information from the BMC signal.

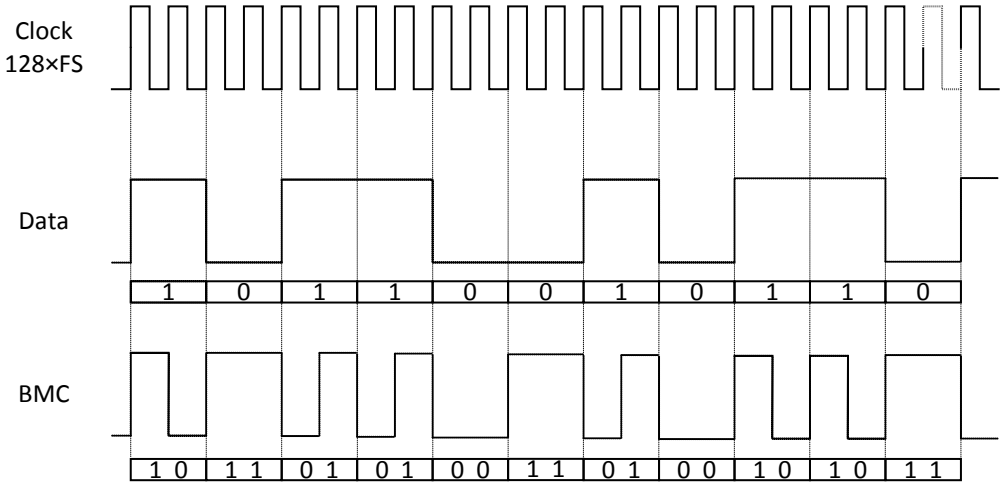

Figure 8- 15. OWA Biphase-Mark Code

Table 8- 7. Biphase-Mark Encoder

Data	Previous State	BMC
0	0	11
0	1	00
1	0	10
1	1	01

8.3.3.4. OWA Transmit Format

The OWA supports digital audio data transfer out and receive in. And it supports full-duplex synchronous work mode. Software can set the work mode by the OWA Control Register.

Every audio sample transmitted in a subframe consists 32-bit, numbered from 0 to 31. Figure 8-16 shows a subframe.

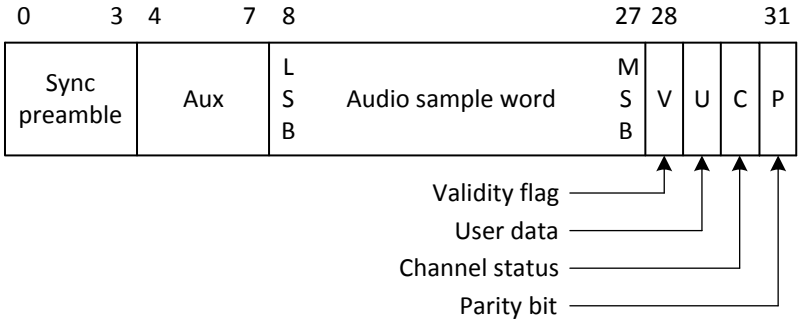

Figure 8- 16. OWA Sub-Frame Format

Bit 0-3 carry one of the four permitted preambles to signify the type of audio sample in the current subframe. The preamble is not encoded in BMC format, and therefore the preamble code can contain more than two consecutive 0 or 1 logical states in a row. See Table 9-8.

Bit 4-27 carry the audio sample word in linear 2s-complement representation. The most-significant bit (MSB) is carried by bit 27. When a 24-bit coding range is used, the least-significant bit (LSB) is in bit 4. When a 20-bit coding range is used, **Bit 8-27** carry the audio sample word with the LSB in bit 8. **Bit 4-7** may be used for other applications and are designated auxiliary sample bits.

If the source provides fewer bits than the interface allows (either 20 or 24), the unused LSBs are set to logical 0. For a nonlinear PCM audio application or a data application, the main data field may carry any other information.

Bit 28 carries the validity bit (V) associated with the main data field in the subframe.

Bit 29 carries the user data channel (U) associated with the main data field in the subframe.

Bit 30 carries the channel status information (C) associated with the main data field in the subframe. The channel status indicates if the data in the subframe is digital audio or some other type of data.

Bit 31 carries a parity bit (P) such that **Bit 4-31** carry an even number of 1s and an even number of 0s (even parity). As shown in Table 8-8, the preambles (**Bit 0-3**) are also defined with even parity.

Table 8- 8. Preamble Codes

Preamble Code	Previous Logical State	Logical State	Description
B(or Z)	0	1110 1000	Start of a block and subframe 1
M(or X)	0	1110 0010	Subframe 1
W(or Y)	0	1110 0100	Subframe 2

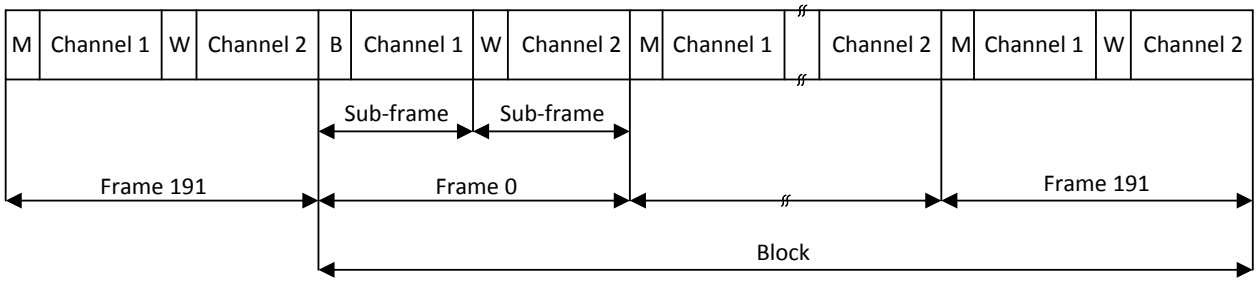

Figure 8- 17. OWA Frame/Block Format

8.3.3.5. Operation Modes

The software operation of the OWA is divided into five steps: system setup, OWA initialization, channel setup, DMA setup and enable/disable module. These five steps are described in detail in the following sections.

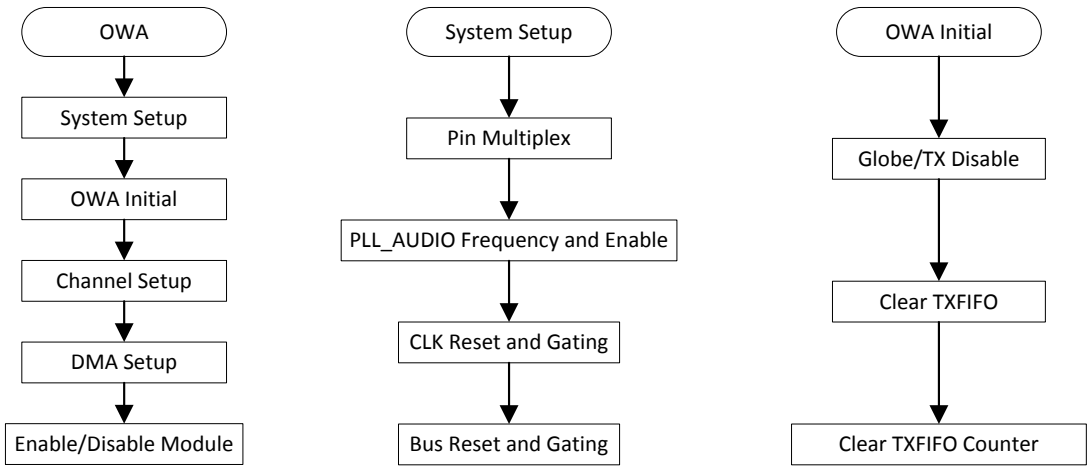

Figure 8- 18. OWA Operation Flow

(1) System Setup and OWA Initialization

The first step in the OWA initialization is properly programming the GPIO. Because the OWA port is a multiplex pin.

You can find the function in the **Port Controller**.

The clock source for the OWA should be followed. At first you must reset the audio PLL in the **CCU**. The second step, you must setup the frequency of the Audio PLL. After that, you must open the OWA gating. At last, you must open the OWA bus gating.

After the system setup, the register of OWA can be setup. At first, you should reset the OWA by writing 1 to **OWA_CTL**[0] and clear the TX FIFO by writing 1 to **OWA_FCTL**[30]. After that you should enable the globe enable bit by writing 1 to **OWA_CTL**[1], and clear the interrupt and TX counter by the **OWA_ISTA** and **OWA_TX_CNT**.

(2) Channel Setup and DMA Setup

The OWA supports three methods to transfer the data. The most common way is DMA, the configuration of DMA can be found in the **DMA**. In this module, you just enable the DRQ by writing the **OWA_INT**[7].

(3) Enable and Disable OWA

To enable the function, you can enable TX by writing the **OWA_TX_CFIG**[0]. After that, you must enable OWA by writing 1 to the **GEN** bit in the **OWA_CTL** register. Writing 0 to the **GEN** bit to disable process.

8.3.4. Register List

Module Name	Base Address
OWA	0x05093000

Register Name	Offset	Description
OWA_GEN_CTL	0x0000	OWA General Control Register
OWA_TX_CFIG	0x0004	OWA TX Configuration Register
OWA_ISTA	0x000C	OWA Interrupt Status Register
OWA_FCTL	0x0014	OWA FIFO Control Register
OWA_FSTA	0x0018	OWA FIFO Status Register
OWA_INT	0x001C	OWA Interrupt Control Register
OWA_TX_FIFO	0x0020	OWA TX FIFO Register
OWA_TX_CNT	0x0024	OWA TX Counter Register
OWA_TX_CHSTA0	0x002C	OWA TX Channel Status Register0
OWA_TX_CHSTA1	0x0030	OWA TX Channel Status Register1

8.3.5. Register Description

8.3.5.1. 0x0000 OWA General Control Register (Default Value: 0x0000_0080)

Offset: 0x0000			Register Name: OWA_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9:5	R/W	0x0	MCLKDIV MCLK Clock Divide Ratio MCLK Divide Ratio from PLL_AUDIO 00000: Divide by 128 00001: Divide by 2 00010: Divide by 4 00011: Divide by 6 00100: Divide by 8 00101: Divide by 10 00110: Divide by 12 00111: Divide by 14 01000: Divide by 16 01001: Divide by 18 01010: Divide by 20 01011: Divide by 22 01100: Divide by 24 …… 11111: Divide by 62
4	/	/	/
3	R/W	0x0	MCLKEN MCLK Enable 0: Disable 1: Enable
2	R/W	0x0	LOOP Loop Back Test 0: Normal Mode 1: Loop Back Test When setting to ‘1’, DOUT and DIN need be connected.
1	R/W	0x0	GEN Global Enable A disable on this bit overrides any other block or channel enables and flushes all FIFOs. 0: Disable 1: Enable
0	R/W	0x0	RST Reset 0: Normal 1: Reset

			Self clear to 0.

8.3.5.2. 0x0004 OWA TX Configure Register (Default Value: 0x0000_00F0)

Offset: 0x0004			Register Name: OWA_TX_CFIG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	TX_SINGLE_MODE Tx Single Channel Mode 0: Disable 1: Enable
30:18	/	/	/
17	R/W	0x0	ASS Audio Sample Select with TX FIFO Underrun when 0: Sending 0 1: Sending the last audio **Note: This bit is only valid in PCM mode.**
16	R/W	0x0	TX_AUDIO TX Data Type 0: Linear PCM (Valid bit of both sub-frame set to 0) 1: Non-audio(Valid bit of both sub-frame set to 1)
15:9	/	/	/
8:4	R/W	0xF	TX_RATIO TX Clock Divide Ratio Clock divide ratio = TX_TATIO +1 Fs= PLL_AUDIO/[(TX_TATIO +1)*64*2]
3:2	R/W	0x0	TX_SF TX Sample Format 00: 16 bits 01: 20 bits 10: 24 bits 11: Reserved
1	R/W	0x0	TX_CHM CHSTMODE 0: Channel status A&B set to 0 1: Channel status A&B generated from TX_CHSTA
0	R/W	0x0	TXEN 0: Disabled 1: Enabled

8.3.5.3. 0x000C OWA Interrupt Status Register (Default Value: 0x0000_0010)

Offset: 0x000C	Register Name: OWA_ISTA

Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6	R/W1C	0x0	TXU_INT TX FIFO Underrun Pending Interrupt 0: No Pending IRQ 1: FIFO Underrun Pending Interrupt Writing “1” to clear this interrupt.
5	R/W1C	0x0	TXO_INT TX FIFO Overrun Pending Interrupt 0: No Pending IRQ 1: FIFO Overrun Pending Interrupt Writing “1” to clear this interrupt.
4	R/W1C	0x1	TXE_INT TX FIFO Empty Pending Interrupt 0: No Pending IRQ 1: FIFO Empty Pending Interrupt Writing “1” to clear this interrupt or automatically clear if the interrupt condition fails.
3:0	/	/	/

8.3.5.4. 0x0014 OWA FIFO Control Register (Default Value: 0x0004_0200)

Offset: 0x0014			Register Name: OWA_FCTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	HUB_EN Audio Hub Enable 0: Disable 1: Enable
30	R/W1C	0x0	FTX Write ‘1’ to flush TXFIFO, self clear to ‘0’.
29:20	/	/	/
19:12	R/W	0x40	TXTL TX FIFO Empty Trigger Level Interrupt and DMA request trigger level for TX FIFO normal condition. Trigger Level = TXTL
11:3	/	/	/
2	R/W	0x0	TXIM TXFIFO Input Mode(Mode0, 1) 0: Valid data at the MSB of TXFIFO Register 1: Valid data at the LSB of TXFIFO Register Example for 20-bit transmitted audio sample: Mode 0: TXFIFO[23:0] = {APB_WDATA[31:12], 4’h0} Mode 1: TXFIFO[23:0] = {APB_WDATA[19:0], 4’h0}

1:0	/	/	/

8.3.5.5. 0x0018 OWA FIFO Status Register (Default Value: 0x8080_0000)

Offset: 0x0018			Register Name: OWA_FSTA
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x1	TXE TXFIFO Empty (indicate TXFIFO is not full) 0: No room for new sample in TXFIFO 1: More than one room for new sample in TXFIFO (>=1 Word)
30:24	/	/	/
23:16	R	0x80	TXE_CNT TXFIFO Empty Space Word Counter
15:0	/	/	/

8.3.5.6. 0x001C OWA Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: OWA_INT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	TX_DRQ TXFIFO Empty DRQ Enable 0: Disable 1: Enable
6	R/W	0x0	TXUI_EN TXFIFO Underrun Interrupt Enable 0: Disable 1: Enable
5	R/W	0x0	TXOI_EN TXFIFO Overrun Interrupt Enable 0: Disable 1: Enable
4	R/W	0x0	TXEI_EN TXFIFO Empty Interrupt Enable 0: Disable 1: Enable
3:0	/	/	/

8.3.5.7. 0x0020 OWA TX FIFO Register (Default Value: 0x0000_0000)

Offset: 0x0020	Register Name: OWA_TXFIFO

Bit	Read/Write	Default/Hex	Description
31:0	W	0x0	TX_DATA Transmitting A, B channel data should be written this register one by one. The A channel data is first and then the B channel data.

8.3.5.8. 0x0024 OWA TX Counter Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: OWA_TX_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	TX_CNT TX Sample Counter This is the audio sample number that sent into TXFIFO. When one sample is put into TXFIFO by DMA or by host IO, the TX sample counter register increases by one. The TX sample counter register can be set to any initial value at any time. After updated by the initial value, the counter register should count on base of this initial value.

8.3.5.9. 0x002C OWA TX Channel Status Register0 (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: OWA_TX_CHSTA0
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 30	/	/	/
29:28	R/W	0x0	CA Clock Accuracy 00: Level 2 01: Level 1 10: Level 3 11: Not matched
27:24	R/W	0x0	FREQ Sample Frequency 0000: 44.1 kHz 0001: Not indicated 0010: 48 kHz 0011: 32 kHz 0100: 22.05 kHz 0101: Reserved 0110: 24 kHz 0111: Reserved 1000: Reserved 1001: 768 kHz 1010: 96 kHz 1011: Reserved 1100: 176.4 kHz

			1101: Reserved 1110: 192 kHz 1111: Reserved
23:20	R/W	0x0	CN Channel Number
19:16	R/W	0x0	SN Source Number
15:8	R/W	0x0	CC Category Code Indicates the kind of equipment that generates the digital audio interface signal.
7:6	R/W	0x0	MODE Mode 00: Default Mode 01~11: Reserved
5:3	R/W	0x0	EMP Emphasis Additional format information For bit 1 = “0”, Linear PCM audio mode: 000: 2 audio channels without pre-emphasis 001: 2 audio channels with 50 µs / 15 µs pre-emphasis 010: Reserved (for 2 audio channels with pre-emphasis) 011: Reserved (for 2 audio channels with pre-emphasis) 100~111: Reserved For bit 1 = “1”, other than Linear PCM applications: 000: Default state 001~111: Reserved
2	R/W	0x0	CP Copyright 0: Copyright is asserted 1: No copyright is asserted
1	R/W	0x0	TYPE Audio Data Type 0: Linear PCM samples 1: Non-linear PCM audio
0	R/W	0x0	PRO Application Type 0: Consumer application 1: Professional application This bit must be fixed to “0”.

8.3.5.10. 0x0030 OWA TX Channel Status Register1 (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: OWA_TX_CHSTA1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9:8	R/W	0x0	CGMS_A 00: Copying is permitted without restriction 01: One generation of copies may be made 10: Condition not be used 11: No copying is permitted
7:4	R/W	0x0	ORIG_FREQ Original Sampling Frequency 0000: Not indicated 0001: 192 kHz 0010: 12 kHz 0011: 176.4 kHz 0100: Reserved 0101: 96 kHz 0110: 8 kHz 0111: 88.2 kHz 1000: 16 kHz 1001: 24 kHz 1010: 11.025 kHz 1011: 22.05 kHz 1100: 32 kHz 1101: 48 kHz 1110: Reserved 1111: 44.1 kHz
3:1	R/W	0x0	WL Sample Word Length For bit 0 = “0”: 000: Not indicated 001: 16 bit 010: 18 bit 100: 19 bit 101: 20 bit 110: 17 bit 111: Reserved For bit 0 = “1”: 000: Not indicated 001: 20 bit 010: 22 bit 100: 23 bit 101: 24 bit

			110: 21 bit 111: Reserved
0	R/W	0x0	MWL Max Word Length 0: Maximum audio sample word length is 20 bits 1: Maximum audio sample word length is 24 bits

8.4. Audio Codec

8.4.1. Overview

The Audio Codec has 2-ch DAC with a high level of mixed-signal integration. The DRC with integrated hardware DAP engine can be used in playback path.

The Audio Codec has the following features:

- Two audio digital-to-analog(DAC) channel
  - Supports 8 kHz to 192 kHz DAC sample rate
  - SNR 95dB ±2dB @A-weight, THD+N -80±3dB, output level more than 0.55Vrms
  - DAC power consumption 3.0 mA@1.8V
  - Supports 16-bit and 20-bit audio sample resolution
- One audio output
  - One differential LINEOUTP/N or single-end LINEOUTL/R output
- Supports Dynamic Range Controller(DRC) adjusting the DAC playback
- One low-noise analog microphone bias output
- One 128x24-bits FIFO for DAC data transmit
- Programmable FIFO thresholds
- Interrupt and DMA support

8.4.2. Block Diagram

Figure 8-19 shows the block diagram of Audio Codec.

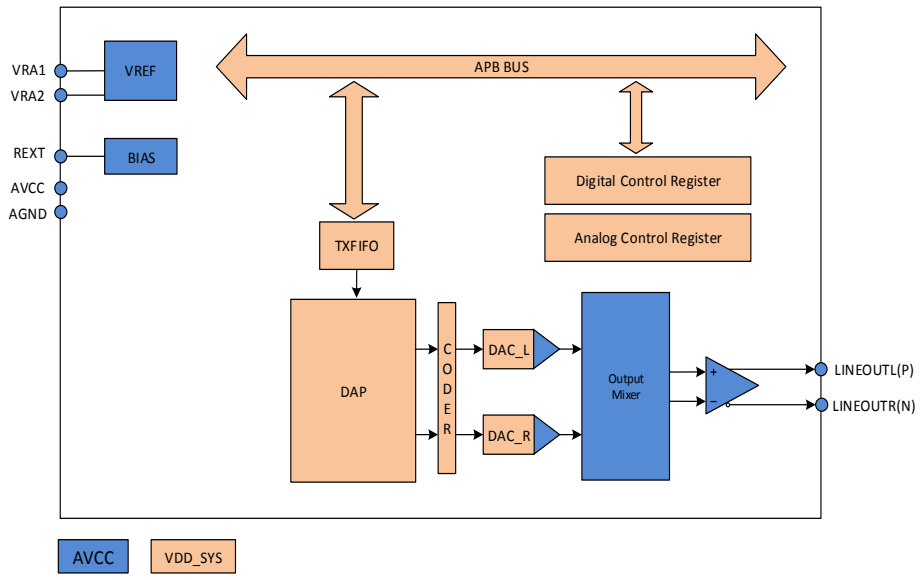

Figure 8- 19. Audio Codec Block Diagram

8.4.3. Operations and Functional Descriptions

8.4.3.1. External Signals

8.4.3.1.1. Analog I/O Pins

Signal	Type	Description
LINEOUTL(P)	AO	Left single-end output for lineout(or differential mono positive output)
LINEOUTR(N)	AO	Right single-end output for lineout(or differential mono negative output)

8.4.3.1.2. Reference

Signal	Type	Description
REXT	AO	External reference pin
VRA1	AO	Internal reference voltage
VRA2	AO	Internal reference voltage

8.4.3.1.3. Power/Ground

Signal	Type	Description
AVCC	P	Analog power
AGND	G	Analog ground

8.4.3.2. Clock Sources

Figure 8-20 describes the Audio Codec clock source. Users can see ***CCU*** for clock setting, configuration and gating information.

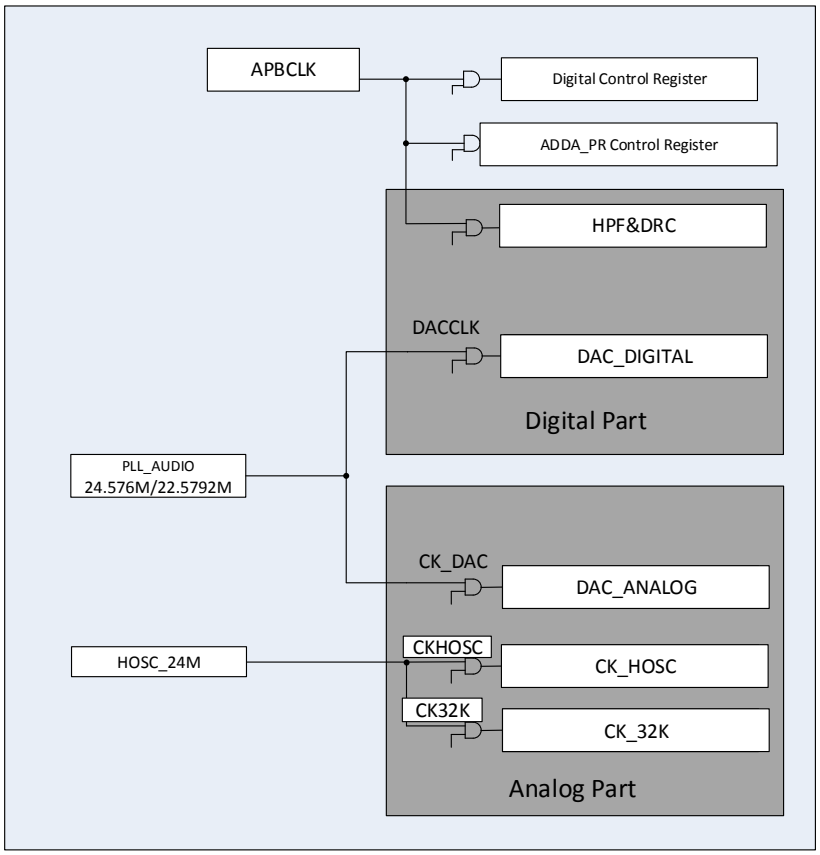

Figure 8- 20. Audio Codec Clock Diagram

The clock of digital part is from PLL_AUDIO(1X). The clock of analog part includes CK_ADC, CK_DAC, CK_DITHER, CK_HOSC, CK_32K. Where, CK_ADC, CK_DAC and CK_DITHER is provided by PLL_AUDIO. CK_HOSC, CK_32K is provided by system oscillator 24M. These clocks need ensure that VDD-SYS is not power-off.

8.4.3.3. Reset System

8.4.3.3.1. Digital Part Reset System

The SYS_RST will be provided by the VDD_SYS domain, which comes from VDD_SYS domain and is produced by RTC domain. Each domain has the de-bounce to confirm whether the reset system is strong. The codec register part, MIX will be reset by the SYS_RST during the power on or the system soft writing the reset control logic. The other parts will be reset by the soft configuration through writing register.

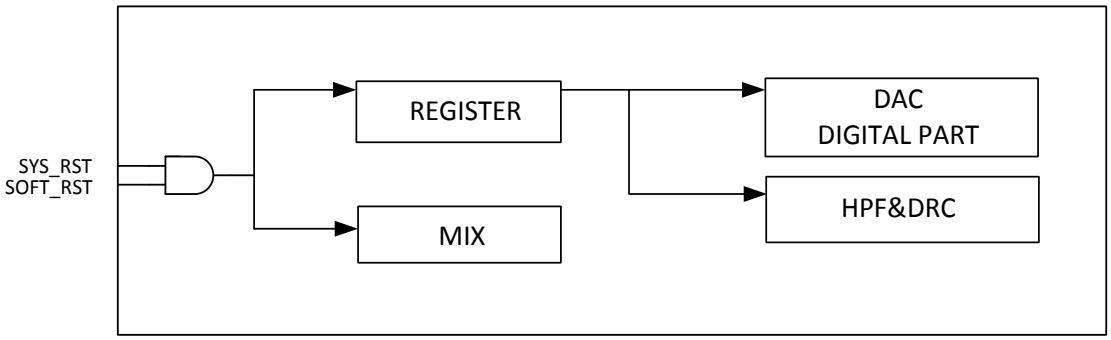

Figure 8- 21. Audio Codec Digital Part Reset System

8.4.3.3.2. Analog Part Reset System

When AVCC is powered on, it will send the AVCC_POR signal. And the AVCC_POR signal passes the level shift and RC filter part to ADDA logic core, which will reset the AVCC analog part.

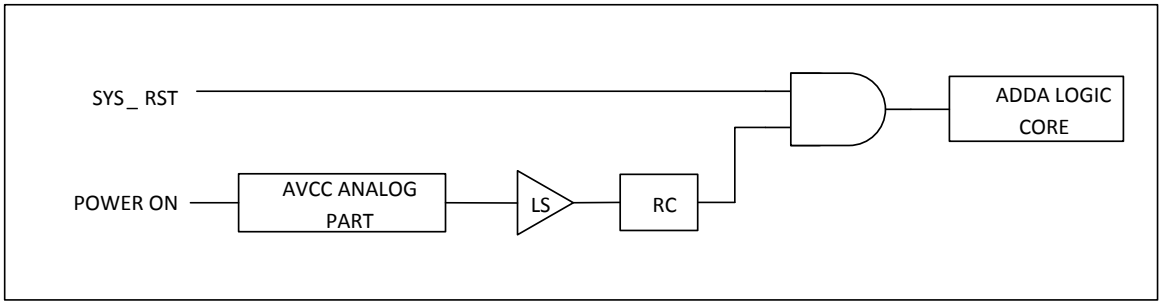

Figure 8- 22. Audio Codec Analog Part Reset System

8.4.3.4. Data Path Diagram

Figure 8-23 shows a data path of the Audio Codec.

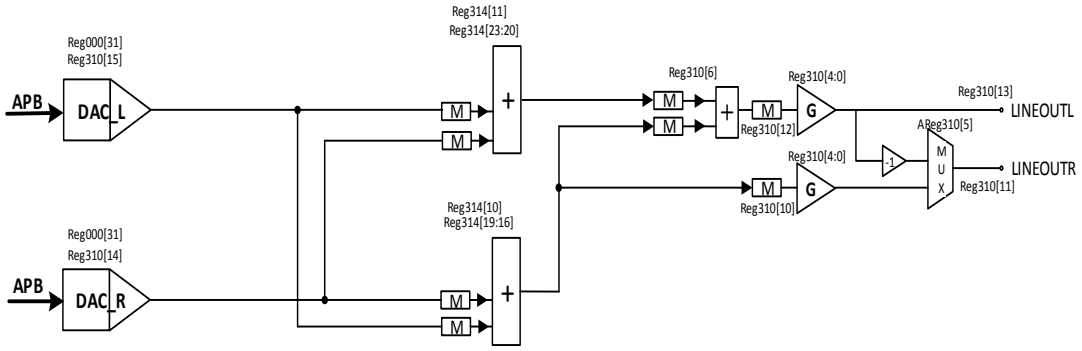

Figure 8- 23. Audio Codec Data Path Diagram

8.4.3.5. Stereo DAC

The stereo DAC sample rate can be configured by setting the register. In order to save power, the DAC can be enabled/disabled by setting the bit[15:14] of the **DAC_REG** register. The digital DAC part can be enabled/disabled by the bit[31] of the **AC_DAC_DPC** register.

8.4.3.6. Analog Audio Output Path

The Audio Codec has one analog output port:

- LINEOUTP/N or LINEOUTL/R

The LINEOUT provides one differential output to drive line level signals to external audio equipment. The LINEOUTL(P) output source is from DACL. The LINEOUTR(N) output source is from DAC differential output. The volume control is logarithmic with a 43.5dB rang in 1.5dB step from -43.5dB to 0dB. The LINEOUT output buffer is powered up or down by the bit[13] and bit[11] of **DAC_REG**.

8.4.3.7. Interrupt

The Audio Codec has two interrupts. Figure 8-24 describes the Audio Codec interrupt system.

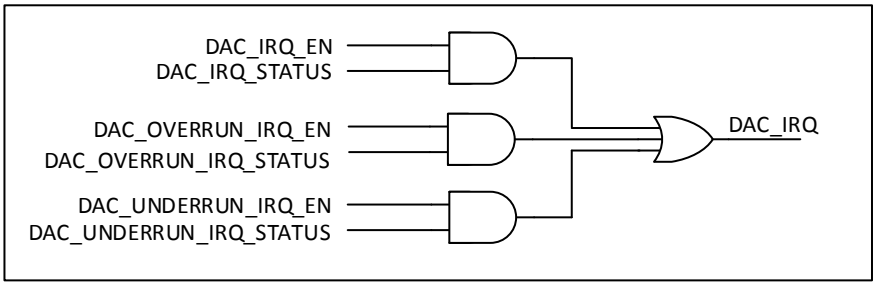

Figure 8- 24. Audio Codec Interrupt System

8.4.3.8. DAP

8.4.3.8.1. DAP Data Flow

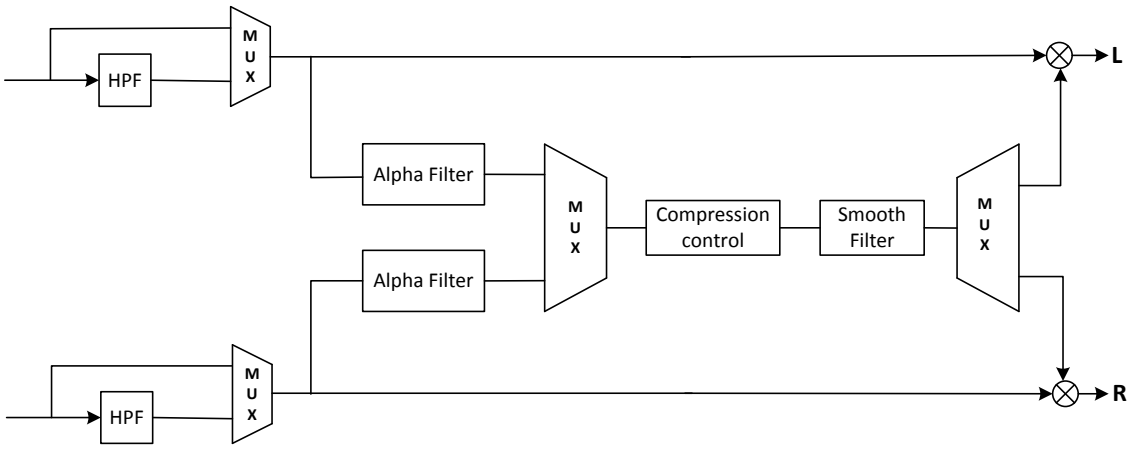

Figure 8- 25. DAP Data Flow

8.4.3.8.2. DRC Function

The DRC scheme has three thresholds, three offset, and four slope (all programmable). There is one ganged DRC for the left/right channels. The diagram of DRC input/output is as follows.

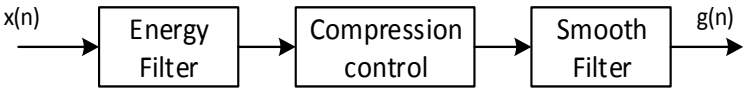

Figure 8- 26. DRC Block Diagram

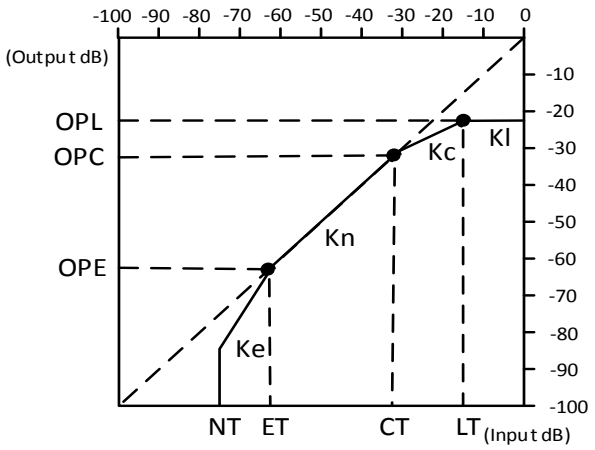

Figure 8- 27. DRC Static Curve Parameters

Professional-quality dynamic range compression automatically adjusts volume to flatten volume level.

One DRC is for left/right, and one DRC is for subwoofer.

Each DRC has adjustable threshold, offset, and compression levels, programmable energy, attack, and decay time constants.

Transparent compression: Compressors can attack fast enough to avoid apparent clipping before engaging, and decay times can be set slow enough to avoid pumping.

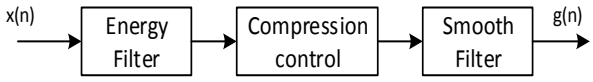

Figure 8- 28. DRC Process

DRC parameter setting:

- **Number format**

The Number format is N.M which means there are N bits to the left of the decimal point including the sign bit and M bits to the right of the decimal point. For example, Numbers formatted 9.23 means that there are 9 bits at the left of the decimal point and 23 bits at the right decimal point.

- **Energy Filter**

The Energy Filter is to estimate the RMS value of the audio data stream into DRC, and has two parameters, which determine the time window over which RMS to be made. The parameter is computed by $\alpha = 1 - e^{-2.2Ts/ta}$.

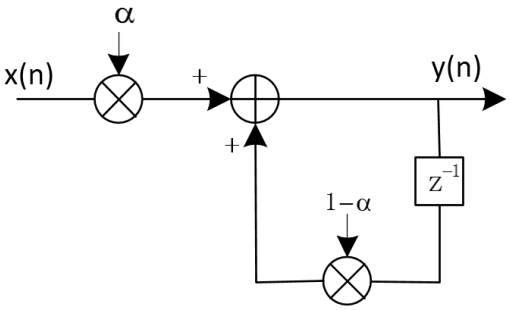

Figure 8- 29. Energy Filter Structure

Compression Control

This element has ten parameters (ET, CT, LT, Ke, Kn, Kc, Kl, OPL, OPC, OPE), which are all programmable, and the computation will be explained as follows.

- **Threshold Parameter Computation(T parameter)**

The threshold is the value that determines the signal to be compressed or not. When the signal's RMS is larger than the threshold, the signal will be compressed. The value of threshold input to the coefficient register is computed by

$$Tin = -\frac{T_{dB}}{6.0206}$$

There, T_{dB} must less than zero, the positive value is illegal.

For example, it is desired to set CT=-40dB, then the Tin require to set CT to -40dB is CT_{in} = - (-40dB)/6.0206 = 6.644, CT_{in} is entered as a 32-bit number in 8.24 format.

Therefore, CT_{in} = 6.644 = 0000 0110.1010 0100 1101 0011 1100 0000 = 0x06A4 D3C0 in 8.24 format.

- **Slope Parameter Computation (K parameter)**

The K is the slope within compression region. For example, a n:1 compression means that an output increase 1dB is for n dB RMS input. The k input to the coefficient ram is computed by $K = \frac{1}{n}$

There, n is from 1 to 50, and must be integer.

For example, it is desired to set 2:1,then the Kc require to set to 2:1 is Kc = 1/2 = 0.5, Kc is entered as a 32-bit number in 8.24 format.

Therefore, Kc = 0.5 = 0000 0000.1000 0000 0000 0000 0000 0000 = 0x0080 0000 in 8.24 format.

- **Gain Smooth Filter**

The Gain Smooth Filter is to smooth the gain and control the ratio of gain increase and decrease. The decay time and attack is shown in Figure 8-30. The structure of the Gain Smooth filter is also the Alpha filter, so the rise time computation is the same as the Energy filter which is $\alpha = 1 - e^{-2.2Ts/ta}$.

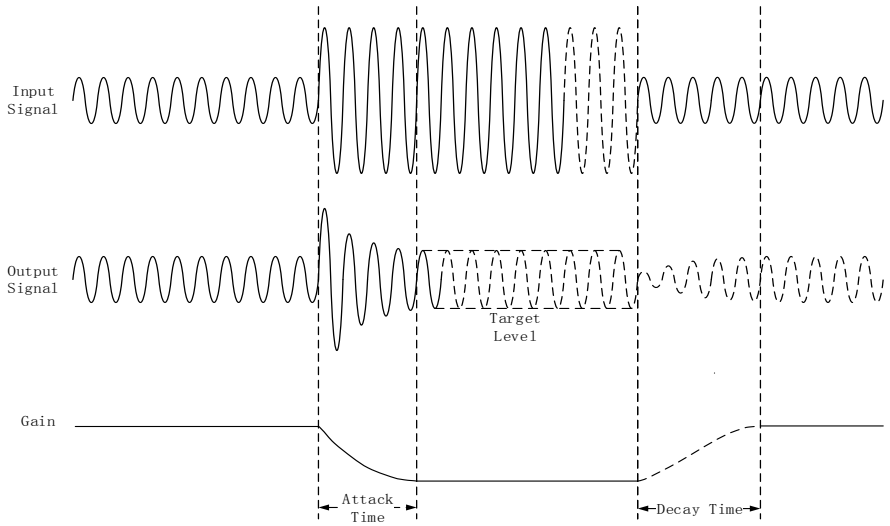

Figure 8- 30. Gain Smooth Filter

8.4.4. Programming Guidelines

8.4.4.1. Playback Process

(1) Codec initial: Open audio codec bus clock gating and de-assert bus reset through **AUDIO_CODEC_BGR_REG**, configure PLL_Audio frequency and enable PLL_Audio through **PLL_AUDIO_CTRL_REG**. Please refer to CCU in chapter 3.3 about detail.
(2) Configure MIX path.
(3) Set sample rate, configure data transfer format, enable DAC.
(4) DMA configure and DMA request.
(5) Enable DAC DRQ and DMA.

8.4.5. Register List

Module Name	Base Address
Audio Codec	0x05096000

Register Name	Offset	Description
AC_DAC_DPC	0x0000	DAC Digital Part Control Register
AC_DAC_FIFOC	0x0010	DAC FIFO Control Register
AC_DAC_FIFOS	0x0014	DAC FIFO Status Register
AC_DAC_TXDATA	0x0020	DAC TX DATA Register
AC_DAC_CNT	0x0024	DAC TX FIFO Counter Register
AC_DAC_DG	0x0028	DAC Debug Register
AC_DAC_DAP_CTRL	0x00F0	DAC DAP Control Register
AC_DAC_DRC_HHPFC	0x0100	DAC DRC High HPF Coef Register
AC_DAC_DRC_LHPFC	0x0104	DAC DRC Low HPF Coef Register
AC_DAC_DRC_CTRL	0x0108	DAC DRC Control Register
AC_DAC_DRC_LPFHAT	0x010C	DAC DRC Left Peak Filter High Attack Time Coef Register
AC_DAC_DRC_LPFLAT	0x0110	DAC DRC Left Peak Filter Low Attack Time Coef Register
AC_DAC_DRC_LPFLAT	0x0118	DAC DRC Left Peak Filter Low Attack Time Coef Register
AC_DAC_DRC_LPFHRT	0x011C	DAC DRC Left Peak Filter High Release Time Coef Register
AC_DAC_DRC_LPFLRT	0x0120	DAC DRC Left Peak Filter Low Release Time Coef Register
AC_DAC_DRC_RPFHRT	0x0124	DAC DRC Right Peak Filter High Release Time Coef Register
AC_DAC_DRC_RPFLRT	0x0128	DAC DRC Right Peak Filter Low Release Time Coef Register
AC_DAC_DRC_LRMSHAT	0x012C	DAC DRC Left RMS Filter High Coef Register
AC_DAC_DRC_LRMSLAT	0x0130	DAC DRC Left RMS Filter Low Coef Register
AC_DAC_DRC_RRMSHAT	0x0134	DAC DRC Right RMS Filter High Coef Register
AC_DAC_DRC_RRMSLAT	0x0138	DAC DRC Right RMS Filter Low Coef Register
AC_DAC_DRC_HCT	0x013C	DAC DRC Compressor Threshold High Setting Register
AC_DAC_DRC_LCT	0x0140	DAC DRC Compressor Slope High Setting Register
AC_DAC_DRC_HKC	0x0144	DAC DRC Compressor Slope High Setting Register
AC_DAC_DRC_LKC	0x0148	DAC DRC Compressor Slope Low Setting Register
AC_DAC_DRC_HOPC	0x014C	DAC DRC Compressor High Output at Compressor Threshold Register
AC_DAC_DRC_LOPC	0x0150	DAC DRC Compressor Low Output at Compressor Threshold Register
AC_DAC_DRC_HLT	0x0154	DAC DRC Limiter Threshold High Setting Register
AC_DAC_DRC_LLT	0x0158	DAC DRC Limiter Threshold Low Setting Register
AC_DAC_DRC_HKl	0x015C	DAC DRC Limiter Slope High Setting Register
AC_DAC_DRC_LKl	0x0160	DAC DRC Limiter Slope Low Setting Register
AC_DAC_DRC_HOPL	0x0164	DAC DRC Limiter High Output at Limiter Threshold
AC_DAC_DRC_LOPL	0x0168	DAC DRC Limiter Low Output at Limiter Threshold
AC_DAC_DRC_HET	0x016C	DAC DRC Expander Threshold High Setting Register
AC_DAC_DRC_LET	0x0170	DAC DRC Expander Threshold Low Setting Register

AC_DAC_DRC_HKE	0x0174	DAC DRC Expander Slope High Setting Register
AC_DAC_DRC_LKE	0x0178	DAC DRC Expander Slope Low Setting Register
AC_DAC_DRC_HOPE	0x017C	DAC DRC Expander High Output at Expander Threshold
AC_DAC_DRC_LOPE	0x0180	DAC DRC Expander Low Output at Expander Threshold
AC_DAC_DRC_HKN	0x0184	DAC DRC Linear Slope High Setting Register
AC_DAC_DRC_LKN	0x0188	DAC DRC Linear Slope Low Setting Register
AC_DAC_DRC_SFHAT	0x018C	DAC DRC Smooth Filter Gain High Attack Time Coef Register
AC_DAC_DRC_SFLAT	0x0190	DAC DRC Smooth Filter Gain Low Attack Time Coef Register
AC_DAC_DRC_SFHRT	0x0194	DAC DRC Smooth Filter Gain High Release Time Coef Register
AC_DAC_DRC_SFLRT	0x0198	DAC DRC Smooth Filter Gain Low Release Time Coef Register
AC_DAC_DRC_MXGHS	0x019C	DAC DRC MAX Gain High Setting Register
AC_DAC_DRC_MXGLS	0x01A0	DAC DRC MAX Gain Low Setting Register
AC_DAC_DRC_MNGHS	0x01A4	DAC DRC MIN Gain High Setting Register
AC_DAC_DRC_MNGLS	0x01A8	DAC DRC MIN Gain Low Setting Register
AC_DAC_DRC_EPSHC	0x01AC	DAC DRC Expander Smooth Time High Coef Register
AC_DAC_DRC_EPSLC	0x01B0	DAC DRC Expander Smooth Time Low Coef Register
AC_DAC_DRC_HPFHGAIN	0x01B8	DAC DRC HPF Gain High Coef Register
AC_DAC_DRC_HPFLGAIN	0x01BC	DAC DRC HPF Gain Low Coef Register
Analog Domain Register		
DAC_REG	0x0310	DAC Analog Control Register
MIXER_REG	0x0314	MIXER Analog Control Register
RAMP_REG	0x031C	RAMP Control Register

8.4.6. Register Description

8.4.6.1. 0x0000 DAC Digital Part Control Register(Default Value: 0x0000_0000)

Offset: 0x0000			**Register Name: AC_DAC_DPC**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	EN_DA DAC Digital Part Enable 0: Disable 1: Enable
30:29	/	/	/
28:25	R/W	0x0	MODQU Internal DAC Quantization Levels Levels=[7*(21+MODQU[3:0])]/128 Default levels=7*21/128=1.15
24	R/W	0x0	DWA DWA Function Disable 0: Enable 1: Disable
23:19	/	/	/

18	R/W	0x0	HPF_EN High Pass Filter Enable 0: Disable 1: Enable
17:12	R/W	0x0	DVOL Digital volume control: DVC, ATT=DVC[5:0]*(-1.16dB) 64 steps, -1.16dB/step
11:1	/	/	/
0	R/W	0x0	HUB_EN Audio Hub Enable 0: Disable 1: Enable

8.4.6.2. 0x0010 DAC FIFO Control Register(Default Value: 0x0000_4000)

Offset: 0x0010			**Register Name: AC_DAC_FIFOC**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	R/W	0x0	DAC_FS Sample Rate of DAC 000: 48 kHz 010: 24 kHz 100: 12 kHz 110: 192 kHz 001: 32 kHz 011: 16 kHz 101: 8 kHz 111: 96 kHz 44.1 kHz/22.05 kHz/11.025 kHz can be supported by Audio PLL Configure Bit
28	R/W	0x0	FIR_VER FIR Version 0: 64-Tap FIR 1: 32-Tap FIR
27	/	/	/
26	R/W	0x0	SEND_LASAT Audio sample select when TX FIFO underrun 0: Sending zero 1: Sending last audio sample
25:24	R/W	0x0	FIFO_MODE For 20-bit transmitted audio sample: 00/10: FIFO_I[19:0] = {TXDATA[31:12]} 01/11: FIFO_I[19:0] = {TXDATA[19:0]} For 16-bit transmitted audio sample: 00/10: FIFO_I[19:0] = {TXDATA[31:16], 4'b0} 01/11: FIFO_I[19:0] = {TXDATA[15:0], 4'b0}

23	/	/	/
22:21	R/W	0x0	DAC_DRQ_CLR_CNT When TX FIFO available room is less than or equal N, DRQ Request will be de-asserted. N is defined here: 00: IRQ/DRQ De-asserted when WLEVEL > TXTL 01: 4 10: 8 11: 16
20:15	/	/	/
14:8	R/W	0x40	TX_TRIG_LEVEL TX FIFO Empty Trigger Level (TXTL[12:0]) Interrupt and DMA request trigger level for TX FIFO normal condition. IRQ/DRQ generated when WLEVEL ≤ TXTL **Note: WLEVEL represents the number of valid samples in the TX FIFO.** **Only TXTL[6:0] valid when TXMODE = 0**
7	/	/	/
6	R/W	0x0	DAC_MONO_EN DAC Mono Enable 0: Stereo, 64 levels FIFO 1: mono, 128 levels FIFO When enabled, L & R channel send same data.
5	R/W	0x0	TX_SAMPLE_BITS Transmitting Audio Sample Resolution 0: 16 bits 1: 24 bits
4	R/W	0x0	DAC_DRQ_EN DAC FIFO Empty DRQ Enable 0: Disable 1: Enable
3	R/W	0x0	DAC_IRQ_EN DAC FIFO Empty IRQ Enable 0: Disable 1: Enable
2	R/W	0x0	FIFO_UNDERRUN_IRQ_EN DAC FIFO Underrun IRQ Enable 0: Disable 1: Enable
1	R/W	0x0	FIFO_OVERRUN_IRQ_EN DAC FIFO Overrun IRQ Enable 0: Disable 1: Enable
0	R/WC	0x0	FIFO_FLUSH DAC FIFO Flush Write ‘1’ to flush TX FIFO, self clear to ‘0’

8.4.6.3. 0x0014 DAC FIFO Status Register(Default Value: 0x0080_8008)

Offset: 0x0014			Register Name: AC_DAC_FIFOS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23	R	0x1	TX_EMPTY TX FIFO Empty 0: No room for new sample in TX FIFO 1: More than one room for new sample in TX FIFO (>= 1 word)
22:8	R	0x80	TXE_CNT TX FIFO Empty Space Word Counter
7:4	/	/	/
3	R/W1C	0x1	TXE_INT TX FIFO Empty Pending Interrupt 0: No Pending IRQ 1: FIFO Empty Pending Interrupt Write ‘1’ to clear this interrupt or automatically clear if interrupt condition fails.
2	R/W1C	0x0	TXU_INT TX FIFO Underrun Pending Interrupt 0: No Pending Interrupt 1: FIFO Underrun Pending Interrupt Write ‘1’ to clear this interrupt
1	R/W1C	0x0	TXO_INT TX FIFO Overrun Pending Interrupt 0: No Pending Interrupt 1: FIFO Overrun Pending Interrupt Write ‘1’ to clear this interrupt
0	/	/	/

8.4.6.4. 0x0020 DAC TX DATA Register(Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: AC_DAC_TXDATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	W	0x0	TX_DATA Transmitting left, right channel sample data should be written this register one by one. The left channel sample data is first and then the right channel sample.

8.4.6.5. 0x0024 DAC TX Counter Register(Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: AC_DAC_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	TX_CNT

			TX Sample Counter The audio sample number of sending into TXFIFO. When one sample is put into TXFIFO by DMA or by host IO, the TX sample counter register increases by one. The TX sample counter register can be set to any initial valve at any time. After been updated by the initial value, the counter register should count on base of this initial value. **Note: It is used for Audio/Video Synchronization**

8.4.6.6. 0x0028 DAC Debug Register(Default Value: 0x0000_0000)

Offset: 0x0028			**Register Name: AC_DAC_DG**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11	R/W	0x0	DAC_MODU_SELECT DAC Modulator Debug 0: DAC Modulator Normal Mode 1: DAC Modulator Debug Mode
10:9	R/W	0x0	DAC_PATTERN_SELECT DAC Pattern Select 00: Normal (Audio Sample from TX FIFO) 01: -6 dB Sin wave 10: -60 dB Sin wave 11: Silent wave
8	R/W	0x0	CODEC_CLK_SELECT CODEC Clock Source Select 0: CODEC Clock from PLL 1: CODEC Clock from OSC (for Debug)
7	/	/	/
6	R/W	0x0	DA_SWP DAC Output Channel Swap Enable 0:Disable 1:Enable
5:3	/	/	/
1:0	R/W	0x0	ADDA_LOOP_MODE ADDA Loop Mode Select 00: Disable 01: ADDA LOOP MODE DACL/DACR connect to ADCL/ADCR 10: ADDA LOOP MODE DACL/DACR connect to ADCX/ADCY 11: Reserved

8.4.6.7. 0x00F0 DAC DAP Control Register (Default Value: 0x0000_0000)

Offset: 0x00F0			Register Name: AC_DAC_DAP_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	DDAP_EN DAP for DRC Enable 0: Bypass 1: Enable
30	/	/	/
29	R/W	0x0	DDAP_DRC_EN DRC enable control 0: Disable 1: Enable
28	R/W	0x0	DDAP_HPF_EN HPF enable control 0: Disable 1: Enable
27:0	/	/	/

8.4.6.8. 0x0100 DAC DRC High HPF Coef Register (Default Value: 0x0000_00FF)

Offset: 0x0100			Register Name: AC_DAC_DRC_HHPFC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R/W	0xFF	HPF coefficient setting and the data is 3.24 format.

8.4.6.9. 0x0104 DAC DRC Low HPF Coef Register (Default Value: 0x0000_FAC1)

Offset: 0x0104			Register Name: AC_DAC_DRC_LHPFC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0xFAC1	HPF coefficient setting and the data is 3.24 format.

8.4.6.10. 0x0108 DAC DRC Control Register (Default Value: 0x0000_0080)

Offset: 0x0108			Register Name: AC_DAC_DRC_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15	R	0x0	DRC delay buffer data output state when drc delay function is enabled and the drc funciton is disabled. After disabled drc function and this bit goes to 0, the user should write the drc delay function bit to 0.

			0: Not completed 1: Completed
14:10	/	/	/
13:8	R/W	0x0	Signal delay time setting 6'h00 : (8x1)fs 6'h01 : (8x2)fs 6'h02 : (8x3)fs ---------- 6'h2e : (8*47)fs 6'h2f : (8*48)fs 6'h30 -- 6'h3f : (8*48)fs Delay time = 8*(n+1)fs, n<6'h30; When the delay function is disabled, the signal delay time is unused.
7	R/W	0x1	The delay buffer use or not when the drc is disabled and the drc buffer data output completely. 0: Don't use the buffer 1: Use the buffer
6	R/W	0x0	DRC gain max limit enable 0: Disable 1: Enable
5	R/W	0x0	DRC gain min limit enable When this function is enabled, it will overwrite the noise detect funciton. 0: Disable 1: Enable
4	R/W	0x0	Control the drc to detect noise when ET enable 0: Disable 1: Enable
3	R/W	0x0	Signal function select 0: RMS filter 1: Peak filter When signal function selects Peak filter, the RMS parameter is unused. (AC_DRC_LRMSHAT/AC_DRC_LRMSLAT/AC_DRC_LRMSHAT/AC_DRC_LRMSLAT) When signal function selects RMS filter, the Peak filter parameter is unused.(AC_DRC_LPFHAT/AC_DRC_LPFLAT/AC_DRC_RPFHAT/AC_DRC_RPFLAT /AC_DRC_LPFHRT/AC_DRC_LPFLRT/AC_DRC_RPFHRT/AC_DRC_RPFLRT)
2	R/W	0x0	Delay function enable 0: Disable 1: Enable When the bit is disabled, the signal delay time is unused.
1	R/W	0x0	DRC LT enable 0: Disable 1: Enable When the bit is disabled, Kl and OPL parameter is unused.
0	R/W	0x0	DRC ET enable 0: Disable

			1: Enable When the bit is disabled, Ke and OPE parameter is unused.

8.4.6.11. 0x010C DAC DRC Left Peak Filter High Attack Time Coef Register (Default Value: 0x0000_000B)

Offset: 0x010C			Register Name: AC_DAC_DRC_LPFHAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R/W	0x000B	The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/ta). The format is 3.24.(The default value is 1ms)

8.4.6.12. 0x0110 DAC DRC Left Peak Filter Low Attack Time Coef Register (Default Value: 0x0000_77BF)

Offset: 0x0110			Register Name: AC_DAC_DRC_LPFLAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x77BF	The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/ta). The format is 3.24.(The default value is 1ms)

8.4.6.13. 0x0118 DAC DRC Left Peak Filter Low Attack Time Coef Register (Default Value: 0x0000_77BF)

Offset: 0x0118			Register Name: AC_DAC_DRC_LPFLAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x77BF	The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/ta). The format is 3.24. (The default value is 1ms)

8.4.6.14. 0x011C DAC DRC Left Peak Filter High Release Time Coef Register (Default Value: 0x0000_00FF)

Offset: 0x011C			Register Name: AC_DAC_DRC_LPFHRT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R/W	0x00FF	The left peak filter release time parameter setting, which is determined by the equation that RT = exp(-2.2Ts/tr). The format is 3.24. (The default value is 100ms)

8.4.6.15. 0x0120 DAC DRC Left Peak Filter Low Release Time Coef Register(Default Value: 0x0000_E1F8)

Offset: 0x0120			Register Name: AC_DAC_DRC_LPFLRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xE1F8	The left peak filter release time parameter setting, which is determined by the equation that RT = exp(-2.2Ts/tr). The format is 3.24. (The default value is 100ms)

8.4.6.16. 0x0124 DAC DRC Right Peak Filter High Release Time Coef Register(Default Value: 0x0000_00FF)

Offset: 0x0124			Register Name: AC_DAC_DRC_RPFHRT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x00FF	The left peak filter attack time parameter setting, which determine by the equation that RT = exp(-2.2Ts/tr). The format is 3.24. (The default value is 100ms)

8.4.6.17. 0x0128 DAC DRC Right Peak Filter Low Release Time Coef Register(Default Value: 0x0000_E1F8)

Offset: 0x0128			Register Name: AC_DAC_DRC_RPFLRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xE1F8	The left peak filter release time parameter setting, which determine by the equation that AT = exp(-2.2Ts/tr). The format is 3.24. (The default value is 100ms)

8.4.6.18. 0x012C DAC DRC Left RMS Filter High Coef Register(Default Value: 0x0000_0001)

Offset: 0x012C			Register Name: AC_DAC_DRC_LRMSHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0001	The left RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/tav). The format is 3.24. (The default value is 10ms)

8.4.6.19. 0x0130 DAC DRC Left RMS Filter Low Coef Register(Default Value: 0x0000_2BAF)

Offset: 0x0130			Register Name: AC_DAC_DRC_LRMSLAT
Bit	Read/Write	Default/Hex	Description

31:16	/	/	/
15:0	R/W	0x2BAF	The left RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/tav). The format is 3.24. (The default value is 10ms)

8.4.6.20. 0x0134 DAC DRC Right RMS Filter High Coef Register(Default Value: 0x0000_0001)

Offset: 0x0134			Register Name: AC_DAC_DRC_RRMSHAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R/W	0x0001	The right RMS filter average time parameter setting, which determine by the equation that AT = 1-exp(-2.2Ts/tav). The format is 3.24.(The default value is 10ms)

8.4.6.21. 0x0138 DAC DRC Right RMS Filter Low Coef Register(Default Value: 0x0000_2BAF)

Offset: 0x0138			Register Name: AC_DAC_DRC_RRMSLAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R/W	0x2BAF	The right RMS filter average time parameter setting, which determine by the equation that AT = 1-exp(-2.2Ts/tav). The format is 3.24.(10ms)

8.4.6.22. 0x013C DAC DRC Compressor Theshold High Setting Register(Default Value: 0x0000_06A4)

Offset: 0x013C			Register Name: AC_DAC_DRC_HCT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x06A4	The compressor threshold setting, which is set by the equation that CTin = -CT/6.0206. The format is 8.24 (The default value is -40dB)

8.4.6.23. 0x0140 DAC DRC Compressor Slope High Setting Register(Default Value: 0x0000_D3C0)

Offset: 0x0140			Register Name: AC_DAC_DRC_LCT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0xD3C0	The compressor threshold setting, which is set by the equation that CTin = -CT/6.0206. The format is 8.24 (The default value is -40dB)

8.4.6.24. 0x0144 DAC DRC Compressor Slope High Setting Register(Default Value: 0x0000_0080)

Offset: 0x0144			Register Name: AC_DAC_DRC_HKC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0080	The slope of the compressor, which is determined by the equation that Kc = 1/R, there, R is the ratio of the compressor, which always is interger. The format is 8.24. (The default value is 2 : 1)

8.4.6.25. 0x0148 DAC DRC Compressor Slope Low Setting Register(Default Value: 0x0000_0000)

Offset: 0x0148			Register Name: AC_DAC_DRC_LKC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0000	The slope of the compressor, which is determined by the equation that Kc = 1/R, there, R is the ratio of the compressor, which always is interger. The format is 8.24. (The default value is 2 : 1)

8.4.6.26. 0x014C DAC DRC Compressor High Output at Compressor Threshold Register (Default Value: 0x0000_F95B)

Offset: 0x014C			Register Name: AC_DAC_DRC_HOPC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0xF95B	The output of the compressor, which is determined by the equation -OPC/6.0206. The format is 8.24 (The default value is -40dB)

8.4.6.27. 0x0150 DAC DRC Compressor Low Output at Compressor Threshold Register(Default Value: 0x0000_2C3F)

Offset: 0x0150			Register Name: AC_DAC_DRC_LOPC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x2C3F	The output of the compressor, which is determined by the equation OPC/6.0206. The format is 8.24 (The default value is -40dB)

8.4.6.28. 0x0154 DAC DRC Limiter Theshold High Setting Register(Default Value: 0x0000_01A9)

Offset: 0x0154			Register Name: AC_DAC_DRC_HLT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/

15:0	R/W	0x01A9	The limiter threshold setting, which is set by the equation that LTin = -LT/6.0206, The format is 8.24. (The default value is -10dB)

8.4.6.29. 0x0158 DAC DRC Limiter Theshold Low Setting Register(Default Value: 0x0000_34F0)

Offset: 0x0158			Register Name: AC_DAC_DRC_LLT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x34F0	The limiter threshold setting, which is set by the equation that LTin = -LT/6.0206. The format is 8.24. (The default value is -10dB)

8.4.6.30. 0x015C DAC DRC Limiter Slope High Setting Register(Default Value: 0x0000_0005)

Offset: 0x015C			Register Name: AC_DAC_DRC_HKl
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0005	The slope of the limiter, which is determined by the equation that Kl = 1/R, there, R is the ratio of the limiter, which always is interger. The format is 8.24. (The default value is <50 :1>)

8.4.6.31. 0x0160 DAC DRC Limiter Slope Low Setting Register(Default Value: 0x0000_1EB8)

Offset: 0x0160			Register Name: AC_DAC_DRC_LKl
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x1EB8	The slope of the limiter, which is determined by the equation that Kl = 1/R, there, R is the ratio of the limiter, which always is interger. The format is 8.24. (The default value is <50 :1>)

8.4.6.32. 0x0164 DAC DRC Limiter High Output at Limiter Threshold Register(Default Value: 0x0000_FBD8)

Offset: 0x0164			Register Name: AC_DAC_DRC_HOPL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0xFBD8	The output of the limiter, which is determined by equation OPT/6.0206. The format is 8.24 (The default value is -25dB)

8.4.6.33. 0x0168 DAC DRC Limiter Low Output at Limiter Threshold Register(Default Value: 0x0000_FBA7)

Offset: 0x0168			Register Name: AC_DAC_DRC_LOPL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFBA7	The output of the limiter, which is determined by equation OPT/6.0206. The format is 8.24 (The default value is -25dB)

8.4.6.34. 0x016C DAC DRC Expander Theshold High Setting Register(Default Value: 0x0000_0BA0)

Offset: 0x016C			Register Name: AC_DAC_DRC_HET
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0BA0	The expander threshold setting, which is set by the equation that ETin = -ET/6.0206, The format is 8.24. (The default value is -70dB)

8.4.6.35. 0x0170 DAC DRC Expander Theshold Low Setting Register(Default Value: 0x0000_7291)

Offset: 0x0170			Register Name: AC_DAC_DRC_LET
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x7291	The expander threshold setting, which is set by the equation that ETin = -ET/6.0206, The format is 8.24. (The default value is -70dB)

8.4.6.36. 0x0174 DAC DRC Expander Slope High Setting Register(Default Value: 0x0000_0500)

Offset: 0x0174			Register Name: AC_DAC_DRC_HKE
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13:0	R/W	0x0500	The slope of the expander, which is determined by the equation that Ke = 1/R, there, R is the ratio of the expander, which always is interger and the ke must larger than 50. The format is 8.24. (The default value is <1:5>)

8.4.6.37. 0x0178 DAC DRC Expander Slope Low Setting Register(Default Value: 0x0000_0000)

Offset: 0x0178			Register Name: AC_DAC_DRC_LKE
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0000	The slope of the expander, which is determined by the equation that Ke = 1/R, there, R is the ratio of the expander, which always is interger and the ke must

			larger than 50. The format is 8.24. (The default value is <1:5>)

8.4.6.38. 0x017C DAC DRC Expander High Output at Expander Threshold Register(Default Value: 0x0000_F45F)

Offset: 0x017C			Register Name: AC_DAC_DRC_HOPE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0xF45F	The output of the expander, which is determined by equation OPE/6.0206. The format is 8.24 (The default value is -70dB)

8.4.6.39. 0x0180 DAC DRC Expander Low Output at Expander Threshold Register(Default Value: 0x0000_8D6E)

Offset: 0x0180			Register Name: AC_DAC_DRC_LOPE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x8D6E	The output of the expander, which is determined by equation OPE/6.0206. The format is 8.24 (The default value is -70dB)

8.4.6.40. 0x0184 DAC DRC Linear Slope High Setting Register(Default Value: 0x0000_0100)

Offset: 0x0184			Register Name: AC_DAC_DRC_HKN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0100	The slope of the linear, which is determined by the equation that Kn = 1/R, there, R is the ratio of the linear, which always is interger . The format is 8.24. (The default value is <1:1>)

8.4.6.41. 0x0188 DAC DRC Linear Slope Low Setting Register(Default Value: 0x0000_0000)

Offset: 0x0188			Register Name: AC_DAC_DRC_LKN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0000	The slope of the linear, which is determined by the equation that Kn = 1/R, there, R is the ratio of the linear, which always is interger . The format is 8.24. (The default value is <1:1>)

8.4.6.42. 0x018C DAC DRC Smooth Filter Gain High Attack Time Coef Register(Default Value: 0x0000_0002)

Offset: 0x018C	Register Name: AC_DAC_DRC_SFHAT

Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0002	The smooth filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 5ms)

8.4.6.43. 0x0190 DAC DRC Smooth Filter Gain Low Attack Time Coef Register(Default Value: 0x0000_5600)

Offset: 0x0190			Register Name: AC_DAC_DRC_SFLAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x5600	The smooth filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 5ms)

8.4.6.44. 0x0194 DAC DRC Smooth Filter Gain High Release Time Coef Register(Default Value: 0x0000_0000)

Offset: 0x0194			Register Name: AC_DAC_DRC_SFHRT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R/W	0x0000	The gain smooth filter release time parameter setting, which is determined by the equation that RT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 200ms)

8.4.6.45. 0x0198 DAC DRC Smooth Filter Gain Low Release Time Coef Register(Default Value: 0x0000_0F04)

Offset: 0x0198			Register Name: AC_DAC_DRC_SFLRT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0F04	The gain smooth filter release time parameter setting, which is determined by the equation that RT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 200ms)

8.4.6.46. 0x019C DAC DRC MAX Gain High Setting Register(Default Value: 0x0000_FE56)

Offset: 0x019C			Register Name: AC_DAC_DRC_MXGHS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0xFE56	The max gain setting, which is determined by equation MXG/6.0206. The format is 8.24 and must -20dB <MXG< 30dB (The default value is -10dB)

8.4.6.47. 0x01A0 DAC DRC MAX Gain Low Setting Register(Default Value: 0x0000_CB0F)

Offset: 0x01A0			Register Name: AC_DAC_DRC_MXGLS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xCB0F	The max gain setting, which is determined by equation MXG/6.0206. The format is 8.24 and must -20dB <MXG < 30dB (The default value is -10dB)

8.4.6.48. 0x01A4 DAC DRC MIN Gain High Setting Register(Default Value: 0x0000_F95B)

Offset: 0x01A4			Register Name: AC_DAC_DRC_MNGHS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xF95B	The min gain setting, which is determined by equation MXG/6.0206. The format is 8.24 and must -60dB ≤MNG ≤ -40dB (The default value is -40dB)

8.4.6.49. 0x01A8 DAC DRC MIN Gain Low Setting Register(Default Value: 0x0000_2C3F)

Offset: 0x01A8			Register Name: AC_DAC_DRC_MNGLS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2C3F	The min gain setting, which is determined by equation MNG/6.0206. The format is 8.24 and must -60dB ≤MNG ≤ -40dB (The default value is -40dB)

8.4.6.50. 0x01AC DAC DRC Expander Smooth Time High Coef Register(Default Value: 0x0000_0000)

Offset: 0x01AC			Register Name: AC_DAC_DRC_EPSHC
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0000	The gain smooth filter release and attack time parameter setting in expander region, which are determined by the equation that RT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 30ms)

8.4.6.51. 0x01B0 DAC DRC Expander Smooth Time Low Coef Register(Default Value: 0x0000_640C)

Offset: 0x01B0			Register Name: AC_DAC_DRC_EPSLC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

15:0	R/W	0x640C	The gain smooth filter release and attack time parameter setting in expander region, which are determined by the equation that RT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 30ms)

8.4.6.52. 0x01B8 DAC DRC HPF Gain High Coef Register(Default Value: 0x0000_0100)

Offset: 0x01B8			Register Name: AC_DAC_DRC_HPFHGAIN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R/W	0x100	The gain of the hpf coefficient setting which format is 3.24.(gain = 1)

8.4.6.53. 0x01BC DAC DRC HPF Gain Low Coef Register(Default Value: 0x0000_0000)

Offset: 0x01BC			Register Name: AC_DAC_DRC_HPFLGAIN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0000	The gain of the hpf coefficient setting which format is 3.24.(gain = 1)

8.4.6.54. 0x0310 DAC Analog Control Register(Default Value: 0x0015_0000)

Offset: 0x0310			Register Name: DAC_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23	R/W	0x0	CURRENT_TEST_SELECT Internal Current Sink Test Enable 0: Normal 1: For Debug
22	/	/	/
21:20	R/W	0x1	IOPVRS VRA2 Buffer OP Bias Current Select 00: 6uA 01: 7uA 10: 8uA 11: 9uA
19:18	R/W	0x1	ILINEOUTAMPS LINEOUTL/R AMP Bias Current Select 00: 6uA 01: 7uA 10: 8uA 11: 9uA
17:16	R/W	0x1	IOPDACS

			OPDAC Bias Current Select 00: 7uA 01: 8uA 10: 9uA 11: 10uA
15	R/W	0x0	DACLEN DACL Enable 0: Disable 1: Enable
14	R/W	0x0	DACREN DACR Enable 0: Disable 1: Enable
13	R/W	0x0	LINEOUTLEN Left Channel LINEOUT Enable 0: Disable 1: Enable
12	R/W	0x0	LMUTE Left MIXER to Left Channel LINEOUT Mute Control 0: Mute 1: Not mute
11	R/W	0x0	LINEOUTREN Right Channel LINEOUT Enable 0: Disable 1: Enable
10	R/W	0x0	RMUTE Right MIXER to Right Channel LINEOUT Mute Control 0: Mute 1: Not mute
9	R/W	0x0	RSWITCH 0: OUTPUT of RAMP_DAC 1: VRA1
8	R/W	0x0	RAMPEN Ramp DAC Enable 0: Disable 1: Enable
7	/	/	/
6	R/W	0x0	Left LIENOUT Source Select 0: Left Output Mixer 1: Left Output Mixer + Right Output Mixer
5	R/W	0x0	Right LIENOUT Source Select 0: Right Output Mixer 1: Left LINEOUT, for Differential Output
4:0	R/W	0x0	LINEOUT Volume Control, Total 30 level from 0x1F to 0x02 with the volume 0dB to -43.5dB, -1.5dB/step, mute when 00000 & 00001.

8.4.6.55. 0x0314 MIXER Analog Control Register(Default Value:0x0000_0133)

Offset: 0x0314			Register Name: MIXER_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:20	R/W	0x0	LMIXMUTE Left Output Mixer Mute Control 0:Mute, 1:Not Mute Bit 23: Reserved Bit 22: Reserved Bit 21: Left Channel DAC Bit 20: Right Channel DAC
19:16	R/W	0x0	RMIXMUTE Right Output Mixer Mute Control 0:Mute, 1:Not Mute Bit 19: Reserved Bit 18: Reserved Bit 17: Right Channel DAC Bit 16: Left Channel DAC
15:12	/	/	/
11	R/W	0x0	LMIXEN Left Analog Output Mixer Enable 0: Disable 1: Enable
10	R/W	0x0	RMIXEN Right Analog Output Mixer Enable 0: Disable 1: Enable
9:8	R/W	0x1	IOPMIXS OPMIX/OPLPF Bias Current Select 00: 6uA 01: 7uA 10: 8uA 11: 9uA
7	/	/	/
6:4	R/W	0x3	Reserved
3	/	/	/
2:0	R/W	0x3	Reserved

8.4.6.56. 0x031C RAMP Control Register(Default Value:0x0000_0000)

Offset: 0x031C	Register Name: RAMP_REG

Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:4	R/W	0x0	RS Ramp Step 000: 3us 001: 6us 010: 12us 011: 24us 100: 48us 101: 60us 110: 96us 111: 120us
3	R/W	0x0	RMDEN Ramp Manual Down Enable 0: Disable 1: Enable
2	R/W	0x0	RMUEN Ramp Manual Up Enable 0: Disable 1: Enable
1	R/W	0x0	RMCEN Ramp Manual Control Enable 0: Disable 1: Enable
0	R/W	0x0	RDEN Ramp Digital Enable 0: Disable 1: Enable

Contents

Figures

Tables

Chapter 9 Interfaces

9.1. TWI

9.1.1. Overview

The TWI is designed as an interface between CPU host and the serial TWI bus. It can support all the standard TWI transfer, including slave and master. The communication of the 2-wire bus is carried out by a byte-wise mode based on interrupt or polled handshaking. The TWI can be operated in standard mode (100 kbit/s) or fast-mode (400 kbit/s). The 10-bit addressing mode is supported for this specified application. General call addressing is also supported in slave mode.

The TWI has the following features:

- Software-programmable for slave or master
- Supports repeated START signal
- Allows 10-bit addressing with TWI bus
- Performs arbitration and clock synchronization
- Own address and general call address detection
- Interrupt on address detection
- Supports speed up to 400 kbit/s (‘fast mode’)
- Allows operation from a wide range of input clock frequency
- TWI driver supports packet transmission and DMA when TWI works in master mode

9.1.2. Block Diagram

Figure 9-1 shows the block diagram of TWI.

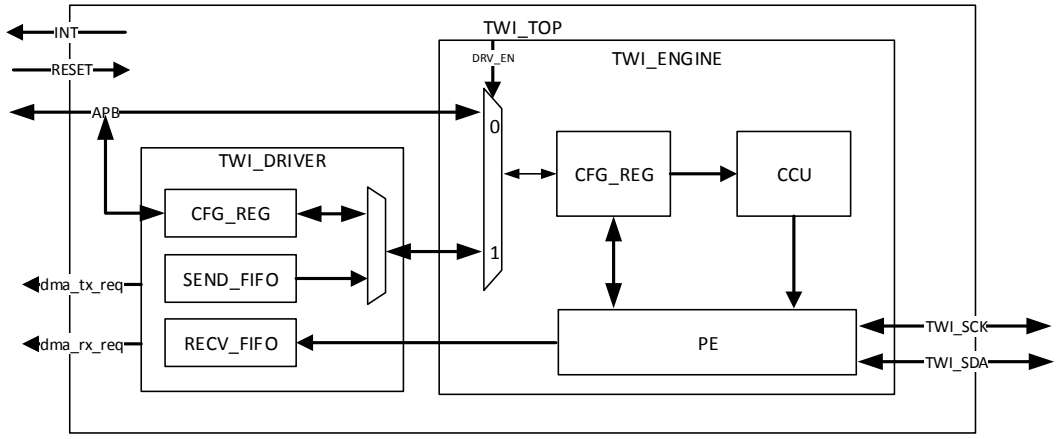

Figure 9- 1. TWI Block Diagram

RESET: Module reset signal

INT: Module output interrupt signal

CFG_REG: Module configuration register in TWI

PE: Packet encoding/decoding

CCU: Module clock controller unit

9.1.3. Operations and Functional Descriptions

9.1.3.1. External Signals

The TWI controller has 6 TWIs. Table 9-1 describes the external signals of TWI. TWI_SCK and TWI_SDA are bidirectional I/O, when TWI is configured as master device, TWI-SCK is output pin; when TWI is configurable as slave device, TWI-SCK is input pin. The unused TWI ports are used as General Purpose I/O ports. For information about General Purpose I/O ports, see **Port Controller** in chapter9.

Table 9- 1. TWI External Signals

Signal	Description	Type
TWI0_SCK	TWI0 Clock Signal	I/O,OD
TWI0_SDA	TWI0 Serial Data	I/O,OD
TWI1_SCK	TWI1 Clock Signal	I/O,OD
TWI1_SDA	TWI1 Serial Data	I/O,OD
TWI2_SCK	TWI2 Clock Signal	I/O,OD
TWI2_SDA	TWI2 Serial Data	I/O,OD
TWI3_SCK	TWI3 Clock Signal	I/O,OD
TWI3_SDA	TWI3 Serial Data	I/O,OD
TWI4_SCK	TWI4 Clock Signal	I/O,OD
TWI4_SDA	TWI4 Serial Data	I/O,OD
S_TWI0_SCK	S_TWI0 Clock Signal	I/O,OD
S_TWI0_SDA	S_TWI0 Serial Data	I/O,OD

9.1.3.2. Clock Sources

Each TWI controller has a fixed clock source. Table 9-2 describes the clock source for TWI. Users can see **Clock Controller Unit(CCU)** in chapter3 and **Power Reset Clock Management(PRCM)** for clock setting, configuration and gating information.

Table 9- 2. TWI Clock Sources

Clock Sources	Description
APB2_CLK	TWI0/1/2/3/4 clock source, for details on APB2 refer to CCU
APBS2_CLK	S_TWI0 clock source, for details on APBS refer to PRCM

After selected a proper clock, for using the TWI0/1/2/3/4, user must open the gating of TWI and release the reset bit. For using the S_TWI0, user also needs to open the gating of R-TWI and release the reset bit .

For more details on the gating/reset register, see CCU and PRCM specification**.**

9.1.3.3. Write/Read Timing in Standard and Extended Address Mode

Figure 9-2 describes the write timing in 7-bit standard address mode.

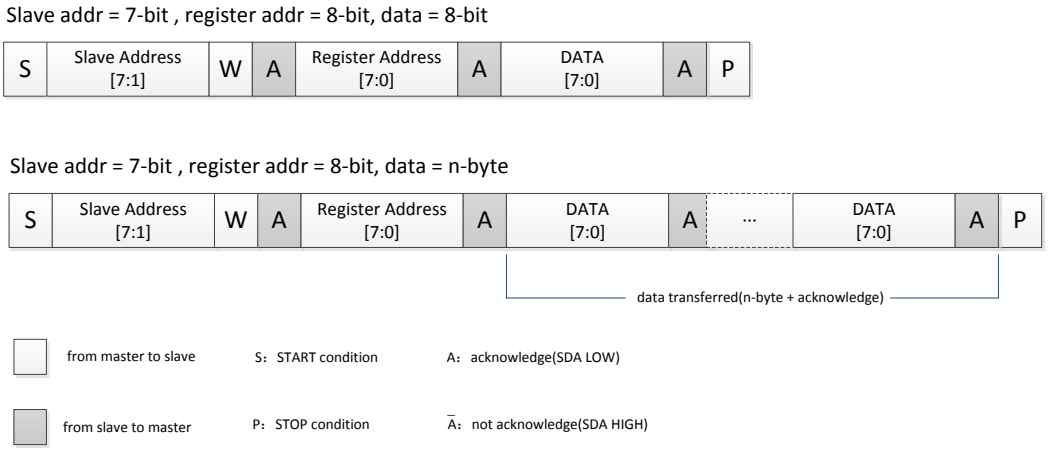

Figure 9- 2. 7-bit Standard Address Write Timing

Figure 9-3 describes the read timing in 7-bit standard address mode.

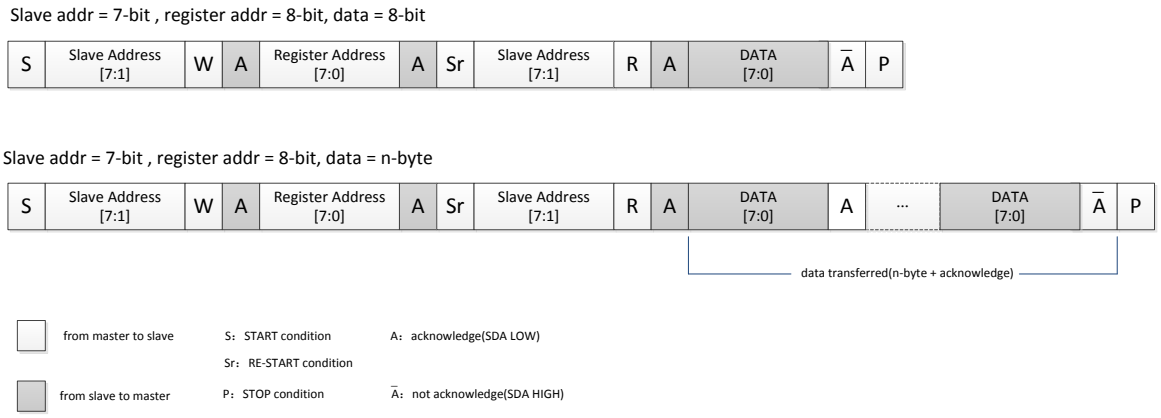

Figure 9- 3. 7-bit Standard Address Read Timing

Figure 9-4 describes the write timing in 10-bit extended address mode.

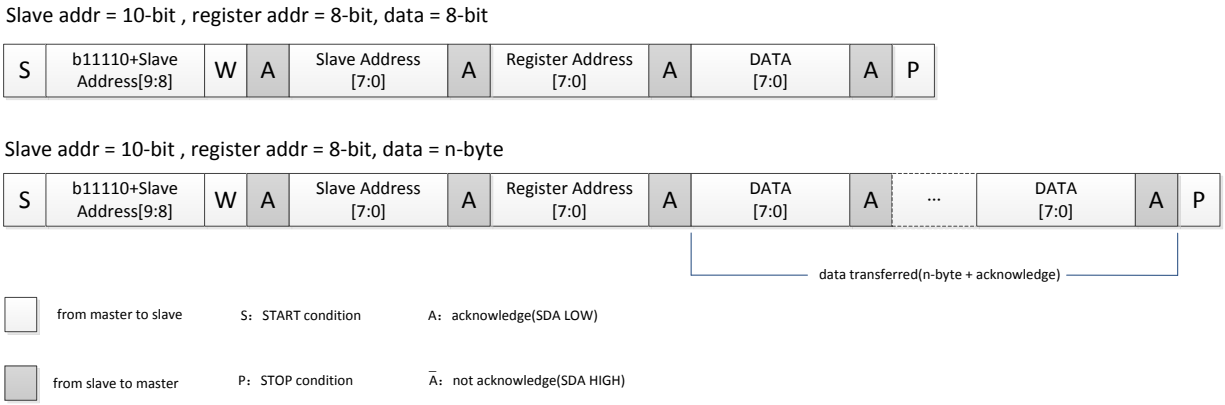

Figure 9- 4. 10-bit Extended Address Write Timing

Figure 9-5 describes the read timing in 10-bit extended address mode.

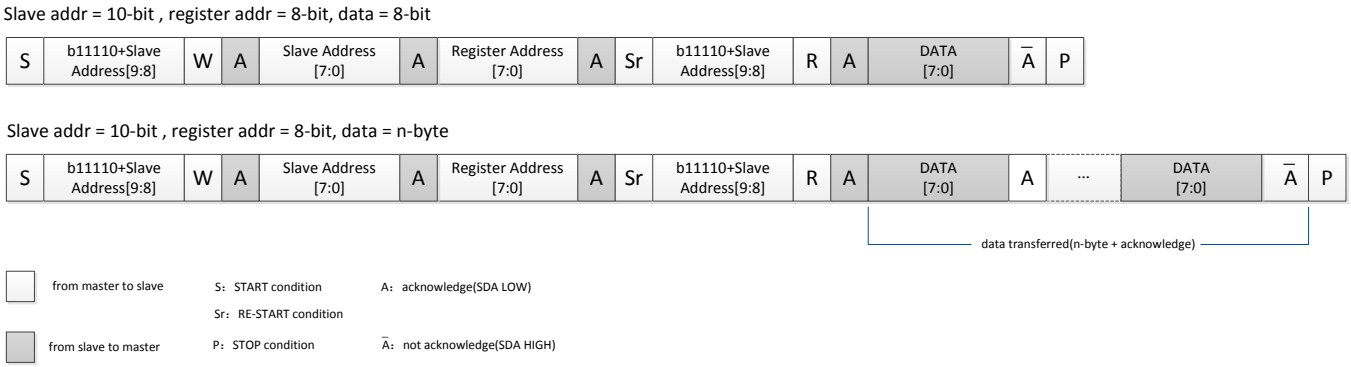

Figure 9- 5. 10-bit Extended Address Read Timing

9.1.3.4. Programming State Diagram

Figure 9-6 shows the TWI programming state diagram. For the value between two states, see TWI_STAT register in section 9.1.6.5.

M_SEND_S: master sends START signal;

M_SEND_ADDR: master sends slave address;

M_SEND_XADD: master sends slave extended address;

M_SEND_SR: master repeated start;

M_SEND_DATA: master sends data;

M_SEND_P: master sends STOP signal;

M_RECV_DATA: master receives data;

ARB_LOST: Arbitration lost;

C_IDLE: Idle;

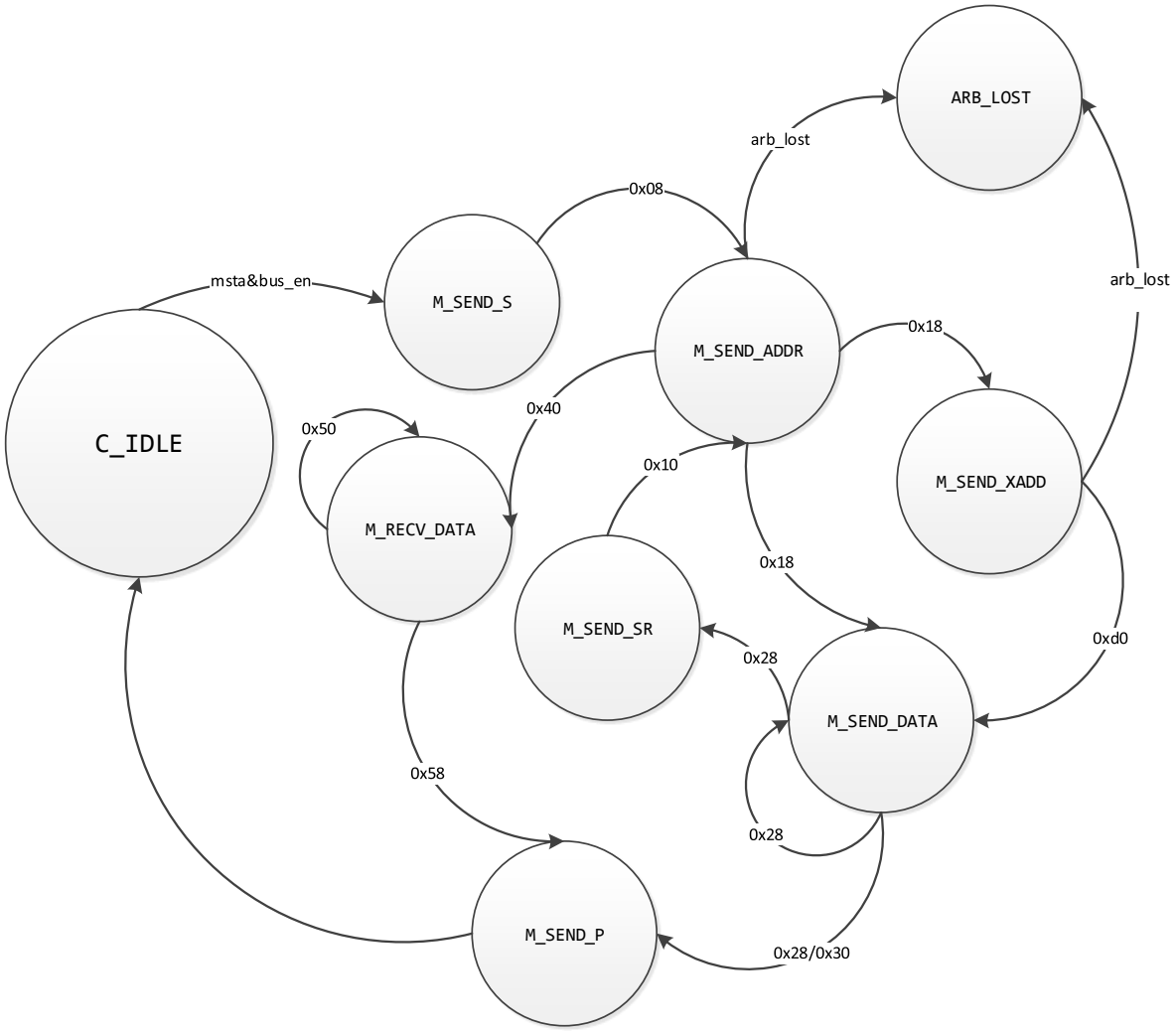

Figure 9- 6. TWI Programming State Diagram

9.1.3.5. TWI Engine Master and Slave Mode

There are four operation modes on the TWI bus. They are Master Transmit, Master Receive, Slave Transmit and Slave

Receive. In general, CPU host controls TWI engine by writing command and data to its registers. TWI engine transmits an interrupt to CPU when each time a byte transfer is done or a START/STOP command is detected. The CPU host can also poll the status register for current status if the interrupt mechanism is not disabled by the CPU host.

When the CPU host wants to start a bus transfer, it initiates a bus START to enter the master mode by setting IM_STA bit of the TWI_CNTR register to high (before it must be low). The TWI engine will assert INT line and INT_FLAG to indicate a completion for the START command and each consequent byte transfer. At each interrupt, the micro-processor needs to check the TWI_STAT register for current status. A transfer has to be concluded with STOP command by setting M_STP bit to high.

In Slave mode, the TWI engine also constantly samples the bus and look for its own slave address during addressing cycles. Once a match is found, it is addressed and interrupt the CPU host with the corresponding status. Upon request, the CPU host should read the status, read/write TWI_DATA data register, and set the TWI_CNTR control register. After each byte transfer, a slave device always stop the operation of remote master by holding the next low pulse on SCL line until the microprocessor responds to the status of previous byte transfer or START command.

9.1.4. Programming Guidelines

The TWI controller operates in 8-bit data format. The data on the TWI_SDA line is always 8 bits long. At first, the TWI controller will sent a start condition. When in the addressing formats of 7-bit, TWI sends out one 8 bits message which includes 7 MSB slave address and 1 LSB read/write flag. The least significant of the salve address indicates the direction of transmission. When TWI works in 10-bit slave address mode, the operation will be divided into two steps, for details on the operation, see the register description in Section 9.1.6.1 and 9.1.6.2.

9.1.4.1. Initialization

To initialize the TWI, perform the following steps:

Step1 Configure corresponding GPIO multiplex function as TWI mode.

Step2 For TWIx, set TWI_BGR_REG[TWIx_GATING] in CCU module to 0 to close TWIx clock;
For S_TWIx, set R_TWI_BGR_REG[R_TWIx_GATING] in PRCM module to 0 to close R_TWIx clock.

Step3 For TWIx, set TWI_BGR_REG[TWIx_RST] in CCU module to 0, then set to 1 to reset TWIx;
For S_TWIx, set R_TWI_BGR_REG[R_TWIx_RST] in PRCM module to 0, then set to 1 to reset R_TWIx.

Step4 For TWIx, set TWI_BGR_REG[TWIx_GATING] in CCU module to 1 to open TWIx clock;
For S_TWIx, set R_TWI_BGR_REG[R_TWIx_GATING] in PRCM module to 1 to open R_TWIx clock.

Step5 Configure TWI_CCR[CLK_M] and TWI_CCR[CLK_N] to get the needed rate(The clock source of TWI is from APB2 or APBS2).

Step6 Configure TWI_CNTR[BUS_EN] and TWI_CNTR[A_ACK], when using interrupt, set TWI_CNTR[BUS_EN] to 1, and register system interrupt through GIC module. In slave mode, configure TWI_ADDR and TWI_XADDR registers to finish TWI initialization configuration.

For PRCM, see the description in ***H616_PRCM_Specification***.

Figure 9-7 shows the process of TWI initialization.

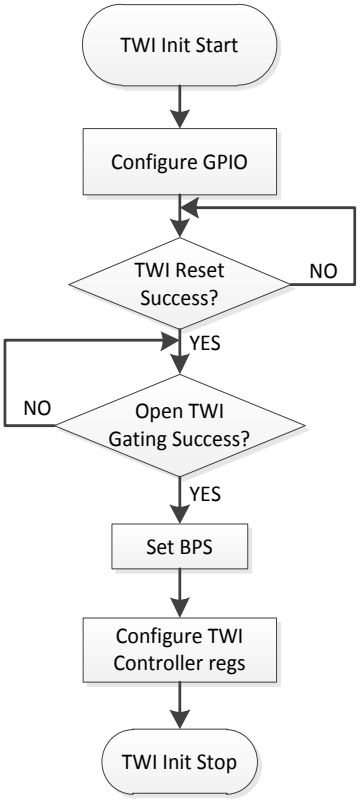

Figure 9- 7. TWI Initialization Process

9.1.4.2. Data Write Operation

To write data to device, perform the following steps:

Step1 Clear TWI_EFR register, and configure TWI_CNTR[M_STA] to 1 to transmit START signal.

Step2 After START signal is transmitted, the first interrupt is triggered, then write device ID to TWI_DATA(For 10-bit device ID, firstly write the first byte ID, secondly write the second byte ID in the next interrupt).

Step3 Interrupt is triggered again after device ID transmission completes, write device data address to be read to TWI_DATA(For 16-bit address, firstly write the first byte address, secondly write the second byte address).

Step4 Interrupt is triggered after data address transmission completes, write data to be transmitted to TWI_DATA(For consecutive write data operation, every byte transmission completion triggers interrupt, during interrupt write the next byte data to TWI_DATA).

Step5 After transmission completes, write TWI_CNTR[M_STP] to 1 to transmit STOP signal and end this write-operation.

Figure 9-8 shows the process of TWI write to device.

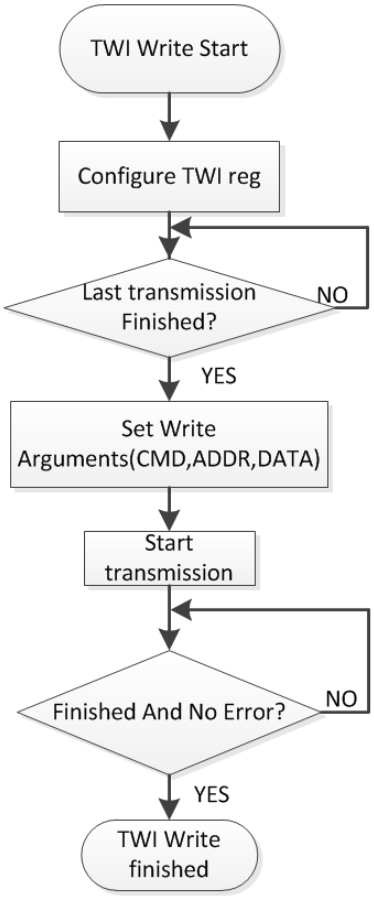

Figure 9- 8. TWI Write Data Process

9.1.4.3. Data Read Operation

To read data from device, perform the following steps:

Step1 Clear TWI_EFR register, set TWI_CNTR[A_ACK] to 1, and configure TWI_CNTR[M_STA] to 1 to transmit START signal.

Step2 After START signal is transmitted, the first interrupt is triggered, then write device ID to TWI_DATA(For 10-bit device ID, firstly write the first byte ID, secondly write the second byte ID in the next interrupt).

Step3 Interrupt is triggered again after device ID transmission completes, write device data address to be read to TWI_DATA(For 16-bit address, firstly write the first byte address, secondly write the second byte address).

Step4 Interrupt is triggered after data address transmission completes, write TWI_CNTR[M_STA] to 1 to transmit new START signal, and after interrupt triggers, write device ID to TWI_DATA to start read-operation.

Step5 After device address transmission completes, each receive completion will trigger interrupt, in turn, read TWI_DATA to get data, when receiving the previous interrupt of the last byte data, clear TWI_CNTR[A_ACK] to stop acknowledge signal of the last byte.

Step6 Write TWI_CNTR[M_STP] to 1 to transmit STOP signal and end this read-operation.

Figure 9-9 shows the process of TWI read from device.

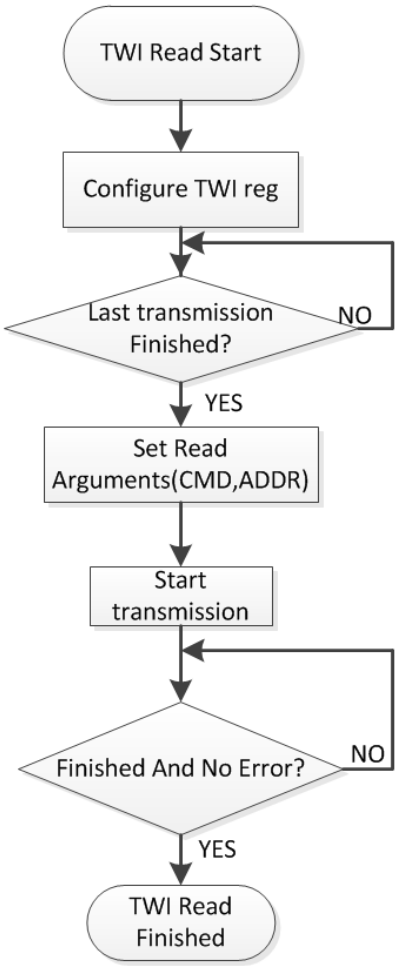

Figure 9- 9. TWI Read Data Process

9.1.4.4. Packet Transmission Operation

Figure 9-10 shows a software operation flow for packet transmission by TWI driver.

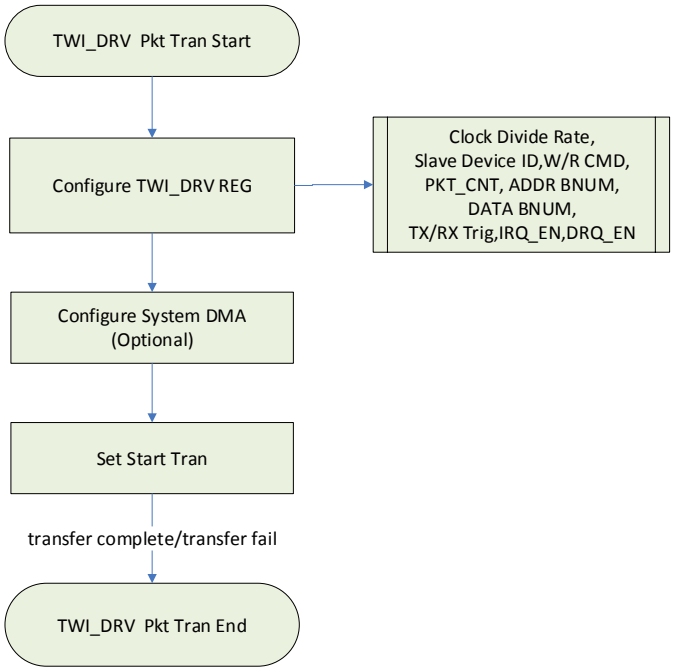

Figure 9- 10. TWI Driver Packet Transmission Process

9.1.5. Register List

Module Name	Base Address
TWI0	0x05002000
TWI1	0x05002400
TWI2	0x05002800
TWI3	0x05002C00
TWI4	0x05003000
R-TWI0	0x07081400

Register Name	Offset	Description
TWI_ADDR	0x0000	TWI Slave Address
TWI_XADDR	0x0004	TWI Extended Slave Address
TWI_DATA	0x0008	TWI Data Byte
TWI_CNTR	0x000C	TWI Control Register
TWI_STAT	0x0010	TWI Status Register
TWI_CCR	0x0014	TWI Clock Control Register
TWI_SRST	0x0018	TWI Software Reset
TWI_EFR	0x001C	TWI Enhance Feature Register
TWI_LCR	0x0020	TWI Line Control Register
TWI_DRV_CTRL	0x0200	TWI_DRV Control Register

TWI_DRV_CFG	0x0204	TWI_DRV Transmission Configuration Register
TWI_DRV_SLV	0x0208	TWI_DRV Slave ID Register
TWI_DRV_FMT	0x020C	TWI_DRV Packet Format Register
TWI_DRV_BUS_CTRL	0x0210	TWI_DRV Bus Control Register
TWI_DRV_INT_CTRL	0x0214	TWI_DRV Interrupt Control Register
TWI_DRV_DMA_CFG	0x0218	TWI_DRV DMA Configure Register
TWI_DRV_FIFO_CON	0x021C	TWI_DRV FIFO Content Register
TWI_DRV_SEND_FIFO_ACC	0x0300	TWI_DRV Send Data FIFO Access Register
TWI_DRV_RECV_FIFO_ACC	0x0304	TWI_DRV Receive Data FIFO Access Register

9.1.6. Register Description

9.1.6.1. 0x0000 TWI Slave Address Register(Default Value:0x0000_0000)

Offset: 0x0000			**Register Name: TWI_ADDR**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:1	R/W	0x0	SLA Slave Address 7-bit addressing: SLA6, SLA5, SLA4, SLA3, SLA2, SLA1, SLA0 10-bit addressing: 1, 1, 1, 1, 0, SLAX[9:8]
0	R/W	0x0	GCE General Call Address Enable 0: Disable 1: Enable

NOTE

For 7-bit addressing:

SLA6 – SLA0 is the 7-bit address of the TWI in slave mode. When the TWI receives this address after a START condition, it will generate an interrupt and enter slave mode. (SLA6 corresponds to the first bit received from the TWI bus.) If GCE is set to '1', the TWI will also recognize the general call address (00h).

For 10-bit addressing:

When the address received starts with 11110b, the TWI recognizes this as the first part of a 10-bit address and if the next two bits match ADDR[2:1] (i.e. SLAX9 and SLAX8 of the device's extended address), it sends an ACK. (The device does not generate an interrupt at this point.) If the next byte of the address matches the XADDR register (SLAX7 – SLAX0), the TWI generates an interrupt and goes into slave mode.

9.1.6.2. 0x0004 TWI Extend Address Register(Default Value:0x0000_0000)

Offset: 0x0004			Register Name: TWI_XADDR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	SLAX Extend Slave Address SLAX[7:0]

9.1.6.3. 0x0008 TWI Data Register(Default Value:0x0000_0000)

Offset: 0x0008			Register Name: TWI_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	TWI_DATA Data byte transmitted or received

9.1.6.4. 0x000C TWI Control Register(Default Value:0x0000_0000)

Offset: 0x000C			Register Name: TWI_CNTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	INT_EN Interrupt Enable 0: The interrupt line always low 1: The interrupt line will go high when INT_FLAG is set.
6	R/W	0x0	BUS_EN TWI Bus Enable 0: The TWI bus ISDA/ISCL is ignored and the TWI Controller will not respond to any address on the bus 1: The TWI will respond to call to its slave address – and to the general call address if the GCE bit in the ADDR register is set. **Note: In master operation mode, this bit should be set to '1'.**
5	R/WAC	0x0	M_STA Master Mode Start When M_STA is set to '1', TWI Controller enters master mode and will transmit a START condition on the bus when the bus is free. If the M_STA bit is set to '1' when the TWI Controller is already in master mode and one or more bytes have been transmitted, then a repeated START condition will be sent. If the M_STA bit is set to '1' when the TWI is being accessed in slave mode, the TWI will complete the data transfer in slave mode then enter master mode when the bus has been released.

			The M_STA bit is cleared automatically after a START condition has been sent. Writing a '0' to this bit has no effect.
4	R/W1C	0x0	M_STP Master Mode Stop If M_STP is set to '1' in master mode, a STOP condition is transmitted on the TWI bus. If the M_STP bit is set to '1' in slave mode, the TWI will indicate if a STOP condition has been received, but no STOP condition will be transmitted on the TWI bus. If both M_STA and M_STP bits are set, the TWI will first transmit the STOP condition (if in master mode) then transmit the START condition. The M_STP bit is cleared automatically: writing a '0' to this bit has no effect.
3	R/W1C	0x0	INT_FLAG Interrupt Flag INT_FLAG is automatically set to '1' when any of 28 (out of the possible 29) states is entered (see 'STAT Register' below). The only state that does not set INT_FLAG is state F8h. If the INT_EN bit is set, the interrupt line goes high when IFLG is set to '1'. If the TWI is operating in slave mode, data transfer is suspended when INT_FLAG is set and the low period of the TWI bus clock line (SCL) is stretched until '1' is written to INT_FLAG. The TWI clock line is then released and the interrupt line goes low.
2	R/W	0x0	A_ACK Assert Acknowledge When A_ACK is set to '1', an Acknowledge (low level on SDA) will be sent during the acknowledge clock pulse on the TWI bus if: (1). Either the whole of a matching 7-bit slave address or the first or the second byte of a matching 10-bit slave address has been received. (2). The general call address has been received and the GCE bit in the ADDR register is set to '1'. (3). A data byte has been received in master or slave mode. When A_ACK is '0', a Not Acknowledge (high level on SDA) will be sent when a data byte is received in master or slave mode. If A_ACK is cleared to '0' in slave transmitter mode, the byte in the DATA register is assumed to be the 'last byte'. After this byte has been transmitted, the TWI will enter state C8h then return to the idle state (status code F8h) when INT_FLAG is cleared. The TWI will not respond as a slave unless A_ACK is set.
1	/	/	/
0	R/W	0x0	CLK_COUNT_MODE 0: scl clock high period count on oscl 1: scl clock high period count on iscl

9.1.6.5. 0x0010 TWI Status Register(Default Value:0x0000_00F8)

Offset: 0x0010			Register Name: TWI_STAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R	0xF8	STA Status Information Byte **Code Status** 0x00: Bus error 0x08: START condition transmitted 0x10: Repeated START condition transmitted 0x18: Address + Write bit transmitted, ACK received 0x20: Address + Write bit transmitted, ACK not received 0x28: Data byte transmitted in master mode, ACK received 0x30: Data byte transmitted in master mode, ACK not received 0x38: Arbitration lost in address or data byte 0x40: Address + Read bit transmitted, ACK received 0x48: Address + Read bit transmitted, ACK not received 0x50: Data byte received in master mode, ACK transmitted 0x58: Data byte received in master mode, not ACK transmitted 0x60: Slave address + Write bit received, ACK transmitted 0x68: Arbitration lost in address as master, slave address + Write bit received, ACK transmitted 0x70: General Call address received, ACK transmitted 0x78: Arbitration lost in address as master, General Call address received, ACK transmitted 0x80: Data byte received after slave address received, ACK transmitted 0x88: Data byte received after slave address received, not ACK transmitted 0x90: Data byte received after General Call received, ACK transmitted 0x98: Data byte received after General Call received, not ACK transmitted 0xA0: STOP or repeated START condition received in slave mode 0xA8: Slave address + Read bit received, ACK transmitted 0xB0: Arbitration lost in address as master, slave address + Read bit received, ACK transmitted 0xB8: Data byte transmitted in slave mode, ACK received 0xC0: Data byte transmitted in slave mode, ACK not received 0xC8: Last byte transmitted in slave mode, ACK received 0xD0: Second Address byte + Write bit transmitted, ACK received 0xD8: Second Address byte + Write bit transmitted, ACK not received 0xF8: No relevant status information, INT_FLAG=0 Others: Reserved

9.1.6.6. 0x0014 TWI Clock Register(Default Value:0x0000_0080)

Offset: 0x0014			Register Name: TWI_CCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x1	CLK_DUTY Setting duty cycle of Clock as Master 0: 50% 1: 40%
6:3	R/W	0x0	CLK_M
2:0	R/W	0x0	CLK_N The TWI bus is sampled by the TWI at the frequency defined by F0: Fsamp = F0 = Fin / 2^CLK_N The TWI OSCL output frequency, in master mode, is F1 / 10: F1 = F0 / (CLK_M + 1) Foscl = F1 / 10 = Fin / (2^CLK_N * (CLK_M + 1)*10) Specially, Foscl = F1/11 when CLK_M=0 and CLK_DUTY=40% due to the delay of SCL sample debounce. For Example： Fin = 24 MHz (APB clock input) For 400 kHz full speed 2Wire, CLK_N = 1, CLK_M=2 F0 = 24 MHz/2^1=12 MHz, F1= F0/(10*(2+1)) = 0.4 MHz For 100 kHz standard speed 2Wire, CLK_N=1, CLK_M=11 F0=24 MHz/2^1=12 MHz, F1=F0/(10*(11+1)) = 0.1 MHz

9.1.6.7. 0x0018 TWI Soft Reset Register(Default Value:0x0000_0000)

Offset: 0x0018			Register Name: TWI_SRST
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/WAC	0x0	SOFT_RST Soft Reset Write ‘1’ to this bit to reset the TWI and clear to ‘0’ when completing Soft Reset operation.

9.1.6.8. 0x001C TWI Enhance Feature Register(Default Value:0x0000_0000)

Offset: 0x001C			Register Name: TWI_EFR
Bit	**Read/Write**	**Default/Hex**	**Description**

31:2	/	/	/
1:0	R/W	0x0	DBN Data Byte Number Follow Read Command Control 00 : No data byte can be written after read command 01 : Only 1 byte data can be written after read command 10 : 2 bytes data can be written after read command 11 : 3 bytes data can be written after read command

9.1.6.9. 0x0020 TWI Line Control Register(Default Value:0x0000_003A)

Offset: 0x0020			Register Name: TWI_LCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:6	/	/	/
5	R	0x1	SCL_STATE Current State of TWI_SCL 0 : Low 1 : High
4	R	0x1	SDA_STATE Current State of TWI_SDA 0 : Low 1 : High
3	R/W	0x1	SCL_CTL TWI_SCL Line State Control Bit When line control mode is enabled (bit[2] set), this bit decides the output level of TWI_SCL. 0 : Output low level 1 : Output high level
2	R/W	0x0	SCL_CTL_EN TWI_SCL Line State Control Enable When this bit is set, the state of TWI_SCL is controlled by the value of bit[3]. 0 : Disable TWI_SCL line control mode 1 : Enable TWI_SCL line control mode
1	R/W	0x1	SDA_CTL TWI_SDA Line State Control Bit When line control mode is enabled (bit[0] set), this bit decides the output level of TWI_SDA. 0 : Output low level 1 : Output high level
0	R/W	0x0	SDA_CTL_EN TWI_SDA Line State Control Enable When this bit is set, the state of TWI_SDA is controlled by the value of bit[1]. 0 : Disable TWI_SDA line control mode 1 : Enable TWI_SDA line control mode

9.1.6.10. 0x0200 TWI_DRV Control Register(Default Value:0x00F8_1000)

Offset: 0x0200			Register Name: TWI_DRV_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/WAC	0x0	START_TRAN 0: Transmission idle 1: Start transmission Automatically cleared to '0' when finished. If slave not respond for the expected status over the time defined by TIMEOUT, current transmission will stop. All format setting and data will be loaded from registers and FIFO when transmission start.
30	/	/	/
29	R/W	0x0	RESTART_MODE 0: RESTART 1: STOP+START Define the TWI_DRV action after sending register address.
28	R/W	0x0	READ_TRAN_MODE 0: send slave_id+W 1: do not send slave_id+W **Setting this bit to 1 if reading from a slave which register width is equal to 0**.
27:24	R	0x0	TRAN_RESULT 000: OK 001: FAIL Other: Reserved
23:16	R	0xf8	TWI_STA 0x00: bus error 0x08: START condition transmitted 0x10: Repeated START condition transmitted 0x18: Address + Write bit transmitted, ACK received 0x20: Address + Write bit transmitted, ACK not received 0x28: Data byte transmitted in master mode, ACK received 0x30: Data byte transmitted in master mode, ACK not received 0x38: Arbitration lost in address or data byte 0x40: Address + Read bit transmitted, ACK received 0x48: Address + Read bit transmitted, ACK not received 0x50: Data byte received in master mode, ACK received 0x58: Data byte received in master mode, ACK not received 0x01: Timeout when sending 9th SCL clk Other: Reserved
15:8	R/W	0x10	TIMEOUT_N When sending the 9th clock, assert fail signal when slave device does not response after **N*F$_{SCL}$** cycles. And software must do a reset to TWI_DRV

			module and send a stop condition to slave.
7:2	/	/	/
1	R/W	0x0	SOFT_RESET 0: normal 1: reset
0	R/W	0x0	TWI_DRV_EN 0: Module disable 1: Module enable (only use in TWI Master Mode)

9.1.6.11. 0x0204 TWI_DRV Transmission Configuration Register(Default Value:0x0000_0001)

Offset: 0x0204			Register Name: TWI_DRV_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	PKT_INTERVAL Define the interval between each packet for PKT_INTERVAL F_{SCL} cycles.
15:0	R/W	0x1	PACKET_CNT FIFO data be transmitted as PACKET_CNT packets in current format.

9.1.6.12. 0x0208 TWI_DRV Slave ID Register(Default Value:0x0000_0000)

Offset: 0x0208			Register Name: TWI_DRV_SLV
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:9	R/W	0x0	SLV_ID Slave device ID • 7-bit addressing SLA6, SLA5, SLA4, SLA3, SLA2, SLA1, SLA0 • 10-bit addressing 1, 1, 1, 1, 0, SLAX[9:8]
8	R/W	0x0	CMD R/W operation to slave device 0: write 1: read
7:0	R/W	0x0	SLV_ID_X SLAX[7:0], low 8 bits for slave device ID with 10-bit addressing

9.1.6.13. 0x020C TWI_DRV Packet Format Register(Default Value:0x0001_0001)

Offset: 0x020C			Register Name: TWI_DRV_FMT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/

23:16	R/W	0x1	ADDR_BYTE How many bytes be sent as slave device reg address 0~255
15:0	R/W	0x1	DATA_BYTE How many bytes be sent/received as data 1~65535

9.1.6.14. 0x0210 TWI_DRV Bus Control Register(Default Value:0x0000_80C0)

Offset: 0x0210			Register Name: TWI_DRV_BUS_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	W	0x0	CLK_COUNT_MODE 0: scl clock high period count on oscl 1: scl clock high period count on iscl
15	R/W	0x1	CLK_DUTY Setting duty cycle of Clock as Master 0: 50% 1: 40%
14:12	R/W	0x0	CLK_N TWI_DRV bus sampling clock F0=24 MHz/2^CLK_N
11:8	R/W	0x0	CLK_M TWI_DRV output SCL frequency is F_{SCL}=F1/10=(F0/(CLK_M+1))/10 Specially,Foscl = F1/11 when CLK_M=0 and CLK_DUTY=40% due to the delay of SCL sample debounce.
7	R	0x1	SCL_STA SCL current status
6	R	0x1	SDA_STA SDA current status
5:4	/	/	/
3	R/W	0x0	SCL_MOV SCL manual output value
2	R/W	0x0	SDA_MOV SDA manual output value
1	R/W	0x0	SCL_MOE SCL manual output en
0	R/W	0x0	SDA_MOE SDA manual output en

9.1.6.15. 0x0214 TWI_DRV Interrupt Control Register(Default Value:0x0000_0000)

Offset: 0x0214	Register Name: TWI_DRV_INT_CTRL

Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
19	R/W	0x0	RX_REQ_INT_EN
18	R/W	0x0	TX_REQ_INT_EN
17	R/W	0x0	TRAN_ERR_INT_EN
16	R/W	0x0	TRAN_COM_INT_EN
15:4	/	/	/
3	R/W1C	0x0	RX_REQ_PD Set when the data byte number in RECV_FIFO reaches RX_TRIG
2	R/W1C	0x0	TX_REQ_PD Set when there is no less than DMA_TX_TRIG empty byte number in SEND_FIFO
1	R/W1C	0x0	TRAN_ERR_PD Packet transmission failed pending
0	R/W1C	0x0	TRAN_COM_PD Packet transmission completed pending

9.1.6.16. 0x0218 TWI_DRV DMA Configure Register(Default Value:0x0010_0010)

Offset: 0x0218			Register Name: TWI_DRV_DMA_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	DMA_RX_EN
23:22	/	/	/
21:16	R/W	0x10	RX_TRIG When DMA_RX_EN set, send DMA RX Req when the data byte number in RECV_FIFO reaches RX_TRIG or Read Packet Transmission completed with RECV_FIFO not empty
15:9	/	/	/
8	R/W	0x0	DMA_TX_EN
7:6	/	/	/
5:0	R/W	0x10	TX_TRIG When DMA_TX_EN set, send DMA TX Req when there is no less than DMA_TX_TRIG empty byte number in SEND_FIFO

9.1.6.17. 0x021C TWI_DRV FIFO Content Register(Default Value:0x0000_0000)

Offset: 0x021C			Register Name: TWI_DRV_FIFO_CON
Bit	**Read/Write**	**Default/Hex**	**Description**
31:23	/	/	/
22	R/WAC	0x0	RECV_FIFO_CLEAR Set this bit to clear RECV_FIFO pointer, and this bit is cleared automatically

21:16	R	0x0	RECV_FIFO_CONTENT The number of data in RECV_FIFO
15:7	/	/	/
6	R/WAC	0x0	SEND_FIFO_CLEAR Set this bit to clear SEND_FIFO pointer, and this bit is cleared automatically
5:0	R	0x0	SEND_FIFO_CONTENT The number of data in SEND_FIFO

9.1.6.18. 0x0300 TWI_DRV Send Data FIFO Access Register(Default Value:0x0000_0000)

Offset: 0x0300			Register Name: TWI_DRV_SEND_FIFO_ACC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	W	0x0	SEND_DATA_FIFO Address of a 32x8 SEND_FIFO, which stores reg address and data sending to slave device

9.1.6.19. 0x0304 TWI_DRV Receive Data FIFO Access Register(Default Value:0x0000_0000)

Offset: 0x0304			Register Name: TWI_DRV_RECV_FIFO_ACC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R	0x0	RECV_DATA_FIFO Address of a 32x8 RECV_FIFO, which stores data received from slave device

9.2. UART

9.2.1. Overview

The UART is used for serial communication with a peripheral, modem (data carrier equipment, DCE) or data set. Data is written from a master (CPU) over the APB bus to the UART and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back.

The UART contains registers to control the character length, baud rate, parity generation/checking, and interrupt generation. Although there is only one interrupt output signal from the UART, there are several prioritized interrupt types that can be responsible for its assertion. Each of the interrupt types can be separately enabled/disabled with the control registers.

The UART has 16450 and 16550 modes of operation, which are compatible with a range of standard software drivers. In 16550 mode, transmit and receive operations are both buffered by FIFOs. In 16450 mode, these FIFOs are disabled.

The UART supports word lengths from five to eight bits, an optional parity bit and 1, 1 ½ or 2 stop bits, and is fully programmable by an AMBA APB CPU interface. A 16-bit programmable baud rate generator and an 8-bit scratch register are included, together with separate transmit and receive FIFOs. Eight modem control lines and a diagnostic loop-back mode are provided.

Interrupts can be generated for a range of TX Buffer/FIFO, RX Buffer/FIFO, Modem Status and Line Status conditions.

For integration in system where Infrared SIR serial data format is required, the UART can be configured to have a software-programmable IrDA SIR mode. If this mode is not selected, only the UART (RS232 standard) serial data format is available.

The UART has the following features:

- Up to 6 UART controllers (UART0, UART1, UART2, UART3, UART4, UART5)
- UART0, UART5: 2-wire; UART1, UART2, UART3, UART4: 4-wire
- 2-wire UART can be used for printing; 4-wire UART can be used for flow control
- Compatible with industry-standard 16550 UARTs
- 256 bytes transmit and receive data FIFOs
- Capable of speed up to 4 Mbit/s with 64 MHz APB clock, and speed up to 1.5 Mbit/s with 24 MHz APB clock
- Supports 5 to 8 data bits and 1/1.5/2 stop bits
- Supports even, odd or no parity
- Supports DMA controller interface
- Supports software/hardware flow control
- Supports IrDA 1.0 SIR
- Supports RS-485/9-bit mode

9.2.2. Block Diagram

Figure 9-11 shows a block diagram of the UART.

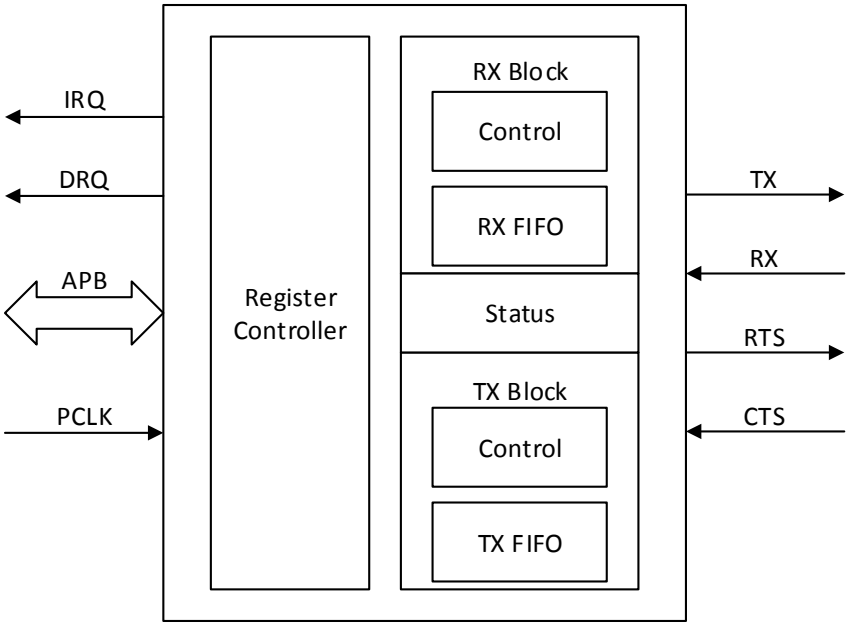

Figure 9- 11. UART Block Diagram

9.2.3. Operations and Functional Descriptions

9.2.3.1. External Signals

Table 9-3 describes the external signals of UART.

Table 9- 3. UART External Signals

Signal	Type	Description
UART0_TX	O	UART0 Data Transmit
UART0_RX	I	UART0 Data Receive
UART1_TX	O	UART1 Data Transmit
UART1_RX	I	UART1 Data Receive
UART1_CTS	I	UART1 Data Clear to Send
UART1_RTS	O	UART1 Data Request to Send
UART2_TX	O	UART2 Data Transmit
UART2_RX	I	UART2 Data Receive
UART2_CTS	I	UART2 Data Clear to Send
UART2_RTS	O	UART2 Data Request to Send
UART3_TX	O	UART3 Data Transmit
UART3_RX	I	UART3 Data Receive
UART3_CTS	I	UART3 Data Clear to Send
UART3_RTS	O	UART3 Data Request to Send

UART4_TX	O	UART4 Data Transmit
UART4_RX	I	UART4 Data Receive
UART4_CTS	I	UART4 Data Clear to Send
UART4_RTS	O	UART4 Data Request to Send
UART5_TX	O	UART5 Data Transmit
UART5_RX	I	UART5 Data Receive

9.2.3.2. Clock Sources

Table 9-4 describes the clock sources of UART.

Table 9- 4. UART Clock Sources

Clock Sources	Description
APB2_CLK	Clock of APB2

9.2.3.3. Typical Application

Figure 9-12 shows the application block diagram of UART.

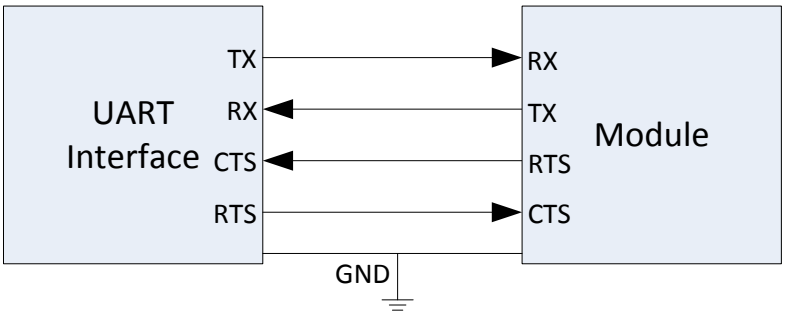

Figure 9- 12. UART Application Diagram

9.2.3.4. UART Timing Diagram

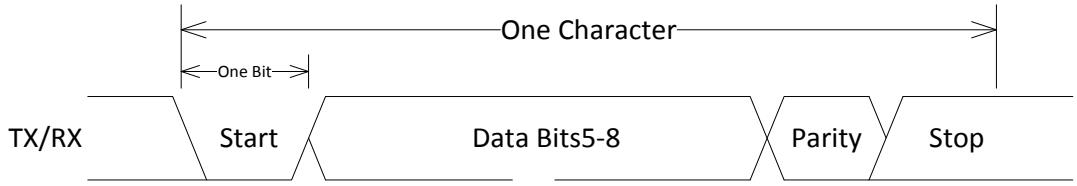

Figure 9- 13. UART Serial Data Format

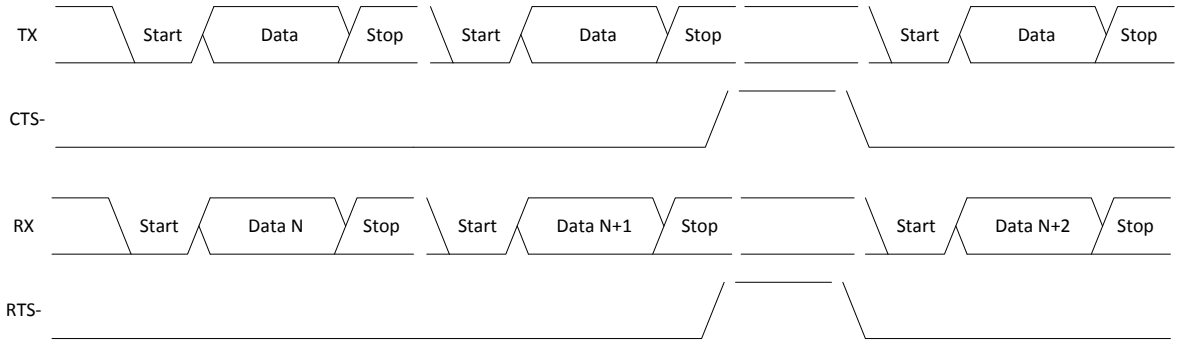

Figure 9- 14. RTS/CTS Autoflow Control Timing

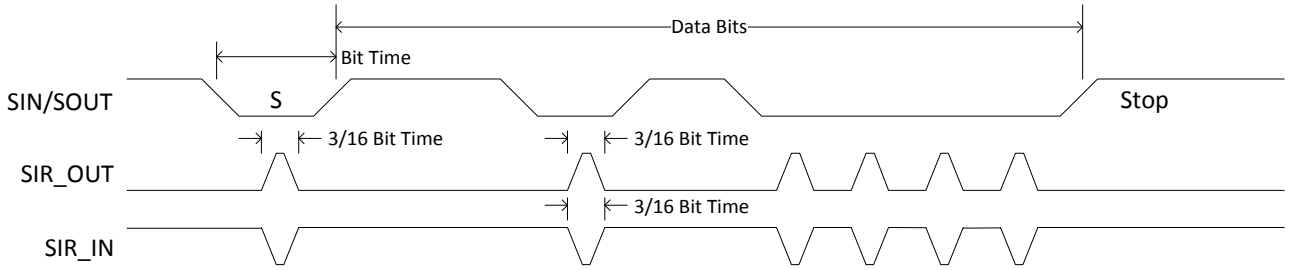

Figure 9- 15. Serial IrDA Data Format

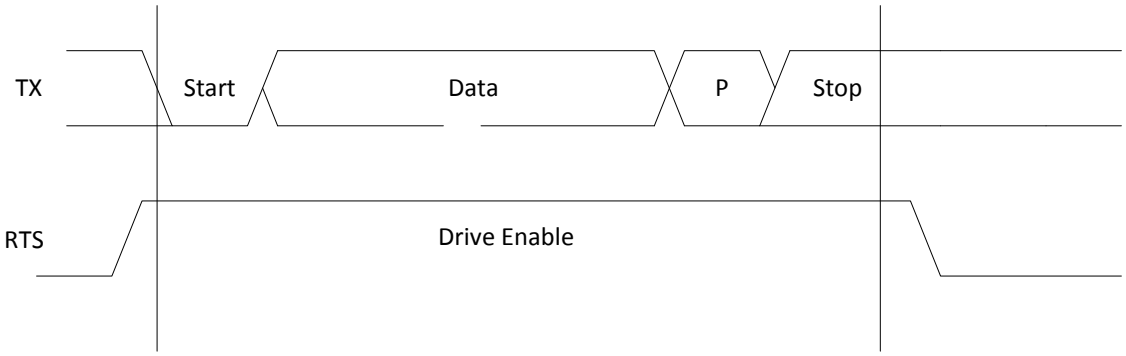

Figure 9- 16. RS-485 Timing

9.2.3.5. UART Operating Mode

9.2.3.5.1. Basic Mode Setting

The **UART_LCR** register can set basic parameter of a data frame: data width(5 to 8 bits), stop bit number(1/1.5/2), parity type.

A frame transfer of the UART includes the start signal, data signal, parity bit and stop signal. The LSB is transmitted first.

- Start signal(start bit): It is the start flag of a data frame. According to UART protocol, the low level of TXD signal indicates the start of a data frame. When the UART transmits data, the level need hold high.
- Data signal(data bit): The data bit width can be configured as 5-bit,6-bit,7-bit,8-bit through different applications.

- Parity bit: It is 1-bit error correction signal. Parity bit includes odd parity, even parity. The UART can enable and disable the parity bit by setting the **UART_LCR** register.
- Stop Signal(stop bit): It is the stop bit of a data frame. The stop bit can be set to 1-bit,1.5-bit and 2-bit by the **UART_LCR** register. The high level of TXD signal indicates the end of a data frame.

9.2.3.5.2. Baud Rate Setting

The baud rate is calculated as follows: Baud rate = SCLK /(16 * divisor). SCLK is usually APB2 and can be set in CCU. Divisor is frequency divider of UART. The frequency divider has 16-bit,the low 8-bit is in the UART_DLL register, the high 8-bit is in the UART_DLH register.
The relationship between different UART mode and error rate is as follows.

Table 9- 5. UART Mode Baud and Error Rates

Clock source	Divisor	Baud rate	Over sampling	Error(%)
24000000	5000	300	16	0
24000000	2500	600	16	0
24000000	1250	1200	16	0
24000000	625	2400	16	0
24000000	313	4800	16	-0.16
24000000	156	9600	16	0.16
24000000	78	19200	16	0.16
24000000	39	38400	16	0.16
24000000	26	57600	16	0.16
24000000	13	115200	16	0.16
48000000	13	230400	16	0.16
64000000	7	576000	16	-0.794
75000000	5	921600	16	1.725
48000000	3	1000000	16	0
24000000	1	1500000	16	0
48000000	1	3000000	16	0
64000000	1	4000000	16	0

Table 9- 6. IrDA Mode Baud and Error Rates

Clock source	Divisor	Baud rate	Encoding	Error(%)
24000000	5000	300	3/16	0
24000000	2500	600	3/16	0
24000000	1250	1200	3/16	0
24000000	625	2400	3/16	0
24000000	313	4800	3/16	-0.16
24000000	156	9600	3/16	0.16
24000000	78	19200	3/16	0.16
24000000	39	38400	3/16	0.16

24000000	26	57600	3/16	0.16
24000000	13	115200	3/16	0.16

Table 9- 7. RS485 Mode Baud and Error Rates

Clock source	Divisor	Baud rate	Encoding	Error(%)
24000000	5000	300	16	0
24000000	2500	600	16	0
24000000	1250	1200	16	0
24000000	625	2400	16	0
24000000	313	4800	16	-0.16
24000000	156	9600	16	0.16
24000000	78	19200	16	0.16
24000000	39	38400	16	0.16
24000000	26	57600	16	0.16
24000000	13	115200	16	0.16

9.2.3.5.3. DLAB Setting

DLAB control bit (**UART_LCR[7]**) is the access control bit of divisor Latch register.
If DLAB is 0, then 0x00 offset address is **TX/RX FIFO** register, 0x04 offset address is **IER** register.
If DLAB is 1, then 0x00 offset address is **DLL** register, 0x04 offset address is **DLH** register.
When UART initial, divisor need be set. That is, writing 1 to DLAB can access the **DLL** and **DLH** register, after finished setting, writing 0 to DLAB can access the **TX/RX FIFO** register.

9.2.3.5.4. CHCFG_AT_BUSY Setting

The function of **CHCFG_AT_BUSY**(UART_HALT[1]) and **CHANGE_UPDATE**(UART_HALT[2]) are as follows.
CHCFG_AT_BUSY(configure at busy): Enable the bit, software can also set UART controller when UART is busy, such as the LCR,DLH,DLL register.
CHANGE_UPDATE(update configuration): If **CHCFG_AT_BUSY** is enabled, and **CHANGE_UPDATE** is written to 1, the configuration of UART controller can be updated. After completed update, the bit is cleared to 0 automatically.

Setting divisor, performs the following steps:
Step1 Write 1 to **CHCFG_AT_BUSY** to enable “configure at busy”.
Step2 Write 1 to **DLAB** , and set **DLH** and **DLL**.
Step3 Write 1 to **CHANGE_UPDATE** to update configuration. The bit is cleared to 0 automatically after completed update.

9.2.3.5.5. UART Busy

UART_USR[0] is a busy flag of UART controller or not.
When TX transmits data, or RX receives data, or TX FIFO is not empty, or RX FIFO is not empty, then the BUSY flag bit can be set to 1 by hardware, which indicates the UART controller is busy.

9.2.4. Programming Guidelines

9.2.4.1. Initialization

Step1 System Initialization

- Configure APB2_CFG_REG in CCU module to set APB2 bus clock(The clock is 24 MHz by default).
- Set UART_BGR_REG[UARTx_GATING] to 1 to enable the module clock, and set UART_BGR_REG[UARTx_RST] to 1 to de-assert the module.

Step2 UART Controller Initialization

- IO configuration: Configure GPIO multiplex as UART function, and set UART pins to internal pull-up mode(For detail, see the description in Port Controller).
- Baud-rate configuration:
  - Set UART baud-rate(refer to section 9.2.3.5.2);
  - Write UART_FCR[FIFOE] to 1 to enable TX/RX FIFO;
  - Write UART_HALT[HALT_TX] to 1 to disable TX transfer;
  - Set UART_LCR[DLAB] to 1, remain default configuration for other bits; set 0x00 offset address to UART_DLL register, set 0x04 offset address to UART_DLH register;
  - Write the high 8-bit of divisor to UART_DLH, and write the low 8-bit of divisor to UART_DLL;
  - Set UART_LCR[DLAB] to 0, remain default configuration for other bits; set 0x00 offset address to UART_RBR/UART_THR register, set 0x04 offset address to UART_IER register;
  - Set UART_HALT[HALT_TX] to 0 to enable TX transfer.

Step3 Controller Parameter Configuration

- Set data width, stop bits and even/odd parity type by writing UART_LCR register.
- Reset, enable FIFO and set FIFO trigger condition by writing UART_FCR register.
- Set flow control parameter by writing UART_MCR register.

Step4 Interrupt Configuration

- Configure UART interrupt vector number to request UART interrupt(please refer to GIC module for interrupt vector number).
- In DMA mode, write UART_IER to 0 to disable interrupt; write UART_HSK[Handshake configuration] to 0xE5 to set DMA handshake mode; write UART_FCR[DMAM] to 1 to set DMA transmission/reception mode; set DMA parameter and request DMA interrupt according to DMA configuration process.
- In Interrupt mode, configure UART_IER to enable corresponding interrupt according to requirements: such as transmit(TX) interrupt, receive(RX) interrupt, receive line status interrupt, RS48 interrupt, etc. (Here TX/RX Interrupt is usually used).

9.2.4.2. Data Transfer/Receive in Query Mode

Data transfer

Step1 Write data to UART_THR to start data transfer.

Step2 Check TX_FIFO status by reading UART_USR[TFNF]. If the bit is 1, data can continue to be written; if the bit is 0, wait data transfer, and data cannot continue to write until FIFO is not full.

Data receive

Step1 Check RX_FIFO status by reading UART_USR[RFNE].

Step2 Read data from UART_RBR if RX_FIFO is not empty.

Step3 If UART_USR[RFNE] is 0, data is received completely.

9.2.4.3. Data Transfer/Receive in Interrupt Mode

Data transfer

Step1 Set UART_IER[ETBEI] to 1 to enable UART transfer interrupt.

Step2 Write data to be transmitted to UART_THR.

Step3 When the data of TX_FIFO meets trigger condition(such as FIFO/2, FIFO/4), UART transfer interrupt is generated.

Step4 Check UART_USR[TFE] and determine whether TX_FIFO is empty. If UART_USR[TFE] is 1, it indicates that the data in TX_FIFO is transmitted completely.

Step5 Clear UART_IER[ETBEI] to 0 to disable transfer interrupt.

Data receive

Step1 Set UART_IER[ERBFI] to 1 to enable UART receive interrupt.

Step2 When the received data from RX_FIFO meets trigger condition(such as FIFO/2, FIFO/4), UART receive interrupt is generated.

Step3 Read data from UART_RBR.

Step4 Check RX_FIFO status by reading UART_USR[RFNE] and determine whether to read data. If the bit is 1, continue to read data from UART_RBR until UART_USR[RFNE] is cleared to 0, which indicates data is received completely.

Figure 9-17 shows the process of UART transmitting and receiving data in interrupt mode.

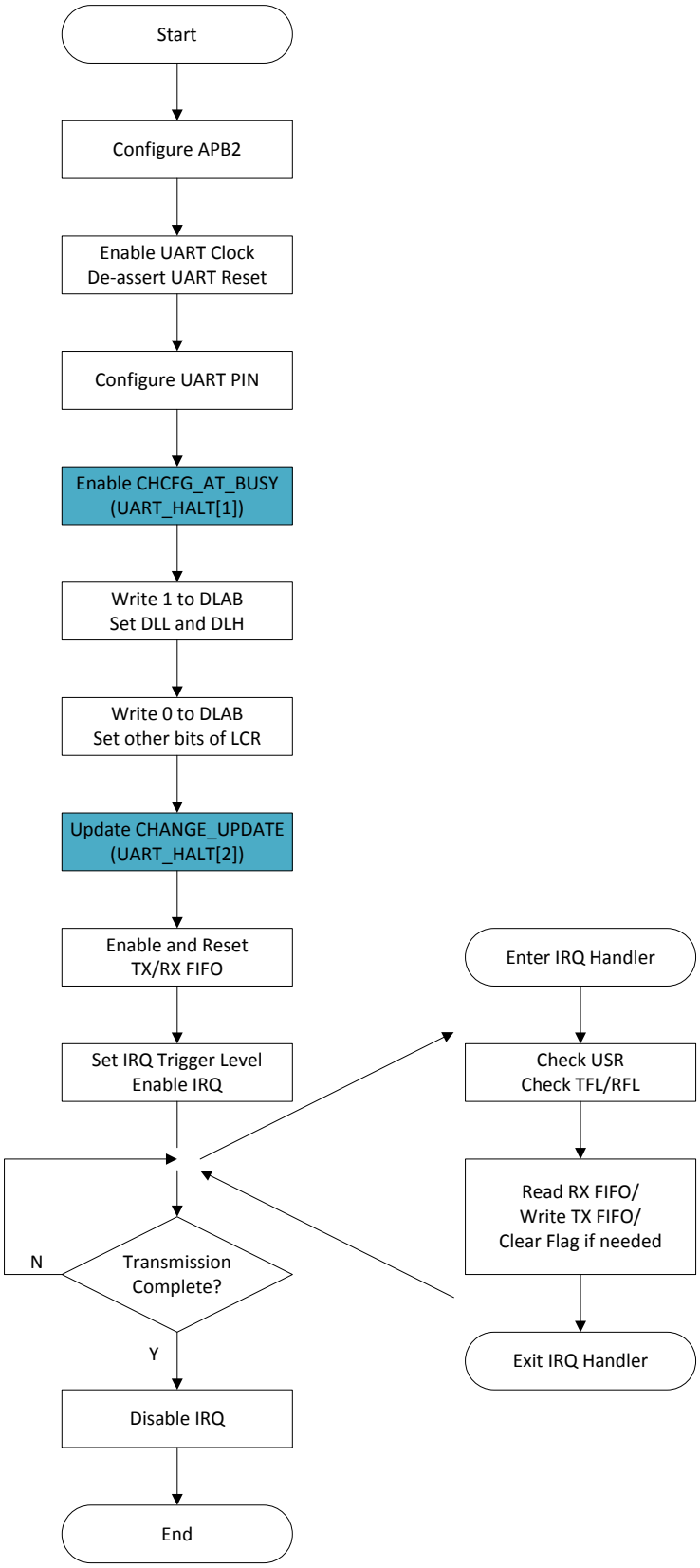

Figure 9- 17. Process of UART Transmitting/Receiving Data in Interrupt Mode

9.2.4.4. Data Transfer/Receive in DMA Mode

Data transfer

Step1 Configure UART DMA interrupt according to initialization process.

Step2 Configure DMA data channel, including transfer source address, transfer destination address, number of data to be transferred, and transfer type, etc(For details, see the description in DMAC module).

Step3 Enable DMA transfer function of the UART by setting the register of DMA module.

Step4 Determine whether UART data is transferred completely based on DMA status. If all data is transferred completely, disable DMA transfer function of the UART.

Data receive

Step1 Configure DMA data channel, including transfer source address, transfer destination address, number of data to be transferred, and transfer type, etc(For details, see the description in DMAC module).

Step2 Enable DMA receive function of the UART by setting the register of DMA module.

Step3 Determine whether UART data is received completely based on DMA status. If all data is received completely, disable DMA receive function of the UART.

Figure 9-18 shows the process of UART transmitting data in DMA mode.

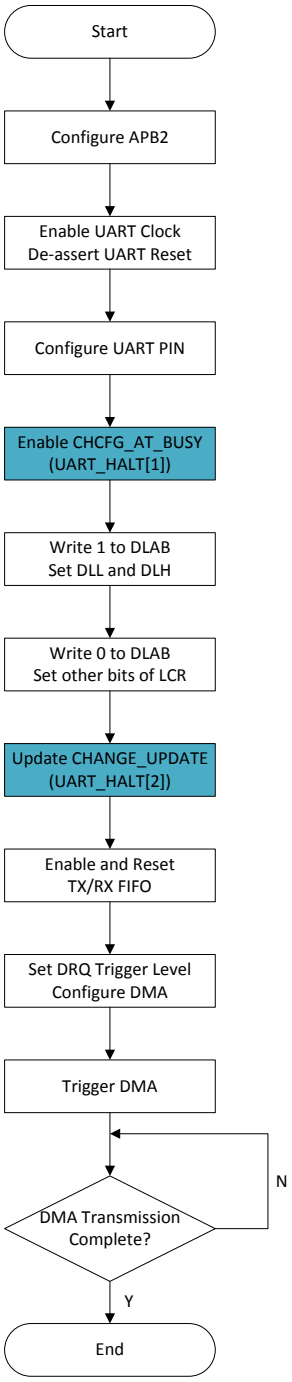

Figure 9- 18. Process of DMA Transmitting Data in DMA Mode

9.2.5. Register List

Module Name	Base Address
UART0	0x05000000
UART1	0x05000400
UART2	0x05000800
UART3	0x05000C00

UART4	0x05001000
UART5	0x05001400

Register Name	Offset	Description
UART_RBR	0x0000	UART Receive Buffer Register
UART_THR	0x0000	UART Transmit Holding Register
UART_DLL	0x0000	UART Divisor Latch Low Register
UART_DLH	0x0004	UART Divisor Latch High Register
UART_IER	0x0004	UART Interrupt Enable Register
UART_IIR	0x0008	UART Interrupt Identity Register
UART_FCR	0x0008	UART FIFO Control Register
UART_LCR	0x000C	UART Line Control Register
UART_MCR	0x0010	UART Modem Control Register
UART_LSR	0x0014	UART Line Status Register
UART_MSR	0x0018	UART Modem Status Register
UART_SCH	0x001C	UART Scratch Register
UART_USR	0x007C	UART Status Register
UART_TFL	0x0080	UART Transmit FIFO Level Register
UART_RFL	0x0084	UART Receive FIFO Level Register
UART_HSK	0x0088	UART DMA Handshake Configuration Register
UART_HALT	0x00A4	UART Halt TX Register
UART_DBG_DLL	0x00B0	UART Debug DLL Register
UART_DBG_DLH	0x00B4	UART Debug DLH Register
UART_485_CTL	0x00C0	UART RS485 Control and Status Register
RS485_ADDR_MATCH	0x00C4	UART RS485 Address Match Register
BUS_IDLE_CHK	0x00C8	UART RS485 Bus Idle Check Register
TX_DLY	0x00CC	UART TX Delay Register

9.2.6. Register Description

9.2.6.1. 0x0000 UART Receiver Buffer Register(Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: UART_RBR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R	0x0	RBR Receiver Buffer Register Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LCR) is set. If in FIFO mode and FIFOs are enabled (FCR[0] set to one), this register

			accesses the head of the receive FIFO. If the receive FIFO is full and this register can not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an overrun error occurs.

9.2.6.2. 0x0000 UART Transmit Holding Register(Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: UART_THR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	W	0x0	THR Transmit Holding Register Data be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, 16 number of characters data may be written to the THR before the FIFO is full. When the FIFO is full, any write data results in the write data being lost.

9.2.6.3. 0x0000 UART Divisor Latch Low Register(Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: UART_DLL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	DLL Divisor Latch Low Lower 8 bits of a 16 bits, read/write, Divisor Latch Register contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that when the Divisor Latch Registers (DLL and DLH) are set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLL is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

9.2.6.4. 0x0004 UART Divisor Latch High Register(Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: UART_DLH
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	DLH Divisor Latch High Upper 8 bits of a 16 bits, read/write, Divisor Latch Register contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that when the Divisor Latch Registers (DLL and DLH) is set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

9.2.6.5. 0x0004 UART Interrupt Enable Register(Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: UART_IER
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	PTIME Programmable THRE Interrupt Mode Enable This is used to enable/disable the generation of THRE Interrupt. 0: Disable 1: Enable
6:5	/	/	/
4	R/W	0x0	RS485_INT_EN RS485 Interrupt Enable 0:Disable 1:Enable
3	R/W	0x0	EDSSI Enable Modem Status Interrupt This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0: Disable 1: Enable
2	R/W	0x0	ELSI Enable Receiver Line Status Interrupt This is used to enable/disable the generation of Receiver Line Status

			Interrupt. This is the highest priority interrupt. 0: Disable 1: Enable
1	R/W	0x0	ETBEI Enable Transmit Holding Register Empty Interrupt This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0: Disable 1: Enable
0	R/W	0x0	ERBFI Enable Received Data Available Interrupt This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupt. 0: Disable 1: Enable

9.2.6.6. 0x0008 UART Interrupt Identity Register(Default Value: 0x0000_0001)

Offset: 0x0008			Register Name: UART_IIR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:6	R	0x0	FEFLAG FIFOs Enable Flag This is used to indicate whether the FIFOs are enabled or disabled. 00: Disable 11: Enable
5:4	/	/	/
3:0	R	0x1	IID Interrupt ID This indicates the highest priority pending interrupt which can be one of the following types: 0000: modem status 0001: no interrupt pending 0010: THR empty 0011: RS485 Interrupt 0100: received data available 0110: receiver line status 0111: busy detect 1100: character timeout The bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt.

Interrupt ID	Priority	Interrupt	Interrupt Source	Interrupt Reset

	Level	Type		
0001	-	None	None	-
0110	Highest	Receiver line status	Overrun/parity/framing errors or break interrupt	Reading the line status register
0011	Second	RS485 Interrupt	In RS485 mode, receives address data and match setting address	Writes 1 to addr flag to reset
0100	Third	Received data available	Receiver data available (non-FIFO mode or FIFOs disabled) or RCVR FIFO trigger level reached (FIFO mode and FIFOs enabled)	Reading the receiver buffer register (non-FIFO mode or FIFOs disabled) or the FIFO drops below the trigger level (FIFO mode and FIFOs enabled)
1100	Fourth	Character timeout indication	No characters in or out of the RCVR FIFO during the last 4 character times and there is at least 1character in it during This time	Reading the receiver buffer register
0010	Fifth	Transmit holding register empty	Transmitter holding register empty (Program THRE mode disabled) or XMIT FIFO at or below threshold (Program THRE mode enabled)	Reading the IIR register (if source of interrupt); or, writing into THR (FIFOs or THRE mode not selected or disabled) or XMIT FIFO above threshold (FIFOs and THRE mode selected and enabled).
0000	Sixth	Modem status	Clear to send or data set ready or ring indicator or data carrier detect. Note that if auto flow control mode is enabled, a change in CTS (that is, DCTS set) does not cause an interrupt.	Reading the Modem status register
0111	Seventh	Busy detect indication	UART_16550_COMPATIBLE = NO and master has tried to write to the Line Control Register while the UART is busy (USR[0] is set to one).	Reading the UART status register

9.2.6.7. 0x0008 UART FIFO Control Register(Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: UART_FCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:6	W	0x0	RT RCVR Trigger This is used to select the trigger level in the receiver FIFO when the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. 00: 1 character in the FIFO

			01: FIFO ¼ full 10: FIFO ½ full 11: FIFO-2 less than full
5:4	W	0x0	TFT TX Empty Trigger This is used to select the empty threshold level when the THRE Interrupts are generated and the mode is active. It also determines when the dma_tx_req_n signal is asserted in certain modes of operation. 00: FIFO empty 01: 2 characters in the FIFO 10: FIFO ¼ full 11: FIFO ½ full
3	W	0x0	DMAM DMA Mode 0: Mode 0 In this mode, if PTE is high and TX FIFO is enabled, the TX DMA request will send when TFL is less than or equal to FIFO Trigger Level. If PTE is high and TX FIFO is disabled, the TX DMA request will send when THRE is empty. If PTE is low, the TX DMA request will send when the TX FIFO is empty. If dma_pte_rx is high and RX FIFO is enabled, the rx drq will send when RFL is equal to or more than FIFO Trigger Level. 1: Mode 1 In this mode, if TX FIFO is enabled and the PTE is high, the TX DMA request will send when TFL is less than or equal to FIFO Trigger Level. If PTE is low, the TX DMA request will send when TX FIFO is empty and the request stops only when TX FIFO is full. If RFL is equal to or more than FIFO Trigger Level, the rx drq will be set to 1, in otherwise, it will be set to 0.
2	W	0x0	XFIFOR XMIT FIFO Reset The bit resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request. It is 'self-clearing'. It is not necessary to clear this bit.
1	W	0x0	RFIFOR RCVR FIFO Reset The bit resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request. It is 'self-clearing'. It is not necessary to clear this bit.
0	W	0x0	FIFOE Enable FIFOs The bit enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed, both the XMIT and RCVR controller portion of FIFOs is reset.

9.2.6.8. 0x000C UART Line Control Register(Default Value: 0x0000_0000)

Offset: 0x000C			Register Name: UART_LCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	DLAB Divisor Latch Access Bit It is writeable only when UART is not busy (USR[0] is zero) and always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers. 0: Select RX Buffer Register (RBR) / TX Holding Register(THR) and Interrupt Enable Register (IER) 1: Select Divisor Latch LS Register (DLL) and Divisor Latch MS Register (DLM)
6	R/W	0x0	BC Break Control Bit This is used to cause a break condition to be transmitted to the receiving device. If setting to 0, the serial output is forced to the spacing (logic 0) state. When not in Loopback mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE is enabled and active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.
5:4	R/W	0x0	EPS Even Parity Select It is writeable only when UART is not busy (USR[0] is zero) and always writable/readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). Setting the LCR[5] is unset to reverse the LCR[4]. 00: Odd Parity 01: Even Parity 1X: Reverse LCR[4] In RS485 mode, it is the 9^{th} bit--address bit. 11:9^{th} bit = 0, indicates that this is a data byte. 10:9^{th} bit = 1, indicates that this is an address byte. **Note: When using this function, PEN(LCR[3]) must set to 1.**
3	R/W	0x0	PEN Parity Enable It is writeable only when UART is not busy (USR[0] is zero) and always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0: Parity disabled 1: Parity enabled
2	R/W	0x0	STOP Number of stop bits

			It is writeable only when UART is not busy (USR[0] is zero) and always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If setting to 0, one stop bit is transmitted in the serial data. If setting to 1 and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 0: 1 stop bit 1: 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit
1:0	R/W	0x0	DLS Data Length Select It is writeable only when UART is not busy (USR[0] is zero) and always readable. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits

9.2.6.9. 0x0010 UART Modem Control Register(Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: UART_MCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:6	R/W	0x0	UART_FUNCTION Select IrDA or RS485 00:UART Mode 01:IrDA SIR Mode 10:RS485 Mode 11:Reserved
5	R/W	0x0	AFCE Auto Flow Control Enable When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled. 0: Auto Flow Control mode disabled 1: Auto Flow Control mode enabled
4	R/W	0x0	LOOP Loop Back Mode 0: Normal Mode 1: Loop Back Mode This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE != Enabled or not active, MCR[6] is set to zero), data on the sout line is held high, while serial data output is looped

			back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE == Enabled AND active, MCR[6] is set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.
3:2	/	/	/
1	R/W	0x0	RTS Request to Send This is used to directly control the Request to Send (rts_n) output. The RTS (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] is set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] is set to one) and FIFOs enable (FCR[0] is set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. 0: rts_n de-asserted (logic 1) 1: rts_n asserted (logic 0) Note that in Loopback mode (MCR[4] is set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input.
0	R/W	0x0	DTR Data Terminal Ready This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n. 0: dtr_n de-asserted (logic 1) 1: dtr_n asserted (logic 0) The Data Terminal Ready output is used to inform the modem or data set that the UART is ready to establish communications. Note that in Loopback mode (MCR[4] is set to one), the dtr_n output is held inactive high while the value of this location is internally looped back to an input.

9.2.6.10. 0x0014 UART Line Status Register(Default Value: 0x0000_0060)

Offset:0x0014			Register Name: UART_LSR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R	0x0	FIFOERR RX Data Error in FIFO When FIFOs are disabled, this bit is always 0. When FIFOs are enabled, this

			bit is set to “1” when there is at least one PE, FE, or BI in the RX FIFO. It is cleared by a read from the LSR register provided, there are no subsequent errors in the FIFO.
6	R	0x1	TEMT Transmitter Empty If the FIFOs are disabled, this bit is set to "1" whenever the TX Holding Register and the TX Shift Register are empty. If the FIFOs are enabled, this bit is set whenever the TX FIFO and the TX Shift Register are empty. In both cases, this bit is cleared when a byte is written to the TX data channel.
5	R	0x1	THRE TX Holding Register Empty If the FIFOs are disabled, this bit is set to "1" whenever the TX Holding Register is empty and ready to accept new data and it is cleared when the CPU writes to the TX Holding Register. If the FIFOs are enabled, this bit is set to "1" whenever the TX FIFO is empty and it is cleared when at least one byte is written to the TX FIFO.
4	R	0x0	BI Break Interrupt This is used to indicate the detection of a break sequence on the serial input data. If in UART mode (SIR_MODE == Disabled), it is set whenever the serial input, sir_in, is held in a logic '0' state for longer than the sum of *start time* + *data bits* + *parity* + *stop bits*. If in infrared mode (SIR_MODE == Enabled), it is set whenever the serial input, sir_in, is continuously pulsed to logic '0' for longer than the sum of *start time* + *data bits* + *parity* + *stop bits*. A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART. In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read.
3	RC	0x0	FE Framing Error This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data,

			and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0: no framing error 1:framing error Reading the LSR clears the FE bit.
2	RC	0x0	PE Parity Error This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0: no parity error 1: parity error Reading the LSR clears the PE bit.
1	RC	0x0	OE Overrun Error This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. 0: no overrun error 1: overrun error Reading the LSR clears the OE bit.
0	R	0x0	DR Data Ready This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0: no data ready 1: data ready This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode.

9.2.6.11. 0x0018 UART Modem Status Register(Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: UART_MSR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R	0x0	DCD

			Line State of Data Carrier Detect This is used to indicate the current state of the modem control line dcd_n. This bit is the complement of dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set. 0: dcd_n input is de-asserted (logic 1) 1: dcd_n input is asserted (logic 0)
6	R	0x0	RI Line State of Ring Indicator This is used to indicate the current state of the modem control line ri_n. This bit is the complement of ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by the modem or data set. 0: ri_n input is de-asserted (logic 1) 1: ri_n input is asserted (logic 0)
5	R	0x0	DSR Line State of Data Set Ready This is used to indicate the current state of the modem control line dsr_n. This bit is the complement of dsr_n. When the Data Set Ready input (dsr_n) is asserted it is an indication that the modem or data set is ready to establish communications with UART. 0: dsr_n input is de-asserted (logic 1) 1: dsr_n input is asserted (logic 0) In Loopback Mode (MCR[4] is set to 1), DSR is the same as MCR[0] (DTR).
4	R	0x0	CTS Line State of Clear To Send This is used to indicate the current state of the modem control line cts_n. This bit is the complement of cts_n. When the Clear to Send input (cts_n) is asserted it is an indication that the modem or data set is ready to exchange data with UART. 0: cts_n input is de-asserted (logic 1) 1: cts_n input is asserted (logic 0) In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS).
3	RC	0x0	DDCD Delta Data Carrier Detect This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read. 0: no change on dcd_n since last read of MSR 1: change on dcd_n since last read of MSR Reading the MSR clears the DDCD bit. **Note: If the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDCD bit is set when the reset is removed if the dcd_n signal remains asserted.**
2	RC	0x0	TERI Trailing Edge Ring Indicator

			This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read. 0: no change on ri_n since last read of MSR 1: change on ri_n since last read of MSR Reading the MSR clears the TERI bit.
1	RC	0x0	DDSR Delta Data Set Ready This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read. 0: no change on dsr_n since last read of MSR 1: change on dsr_n since last read of MSR Reading the MSR clears the DDSR bit. In Loopback Mode (MCR[4] = 1), DDSR reflects changes on MCR[0] (DTR). **Note: If the DDSR bit is not set and the dsr_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDSR bit is set when the reset is removed if the dsr_n signal remains asserted.**
0	RC	0x0	DCTS Delta Clear to Send This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read. 0: no change on ctsdsr_n since last read of MSR 1: change on ctsdsr_n since last read of MSR Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] = 1), DCTS reflects changes on MCR[1] (RTS). **Note: If the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or otherwise), then the DCTS bit is set when the reset is removed if the cts_n signal remains asserted.**

9.2.6.12. 0x001C UART Scratch Register(Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: UART_SCH
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	SCRATCH_REG Scratch Register This register is for programmers to use as a temporary storage space. It has no defined purpose in the UART.

9.2.6.13. 0x007C UART Status Register(Default Value: 0x0000_0006)

Offset: 0x007C			Register Name: UART_USR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/

4	R	0x0	RFF Receive FIFO Full This is used to indicate that the receive FIFO is completely full. 0: Receive FIFO not full 1: Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.
3	R	0x0	RFNE Receive FIFO Not Empty This is used to indicate that the receive FIFO contains one or more entries. 0: Receive FIFO is empty 1: Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.
2	R	0x1	TFE Transmit FIFO Empty This is used to indicate that the transmit FIFO is completely empty. 0: Transmit FIFO is not empty 1: Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty.
1	R	0x1	TFNF Transmit FIFO Not Full This is used to indicate that the transmit FIFO in not full. 0: Transmit FIFO is full 1: Transmit FIFO is not full This bit is cleared when the TX FIFO is full.
0	R	0x0	BUSY UART Busy Bit 0: Idle or inactive 1: Busy

9.2.6.14. 0x0080 UART Transmit FIFO Level Register(Default Value: 0x0000_0000)

Offset: 0x0080			Register Name: UART_TFL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8:0	R	0x0	TFL Transmit FIFO Level The bit indicates the number of data entries in the transmit FIFO.

9.2.6.15. 0x0084 UART Receive FIFO Level Register(Default Value: 0x0000_0000)

Offset: 0x0084			Register Name: UART_RFL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/

8:0	R	0x0	RFL Receive FIFO Level The bit indicates the number of data entries in the receive FIFO.

9.2.6.16. 0x0088 UART DMA Handshake Configuration Register(Default Value: 0x0000_00E5)

Offset: 0x0088			Register Name: UART_HSK
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0xE5	Handshake configuration 0xA5: DMA wait cycle mode 0xE5: DMA handshake mode

9.2.6.17. 0x00A4 UART Halt TX Register(Default Value: 0x0000_0000)

Offset: 0x00A4			Register Name: UART_HALT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	PTE The sending of TX_REQ. In DMA1 mode (FIFO on), if PTE is set to 1, when TFL is less than trig, send the DMA request. If PTE is set to 0, when FIFO is empty, send the DMA request. The DMA request will stop when FIFO is full. In DMA0 mode, if PTE is set to 1 and FIFO is on, when TFL is less than trig, send DMA request. If PTE is set to 1 and FIFO is off, when THRE is empty, send DMA request. If PTE is set to 0, when FIFO is empty, send DMA request.
6	R/W	0x0	DMA_PTE_RX The sending of RX_DRQ. In DMA1 mode, when RFL is more than or equal to trig or receive timeout, send DRQ. In DMA0 mode, if DMA_PTE_RX is 1 and FIFO is on, when RFL is more than trig, send DRQ. In other cases, once the receive data is valid, send DRQ.
5	R/W	0x0	SIR_RX_INVERT SIR Receiver Pulse Polarity Invert 0: Not invert receiver signal 1: Invert receiver signal
4	R/W	0x0	SIR_TX_INVERT SIR Transmit Pulse Polarity Invert 0: Not invert transmit pulse 1: Invert transmit pulse

3	/	/	/
2	R/WAC	0x0	CHANGE_UPDATE After the user uses HALT[1] to change the baud rate or LCR configuration, write 1 to update the configuration and wait this bit to self-clear to 0 to finish update process. Writing 0 to this bit has no effect. 1: Update trigger, Self clear to 0 when finish update.
1	R/W	0x0	CHCFG_AT_BUSY This is an enable bit for the user to change LCR register configuration and baud rate register (DLH and DLL) when the UART is busy. 1: Enable change when busy
0	R/W	0x0	HALT_TX Halt TX This register is used to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 0 : Halt TX disabled 1 : Halt TX enabled **Note: If FIFOs are not enabled, the setting has no effect on operation.**

9.2.6.18. 0x00B0 UART DBG DLL Register(Default Value: 0x0000_0000)

Offset: 0x00B0			**Register Name: UART_DBG_DLL**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	DEBUG DLL

9.2.6.19. 0x00B4 UART DBG DLH Register(Default Value: 0x0000_0000)

Offset: 0x00B4			**Register Name: UART_DBG_DLH**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	DEBUG DLH

9.2.6.20. 0x00C0 UART RS485 Control and Status Register(Default Value: 0x0000_0000)

Offset: 0x00C0			**Register Name: UART_485_CTL**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	Reserved
6	R/W1C	0x0	AAD_ADDR_F In AAD mode, when UART receives an address byte and the byte is the same as RS485_ADDR_MATCH, this bit will be set to 1. If RS485 interrupt is enabled, the RS485 interrupt will arrive.

			Write 1 to clear this bit and reset the RS485 interrupt.
5	R/W1C	0x0	RS485_ADDR_DET_F This is a flag of the detected address bytes. When UART receives an address byte, this bit will be set to 1. If the RS485 Interrupt is enabled, the RS485 interrupt will arrive. 1:An address byte is detected 0:No address byte is detected Write 1 to clear this bit and reset the RS485 interrupt.
4	/	/	/
3	R/W	0x0	RX_BF_ADDR In NMM mode, If setting this bit to 1, UART will receive all the bytes into FIFO before receiving an address byte. If setting to 0, it will not. 1:Receive 0:Not Receive
2	R/W	0x0	RX_AF_ADDR In NMM mode, if setting this bit to 1, UART will receive all the bytes into FIFO after receiving an address byte. If setting to 0, it will not. 1:Receive 0:Not Receive
1:0	R/W	0x0	RS485_SLAVE_MODE_SEL RS485 Slave Mode 00: Normal Multidrop Operation(NMM) 01: Auto Address Detection Operation(AAD) 10: Reserved 11: Reserved

9.2.6.21. 0x00C4 UART RS485 Address Match Register(Default Value: 0x0000_0000)

Offset: 0x00C4			**Register Name: RS485_ADDR_MATCH**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	ADDR_MATCH The matching address uses in AAD mode. **Note: It is only available for AAD.**

9.2.6.22. 0x00C8 UART RS485 Bus Idle Check Register(Default Value: 0x0000_0000)

Offset: 0x00C8			**Register Name: BUS_IDLE_CHK**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	BUS_IDLE_CHK_EN 0: Disable bus idle check function 1: Enable bus idle check function

6	R	0x0	BUS_STATUS The Flag of Bus Status 0:Idle 1:Busy
5:0	R	0x0	ADJ_TIME Bus Idle Time The unit is 8*16*Tclk.

9.2.6.23. 0x00CC UART TX Delay Register(Default Value: 0x0000_0000)

Offset: 0x00CC			Register Name: TX_DLY
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	DLY The delay time between the last stop bit and the next start bit. The unit is 16*Tclk. It is used to control the space between two bytes in TX.

9.3. SPI

9.3.1. Overview

The SPI is a full-duplex, synchronous, serial communication interface which allows rapid data communication with fewer software interrupts. The SPI controller contains one 64x8 bits receiver buffer (RXFIFO) and one 64x8 bits transmit buffer (TXFIFO). It can work at master mode and slave mode.

The SPI has the following features:

- Full-duplex synchronous serial interface
- Master/slave configurable
- 8-bit wide by 64-entry FIFO for both transmit and receive data
- Polarity and phase of the Chip Select (SPI_SS) and SPI Clock (SPI_SCLK) are configurable
- Interrupt or DMA support
- Supports mode0, mode1, mode2 and mode3
- Supports 3-wire/4-wire SPI
- Supports programmable serial data frame length: 1 bit to 32 bits
- Supports the SPI NAND flash and SPI NOR flash
- Supports standard SPI, dual-output/dual-input SPI, dual I/O SPI, quad-output/quad-input SPI
- Supports maximum IO rate: 100 MHz

9.3.2. Block Diagram

Figure 9-19 shows a block diagram of the SPI.

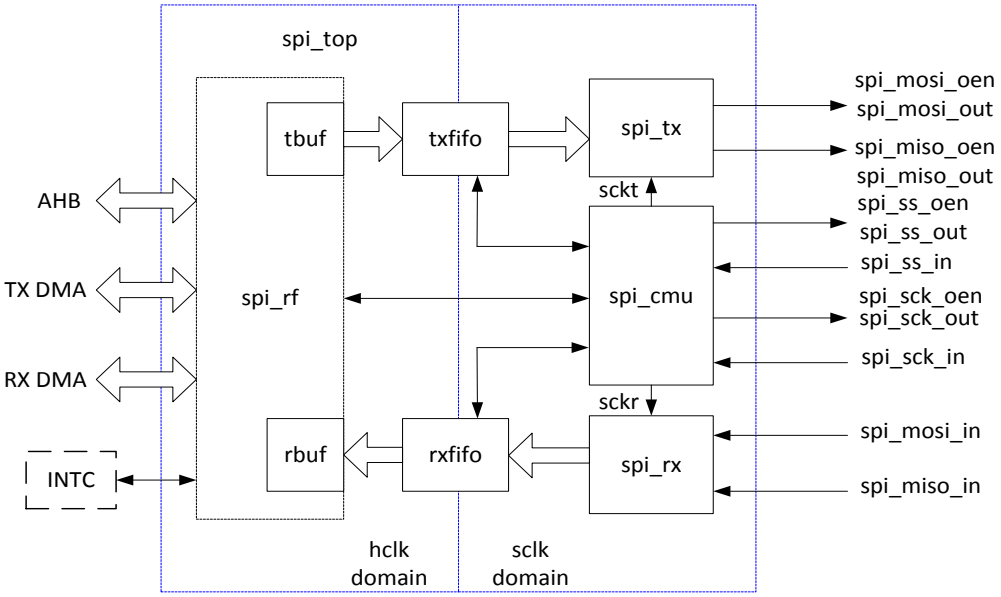

Figure 9- 19. SPI Block Diagram

The SPI comprises with:
spi_rf: Responsible for implementing the internal register, interrupt and DMA Request.
spi_tbuf: The data length transmitted from AHB to txfifo is converted into 8bits,then the data is written into the rxfifo.
spi_rbuf: The block is used to convert the rxfifo data into read data length of AHB.
txfifo, rxfifo: For transmit and receive transfers, data transmitted from the SPI to the external serial device is written into the txfifo; data received from the external serial device into SPI is pushed into the rxfifo.
spi_cmu: Responsible for implementing SPI bus clock, chip select, internal sample and the generation of transfer clock.
spi_tx: Responsible for implementing SPI data transfer, the interface of the internal txfifo and status register.
spi_rx: Responsible for implementing SPI data receive, the interface of the internal rxfifo and status register.

9.3.3. Operations and Functional Descriptions

9.3.3.1. External Signals

Table 9-8 describes the external signals of SPI. MOSI and MISO are bidirectional I/O, when SPI is configured as master device, CLK and CS is output pin; when SPI is configurable as slave device, CLK and CS is input pin. The unused SPI ports are used as General Purpose I/O ports.

Table 9- 8. SPI External Signals

Signal	Description	Type
SPI0_CS0	SPI0 Chip Select Signal0, Low Active	I/O
SPI0_CS1	SPI0 Chip Select Signal1, Low Active	I/O
SPI0_CLK	SPI0 Clock Signal	I/O
SPI0_MOSI	SPI0 Master Data Out, Slave Data In	I/O
SPI0_MISO	SPI0 Master Data In, Slave Data Out	I/O
SPI0_WP	Write protection and active low or Serial Data Input and Output for Quad Input or Quad Output	I/O
SPI0_HOLD	The HOLD pin is used to temporarily pause serial communication without deselecting or resetting the device. While the HOLD pin is asserted, transitions on the SCK pin and data on the SI pin will be ignored, or Serial Data Input and Output for Quad Input or Quad Output	I/O
SPI1_CS0	SPI1 Chip Select Signal0, Low Active	I/O
SPI1_CS1	SPI1 Chip Select Signal1, Low Active	I/O
SPI1_CLK	SPI1 Clock Signal	I/O
SPI1_MOSI	SPI1 Master Data Out, Slave Data In	I/O
SPI1_MISO	SPI1 Master Data In, Slave Data Out	I/O

9.3.3.2. Clock Sources

The SPI controller get 5 different clock sources, users can select one of them to make SPI clock source. Table 9-9

describes the clock sources for SPI.

Table 9- 9. SPI Clock Sources

Clock Sources	Description
OSC24M	24 MHz Crystal
PLL_PERIPH0(1X)	Peripheral Clock, default value is 600 MHz
PLL_PERIPH1(1X)	Peripheral Clock, default value is 600 MHz
PLL_PERIPH0(2X)	Peripheral Clock, default value is 1200 MHz
PLL_PERIPH1(2X)	Peripheral Clock, default value is 1200 MHz

9.3.3.3. Typical Application

Figure 9-20 shows the application block diagram when the SPI master device is connected to a slave device.

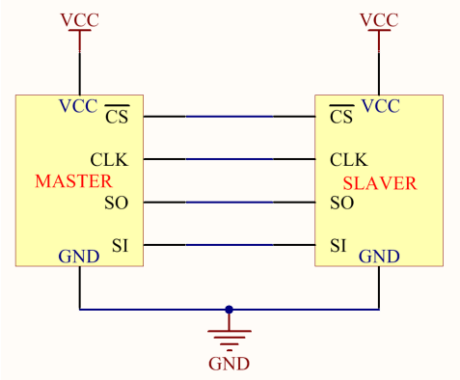

Figure 9- 20. SPI Application Block Diagram

9.3.3.4. SPI Transmit Format

The SPI supports 4 different formats for data transfer. Software can select one of the four modes in which the SPI works by setting the bit1(Polarity) and bit0(Phase) of **SPI Transfer Control Register**. The SPI controller master uses the SPI_SCLK signal to transfer data in and out of the shift register. Data is clocked using any one of four programmable clock phase and polarity combinations.

During Phase 0, Polarity 0 and Phase 1, Polarity 1 operations, output data changes on the falling clock edge and input data is shifted in on the rising edge.

During Phase 1, Polarity 0 and Phase 0, Polarity 1 operations, output data changes on the rising edges of the clock and is shifted in on falling edges.

The POL defines the signal polarity when SPI_SCLK is in idle state. The SPI_SCLK is high level when POL is ‘1’ and it is low level when POL is ‘0’. The PHA decides whether the leading edge of SPI_SCLK is used for setup or sample data. The leading edge is used for setup data when PHA is ‘1’ and for sample data when PHA is ‘0’. The four modes are listed in

Table 9-10.

Table 9- 10. SPI Transmit Format

SPI Mode	POL	PHA	Leading Edge	Trailing Edge
0	0	0	Rising, Sample	Falling, Setup
1	0	1	Rising, Setup	Falling, Sample
2	1	0	Falling, Sample	Rising, Setup
3	1	1	Failing, Setup	Rising, Sample

Figure 9-21 and Figure 9-22 describe four waveforms for SPI_SCLK.

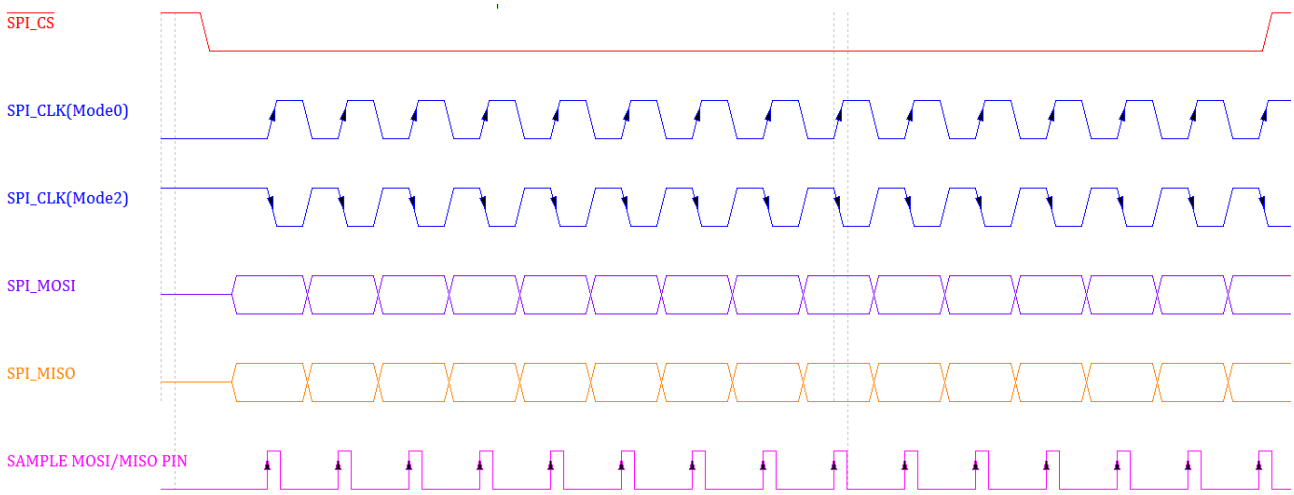

Figure 9- 21. SPI Phase 0 Timing Diagram

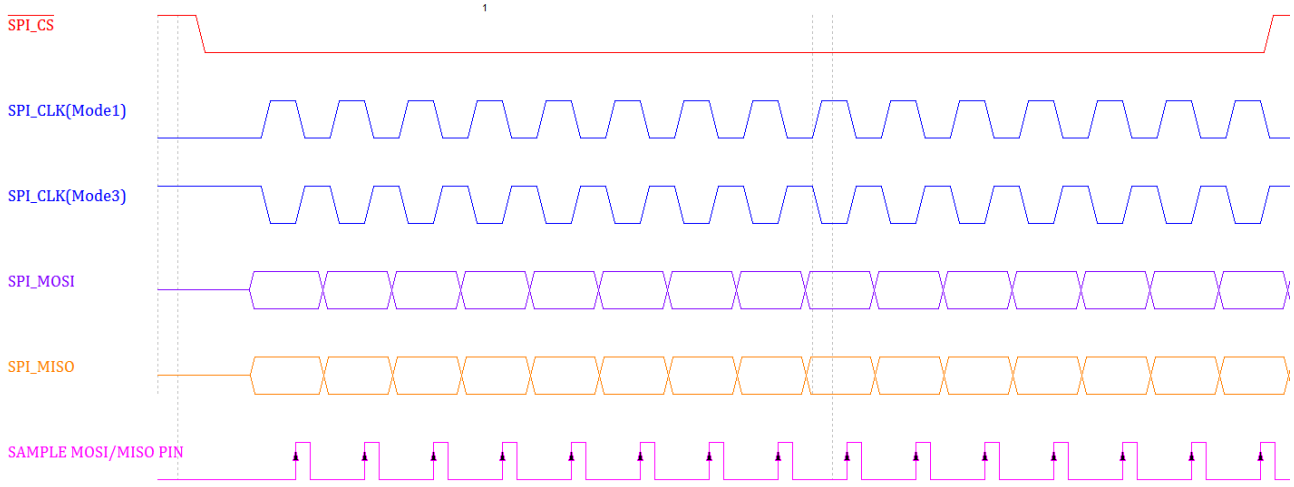

Figure 9- 22. SPI Phase 1 Timing Diagram

9.3.3.5. SPI Master and Slave Mode

The SPI controller can be configured to a master or slave device. Master mode is selected by setting the **MODE** bit in

the **SPI Global Control Register;** slave mode is selected by clearing the **MODE** bit in the **SPI Global Control Register**.

In master mode, SPI_CLK is generated and transmitted to external device, and data from the TX FIFO is transmitted on the MOSI pin, the data from slave is received on the MISO pin and sent to RX FIFO. Chip Select(SPI_SS) is active low signal. SPI_SS must be set low before data are transmitted or received. SPI_SS can be selected SPI auto control or software manual control. When using auto control, **SS_OWNER**(the bit 6 in the **SPI Transfer Control Register**) must be cleared(default value is 0);when using manual control, **SS_OWNER** must be set, Chip Select level is controlled by **SS_LEVEL** bit(the bit 7 in the **SPI Transfer Control Register**).

In slave mode, after software selects the **MODE** bit to '0',it waits for master initiate a transaction. When the master asserts SPI_SS and SPI_CLK is transmitted to the slave, the slave data is transmitted from TX FIFO on MISO pin and data from MOSI pin is received in RX FIFO.

9.3.3.6. SPI 3-Wire Mode

The SPI 3-wire mode is only valid when the SPI controller work in master mode, and is selected when the **Work Mode Select(bit[1:0])** is equal to 0x2 in the **SPI Bit-Aligned Transfer Configure Register**. And in the 3-wire mode, the input data and the output data use the same single data line. The following figure describes this mode.

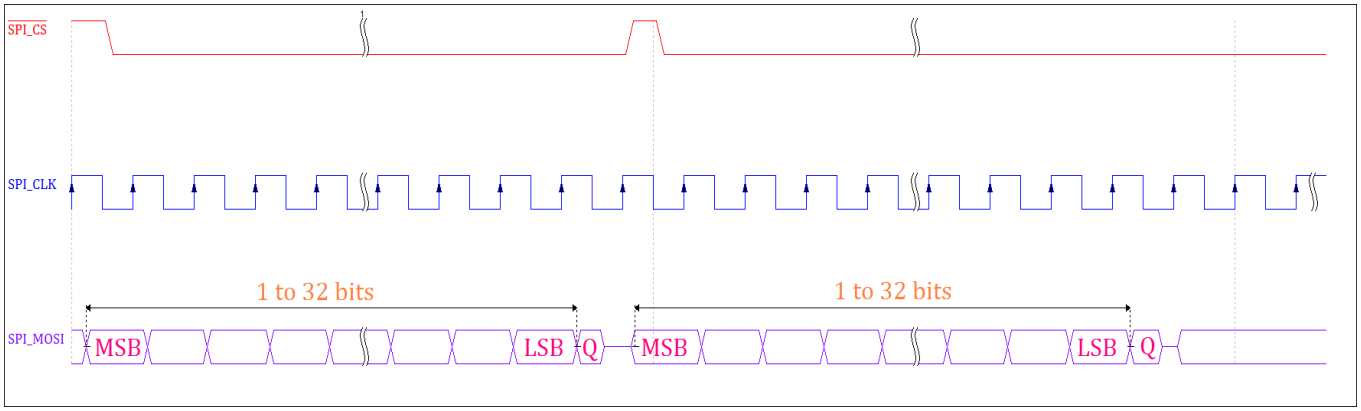

Figure 9- 23. SPI 3-Wire Mode

9.3.3.7. SPI Dual-Input/Dual-Output and Dual I/O Mode

The dual read mode(SPI x2) is selected when the **DRM**(bit28) is set in the **SPI Master Burst Control Counter Register**. Using the dual mode allows data to be transferred to or from the device at double the rate of standard single mode SPI devices, data can be read at fast speed using two data bits(MOSI and MISO) at a time. The following figure describes the dual-input/dual-output SPI(Figure 9-24) and the dual I/O SPI(Figure 9-25).

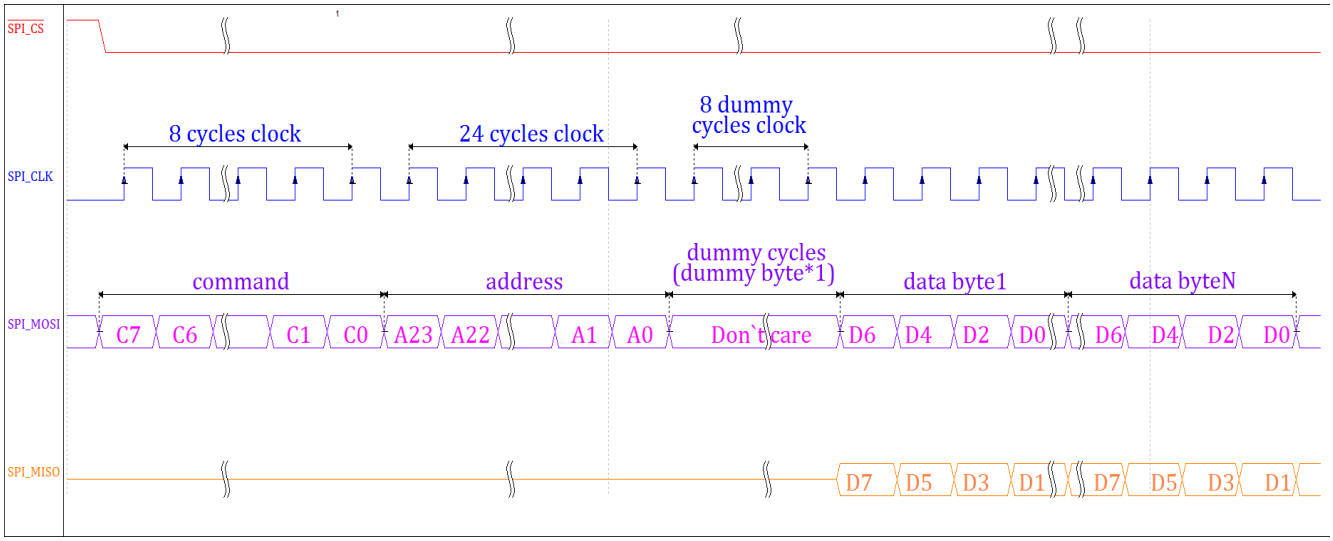

Figure 9- 24. SPI Dual-Input/Dual-Output Mode

In the dual-input/dual-output SPI, the command, address, and the dummy bytes output in unit of a single bit in serial mode through SPI_MOSI line, only the data bytes are output(write) and input(read) in unit of dual bits through the SPI_MOSI and SPI_MISO.

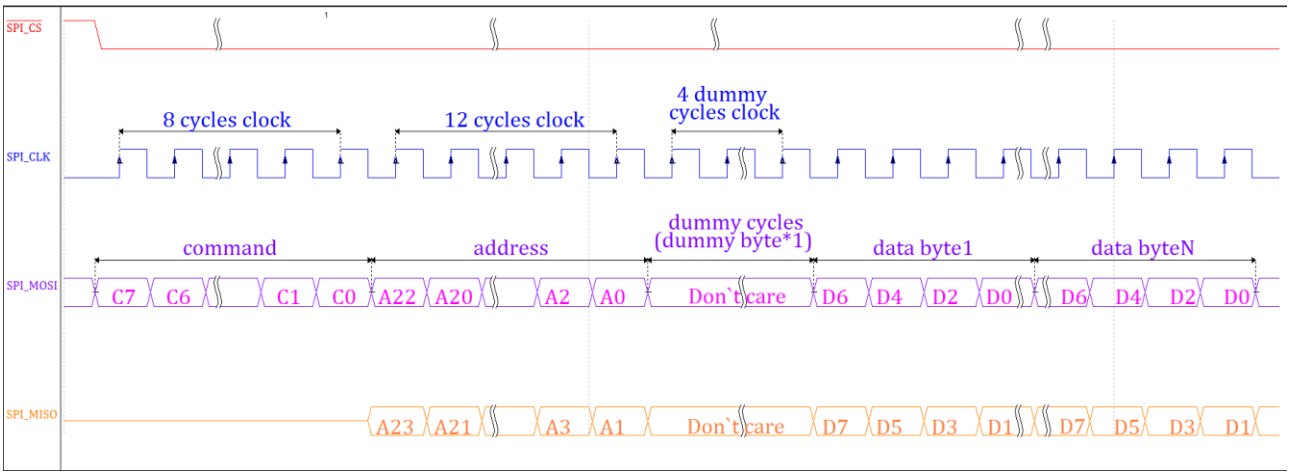

Figure 9- 25. SPI Dual I/O Mode

In the dual I/O SPI, only the command bytes are output in unit of a single bit in serial mode through SPI_MOSI line. The address bytes and the dummy bytes are output in unit of dual bits through the SPI_MOSI and SPI_MISO. And the data bytes are output(write) and input(read) in unit of dual bits through the SPI_MOSI and SPI_MISO.

9.3.3.8. SPI Quad-Input/Quad-Output Mode

The quad read mode(SPI x4) is selected when the **Quad_EN**(bit29) is set in the **SPI Master Burst Control Counter Register**. Using the quad mode allows data to be transferred to or from the device at 4 times the rate of standard single mode SPI devices, data can be read at fast speed using four data bits(MOSI, MISO, IO2(WP#)and IO3(HOLD#)) at the same time. The following figure describes the quad-input/quad-output SPI.

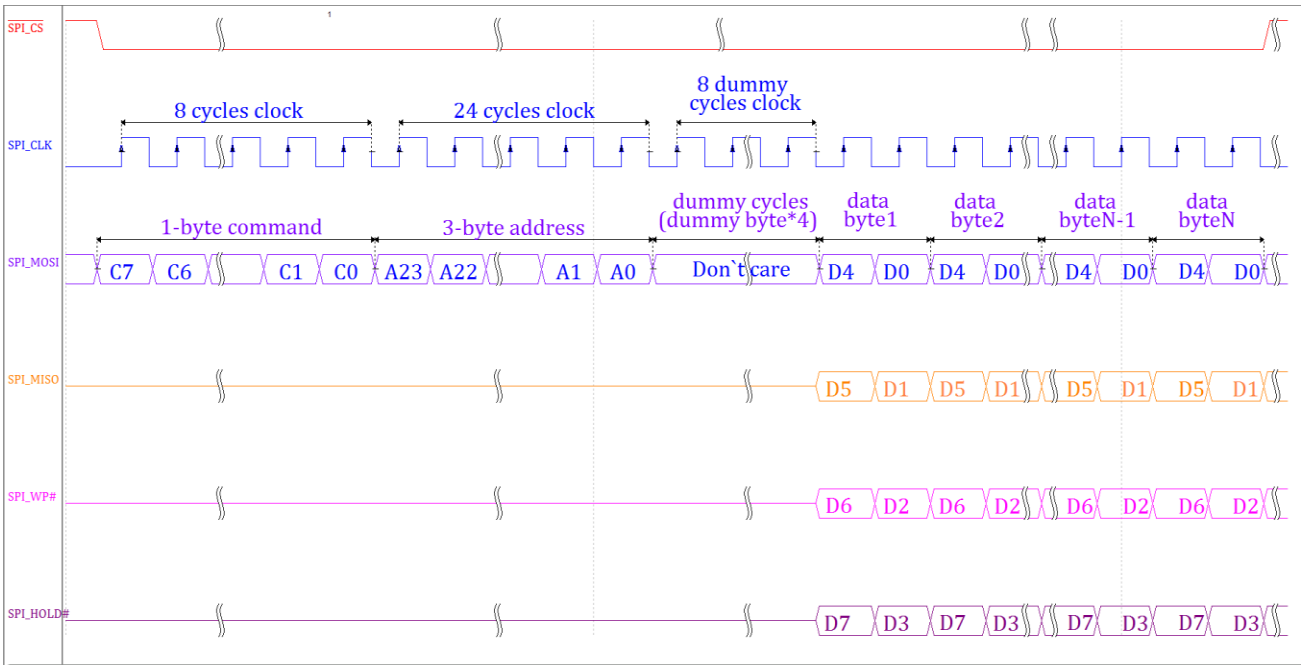

Figure 9- 26. SPI Quad-Input/Quad-Output Mode

In the quad-input/quad-output SPI, the command, address, and the dummy bytes are output in unit of a single bit in serial mode through SPI_MOSI line. Only the data bytes are output(write) and input(read) in unit of quad bits through the SPI_MOSI, SPI_MISO, SPI_WP# and SPI_HOLD#.

9.3.3.9. Transmit/Receive Burst in Master Mode

In SPI master mode, the transmit and receive burst(byte in unit) are configured before the SPI transfers serial data between the processor and external device. The transmit bursts are written in MWTC(bit[23:0]) of **SPI Master Transmit Counter Register.** The transmit bursts in single mode before automatically sending dummy burst are written in STC(bit[23:0]) of **SPI Master Burst Control Counter Register.** For dummy data, SPI controller can automatically sent before receiving by writing DBC(bit[27:24]) in **SPI Master Burst Control Counter Register**. If users donot use SPI controller to sent dummy data automatically, then the dummy bursts are used as the transmit counters to write together in MWTC(bit[23:0]) of **SPI Master Transmit Counter Register.** In master mode, the total burst numbers are written in MBC(bit[23:0]) of **SPI Master Burst Counter Register.** When all transmit burst and receive burst are transferred, SPI controller will send an completed interrupt, at the same time, SPI controller will clear DBC,MWTC and MBC.

9.3.3.10. SPI Sample Mode and Run Clock Configuration

The SPI controller runs at 3 kHz~100 MHz at its interface to external SPI devices. The internal SPI clock should run at the same frequency as the outgoing clock in master mode. The SPI clock is selected different clock sources, SPI must configure different work mode. There are three work modes: normal sample mode, delay half cycle sample mode, delay one cycle sample mode. Delay half cycle sample mode is the default mode of SPI controller. When SPI runs at 40 MHz or below 40 MHz, SPI can work at normal sample mode or delay half cycle sample mode. When SPI runs over 60 MHz, setting the **SDC** bit in **SPI Transfer Control Register** to ‘1’ makes the internal read sample point with a half cycle

delay of SPI_CLK, which is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK propagating between master and slave. The different configuration of SPI sample mode shows in Table 9-11.

Table 9- 11. SPI Sample Mode and Run Clock

SPI Sample Mode	SDM(bit13)	SDC(bit11)	Run Clock
normal sample	1	0	<=24 MHz
delay half cycle sample	0	0	<=40 MHz
delay one cycle sample	0	1	>=60 MHz

9.3.3.11. SPI Error Conditions

If any error conditions occur, hardware will set the corresponding status bits in the **SPI Interrupt Status Register** and stop the transfer. For the SPI controller, the following error scenarios can happen.

(1) TX_FIFO Underrun

TX_FIFO underrun happens when the CPU/DMA reads from TX FIFO when it is empty. In the case, the SPI controller will end the transaction and flag the error bit along with the TF_UDF bit in the **SPI Interrupt Status Register**. The SPI controller will generate an interrupt if interrupts are enabled. Software has to clear the error bit and the TF_UDF bit. To start a new transaction, software has to reset the FIFO by writing to the SRST(soft reset) bit in the **SPI Global Controller Register.**

(2) TX_FIFO Overflow

TX_FIFO overflow happens when the CPU/DMA writes into the TX FIFO when it is full. In the case, the SPI controller will end the transaction and flag the error bit along with the TF_OVF bit in the **SPI Interrupt Status Register**. The SPI controller will generate an interrupt if interrupts are enabled. Software has to clear the error bit and the TF_OVF bit. To start a new transaction, software has to reset the FIFO by writing to the SRST(soft reset) bit in the **SPI Global Controller Register.**

(3) RX_FIFO Underrun

RX_FIFO underrun happens when the CPU/DMA reads from RX FIFO when it is empty. In the case, the SPI controller will end the transaction and flag the error bit along with the RF_UDF bit in the **SPI Interrupt Status Register**. The SPI controller will generate an interrupt if interrupts are enabled. Software has to clear the error bit and the RF_UDF bit. To start a new transaction, software has to reset the FIFO by writing to the SRST(soft reset) bit in the **SPI Global Controller Register.**

(4) RX_FIFO Overflow

RX_FIFO overflow happens when the CPU/DMA writes into the RX FIFO when it is full. In the case, the SPI controller will end the transaction and flag the error bit along with the RF_OVF bit in the **SPI Interrupt Status Register**. The SPI controller will generate an interrupt if interrupts are enabled. Software has to clear the error bit and the RF_OVF bit. To start a new transaction, software has to reset the FIFO by writing to the SRST(soft reset) bit in the **SPI Global Controller Register.**

9.3.4. Programming Guidelines

9.3.4.1. CPU or DMA Operation

The SPI transfers serial data between the processor and external device. CPU and DMA are the two main operational modes for SPI. For each SPI, data is simultaneously transmitted(shifted out serially) and received (shifted in serially).SPI has 2 channels, TX channel and RX channel. TX channel has the path from TX FIFO to external device. RX channel has the path from external device to RX FIFO.

Write Data: CPU or DMA must write data on the register SPI_TXD, data on the register are automatically moved to TX FIFO.

Read Data: To read data from RX FIFO, CPU or DMA must access the register SPI_RXD and data are automatically sent to the register SPI_RXD.

In CPU or DMA mode, the SPI sends an completed interrupt(the TC bit in SPI Interrupt Status Register) to the processor at the end of each transfer.

(1).CPU Mode

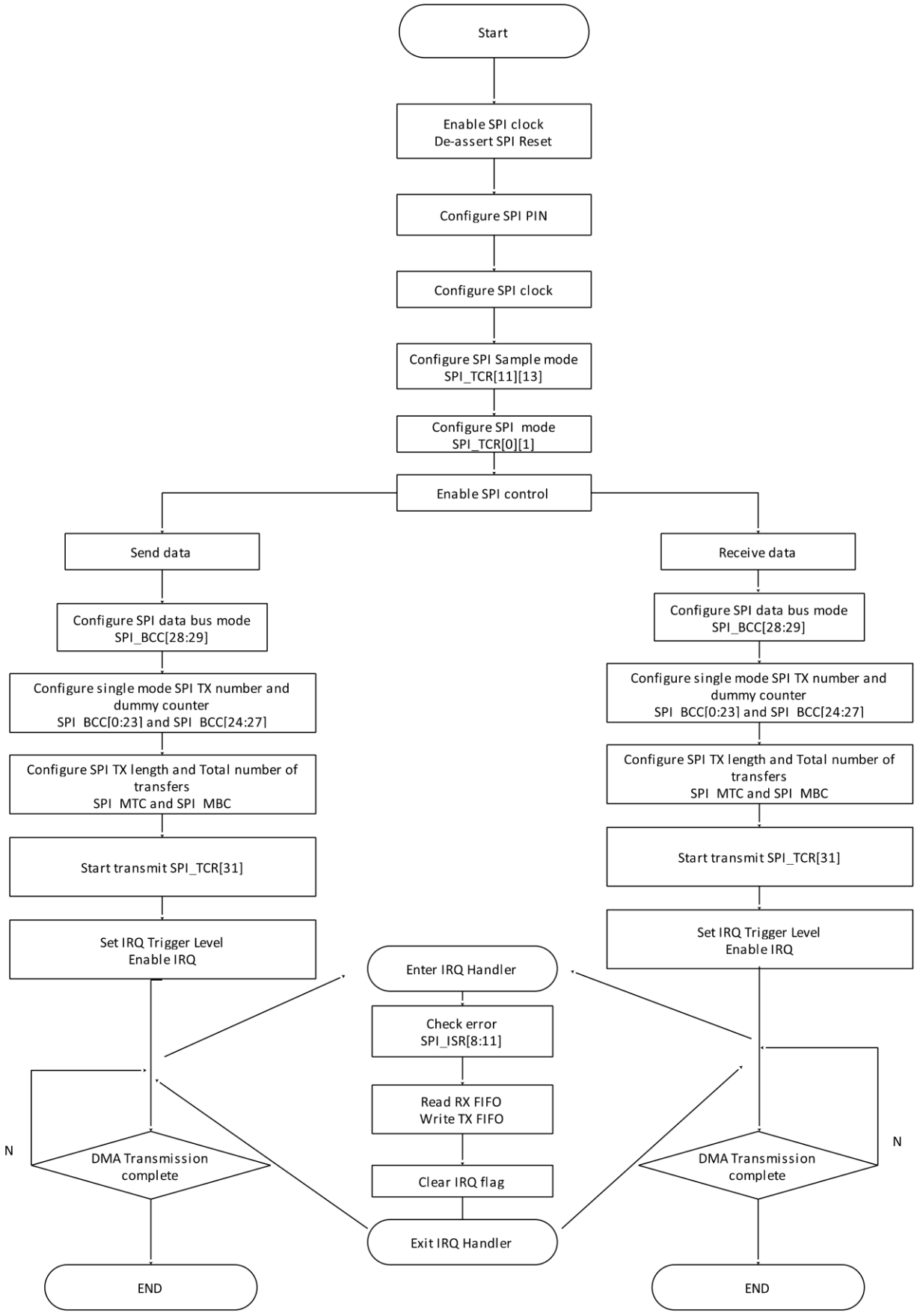

Figure 9- 27. SPI Write/Read Data in CPU Mode

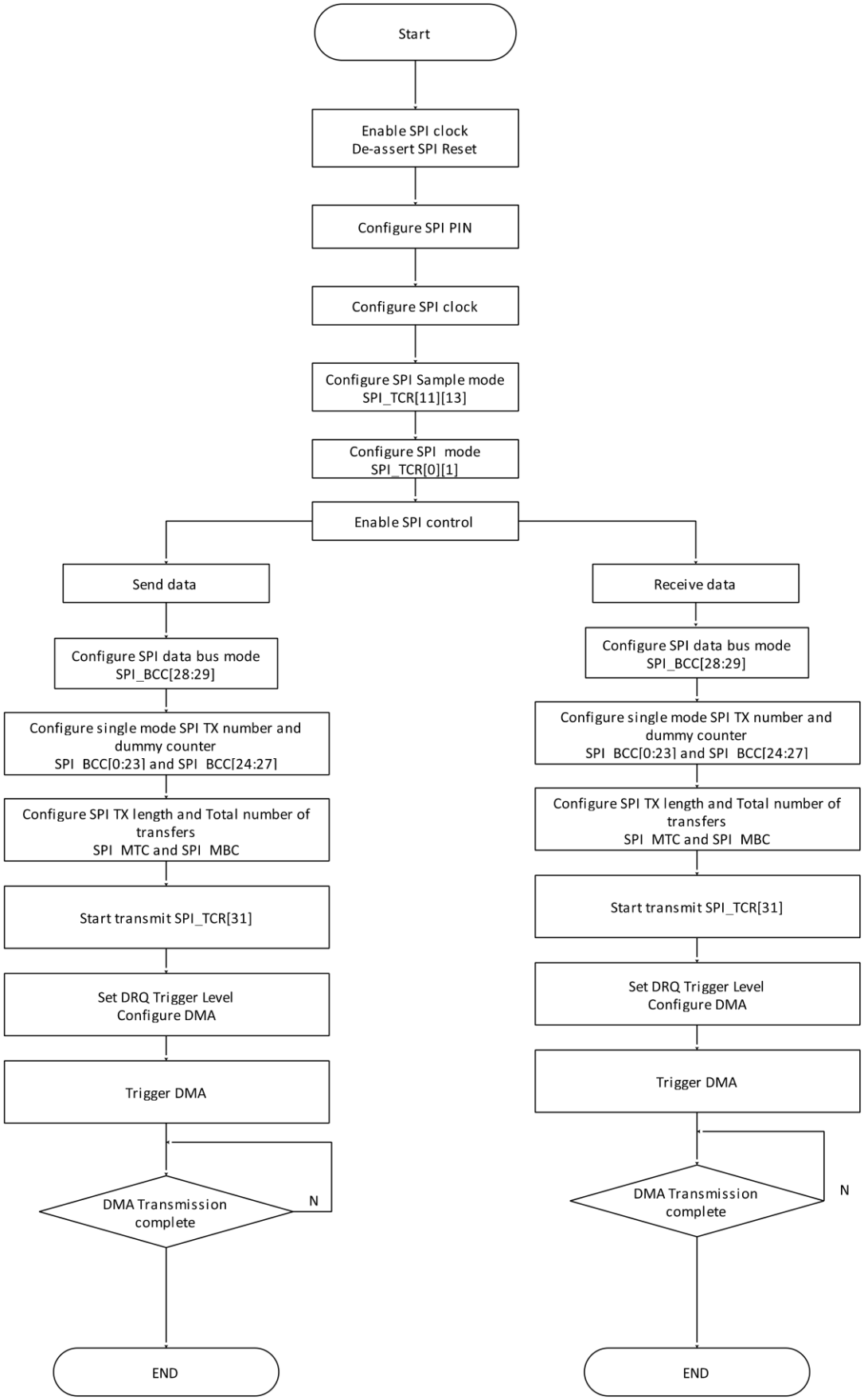

Figure 9- 28. SPI Write/Read Data in DMA Mode

9.3.4.2. Calibrate Delay Chain

The SPI has one delay chain, which is used to generate delay to make proper timing between internal SPI clock signal and data signals. Delay chain is made up with 64 delay cells. The delay time of one delay cell can be estimated through delay chain calibration.

The steps to calibrate delay chain are as follows:

Step1: Enable SPI. In order to calibrate delay chain by operation registers in SPI, SPI must be enabled through AHB reset and AHB clock gating control registers.

Step2: Configure a proper clock for SPI. Calibration delay chain is based on the clock for SPI from CCU.

Step3: Set proper initial delay value. Write 0xA0 to delay control register to set initial delay value 0x20 to delay chain. Then write 0x0 to delay control register to clear this value.

Step4: Write 0x8000 to delay control register to start calibrate delay chain.

Step5: Wait until the flag(Bit14 in delay control register) of calibration done is set. The number of delay cells is shown at Bit8~Bit14 in delay control register. The delay time generated by these delay cells is equal to the cycle of SPI's clock nearly. This value is the result of calibration.

Step6: Calculate the delay time of one delay cell according to the cycle of SPI's clock and the result of calibration.

9.3.5. Register List

Module Name	Base Address
SPI0	0x05010000
SPI1	0x05011000

Register Name	Offset	Description
SPI_GCR	0x0004	SPI Global Control Register
SPI_TCR	0x0008	SPI Transfer Control Register
SPI_IER	0x0010	SPI Interrupt Control Register
SPI_ISR	0x0014	SPI Interrupt Status Register
SPI_FCR	0x0018	SPI FIFO Control Register
SPI_FSR	0x001C	SPI FIFO Status Register
SPI_WCR	0x0020	SPI Wait Clock Counter Register
SPI_CCR	0x0024	SPI Clock Rate Control Register
SPI_SAMP_DL	0x0028	SPI Sample Delay Control Register
SPI_MBC	0x0030	SPI Burst Counter Register

SPI_MTC	0x0034	SPI Transmit Counter Register
SPI_BCC	0x0038	SPI Burst Control Register
SPI_BATCR	0x0040	SPI Bit-Aligned Transfer Configure Register
SPI_3W_CCR	0x0044	SPI Bit-Aligned CLOCK Configuration Register
SPI_TBR	0x0048	SPI TX Bit Register
SPI_RBR	0x004C	SPI RX Bit Register
SPI_NDMA_MODE_CTL	0x0088	SPI Normal DMA Mode Control Register
SPI_TXD	0x0200	SPI TX Data Register
SPI_RXD	0x0300	SPI RX Data Register

9.3.6. Register Description

9.3.6.1. 0x0004 SPI Global Control Register(Default Value: 0x0000_0080)

Offset:0x0004			**Register Name: SPI_GCR**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/WAC	0x0	SRST Soft reset Writing '1' to this bit will clear the SPI controller, and auto clear to '0' when reset operation completes. Writing '0' has no effect.
30:8	/	/	/
7	R/W	0x1	TP_EN Transmit Pause Enable In master mode, it is used to control transmit state machine to stop smart burst sending when RX FIFO is full. 0: Normal operation, ignore RXFIFO status 1: Stop transmit data when RXFIFO full Cannot be written when XCH=1
6:2	/	/	/
1	R/W	0x0	MODE SPI Function Mode Select 0: Slave mode 1: Master mode Cannot be written when XCH=1
0	R/W	0x0	EN SPI Module Enable Control 0: Disable 1: Enable After transforming from bit_mode to byte_mode, it must enable the SPI module again.

9.3.6.2. 0x0008 SPI Transfer Control Register(Default Value: 0x0000_0087)

Offset: 0x0008			Register Name: SPI_TCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/WAC	0x0	XCH Exchange Burst In master mode it is used to start SPI burst 0: Idle 1: Initiates exchange. Writing “1” to this bit will start the SPI burst, and will auto clear after finishing the bursts transfer specified by BC. Writing “1” to SRST will also clear this bit. Writing ‘0’ to this bit has no effect. Cannot be written when XCH=1.
30:15	/	/	/
14	R/W	0x0	SDDM Sending Data Delay Mode 0:Normal sending 1:Delay sending Set the bit to "1" to make the data that should be sent with a delay of half cycle of SPI_CLK in dual IO mode for SPI mode 0. Cannot be written when XCH=1.
13	R/W	0x0	SDM Master Sample Data Mode 0: Delay sample mode 1: Normal sample mode In normal sample mode, SPI master samples the data at the correct edge for each SPI mode; In delay sample mode, SPI master samples data at the edge that is half cycle delayed by the correct edge defined in respective SPI mode. Cannot be written when XCH=1.
12	R/W	0x0	FBS First Transmit Bit Select 0: MSB first 1: LSB first Cannot be written when XCH=1.
11	R/W	0x0	SDC Master Sample Data Control Set this bit to ‘1’ to make the internal read sample point with a delay of half cycle of SPI_CLK. It is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK propagating between master and slave. 0: Normal operation, do not delay internal read sample point 1: Delay internal read sample point Cannot be written when XCH=1.
10	R/W	0x0	RPSM Rapids Mode Select

			Select rapid mode for high speed write. 0: Normal write mode 1: Rapid write mode Cannot be written when XCH=1.
9	R/W	0x0	DDB Dummy Burst Type 0: The bit value of dummy SPI burst is zero 1: The bit value of dummy SPI burst is one Cannot be written when XCH=1.
8	R/W	0x0	DHB Discard Hash Burst In master mode it controls whether discarding unused SPI bursts 0: Receiving all SPI bursts in BC period 1: Discard unused SPI bursts, only fetching the SPI bursts during dummy burst period. The bursts number is specified by TC. Cannot be written when XCH=1.
7	R/W	0x1	SS_LEVEL When control SS signal manually (SPI_CTRL_REG.SS_CTRL==1), set this bit to '1' or '0' to control the level of SS signal. 0: set SS to low 1: set SS to high Cannot be written when XCH=1.
6	R/W	0x0	SS_OWNER SS Output Owner Select Usually, controller sends SS signal automatically with data together. When this bit is set to 1, software must manually write SPI_CTL_REG.SS_LEVEL to 1 or 0 to control the level of SS signal. 0: SPI controller 1: Software Cannot be written when XCH=1.
5:4	R/W	0x0	SS_SEL SPI Chip Select Select one of four external SPI Master/Slave Devices 00: SPI_SS0 will be asserted 01: SPI_SS1 will be asserted 10: SPI_SS2 will be asserted 11: SPI_SS3 will be asserted Cannot be written when XCH=1.
3	R/W	0x0	SSCTL In master mode, this bit selects the output wave form for the SPI_SSx signal. Only valid when SS_OWNER = 0. 0: SPI_SSx remains asserted between SPI bursts 1: Negate SPI_SSx between SPI bursts Cannot be written when XCH=1.
2	R/W	0x1	SPOL

			SPI Chip Select Signal Polarity Control 0: Active high polarity (0 = Idle) 1: Active low polarity (1 = Idle) Cannot be written when XCH=1.
1	R/W	0x1	CPOL SPI Clock Polarity Control 0: Active high polarity (0 = Idle) 1: Active low polarity (1 = Idle) Cannot be written when XCH=1.
0	R/W	0x1	CPHA SPI Clock/Data Phase Control 0: Phase 0 (Leading edge for sample data) 1: Phase 1 (Leading edge for setup data) Cannot be written when XCH=1.

9.3.6.3. 0x0010 SPI Interrupt Control Register(Default Value: 0x0000_0000)

Offset: 0x0010			**Register Name: SPI_IER**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:14	/	/	/
13	R/W	0x0	SS_INT_EN SSI Interrupt Enable Chip select signal (SSx) from valid state to invalid state 0: Disable 1: Enable
12	R/W	0x0	TC_INT_EN Transfer Completed Interrupt Enable 0: Disable 1: Enable
11	R/W	0x0	TF_UDR_INT_EN TXFIFO Underrun Interrupt Enable 0: Disable 1: Enable
10	R/W	0x0	TF_OVF_INT_EN TX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
9	R/W	0x0	RF_UDR_INT_EN RXFIFO Underrun Interrupt Enable 0: Disable 1: Enable
8	R/W	0x0	RF_OVF_INT_EN RX FIFO Overflow Interrupt Enable 0: Disable

			1: Enable
7	/	/	/
6	R/W	0x0	TF_FUL_INT_EN TX FIFO Full Interrupt Enable 0: Disable 1: Enable
5	R/W	0x0	TX_EMP_INT_EN TX FIFO Empty Interrupt Enable 0: Disable 1: Enable
4	R/W	0x0	TX_ERQ_INT_EN TX FIFO Empty Request Interrupt Enable 0: Disable 1: Enable
3	/	/	/
2	R/W	0x0	RF_FUL_INT_EN RX FIFO Full Interrupt Enable 0: Disable 1: Enable
1	R/W	0x0	RX_EMP_INT_EN RX FIFO Empty Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	RF_RDY_INT_EN RX FIFO Ready Request Interrupt Enable 0: Disable 1: Enable

9.3.6.4. 0x0014 SPI Interrupt Status Register(Default Value: 0x0000_0032)

Offset: 0x0014			**Register Name: SPI_ISR**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:14	/	/	/
13	R/W1C	0x0	SSI SS Invalid Interrupt When SSI is 1, it indicates that SS has changed from valid state to invalid state. Writing 1 to this bit clears it.
12	R/W1C	0x0	TC Transfer Completed In master mode, it indicates that all bursts specified by BC have been exchanged. In other condition, When set, this bit indicates that all the datas in TXFIFO have been loaded in the Shift register, and the Shift register has shifted out all the bits. Writing 1 to this bit clears it. 0: Busy

			1: Transfer completed
11	R/W1C	0x0	TF_UDF TXFIFO Underrun This bit is set when if the TXFIFO is underrun. Writing 1 to this bit clears it. 0: TXFIFO is not underrun 1: TXFIFO is underrun
10	R/W1C	0x0	TF_OVF TXFIFO Overflow This bit is set when if the TXFIFO is overflow. Writing 1 to this bit clears it. 0: TXFIFO is not overflow 1: TXFIFO is overflowed
9	R/W1C	0x0	RX_UDF RXFIFO Underrun When set, this bit indicates that RXFIFO has underrun. Writing 1 to this bit clears it.
8	R/W1C	0x0	RX_OVF RXFIFO Overflow When set, this bit indicates that RXFIFO has overflowed. Writing 1 to this bit clears it. 0: RXFIFO is available 1: RXFIFO is overflowed
7	/	/	/
6	R/W1C	0x0	TX_FULL TXFIFO Full This bit is set when if the TXFIFO is full . Writing 1 to this bit clears it. 0: TXFIFO is not Full 1: TXFIFO is Full
5	R/W1C	0x1	TX_EMP TXFIFO Empty This bit is set if the TXFIFO is empty. Writing 1 to this bit clears it. 0: TXFIFO contains one or more words. 1: TXFIFO is empty
4	R/W1C	0x1	TX_READY TXFIFO Ready 0: TX_WL > TX_TRIG_LEVEL 1: TX_WL <= TX_TRIG_LEVEL This bit is set any time if TX_WL <= TX_TRIG_LEVEL. Writing “1” to this bit clears it. TX_WL is the water level of TXFIFO.
3	/	/	/
2	R/W1C	0x0	RX_FULL RXFIFO Full This bit is set when the RXFIFO is full . Writing 1 to this bit clears it. 0: Not Full 1: Full
1	R/W1C	0x1	RX_EMP

			RXFIFO Empty This bit is set when the RXFIFO is empty . Writing 1 to this bit clears it. 0: Not empty 1: empty
0	R/W1C	0x0	RX_RDY RXFIFO Ready 0: RX_WL < RX_TRIG_LEVEL 1: RX_WL >= RX_TRIG_LEVEL This bit is set any time if RX_WL >= RX_TRIG_LEVEL. Writing “1” to this bit clears it. RX_WL is the water level of RXFIFO.

9.3.6.5. 0x0018 SPI FIFO Control Register(Default Value: 0x0040_0001)

Offset: 0x0018			**Register Name: SPI_FCR**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/WAC	0x0	TX_FIFO_RST TX FIFO Reset Writing ‘1’ to this bit will reset the control portion of the TX FIFO and auto clear to ‘0’ when completing reset operation, writing to ‘0’ has no effect.
30	R/W	0x0	TF_TEST_ENB TX Test Mode Enable 0: Disable 1: Enable **Note: In normal mode, TX FIFO can only be read by SPI controller, writing ‘1’ to this bit will switch TX FIFO read and write function to AHB bus. This bit is used to test the TX FIFO, donot set in normal operation and donot set RF_TEST and TF_TEST at the same time.**
29:25	/	/	/
24	R/W	0x0	TF_DRQ_EN TX FIFO DMA Request Enable 0: Disable 1: Enable
23:16	R/W	0x40	TX_TRIG_LEVEL TX FIFO Empty Request Trigger Level
15	R/WAC	0x0	RF_RST RXFIFO Reset Writing ‘1’ to this bit will reset the control portion of the receiver FIFO, and auto clear to ‘0’ when completing reset operation, writing ‘0’ to this bit has no effect.
14	R/W	0x0	RF_TEST RX Test Mode Enable 0: Disable 1: Enable **Note: In normal mode, RX FIFO can only be written by SPI controller, writing**

			'1' to this bit will switch RX FIFO read and write function to AHB bus. This bit is used to test the RX FIFO, donot set in normal operation and donot set RF_TEST and TF_TEST at the same time.
13:9	/	/	/
8	R/W	0x0	RF_DRQ_EN RX FIFO DMA Request Enable 0: Disable 1: Enable
7:0	R/W	0x1	RX_TRIG_LEVEL RX FIFO Ready Request Trigger Level

9.3.6.6. 0x001C SPI FIFO Status Register(Default Value: 0x0000_0000)

Offset: 0x001C			**Register Name: SPI_FSR**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x0	TB_WR TX FIFO Write Buffer Write Enable
30:28	R	0x0	TB_CNT TX FIFO Write Buffer Counter These bits indicate the number of words in TX FIFO Write Buffer
27:24	/	/	/
23:16	R	0x0	TF_CNT TX FIFO Counter These bits indicate the number of words in TX FIFO 0: 0 byte in TX FIFO 1: 1 byte in TX FIFO … 64: 64 bytes in TX FIFO other: Reserved
15	R	0x0	RB_WR RX FIFO Read Buffer Write Enable
14:12	R	0x0	RB_CNT RX FIFO Read Buffer Counter These bits indicate the number of words in RX FIFO Read Buffer
11:8	/	/	/
7:0	R	0x0	RF_CNT RX FIFO Counter These bits indicate the number of words in RX FIFO 0: 0 byte in RX FIFO 1: 1 byte in RX FIFO … 64: 64 bytes in RX FIFO other: Reserved

9.3.6.7. 0x0020 SPI Wait Clock Register(Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: SPI_WCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/
19:16	R/W	0x0	SWC Dual mode direction switch wait clock counter (for master mode only). Cannot be written when XCH=1. 0: No wait states inserted n: n SPI_SCLK wait states inserted **Note: These bits control the number of wait states to be inserted before start dual data transfer in dual SPI mode. The SPI module counts SPI_SCLK by SWC for delaying next word data transfer.**
15:0	R/W	0x0	WCC Wait Clock Counter (In master mode) These bits control the number of wait states to be inserted in data transfers. The SPI module counts SPI_SCLK by WCC for delaying next word data transfer. 0: No wait states inserted N: N SPI_SCLK wait states inserted

9.3.6.8. 0x0024 SPI Clock Control Register(Default Value: 0x0000_0002)

Offset: 0x0024			Register Name: SPI_CCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
12	R/W	0x0	DRS Divide Rate Select (Master Mode Only) 0: Select Clock Divide Rate 1 1: Select Clock Divide Rate 2 Cannot be written when XCH=1.
11:8	R/W	0x0	CDR1_M Clock Divide Rate 1 (Master Mode Only) The SPI_SCLK is determined according to the following equation: SPI_CLK = Source_CLK / (2^CDR1_M). Cannot be written when XCH=1.
7:0	R/W	0x2	CDR2_N Clock Divide Rate 2 (Master Mode Only) The SPI_SCLK is determined according to the following equation: SPI_CLK = Source_CLK / (2*(CDR2_N + 1)). Cannot be written when XCH=1.

9.3.6.9. 0x0030 SPI Master Burst Counter Register(Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: SPI_MBC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:0	R/W	0x0	MBC Master Burst Counter In master mode, this field specifies the total burst number. 0: 0 burst 1: 1 burst … N: N bursts Cannot be written when XCH=1. **Note: Total transfer data, includes the TXD, RXD and dummy burst.**

9.3.6.10. 0x0034 SPI Master Transmit Counter Register(Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: SPI_MTC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:0	R/W	0x0	MWTC Master Write Transmit Counter In master mode, this field specifies the burst number that should be sent to TXFIFO before automatically sending dummy burst. For saving bus bandwidth, the dummy burst (all zero bits or all one bits) is sent by SPI Controller automatically. 0: 0 burst 1: 1 burst … N: N bursts Can't be written when XCH=1.

9.3.6.11. 0x0038 SPI Master Burst Control Counter Register(Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: SPI_BCC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29	R/W	0x0	Quad_EN Quad_Mode_EN 0: Quad mode disable 1: Quad mode enable Cannot be written when XCH=1. **Note: Quad mode includes Quad-Input and Quad-Output.**

28	R/W	0x0	DRM Master Dual Mode RX Enable 0: RX use single-bit mode 1: RX use dual mode Cannot be written when XCH=1. It is only valid when Quad_Mode_EN=0.
27:24	R/W	0x0	DBC Master Dummy Burst Counter In master mode, this field specifies the burst number that should be sent before receive in dual SPI mode. The data does not care by the device. 0: 0 burst 1: 1 burst … N: N bursts Cannot be written when XCH=1.
23:0	R/W	0x0	STC Master Single Mode Transmit Counter In master mode, this field specifies the burst number that should be sent in single mode before automatically sending dummy burst. This is the first transmit counter in all bursts. 0: 0 burst 1: 1 burst … N: N bursts Cannot be written when XCH=1.

9.3.6.12. 0x0040 SPI Bit-Aligned Transfer Configure Register(Default Value: 0x0000_00A0)

Offset: 0x0040			**Register Name: SPI_BATC**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/WAC	0x0	TCE Transfer Control Enable In master mode, it is used to start to transfer the serial bits frame, it is only valid when ***Work Mode Select==0x10/0x11.*** 0: Idle 1: Initiates transfer Writing “1” to this bit will start to transfer serial bits frame(the value comes from the ***SPI TX Bit Register*** or ***SPI RX Bit Register***), and will auto clear after the bursts transfer completely. Writing ‘0’ to this bit has no effect.
30	R/W	0x0	MSMS Master Sample Standard 0: Delay Sample Mode 1: Standard Sample Mode

			In Standard Sample Mode, SPI master samples the data at the standard rising edge of SCLK for each SPI mode; In Delay Sample Mode, SPI master samples data at the edge that is half cycle delayed by the standard rising edge of SCLK defined in respective SPI mode.
29:26	/	/	/
25	R/W1C	0x0	TBC Transfer Bits Completed When set, this bit indicates that the last bit of the serial data frame in ***SPI TX Bit Register****(or* ***SPI RX Bit Register****)* has been transferred completely. Writing 1 to this bit clears it. 0: Busy 1: Transfer Completed It is only valid when ***Work Mode Select==0x10/0x11.***
24	R/W	0x0	TBC_INT_EN Transfer Bits Completed Interrupt Enable 0: Disable 1: Enable It is only valid when ***Work Mode Select==0x10/0x11.***
23:22	/	/	/
21:16	R/W	0x00	Configure the length of serial data frame(burst) of RX 000000: 0bit 000001: 1bit … 100000: 32bits Other values: reserved It is only valid when ***Work Mode Select==0x10/0x11,*** and cannot be written when TCE=1.
15:14	/	/	/
13:8	R/W	0x00	Configure the length of serial data frame(burst) of TX 000000: 0bit 000001: 1bit … 100000: 32bits Other values: reserved It is only valid when ***Work Mode Select==0x10/0x11,*** and cannot be written when TCE=1.
7	R/W	0x1	SS_LEVEL When control SS signal manually , set this bit to ‘1’ or ‘0’ to control the level of SS signal. 0: Set SS to low 1: Set SS to high It is only valid when ***Work Mode Select==0x10/0x11,*** and only ***work in Mode0***, cannot be written when TCE=1.
6	R/W	0x0	SS_OWNER SS Output Owner Select

			Usually, controller sends SS signal automatically with data together. When this bit is set to 1, software must manually write SPI_CTL_REG.SS_LEVEL to 1 or 0 to control the level of SS signal. 0: SPI controller 1: Software It is only valid when ***Work Mode Select==0x10/0x11,*** and only ***work in Mode0***, cannot be written when TCE=1.
5	R/W	0x1	SPOL SPI Chip Select Signal Polarity Control 0: Active high polarity (0 = Idle) 1: Active low polarity (1 = Idle) It is only valid when ***Work Mode Select==0x10/0x11,*** and only ***work in Mode0***, cannot be written when TCE=1.
4	/	/	/
3:2	R/W	0x0	SS_SEL SPI Chip Select Select one of four external SPI Master/Slave Devices 00: SPI_SS0 will be asserted 01: SPI_SS1 will be asserted 10: SPI_SS2 will be asserted 11: SPI_SS3 will be asserted It is only valid when ***Work Mode Select= =0x10/0x11,*** and only ***work in Mode0***, cannot be written when TCE=1.
1:0	R/W	0x0	Work Mode Select 00: Data frame is byte aligned in standard SPI, dual-output/dual input SPI, dual IO SPI and quad-output/quad-input SPI. 01: Reserved 10: Data frame is bit aligned in 3-wire SPI 11: Data frame is bit aligned in standard SPI

9.3.6.13. 0x0044 SPI Bit-Aligned CLOCK Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0044			**Register Name: SPI_BA_CCR**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	CDR_N Clock Divide Rate (Master Mode Only) The SPI_SCLK is determined according to the following equation: SPI_CLK = Source_CLK / (2*(CDR_N + 1)).

NOTE

This register is only valid when *Work Mode Select==0x10/0x11*.

9.3.6.14. 0x0048 SPI TX Bit Register(Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: SPI_TBR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	VTB The Value of the Transmit Bits This register is used to store the value of the transmitted serial data frame. In the process of transmission, the LSB is transmitted first.

NOTE

This register is only valid when *Work Mode Select==0x10/0x11*.

9.3.6.15. 0x004C SPI RX Bit Register(Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: SPI_RBR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	VRB The Value of the Receive Bits This register is used to store the value of the received serial data frame. In the process of transmission, the LSB is transmitted first.

NOTE

This register is only valid when *Work Mode Select==0x10/0x11*.

9.3.6.16. 0x0088 SPI Normal DMA Mode Control Register(Default Value: 0x0000_00E5)

Offset: 0x0088			Register Name: NDFC_NDMA_MODE_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:6	R/W	0x11	00:dma_active is low 01:dma_active is high 10:dma_active is controlled by dma_request(DRQ) 11:dma_active is controlled by controller
5	R/W	0x1	0: active fall do not care ack 1: active fall must after detect ack is high
4:0	R/W	0x05	Delay Cycles The counts of hold cycles from DMA last signal high to dma_active high

9.3.6.17. 0x0200 SPI TX Data Register(Default Value: 0x0000_0000)

Offset: 0x0200	Register Name: SPI_TXD

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	TDATA Transmit Data This register can be accessed in byte, half-word or word unit by AHB. In byte accessing method, if there are rooms in TXFIFO, one burst data is written to TXFIFO and the depth is increased by 1. In half-word accessing method, two SPI burst data are written and the TXFIFO depth is increased by 2. In word accessing method, four SPI burst data are written and the TXFIFO depth is increased by 4. **Note: This address is writable-only if TF_TEST is '0', and if TF_TEST is set to '1', this address is readable and writable to test the TX FIFO through the AHB bus.**

9.3.6.18. 0x0300 SPI RX Data Register(Default Value: 0x0000_0000)

Offset: 0x0300			Register Name: SPI_RXD
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	RDATA Receive Data This register can be accessed in byte, half-word or word unit by AHB. In byte accessing method, if there are data in RXFIFO, the top word is returned and the RXFIFO depth is decreased by 1. In half-word accessing method, two SPI bursts are returned and the RXFIFO depth is decreased by 2. In word accessing method, the four SPI bursts are returned and the RXFIFO depth is decreased by 4. **Note: This address is readable-only if RF_TEST is '0', and if RF_TEST is set to '1', this address is readable and writable to test the RX FIFO through the AHB bus.**

9.4. USB2.0 OTG

9.4.1. Overview

The USB2.0 OTG is a dual-role device controller, which supports both device and host functions which can also be configured as a Host-only or Device-only controller, fully compliant with the USB2.0 Specification. It can support high-speed (HS, 480 Mbit/s), full-speed (FS, 12 Mbit/s), and low-speed (LS, 1.5 Mbit/s) transfers in Host mode. It can support high-speed (HS, 480 Mbit/s), and full-speed (FS, 12 Mbit/s) in Device mode. Standard USB transceiver can be used through its UTMI+PHY Level3 interface. The UTMI+PHY interface is bidirectional with 8-bit data bus. For saving CPU bandwidth, USB-OTG DMA interface can support external DMA controller to take care of the data transfer between the memory and USB-OTG FIFO. The USB-OTG core also supports USB power saving functions.

The USB2.0 OTG has the following features:

- One USB 2.0 OTG(USB0), with integrated USB 2.0 analog PHY
- Complies with USB2.0 Specification
- Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s), and Low-Speed (LS, 1.5 Mbit/s) in Host mode
- Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s) in Device mode
- Supports the UTMI+ Level 3 interface. The 8-bit bidirectional data buses are used
- Supports bi-directional endpoint0 for Control transfer
- Supports up to 8 User-Configurable Endpoints for Bulk, Isochronousl and Interrupt bi-directional transfer
- Supports up to (4 KB+64 bytes) FIFO for all EPs (including EP0)
- Supports High-Bandwidth Isochronous & Interrupt transfers
- Automated splitting/combining of packets for Bulk transfers
- Supports point-to-point and point-to-multipoint transfer in both Host and Peripheral mode
- Includes automatic ping capabilities
- Soft connect/disconnect function
- Performs all transaction scheduling in hardware
- Power optimization and power management capabilities
- Includes interface to an external Normal DMA controller for every EPs

9.4.2. Block Diagram

Figure 9-29 shows the block diagram of USB2.0 OTG Controller.

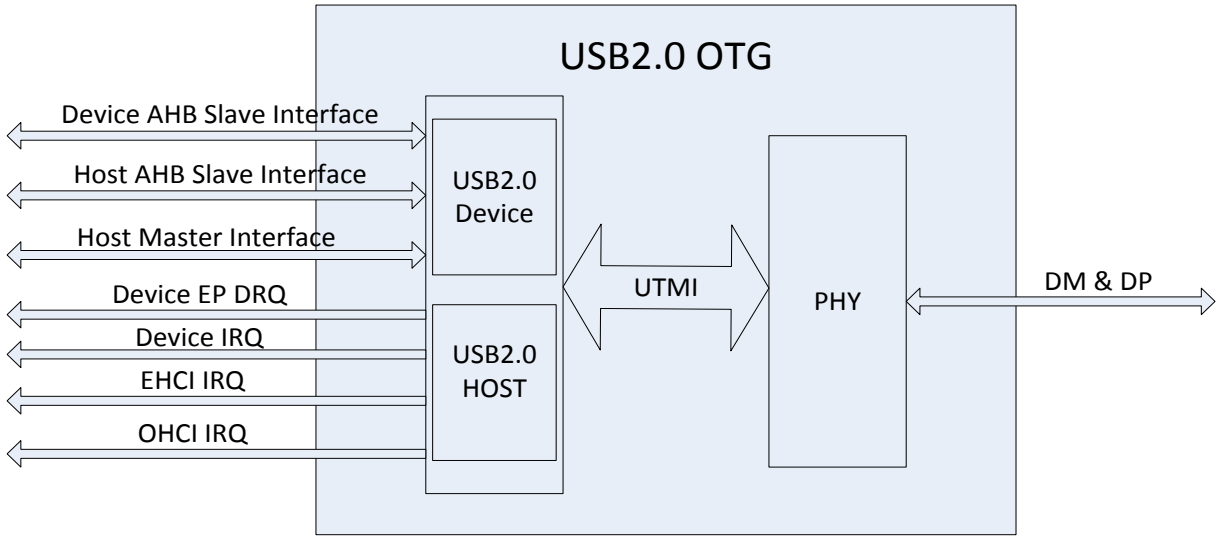

Figure 9- 29. USB2.0 OTG Controller Block Diagram

9.4.3. Operations and Functional Descriptions

9.4.3.1. External Signals

Table 9- 12. USB2.0 OTG External Signals

Signal	Description	Type
USB0_DP	USB2.0 OTG differential signal positive	AI/O
USB0_DM	USB2.0 OTG differential signal negative	AI/O

9.4.3.2. Controller and PHY Connection Diagram

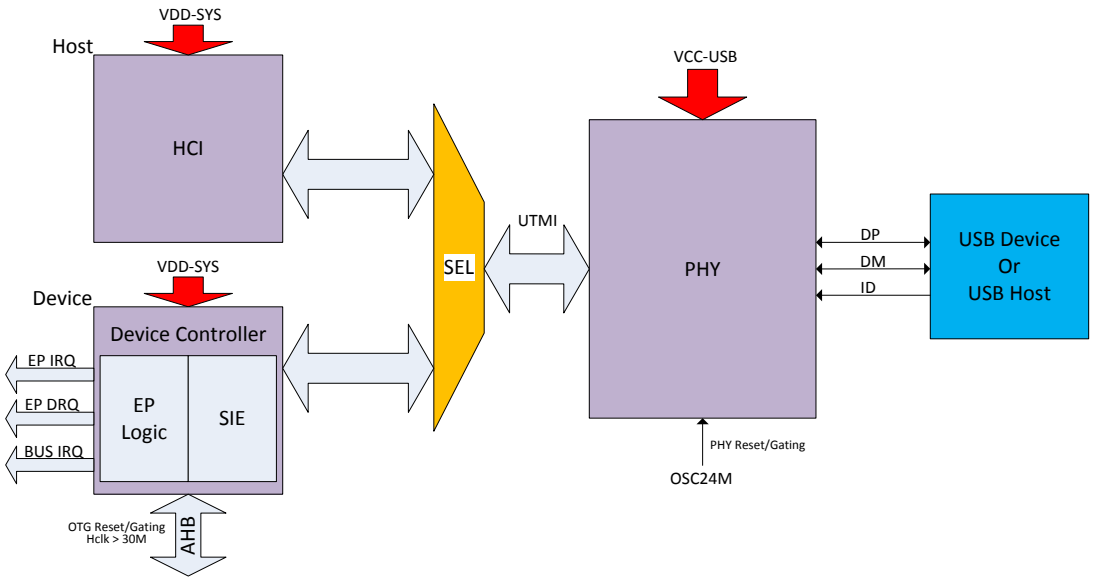

Figure 9- 30. USB2.0 OTG Controller and PHY Connection Diagram

9.5. USB2.0 Host Controller

9.5.1. Overview

USB2.0 Host Controller is fully compliant with the USB 2.0 specification, Enhanced Host Controller Interface (EHCI) Specification, Revision 1.0, and the Open Host Controller Interface (OHCI) Specification Release 1.0a. The controller supports high-speed, 480 Mbit/s transfers (40 times faster than USB 1.1 full-speed mode) using an EHCI Host Controller, as well as full and low speeds through one or more integrated OHCI Host Controllers.

The USB2.0 host controller includes the following features:

- Three USB 2.0 HOST(USB1, USB2, USB3), with integrated USB 2.0 analog PHY
- Only USB2 supports USB standby
- Supports industry-standard AMBA High-Performance Bus (AHB) and it is fully compliant with the AMBA Specification, Revision 2.0
- Supports 32-bit Little Endian AMBA AHB Slave Bus for Register Access
- Supports 32-bit Little Endian AMBA AHB Master Bus for Memory Access
- Including an internal DMA Controller for data transfer with memory
- Complies with Enhanced Host Controller Interface (EHCI) Specification, Version 1.0, and the Open Host Controller Interface (OHCI) Specification, Version 1.0a
- Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s), and Low-Speed (LS, 1.5 Mbit/s) Device
- Supports the UTMI+ Level 3 interface. The 8-bit bidirectional data buses are used
- Supports only 1 USB Root Port shared between EHCI and OHCI

9.5.2. Operations and Functional Descriptions

9.5.2.1. External Signals

Table 9- 13. USB2.0 HOST External Signals

Signal	Description	Type
USB1_DP	USB2.0 HOST1 differential signal positive	AI/O
USB1_DM	USB2.0 HOST1 differential signal negative	AI/O
USB2_DP	USB2.0 HOST2 differential signal positive	AI/O
USB2_DM	USB2.0 HOST2 differential signal negative	AI/O
USB3_DP	USB2.0 HOST3 differential signal positive	AI/O
USB3_DM	USB2.0 HOST3 differential signal negative	AI/O

9.5.2.2. Controller and PHY Connection Diagram

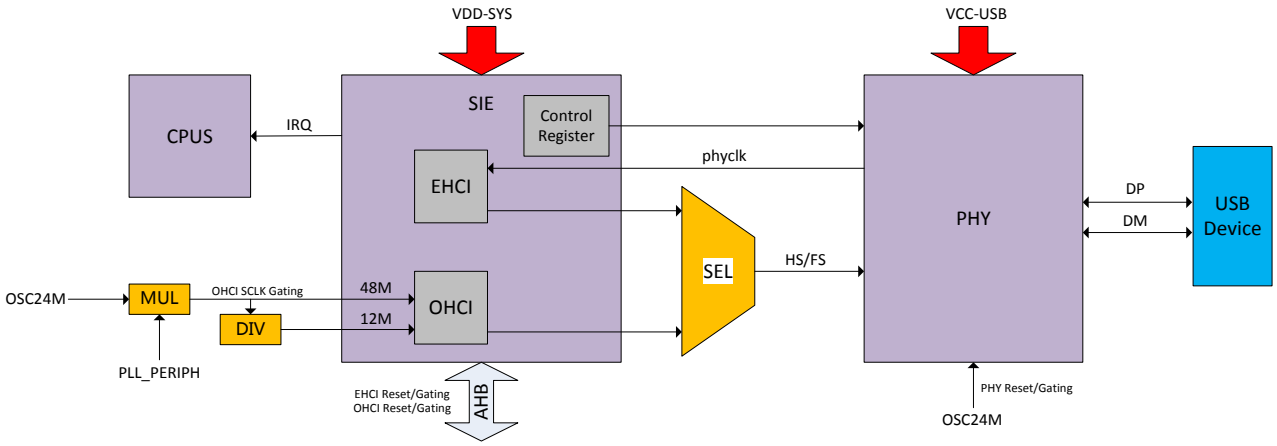

Figure 9- 31. USB2.0 HOST Controller and PHY Connection Diagram

9.5.3. USB Host Register List

Module Name	Base Address
USB1	0x05200000
USB2	0x05310000
USB3	0x05311000

Register Name	Offset	Description
EHCI Capability Register		
E_CAPLENGTH	0x0000	EHCI Capability Register Length Register
E_HCIVERSION	0x0002	EHCI Host Interface Version Number Register
E_HCSPARAMS	0x0004	EHCI Host Control Structural Parameter Register
E_HCCPARAMS	0x0008	EHCI Host Control Capability Parameter Register
E_HCSPPORTROUTE	0x000C	EHCI Companion Port Route Description
EHCI Operational Register		
E_USBCMD	0x0010	EHCI USB Command Register
E_USBSTS	0x0014	EHCI USB Status Register
E_USBINTR	0x0018	EHCI USB Interrupt Enable Register
E_FRINDEX	0x001C	EHCI USB Frame Index Register
E_CTRLDSSEGMENT	0x0020	EHCI 4G Segment Selector Register
E_PERIODICLISTBASE	0x0024	EHCI Frame List Base Address Register
E_ASYNCLISTADDR	0x0028	EHCI Next Asynchronous List Address Register
E_TIMERINTERRUPT	0x0030	EHCI Timer Configured And Standby Interrupt Status Register
E_CONFIGFLAG	0x0050	EHCI Configured Flag Register
E_PORTSC	0x0054	EHCI Port Status/Control Register
OHCI Control and Status Partition Register		
O_HcRevision	0x0400	OHCI Revision Register
O_HcControl	0x0404	OHCI Control Register

O_HcCommandStatus	0x0408	OHCI Command Status Register
O_HcInterruptStatus	0x040C	OHCI Interrupt Status Register
O_HcInterruptEnable	0x0410	OHCI Interrupt Enable Register
O_HcInterruptDisable	0x0414	OHCI Interrupt Disable Register
OHCI Memory Pointer Partition Register		
O_HcHCCA	0x0418	OHCI HCCA Base
O_HcPeriodCurrentED	0x041C	OHCI Period Current ED Base
O_HcControlHeadED	0x0420	OHCI Control Head ED Base
O_HcControlCurrentED	0x0424	OHCI Control Current ED Base
O_HcBulkHeadED	0x0428	OHCI Bulk Head ED Base
O_HcBulkCurrentED	0x042C	OHCI Bulk Current ED Base
O_HcDoneHead	0x0430	OHCI Done Head Base
OHCI Frame Counter Partition Register		
O_HcFmInterval	0x0434	OHCI Frame Interval Register
O_HcFmRemaining	0x0438	OHCI Frame Remaining Register
O_HcFmNumber	0x043C	OHCI Frame Number Register
O_HcPerioddicStart	0x0440	OHCI Periodic Start Register
O_HcLSThreshold	0x0444	OHCI LS Threshold Register
OHCI Root Hub Partition Register		
O_HcRhDescriptorA	0x0448	OHCI Root Hub Descriptor Register A
O_HcRhDesriptorB	0x044C	OHCI Root Hub Descriptor Register B
O_HcRhStatus	0x0450	OHCI Root Hub Status Register
O_HcRhPortStatus	0x0454	OHCI Root Hub Port Status Register
HCI Controller and PHY Interface Register		
HCI Interface	0x0800	HCI Interface Register
PHY Control	0x0810	PHY Control Register
HSIC PHY tune1	0x081C	HSIC PHY Tune1 Register
HSIC PHY tune2	0x0820	HSIC PHY Tune2 Register
HSIC PHY tune3	0x0824	HSIC PHY Tune3 Register
HCI SIE Port Disable Control	0x0828	HCI SIE Port Disable Control Register

9.5.4. EHCI Register Description

9.5.4.1. 0x0000 EHCI Identification Register(Default Value:0x10)

Offset:0x0000			**Register Name: CAPLENGTH**
Bit	**Read/Write**	**Default/Hex**	**Description**
7:0	R	0x10	**CAPLENGTH** The value in these bits indicates an offset to add to register base to find the beginning of the Operational Register Space.

9.5.4.2. 0x0002 EHCI Host Interface Version Number Register(Default Value:0x0100)

Offset: 0x0002			Register Name: HCIVERSION
Bit	**Read/Write**	**Default/Hex**	**Description**
15:0	R	0x0100	**HCIVERSION** This is a 16-bits register containing a BCD encoding of the EHCI revision number supported by this host controller. The most significant byte of this register represents a major revision and the least significant byte is the minor revision.

9.5.4.3. 0x0004 EHCI Host Control Structural Parameter Register(Default Value:0x0000_0004)

Offset: 0x0004			Register Name: HCSPARAMS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:20	R	0x0	**Debug Port Number** This register identifies which of the host controller ports is the debug port. The value is the port number (one based) of the debug port. This field will always be '0'.
19:16	/	/	/
15:12	R	0x0	**Number of Companion Controller (N_CC)** This field indicates the number of companion controllers associated with this USB2.0 host controller. A zero in this field indicates there are no companion host controllers. And a value larger than zero in this field indicates there are companion USB1.1 host controller(s). This field will always be '0'.
11:8	R	0x0	**Number of Port per Companion Controller(N_PCC)** This field indicates the number of ports supported per companion host controller host controller. It is used to indicate the port routing configuration to system software. This field will always fix with '0'.
7	R	0x0	**Port Routing Rules** This field indicates the method used by this implementation for how all ports are mapped to companion controllers. The value of this field has the following interpretation: Value: 0 — The first N_PCC ports are routed to the lowest numbered function companion host controller, the next N_PCC port are routed to the next lowest function companion controller, and so on. Value: 1 — The port routing is explicitly enumerated by the first N_PORTS elements of the HCSP-PORTTOUTE array. This field will always be '0'.

6:4	/	/	/
3:0	R	0x1	**N_PORTS** This field specifies the number of physical downstream ports implemented on this host controller. The value of this field determines how many port registers are addressable in the Operational Register Space. Valid values are in the range of 0x1 to 0x0f. This field is always 1.

9.5.4.4. 0x0008 EHCI Host Control Capability Parameter Register(Default Value:0x0000_0008)

Offset: 0x0008			**Register Name: HCCPARAMS**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:8	R	0x0	**EHCI Extended Capabilities Pointer (EECP)** This optional field indicates the existence of a capabilities list. A value of 00b indicates no extended capabilities are implemented. A non-zero value in this register indicates the offset in PCI configuration space of the first EHCI extended capabiliby. The pointer value must be 40h or greater if implemented to maintain to consistency of the PCI header defined for this calss of device. The value of this field is always '00b'.
7:4	R	0x0	**Isochronous Scheduling Threshold** This field indicates, relative to the current position of the executing host controller, where software can reliably update the isochronous schedule. When bit[7] is zero, the value of the least significant 3 bits indicates the number of micro-frames a host controller can hold a set of isochronous data structures(one or more) before flushing the state. When bit[7] is a one, then host software assumes the host controller may cache an isochronous data structure for an entire frame.
3	R	0x1	/
2	R	0x0	**Asynchronous Schedule Park Capability** If this bit is set to a one, then the host controller supports the park feature for high-speed queue heads in the Asynchronous Schedule. The feature can be disabled or enabled and set to a specific level by using the Asynchronous Schedule Park Mode Enable and Asynchronous Schedule Park Mode Count fields in the USBCMD register.
1	R	0x0	**Programmable Frame List Flag** If this bit is set to a zero, then system software must use a frame list length of 1024 elements with this host controller. The USBCMD register Frame List Size field is a read-only register and should be set to zero. If set to 1, then system software can specify and use the frame list in the USBCMD register Frame List Size field to cofigure the host controller. The frame list must always aligned on a 4K page boundary. This requirement ensures that the frame list is always physically contiguous.

0	/	/	/

9.5.4.5. 0x000C EHCI Companion Port Route Description(Default Value:0x0000_0000)

Offset: 0x000C			Register Name: HCSP-PORTROUTE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	**HCSP-PORTROUTE** This optional field is valid only if Port Routing Rules field in HCSPARAMS register is set to a one. This field is used to allow a host controller implementation to explicitly describe to which companion host controller each implemented port is mapped. This field is a 15-element nibble array (each 4 bit is one array element). Each array location corresponds one-to-one with a physical port provided by the host controller (e.g. PORTROUTE [0] corresponds to the first PORTSC port, PORTROUTE [1] to the second PORTSC port, etc.). The value of each element indicates to which the companion host controllers this port is routed. Only the first N_PORTS elements have valid information. A value of zero indicates that the port is routed to the lowest numbered function companion host controller. A value of one indicates that the port is routed to the next lowest numbered function companion host controller, and so on.

9.5.4.6. 0x0010 EHCI USB Command Register(Default Value:0x0008_0000)

Offset: 0x0010			Register Name: USBCMD
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:16	R/W	0x08	**Interrupt Threshold Control** The value in this field is used by system software to select the maximum rate at which the host controller will issue interrupts. The only valid values are defined below: Value – Minimum Interrupt Interval 0x00 – Reserved 0x01 – 1 micro-frame 0x02 – 2 micro-frame 0x04 – 4 micro-frame 0x08 – 8 micro-frame(default, equates to 1 ms) 0x10 – 16 micro-frame(2ms) 0x20 – 32 micro-frame(4ms) 0x40 – 64 micro-frame(8ms) Any other value in this register yields undefined results. The default value in this field is 0x08 . Software modifications to this bit while HC Halted bit is equal to zero results

			in undefined behavior.
15:12	/	/	/
11	R	0x0	**Asynchronous Schedule Park Mode Enable(OPTIONAL)** If the Asynchronous Park Capability bit in the HCCPARAMS register is a one, then this bit defaults to a 1 and is R/W. Otherwise the bit must be a zero and is Read Only. Software uses this bit to enable or disable Park mode. When this bit is one, Park mode is enabled. When this bit is zero, Park mode is disabled.
10	/	/	/
9:8	R	0x0	**Asynchronous Schedule Park Mode Count(OPTIONAL)** Asynchronous Park Capability bit in the HCCPARAMS register is a one, Then this field defaults to 0x3 and is W/R. Otherwise it defaults to zero and is R. It contains a count of the number of successive transactions the host controller is allowed to execute from a high-speed queue head on the Asynchronous schedule before continuing traversal of the Asynchronous schedule. Valid value are 0x1 to 0x3.Software must not write a zero to this bit when Park Mode Enable is a one as it will result in undefined behavior.
7	R/W	0x0	**Light Host Controller Reset(OPTIONAL)** This control bit is not required. If implemented, it allows the driver to reset the EHCI controller without affecting the state of the ports or relationship to the companion host controllers. For example, the PORSTC registers should not be reset to their default values and the CF bit setting should not go to zero (retaining port ownership relationships). A host software read of this bit as zero indicates the Light Host Controller Reset has completed and it si safe for software to re-initialize the host controller. A host software read of this bit as a one indicates the Light Host
6	R/W	0x0	**Interrupt on Async Advance Doorbell** This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Soft- Ware must write a 1 to this bit to ring the doorbell. When the host controller has evicted all appropriate cached schedule state, it sets the Interrupt on Async Advance status bit in the USBSTS. if the Interrupt on Async Advance Enable bit in the USBINTR register is a one then the host controller will assert an interrupt at the next interrupt threshold. The host controller sets this bit to a zero after it has set the Interrupt on Async Advance status bit in the USBSTS register to a one. Software should not write a one to this bit when the asynchronous schedule is disabled. Doing so will yield undefined results.
5	R/W	0x0	**Asynchronous Schedule Enable** This bit controls whether the host controller skips processing the Asynchronous Schedule. Values mean: Bit Value | Meaning

			0 Do not process the Asynchronous Schedule. 1 Use the ASYNLISTADDR register to access the Asynchronous Schedule. The default value of this field is '0b'.
4	R/W	0x0	**Periodic Schedule Enable** This bit controls whether the host controller skips processing the Periodic Schedule. Values mean: Bit Value: Meaning 0: Do not process the Periodic Schedule. 1: Use the PERIODICLISTBASE register to access the Periodic Schedule. The default value of this field is '0b'.
3:2	R/W	0x0	**Frame List Size** This field is R/W only if Programmable Frame List Flag in the HCCPARAMS registers is set to a one. This field specifies the size of the Frame list. The size the frame list controls which bits in the Frame Index Register should be used for the Frame List Current index. Values mean: Bits: Meaning 00b: 1024 elements(4096 bytes)Default value 01b: 512 elements(2048 byts) 10b: 256 elements(1024 bytes)For resource-constrained condition 11b: reserved The default value is '00b'.
1	R/W	0x0	**Host Controller Reset** This control bit is used by software to reset the host controller. The effects of this on Root Hub registers are similar to a Chip Hardware Reset. When software writes a one to this bit, the Host Controller resets its internal pipelines, timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports. All operational registers, including port registers and port state machines are set to their initial values. Port ownership reverts to the companion host controller(s). Software must reinitialize the host controller as described in Section 4.1 of the CHEI Specification in order to return the host controller to an operational state. This bit is set to zero by the Host Controller when the reset process is complete. Software cannot terminate the reset process early by writing a zero to this register. Software should not set this bit to a one when the HC Halted bit in the USBSTS register is a zero. Attempting to reset an actively running host controller will result in undefined behavior.
0	R/W	0x0	**Run/Stop** When set to a 1, the Host Controller proceeds with execution of the

			schedule. When set to 0, the Host Controller completes the current and any actively pipelined transactions on the USB and then halts. The Host Controller must halt within 16 micro-frames after software clears this bit. The HC Halted bit indicates when the Host Controller has finished its pending pipelined transactions and has entered the stopped state. Software must not write a one to this field unless the Host Controller is in the Halt State. The default value is 0x0.

9.5.4.7. 0x0014 EHCI USB Status Register(Default Value:0x0000_1000)

Offset: 0x0014			**Register Name: USBSTS**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15	R	0x0	**Asynchronous Schedule Status** The bit reports the current real status of Asynchronous Schedule. If this bit is a zero then the status of the Asynchronous Schedule is disabled. If this bit is a one then the status of the Asynchronous Schedule is enabled. The Host Controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions the Asynchronous Schedule Enable bit in the USBCMD register. When this bit and the Asynchronous Schedule Enable bit are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0).
14	R	0x0	**Periodic Schedule Status** The bit reports the current real status of the Periodic Schedule. If this bit is a zero then the status of the Periodic Schedule is disabled. If this bit is a one then the status of the Periodic Schedule is enabled. The Host Controller is not required to *immediately* disable or enable the Periodic Schedule when software transitions the *Periodic Schedule Enable* bit in the USBCMD register. When this bit and the *Periodic Schedule Enable* bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0).
13	R	0x0	**Reclamation** This is a read-only status bit, which is used to detect an empty asynchronous schedule.
12	R	0x1	**HC Halted** This bit is a zero whenever the Run/Stop bit is a one. The Host Controller Sets this bit to one after it has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host Controller Hardware (e.g. internal error). The default value is '1'.
11:6	/	/	/
5	R/WC	0x0	**Interrupt on Async Advance** System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a

			one to the Interrupt on Async Advance Doorbell bit in the USBCMD register. This status bit indicates the assertion of that interrupt source.
4	R/WC	0x0	**Host System Error** The Host Controller set this bit to 1 when a serious error occurs during a host system access involving the Host Controller module. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs.
3	R/WC	0x0	**Frame List Rollover** The Host Controller sets this bit to a one when the Frame List Index rolls over from its maximum value to zero. The exact value at which the rollover occurs depends on the frame list size. For example, if the frame list size is 1024, the Frame Index Register rolls over every time FRINDEX [13] toggles. Similarly, if the size is 512, the Host Controller sets this bit to a one every time FRINDEX [12] toggles.
2	R/WC	0x0	**Port Change Detect** The Host Controller sets this bit to a one when any port for which the Port Owner bit is set to zero has a change bit transition from a zero to a one or a Force Port Resume bit transition from a zero to a one as a result of a J-K transition detected on a suspended port. This bit will also be set as a result of the Connect Status Chang being set to a one after system software has relinquished ownership of a connected port by writing a one to a port's Port Owner bit.
1	R/WC	0x0	**USB Error Interrupt(USBERRINT)** The Host Controller sets this bit to 1 when completion of USB transaction results in an error condition(e.g. error counter underflow).If the TD on which the error interrupt occurred also had its IOC bit set, both. This bit and USBINT bit are set.
0	R/WC	0x0	**USB Interrupt(USBINT)** The Host Controller sets this bit to a one on the completion of a USB transaction, which results in the retirement of a Transfer Descriptor that had its IOC bit set. The Host Controller also sets this bit to 1 when a short packet is detected (actual number of bytes received was less than the expected number of bytes)

9.5.4.8. 0x0018 EHCI USB Interrupt Enable Register(Default Value:0x0000_0000)

Offset: 0x0018			**Register Name: USBINTR**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:6	/	/	/
5	R/W	0x0	**Interrupt on Async Advance Enable** When this bit is 1, and the Interrupt on Async Advance bit in the USBSTS register is 1, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the Interrupt

			on Async Advance bit.
4	R/W	0x0	**Host System Error Enable** When this bit is 1, and the Host System Error Status bit in the USBSTS register is 1, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Host System Error bit.
3	R/W	0x0	**Frame List Rollover Enable** When this bit is 1, and the Frame List Rollover bit in the USBSTS register is 1, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Frame List Rollover bit.
2	R/W	0x0	**Port Change Interrupt Enable** When this bit is 1, and the Port Chang Detect bit in the USBSTS register is 1, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Port Chang Detect bit.
1	R/W	0x0	**USB Error Interrupt Enable** When this bit is 1, and the USBERRINT bit in the USBSTS register is 1,the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the USBERRINT bit.
0	R/W	0x0	**USB Interrupt Enable** When this bit is 1, and the USBINT bit in the USBSTS register is 1,the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the USBINT bit

9.5.4.9. 0x001C EHCI Frame Index Register(Default Value:0x0000_0000)

Offset: 0x001C			**Register Name: FRINDEX**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:14	/	/	/
13:0	R/W	0	**Frame Index** The value in this register increment at the end of each time frame (e.g. micro-frame).Bits[N:3] are used for the Frame List current index. It means that each location of the frame list is accessed 8 times(frames or Micro-frames) before moving to the next index. The following illustrates values of N based on the value of the Frame List Size field in the USBCMD register.

USBCMD[Frame List Size]	Number Elements	N
00b	1024	12
01b	512	11
10b	256	10
11b	Reserved	

NOTE

This register must be written as a DWord. Byte writes produce undefined results.

9.5.4.10. 0x0024 EHCI Periodic Frame List Base Address Register(Default Value:0x0000_0000)

Offset: 0x0024			Register Name: PERIODICLISTBASE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	R/W	0x0	**Base Address** These bits correspond to memory address signals [31:12], respectively. This register contains the beginning address of the Periodic Frame List in the system memory. System software loads this register prior to starting the schedule execution by the Host Controller. The memory structure referenced by this physical memory pointer is assumed to be 4 KB aligned. The contents of this register are combined with the Frame Index Register (FRINDEX) to enable the Host Controller to step through the Periodic Frame List in sequence.
11:0	/	/	/

NOTE

Writes must be Dword Writes.

9.5.4.11. 0x0028 EHCI Current Asynchronous List Address Register(Default Value:0x0000_0000)

Offset: 0x0028			Register Name: ASYNCLISTADDR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	R/W	0x0	**Link Pointer (LP)** This field contains the address of the next asynchronous queue head to be executed. These bits correspond to memory address signals [31:5], respectively.
4:0	/	/	/

NOTE

Write must be DWord Writes.

9.5.4.12. 0x0030 EHCI Timer Configured & Standby Interrupt Status register (Default Value:0x0000_0000)

Offset: 0x0030			Register Name: TIMERINTERRUPT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9:8	R/W	0x0	Timer value configured 00: 8ms 01: 16ms 10: 32ms

			11: 64ms
7:3	/	/	/
2	R/W	0x0	Standby irq enable, when usb in standby operation, open this bit, and close it after quitting standby mode, high active
1	R/WC	0x0	Standby irq status bit, write “1” to clear or timer auto clear when open timer enable bit, configured timer value see bit9:8.
0	R/W	0x0	Timer enable, high active

9.5.4.13. 0x0050 EHCI Configure Flag Register(Default Value:0x0000_0000)

Offset: 0x0050			Register Name: CONFIGFLAG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	**Configure Flag(CF)** Host software sets this bit as the last action in its process of configuring the Host Controller. This bit controls the default port-routing control logic as follow: Value: 0 — Port routing control logic default-routs each port to an implementation dependent classic host controller. Value: 1 — Port routing control logic default-routs all ports to this host controller. The default value of this field is ‘0’.

NOTE

This register is not used in the normal implementation.

9.5.4.14. 0x0054 EHCI Port Status and Control Register(Default Value:0x0000_2000)

Offset: 0x0054			Register Name: PORTSC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21	R/W	0x0	**Wake on Disconnect Enable(WKDSCNNT_E)** Writing this bit to a one enables the port to be sensitive to device disconnects as wake-up events. This field is zero if Port Power is zero. The default value in this field is ‘0’.
20	R/W	0x0	**Wake on Connect Enable(WKCNNT_E)** Writing this bit to a one enable the port to be sensitive to device connects as wake-up events. This field is zero if Port Power is zero. The default value in this field is ‘0’.

<table>
<tr><td>19:16</td><td>R/W</td><td>0x0</td><td>Port Test Control
The value in this field specifies the test mode of the port. The encoding of the test mode bits are as follows:<table><tr><td>Bits</td><td>Test Mode</td></tr><tr><td>0000b</td><td>The port is NOT operating in a test mode.</td></tr><tr><td>0001b</td><td>Test J_STATE</td></tr><tr><td>0010b</td><td>Test K_STATE</td></tr><tr><td>0011b</td><td>Test SE0_NAK</td></tr><tr><td>0100b</td><td>Test Packet</td></tr><tr><td>0101b</td><td>Test FORCE_ENABLE</td></tr><tr><td>0110b-1111b</td><td>Reserved</td></tr></table>The default value in this field is ‘0000b’.</td></tr>
<tr><td>15:14</td><td>/</td><td>/</td><td>/</td></tr>
<tr><td>13</td><td>R/W</td><td>0x1</td><td>Port Owner
This bit unconditionally goes to a 0b when the Configured bit in the CONFIGFLAG register makes a 0b to 1b transition. This bit unconditionally goes to 1b whenever the Configured bit is zero.
System software uses this field to release ownership of the port to selected host controller (in the event that the attached device is not a high-speed device).Software writes a one to this bit when the attached device is not a high-speed device. A one in this bit means that a companion host controller owns and controls the port.
Default Value = 1b.</td></tr>
<tr><td>12</td><td>/</td><td>/</td><td>/</td></tr>
<tr><td>11:10</td><td>R</td><td>0x0</td><td>Line Status
These bits reflect the current logical levels of the D+ (bit11) and D-(bit10) signal lines. These bits are used for detection of low-speed USB devices prior to port reset and enable sequence. This read only field is valid only when the port enable bit is zero and the current connect status bit is set to a one.
The encoding of the bits are:<table><tr><td>Bit[11:10]</td><td>USB State</td><td>Interpretation</td></tr><tr><td>00b</td><td>SE0</td><td>Not Low-speed device, perform EHCI reset.</td></tr><tr><td>10b</td><td>J-state</td><td>Not Low-speed device, perform EHCI reset.</td></tr><tr><td>01b</td><td>K-state</td><td>Low-speed device, release ownership of port.</td></tr><tr><td>11b</td><td>Undefined</td><td>Not Low-speed device, perform EHCI reset.</td></tr></table>This value of this field is undefined if Port Power is zero.</td></tr>
<tr><td>9</td><td>/</td><td>/</td><td>/</td></tr>
<tr><td>8</td><td>R/W</td><td>0x0</td><td>Port Reset
1=Port is in Reset. 0=Port is not in Reset. Default value = 0.</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>When software writes a one to this bit (from a zero), the bus reset sequence as defined in the USB Specification Revision 2.0 is started. Software writes a zero to this bit to terminate the bus reset sequence. Software must keep this bit at a one long enough to ensure the reset sequence, as specified in the USB Specification Revision 2.0, completes. Notes: when software writes this bit to a one , it must also write a zero to the Port Enable bit.
Note that when software writes a zero to this bit there may be a delay before the bit status changes to a zero. The bit status will not read as a zero until after the reset has completed. If the port is in high-speed mode after reset is complete, the host controller will automatically enable this port (e.g. set the Port Enable bit to a one). A host controller must terminate the reset and stabilize the state of the port within 2 milliseconds of software transitioning this bit from a one to a zero. For example: if the port detects that the attached device is high-speed during reset, then the host controller must have the port in the enabled state with 2ms of software writing this bit to a zero.
The HC Halted bit in the USBSTS register should be a zero before software attempts to use this bit. The host controller may hold Port Reset asserted to a one when the HC Halted bit is a one.
This field is zero if Port Power is zero.</td></tr>
<tr><td>7</td><td>R/W</td><td>0x0</td><td>Suspend
Port Enabled Bit and Suspend bit of this register define the port states as follows:
<table>
<tr><td>Bits[Port Enables, Suspend]</td><td>Port State</td></tr>
<tr><td>0x</td><td>Disable</td></tr>
<tr><td>10</td><td>Enable</td></tr>
<tr><td>11</td><td>Suspend</td></tr>
</table>
When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Not that the bit status does not change until the port is suspend and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB.
A write of zero to this bit is ignored by the host controller. The host controller will unconditionally set this bit to a zero when:
① Software sets the Force Port Resume bit to a zero(from a one).
② Software sets the Port Reset bit to a one(from a zero).
If host software sets this bit to a one when the port is not enabled(i.e. Port enabled bit is a zero), the results are undefined.
This field is zero if Port Power is zero.
The default value in this field is ‘0’.</td></tr>
<tr><td>6</td><td>R/W</td><td>0x0</td><td>Force Port Resume
1 = Resume detected/driven on port. 0 = No resume (K-state) detected/ driven on port. Default value = 0.</td></tr>
</table>

			This functionality defined for manipulating this bit depends on the value of the Suspend bit. For example, if the port is not suspend and software transitions this bit to a one, then the effects on the bus are undefined. Software sets this bit to a 1 drive resume signaling. The Host Controller sets this bit to a 1 if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to a one because a J-to-K transition is detected, the Port Change Detect bit in the USBSTS register is also set to a one. If software sets this bit to a one, the host controller must not set the Port Change Detect bit. Note that when the EHCI controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this remains a one. Software must appropriately time the Resume and set this bit to a zero when the appropriate amount of time has elapsed. Writing a zero (from one) causes the port to return high-speed mode (forcing the bus below the port into a high-speed idle). This bit will remain a one until the port has switched to high-speed idle. The host controller must complete this transition within 2 milliseconds of software setting this bit to a zero. This field is zero if Port Power is zero.
5	R/WC	0x0	**Over-current Change** Default = 0. This bit gets set to a one when there is a change to Over-current Active. Software clears this bit by writing a one to this bit position.
4	R	0x0	**Over-current Active** 0 = This port does not have an over-current condition. 1 = This port currently has an over-current condition. This bit will automatically transition from a one to a zero when the over current condition is removed. The default value of this bit is '0'.
3	R/WC	0x0	**Port Enable/Disable Change** Default = 0. 1 = Port enabled/disabled status has changed. 0 = No change. For the root hub, this bit gets set to a one only when a port is disabled due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification for the definition of a Port Error). Software clears this bit by writing a 1 to it. This field is zero if Port Power is zero.
2	R/W	0x0	**Port Enabled/Disabled** 1=Enable, 0=Disable. Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a one to this field. The host controller will only set this bit to a one when the reset sequence determines that the attached device is a high-speed device. Ports can be disabled by either a fault condition(disconnect event or other fault condition) or by host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events. When the port is disabled, downstream propagation of data is blocked on this port except for reset.

			The default value of this field is '0'. This field is zero if Port Power is zero.
1	R/WC	0x0	**Connect Status Change** 1=Change in Current Connect Status, 0=No change, Default=0. Indicates a change has occurred in the port's Current Connect Status. The host controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be "setting" an already-set bit. Software sets this bit to 0 by writing a 1 to it. This field is zero if Port Power is zero.
0	R	0x0	**Current Connect Status** Device is present on port when the value of this field is a one, and no device is present on port when the value of this field is a zero. This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change(Bit 1) to be set. This field is zero if Port Power zero.

NOTE

This register is only reset by hardware or in response to a host controller reset.

9.5.5. OHCI Register Description

9.5.5.1. 0x0400 HcRevision Register(Default Value:0x0000_0010)

Offset: 0x0400				**Register Name: HcRevision**
Bit	**Read/Write**		**Default/Hex**	**Description**
	HCD	**HC**		
31:8	/	/	/	/
7:0	R	R	0x10	**Revision** This read-only field contains the BCD representation of the version of the HCI specification that is implemented by this HC. For example, a value of 0x11 corresponds to version 1.1. All of the HC implementations that are compliant with this specification will have a value of 0x10.

9.5.5.2. 0x0404 HcControl Register(Default Value:0x0000_0000)

Offset: 0x0404				**Register Name: HcRevision**
Bit	**Read/Write**		**Default/Hex**	**Description**
	HCD	**HC**		
31:11	/	/	/	/
10	R/W	R	0x0	**RemoteWakeupEnable**

				This bit is used by HCD to enable or disable the remote wakeup feature upon the detection of upstream resume signaling. When this bit is set and the **ResumeDetected** bit in ***HcInterruptStatus*** is set, a remote wakeup is signaled to the host system. Setting this bit has no impact on the generation of hardware interrupt.
9	R/W	R/W	0x0	**RemoteWakeupConnected** This bit indicates whether HC supports remote wakeup signaling. If remote wakeup is supported and used by the system, it is the responsibility of system firmware to set this bit during POST. HC clear the bit upon a hardware reset but does not alter it upon a software reset. Remote wakeup signaling of the host system is host-bus-specific and is not described in this specification.
8	R/W	R	0x0	**InterruptRouting** This bit determines the routing of interrupts generated by events registered in HcInterruptStatus. If clear, all interrupt are routed to the normal host bus interrupt mechanism. If set interrupts are routed to the System Management Interrupt. HCD clears this bit upon a hardware reset, but it does not alter this bit upon a software reset. HCD uses this bit as a tag to indicate the ownership of HC.
7:6	R/W	R/W	0x0	HostControllerFunctionalState for USB 00b USBReset 01b USBResume 10b USBOperational 11b USBSuspend A transition to USBOperational from another state causes SOF generation to begin 1 ms later. HCD may determine whether HC has begun sending SOFs by reading the StartoFrame field of HcInterruptStatus. This field may be changed by HC only when in the USBSUSPEND state. HC may move from the USBSUSPEND state to the USBRESUME state after detecting the resume signaling from a downstream port. HC enters USBSUSPEND after a software reset, whereas it enters USBRESET after a hardware reset. The latter also resets the Root Hub and asserts subsequent reset signaling to downstream ports.
5	R/W	R	0x0	**BulkListEnable** This bit is set to enable the processing of the Bulk list in the next Frame. If cleared by HCD, processing of the Bulk list does not occur after the next SOF. HC checks this bit whenever it determines to process the list. When disabled, HCD may modify the list. If *HcBulkCurrentED* is pointing to an ED to be removed, HCD must advance the pointer by updating *HcBulkCurrentED* before re-enabling processing of the list.
4	R/W	R	0x0	**ControlListEnable** This bit is set to enable the processing of the Control list in the next Frame. If cleared by HCD, processing of the Control list does not occur after the next SOF. HC must check this bit whenever it determines to process the list.

				When disabled, HCD may modify the list. If *HcControlCurrentED* is pointing to an ED to be removed, HCD must advance the pointer by updating *HcControlCurrentED* before re-enabling processing of the list.
3	R/W	R	0x0	**IsochronousEnable** This bit is used by HCD to enable/disable processing of isochronous EDs. While processing the periodic list in a Frame, HC checks the status of this bit when it finds an Isochronous ED (F=1). If set (enabled), HC continues processing the EDs. If cleared (disabled), HC halts processing of the periodic list (which now contains only isochronous EDs) and begins processing the Bulk/Control lists. Setting this bit is guaranteed to take effect in the next Frame (not the current Frame).
2	R/W	R	0x0	**PeriodicListEnable** This bit is set to enable the processing of periodic list in the next Frame. If cleared by HCD, processing of the periodic list does not occur after the next SOF. HC must check this bit before it starts processing the list.
1:0	R/W	R	0x0	**ControlBulkServiceRatio** This specifies the service ratio between Control and Bulk EDs. Before processing any of the nonperiodic lists, HC must compare the ratio specified with its internal count on how many nonempty Control EDs have been processed, in determining whether to continue serving another Control ED or switching to Bulk EDs. The internal count will be retained when crossing the frame boundary. In case of reset, HCD is responsible for restoring this value. CBSR: 0 → No. of Control EDs Over Bulk EDs Served: 1:1 CBSR: 1 → 2:1 CBSR: 2 → 3:1 CBSR: 3 → 4:1 The default value is 0x0.

9.5.5.3. 0x0408 HcCommandStatus Register(Default Value:0x0000_0000)

Offset: 0x0408				**Register Name: HcCommandStatus**
Bit	**Read/Write**		**Default/Hex**	**Description**
	HCD	**HC**		
31:18	/	/	0x0	**Reserved**
17:16	R	R/W	0x0	**SchedulingOverrunCount** These bits are incremented on each scheduling overrun error. It is initialized to 00b and wraps around at 11b. This will be incremented when a scheduling overrun is detected even if **SchedulingOverrun** in *HcInterruptStatus* has already been set. This is used by HCD to monitor any persistent scheduling problem.

15:4	/	/	/	/
3	R/W	R/W	0x0	**OwershipChangeRequest** This bit is set by an OS HCD to request a change of control of the HC. When set HC will set the **OwnershipChange** field in *HcInterruptStatus*. After the changeover, this bit is cleared and remains so until the next request from OS HCD.
2	R/W	R/W	0x0	**BulklListFilled** This bit is used to indicate whether there are any TDs on the Bulk list. It is set by HCD whenever it adds a TD to an ED in the Bulk list. When HC begins to process the head of the Bulk list, it checks BLF. As long as **BulkListFilled** is 0, HC will not start processing the Bulk list. If **BulkListFilled** is 1, HC will start processing the Bulk list and will set BF to 0. If HC finds a TD on the list, then HC will set **BulkListFilled** to 1 causing the Bulk list processing to continue. If no TD is found on the Bulk list, and if HCD does not set **BulkListFilled**, then **BulkListFilled** will still be 0 when HC completes processing the Bulk list and Bulk list processing will stop.
1	R/W	R/W	0x0	**ControlListFilled** This bit is used to indicate whether there are any TDs on the Control list. It is set by HCD whenever it adds a TD to an ED in the Control list. When HC begins to process the head of the Control list, it checks CLF. As long as **ControlListFilled** is 0, HC will not start processing the Control list. If CF is 1, HC will start processing the Control list and will set **ControlListFilled** to 0. If HC finds a TD on the list, then HC will set **ControlListFilled** to 1 causing the Control list processing to continue. If no TD is found on the Control list, and if the HCD does not set **ControlListFilled**, then **ControlListFilled** will still be 0 when HC completes processing the Control list and Control list processing will stop.
0	R/W	R/E	0x0	**HostControllerReset** This bit is by HCD to initiate a software reset of HC. Regardless of the functional state of HC, it moves to the USBSuspend state in which most of the operational registers are reset except those stated otherwise; e.g, the InteruptRouting field of HcControl, and no Host bus accesses are allowed. This bit is cleared by HC upon the completion of the reset operation. The reset operation must be completed within 10 ms. This bit, when set, should not cause a reset to the Root Hub and no subsequent reset signaling should be asserted to its downstream ports.

9.5.5.4. 0x040C HcInterruptStatus Register(Default Value:0x0000_0000)

Offset: 0x040C				**Register Name: HcInterruptStatus**
Bit	**Read/Write**		**Default/Hex**	**Description**
	HCD	**HC**		
31:7	/	/	/	/
6	R/W	R/W	0x0	**RootHubStatusChange**

				This bit is set when the content of *HcRhStatus* or the content of any of *HcRhPortStatus*[**NumberofDownstreamPort**] has changed.
5	R/W	R/W	0x0	**FrameNumberOverflow** This bit is set when the MSb of *HcFmNumber* (bit 15) changes value, from 0 to 1 or from 1 to 0, and after *HccaFrameNumber* has been updated.
4	R/W	R/W	0x0	**UnrecoverableError** This bit is set when HC detects a system error not related to USB. HC should not proceed with any processing nor signaling before the system error has been corrected. HCD clears this bit after HC has been reset.
3	R/W	R/W	0x0	**ResumeDetected** This bit is set when HC detects that a device on the USB is asserting resume signaling. It is the transition from no resume signaling to resume signaling causing this bit to be set. This bit is not set when HCD sets the USBRseume state.
2	R/W	R/W	0x0	**StartofFrame** This bit is set by HC at each start of frame and after the update of HccaFrameNumber. HC also generates a SOF token at the same time.
1	R/W	R/W	0x0	**WritebackDoneHead** This bit is set immediately after HC has written *HcDoneHead* to *HccaDoneHead*. Further updates of the *HccaDoneHead* will not occur until this bit has been cleared. HCD should only clear this bit after it has saved the content of *HccaDoneHead*.
0	R/W	R/W	0x0	**SchedulingOverrun** This bit is set when the USB schedule for the current Frame overruns and after the update of *HccaFrameNumber*. A scheduling overrun will also cause the **SchedulingOverrunCount** of *HcCommandStatus* to be incremented.

9.5.5.5. 0x0410 HcInterruptEnable Register(Default Value:0x0000_0000)

Offset: 0x0410				**Register Name: HcInterruptEnable Register**
Bit	**Read/Write**		**Default/Hex**	**Description**
	HCD	**HC**		
31	R/W	R	0x0	**MasterInterruptEnable** A '0' writtern to this field is ignored by HC. A '1' written to this field enables interrupt generation due to events specified in the other bits of this register. This is used by HCD as Master Interrupt Enable.
30:7	/	/	/	/
6	R/W	R	0x0	**RootHubStatusChange Interrupt Enable** 0 Ignore; 1 Enable interrupt generation due to Root Hub Status Change;
5	R/W	R	0x0	**FrameNumberOverflow Interrupt Enable** 0 Ignore;

				1 Enable interrupt generation due to Frame Number Over Flow;
4	R/W	R	0x0	**UnrecoverableError Interrupt Enable** 0 Ignore; 1 Enable interrupt generation due to Unrecoverable Error;
3	R/W	R	0x0	**ResumeDetected Interrupt Enable** 0 Ignore; 1 Enable interrupt generation due to Resume Detected;
2	R/W	R	0x0	**StartofFrame Interrupt Enable** 0 Ignore; 1 Enable interrupt generation due to Start of Flame;
1	R/W	R	0x0	**WritebackDoneHead Interrupt Enable** 0 Ignore; 1 Enable interrupt generation due to Write back Done Head;
0	R/W	R	0x0	**SchedulingOverrun Interrupt Enable** 0 Ignore; 1 Enable interrupt generation due to Scheduling Overrun;

9.5.5.6. 0x0414 HcInterruptDisable Register(Default Value:0x0000_0000)

Offset: 0x0414				Register Name: HcInterruptDisable Register
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31	R/W	R	0x0	**MasterInterruptEnable** A written '0' to this field is ignored by HC. A '1' written to this field disables interrupt generation due events specified in the other bits of this register. This field is set after a hardware or software reset.
30:7	/	/	/	/
6	R/W	R	0x0	**RootHubStatusChange Interrupt Disable** 0 Ignore; 1 Disable interrupt generation due to Root Hub Status Change;
5	R/W	R	0x0	**FrameNumberOverflow Interrupt Disable** 0 Ignore; 1 Disable interrupt generation due to Frame Number Over Flow;
4	R/W	R	0x0	**UnrecoverableError Interrupt Disable** 0 Ignore; 1 Disable interrupt generation due to Unrecoverable Error;
3	R/W	R	0x0	**ResumeDetected Interrupt Disable** 0 Ignore; 1 Disable interrupt generation due to Resume Detected;
2	R/W	R	0x0	**StartofFrame Interrupt Disable** 0 Ignore;

				1 Disable interrupt generation due to Start of Flame;
1	R/W	R	0x0	**WritebackDoneHead Interrupt Disable** 0 Ignore; 1 Disable interrupt generation due to Write back Done Head;
0	R/w	R	0x0	**SchedulingOverrun Interrupt Disable** 0 Ignore; 1 Disable interrupt generation due to Scheduling Overrun;

9.5.5.7. 0x0418 HcHCCA Register(Default Value:0x0000_0000)

Offset: 0x0418				Register Name: HcHCCA
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:8	R/W	R	0x0	**HCCA[31:8]** This is the base address of the Host Controller Communication Area. This area is used to hold the control structures and the Interrupt table that are accessed by both the Host Controller and the Host Controller Driver.
7:0	R	R	0x0	**HCCA[7:0]** The alignment restriction in HcHCCA register is evaluated by examining the number of zeros in the lower order bits. The minimum alignment is 256 bytes, therefore, bits 0 through 7 must always return 0 when read.

9.5.5.8. 0x041C HcPeriodCurrentED Register(Default Value:0x0000_0000)

Offset: 0x041C				Register Name: HcPeriodCurrentED(PCED)
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:4	R	R/W	0x0	**PCED[31:4]** This is used by HC to point to the head of one of the Periodec list which will be processed in the current Frame. The content of this register is updated by HC after a periodic ED has been processed. HCD may read the content in determining which ED is currently being processed at the time of reading.
3:0	R	R	0x0	**PCED[3:0]** Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field.

9.5.5.9. 0x0420 HcControlHeadED Register(Default Value:0x0000_0000)

Offset: 0x0420	Register Name: HcControlHeadED[CHED]

Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:4	R/W	R	0x0	**EHCD[31:4]** The HcControlHeadED register contains the physical address of the first Endpoint Descriptor of the Control list. HC traverse the Control list starting with the HcControlHeadED pointer. The content is loaded from HCCA during the initialization of HC.
3:0	R	R	0x0	**EHCD[3:0]** Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field.

9.5.5.10. 0x0424 HcControlCurrentED Register

Offset: 0x0424				Register Name: HcControlCurrentED[CCED]
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:4	R/W	R/W	0x0	**CCED[31:4]** The pointer is advanced to the next ED after serving the present one. HC will continue processing the list from where it left off in the last Frame. When it reaches the end of the Control list, HC checks the ControlListFilled of in HcCommandStatus. If set, it copies the content of HcControlHeadED to HcControlCurrentED and clears the bit. If not set, it does nothing. HCD is allowed to modify this register only when the ControlListEnable of HcControl is cleared. When set, HCD only reads the instantaneous value of this register. Initially, this is set to zero to indicate the end of the Control list.
3:0	R	R	0x0	**CCED[3:0]** Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field.

9.5.5.11. 0x0428 HcBulkHeadED Register(Default Value:0x0000_0000)

Offset: 0x0428				Register Name: HcBulkHeadED[BHED]
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:4	R/W	R	0x0	**BHED[31:4]** The HcBulkHeadED register contains the physical address of the first Endpoint Descriptor of the Bulk list. HC traverses the Bulk list starting with the HcBulkHeadED pointer. The content is loaded from HCCA during the initialization of HC.

3:0	R	R	0x0	**BHED[3:0]** Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field.

9.5.5.12. 0x042C HcBulkCurrentED Register(Default Value:0x0000_0000)

Offset: 0x042C				Register Name: HcBulkCurrentED [BCED]
Bit	**Read/Write**		**Default/Hex**	**Description**
	HCD	**HC**		
31:4	R/W	R/W	0x0	**BulkCurrentED[31:4]** This is advanced to the next ED after the HC has served the present one. HC continues processing the list from where it left off in the last Frame. When it reaches the end of the Bulk list, HC checks the **ControlListFilled** of HcControl. If set, it copies the content of *HcBulkHeadED* to *HcBulkCurrentED* and clears the bit. If it is not set, it does nothing. HCD is only allowed to modify this register when the **BulkListEnable** of *HcControl* is cleared. When set, the HCD only reads the instantaneous value of this register. This is initially set to zero to indicate the end of the Bulk list.
3:0	R	R	0x0	**BulkCurrentED [3:0]** Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field.

9.5.5.13. 0x0430 HcDoneHead Register(Default Value:0x0000_0000)

Offset: 0x0430				Register Name: HcDoneHead
Bit	**Read/Write**		**Default/Hex**	**Description**
	HCD	**HC**		
31:4	R	R/W	0x0	**HcDoneHead[31:4]** When a TD is completed, HC writes the content of *HcDoneHead* to the NextTD field of the TD. HC then overwrites the content of *HcDoneHead* with the address of this TD. This is set to zero whenever HC writes the content of this register to HCCA. It also sets the **WritebackDoneHead** of *HcInterruptStatus*.
3:0	R	R	0x0	**HcDoneHead[3:0]** Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field.

9.5.5.14. 0x0434 HcFmInterval Register(Default Value:0x0000_2EDF)

Offset: 0x0434				Register Name: HcFmInterval Register
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31	R/W	R	0x0	**FrameIntervalToggler** HCD toggles this bit whenever it loads a new value to **FrameInterval**.
30:16	R/W	R	0x0	**FSLargestDataPacket** This field specifies a value which is loaded into the Largest Data Packet Counter at the beginning of each frame. The counter value represents the largest amount of data in bits which can be sent or received by the HC in a single transaction at any given time without causing scheduling overrun. The field value is calculated by the HCD.
15:14	/	/	/	/
13:0	R/W	R	0x2edf	**FrameInterval** This specifies the interval between two consecutive SOFs in bit times. The nominal value is set to be 11,999. HCD should store the current value of this field before resetting HC. By setting the **HostControllerReset** field of *HcCommandStatus* as this will cause the HC to reset this field to its nominal value. HCD may choose to restore the stored value upon the completion of the Reset sequence.

9.5.5.15. 0x0438 HcFmRemaining Register(Default Value:0x0000_0000)

Offset: 0x0438				Register Name: HcFmRemaining
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31	R	R/W	0x0	**FrameRemaining Toggle** This bit is loaded from the **FrameIntervalToggle** field of *HcFmInterval* whenever **FrameRemaining** reaches 0. This bit is used by HCD for the synchronization between **FrameInterval** and **FrameRemaining**.
30:14	/	/	/	**/**
13:0	R	RW	0x0	**FramRemaining** This counter is decremented at each bit time. When it reaches zero, it is reset by loading the FrameInterval value specified in *HcFmInterval* at the next bit time boundary. When entering the USBOPERATIONAL state, HC re-loads the content with the FrameInterval of *HcFmInterval* and uses the updated value from the next SOF.

9.5.5.16. 0x043C HcFmNumber Register(Default Value:0x0000_0000)

Offset: 0x043C	Register Name: HcFmNumber

Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:16	/	/	/	/
15:0	R	R/W	0x0	**FrameNumber** This is incremented when *HcFmRemaining* is re-loaded. It will be rolled over to 0x0 after 0x0ffff. When entering the USBOPERATIONAL state, this will be incremented automatically. The content will be written to HCCA after HC has incremented the **FrameNumber** at each frame boundary and sent a SOF but before HC reads the first ED in that Frame. After writing to HCCA, HC will set the **StartofFrame** in *HcInterruptStatus.*

9.5.5.17. 0x0440 HcPeriodicStart Register(Default Value:0x0000_0000)

Offset: 0x0440				Register Name: HcPeriodicStatus
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:14	/	/	/	/
13:0	R/W	R	0x0	**PeriodicStart** After a hardware reset, this field is cleared. This is then set by HCD during the HC initialization. The value is calculated roughly as 10% off from *HcFmInterval*. A typical value will be 0x2A3F (0x3e67??). When *HcFmRemaining* reaches the value specified, processing of the periodic lists will have priority over Control/Bulk processing. HC will therefore start processing the Interrupt list after completing the current Control or Bulk transaction that is in progress.

9.5.5.18. 0x0444 HcLSThreshold Register(Default Value:0x0000_0628)

Offset: 0x0444				Register Name: HcLSThreshold
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:12	/	/	/	/
11:0	R/W	R	0x0628	**LSThreshold** This field contains a value which is compared to the **FrameRemaining** field prior to initiating a Low Speed transaction. The transaction is started only if **FrameRemaining** ^{3} this field. The value is calculated by HCD with the consideration of transmission and setup overhead.

9.5.5.19. 0x0448 HcRhDescriptorA Register(Default Value:0x0200_1201)

Offset: 0x0448	Register Name: HcRhDescriptorA

Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:24	R/W	R	0x2	**PowerOnToPowerGoodTime[POTPGT]** This byte specifies the duration HCD has to wait before accessing a powered-on port of the Root Hub. It is implementation-specific. The unit of time is 2 ms. The duration is calculated as POTPGT * 2ms.
23:13	/	/	/	/
12	R/W	R	1	**NoOverCurrentProtection** This bit describes how the overcurrent status for the Root Hub ports are reported. When this bit is cleared, the **OverCurrentProtectionMode** field specifies global or per-port reporting. **0** Over-current status is reported collectively for all downstream ports. **1** No overcurrent protection supported.
11	R/W	R	0	**OverCurrentProtectionMode** This bit describes how the overcurrent status for the Root Hub ports are reported. At reset, these fields should reflect the same mode as **PowerSwitchingMode**. This field is valid only if the **NoOverCurrentProtection** field is cleared. 0 Over-current status is reported collectively for all downstream ports. 1 Over-current status is reported on per-port basis.
10	R	R	0x0	**Device Type** This bit specifies that the Root Hub is not a compound device. The Root Hub is not permitted to be a compound device. This field should always read/write 0.
9	R/W	R	1	**PowerSwitchingMode** This bit is used to specify how the power switching of the Root Hub ports is controlled. It is implementation-specific. This field is only valid if the **NoPowerSwitching** field is cleared. 0 All ports are powered at the same time. 1 Each port is powered individually. This mode allows port power to be controlled by either the global switch or per-port switching. If the **PortPowerControlMask** bit is set, the port responds only to port power commands (**Set/ClearPortPower**). If the port mask is cleared, then the port is controlled only by the global power switch (**Set/ClearGlobalPower**).

8	R/W	R	0	**NoPowerSwithcing** These bits are used to specify whether power switching is supported or ports are always powered. It is implementation-specific. When this bit is cleared, the **PowerSwitchingMode** specifies global or per-port switching. **0** Ports are power switched. **1** Ports are always powered on when the HC is powered on.
7:0	R	R	0x01	**NumberDownstreamPorts** These bits specify the number of downstream ports supported by the Root Hub. It is implementation-specific. The minimum number of ports is 1. The maximum number of ports supported.

9.5.5.20. 0x044C HcRhDescriptorB Register (Default Value:0x0000_0000)

Offset: 0x044C				**Register Name: HcRhDescriptorB Register**
Bit	**Read/Write**		**Default/Hex**	**Description**
	HCD	**HC**		
31:16	R/W	R	0x0	**PortPowerControlMask** Each bit indicates if a port is affected by a global power control command when **PowerSwitchingMode** is set. When set, the port's power state is only affected by per-port power control (**Set/ClearPortPower**). When cleared, the port is controlled by the global power switch (**Set/ClearGlobalPower**). If the device is configured to global switching mode (**PowerSwitchingMode** = 0), this field is not valid. Bit0 Reserved Bit1 Ganged-power mask on Port #1. Bit2 Ganged-power mask on Port #2. … Bit15 Ganged-power mask on Port #15.
15:0	R/W	R	0x0	**DeviceRemovable** Each bit is dedicated to a port of the Root Hub. When cleared, the attached device is removable. When set, the attached device is not removable. Bit0 Reserved Bit1 Device attached to Port #1. Bit2 Device attached to Port #2. … Bit15 Device attached to Port #15.

9.5.5.21. 0x0450 HcRhStatus Register(Default Value:0x0000_0000)

Offset: 0x0450	**Register Name: HcRhStatus Register**

Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31	W	R	0x0	**(write)ClearRemoteWakeupEnable** Write a '1' clears **DeviceRemoteWakeupEnable.** Writing a '0' has no effect.
30:18	/	/	/	**/**
17	R/W	R	0x0	**OverCurrentIndicatorChang** This bit is set by hardware when a change has occurred to the **OverCurrentIndicator** field of this register. The HCD clears this bit by writing a '1'. Writing a '0' has no effect.
16	R/W	R	0x0	**(read)LocalPowerStartusChange** The Root Hub does not support the local power status features, thus, this bit is always read as '0'. **(write)SetGlobalPower** In global power mode (**PowerSwitchingMode**=0), This bit is written to '1' to turn on power to all ports (clear **PortPowerStatus**). In per-port power mode, it sets **PortPowerStatus** only on ports whose **PortPowerControlMask** bit is not set. Writing a '0' has no effect.
15	R/W	R	0x0	**(read)DeviceRemoteWakeupEnable** This bit enables a **ConnectStatusChange** bit as a resume event, causing a USBSUSPEND to USBRESUME state transition and setting the **ResumeDetected** interrupt. 0 – **ConnectStatusChange** is not a remote wakeup event. 1 – **ConnectStatusChange** is a remote wakeup event. **(write)SetRemoteWakeupEnable** Writing a '1' sets DeviceRemoveWakeupEnable. Writing a '0' has no effect.
14:2	/	/	/	**/**
1	R	R/W	0x0	**OverCurrentIndicator** This bit reports overcurrent conditions when the global reporting is implemented. When set, an overcurrent condition exists. When cleared, all power operations are normal. If per-port overcurrent protection is implemented this bit is always '0'
0	R/W	R	0x0	**(Read)LocalPowerStatus** When read, this bit returns the LocalPowerStatus of the Root Hub. The Root Hub does not support the local power status feature; thus, this bit is always read as '0'. **(Write)ClearGlobalPower** When write, this bit is operated as the ClearGlobalPower. In global power mode (**PowerSwitchingMode**=0), This bit is written to '1' to turn off power to all ports (clear **PortPowerStatus**). In per-port power mode, it clears **PortPowerStatus** only on ports whose **PortPowerControlMask** bit is not set. Writing a '0' has no effect.

9.5.5.22. 0x0454 HcRhPortStatus Register(Default Value:0x0000_0100)

Offset: 0x0454				Register Name: HcRhPortStatus
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:21	/	/	/	/
20	R/W	R/W	0x0	**PortResetStatusChange** This bit is set at the end of the 10-ms port reset signal. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. 0 port reset is not complete 1 port reset is complete
19	R/W	R/W	0x0	**PortOverCurrentIndicatorChange** This bit is valid only if overcurrent conditions are reported on a per-port basis. This bit is set when Root Hub changes the **PortOverCurrentIndicator** bit. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. 0 no change in **PortOverCurrentIndicator** 1 **PortOverCurrentIndicator** has changed
18	R/W	R/W	0x0	**PortSuspendStatusChange** This bit is set when the full resume sequence has been completed. This sequence includes the 20-s resume pulse, LS EOP, and 3-ms resychronization delay. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. This bit is also cleared when **ResetStatusChange** is set. 0 resume is not completed 1 resume completed
17	R/W	R/W	0x0	**PortEnableStatusChange** This bit is set when hardware events cause the **PortEnableStatus** bit to be cleared. Changes from HCD writes do not set this bit. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. 0 no change in **PortEnableStatus** 1 change in **PortEnableStatus**
16	R/W	R/W	0x0	**ConnectStatusChange** This bit is set whenever a connect or disconnect event occurs. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. If **CurrentConnectStatus** is cleared when a **SetPortReset,SetPortEnable**, or **SetPortSuspend** write occurs, this bit is set to force the driver to re-evaluate the connection status since these writes should not occur if the port is disconnected. 0 no change in **PortEnableStatus** 1 change in **PortEnableStatus** **Note: If the DeviceRemovable[NDP] bit is set, this bit is set only after a Root Hub reset to inform the system that the device is attached.**
15:10	/	/	/	/

<table>
<tr><td>9</td><td>R/W</td><td>R/W</td><td>0x0</td><td>(read)LowSpeedDeviceAttached
This bit indicates the speed of the device attached to this port. When set, a Low Speed device is attached to this port. When clear, a Full Speed device is attached to this port. This field is valid only when the CurrentConnectStatus is set.
<table><tr><td>0</td><td>full speed device attached</td></tr><tr><td>1</td><td>low speed device attached</td></tr></table>
(write)ClearPortPower
The HCD clears the PortPowerStatus bit by writing a ‘1’ to this bit. Writing a ‘0’ has no effect.</td></tr>
<tr><td>8</td><td>R/W</td><td>R/W</td><td>0x1</td><td>(read)PortPowerStatus
This bit reflects the port’s power status, regardless of the type of power switching implemented. This bit is cleared if an overcurrent condition is detected. HCD sets this bit by writing SetPortPower or SetGlobalPower. HCD clears this bit by writing ClearPortPower or ClearGlobalPower. Which power control switches are enabled is determined by PowerSwitchingMode and PortPortControlMask[NumberDownstreamPort]. In global switching mode(PowerSwitchingMode=0), only Set/ClearGlobalPower controls this bit. In per-port power switching (PowerSwitchingMode=1), if the PortPowerControlMask[NDP] bit for the port is set, only Set/ClearPortPower commands are enabled. If the mask is not set, only Set/ClearGlobalPower commands are enabled. When port power is disabled, CurrentConnectStatus, PortEnableStatus, PortSuspendStatus, and PortResetStatus should be reset.
<table><tr><td>0</td><td>port power is off</td></tr><tr><td>1</td><td>port power is on</td></tr></table>
(write)SetPortPower
The HCD writes a ‘1’ to set the PortPowerStatus bit. Writing a ‘0’ has no effect.

Note: This bit is always reads ‘1b’ if power switching is not supported.</td></tr>
<tr><td>7:5</td><td>/</td><td>/</td><td>/</td><td>/</td></tr>
<tr><td>4</td><td>R/W</td><td>R/W</td><td>0x0</td><td>(read)PortResetStatus
When this bit is set by a write to SetPortReset, port reset signaling is asserted. When reset is completed, this bit is cleared when PortResetStatusChange is set. This bit cannot be set if CurrentConnectStatus is cleared.
<table><tr><td>0</td><td>port reset signal is not active</td></tr><tr><td>1</td><td>port reset signal is active</td></tr></table>
(write)SetPortReset
The HCD sets the port reset signaling by writing a ‘1’ to this bit. Writing a</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td>‘0’ has no effect. If CurrentConnectStatus is cleared, this write does not set PortResetStatus, but instead sets ConnectStatusChange. This informs the driver that it attempted to reset a disconnected port.</td></tr>
<tr><td>3</td><td>R/W</td><td>R/W</td><td>0x0</td><td>(read)PortOverCurrentIndicator
This bit is only valid when the Root Hub is configured in such a way that overcurrent conditions are reported on a per-port basis. If per-port overcurrent reporting is not supported, this bit is set to 0. If cleared, all power operations are normal for this port. If set, an overcurrent condition exists on this port. This bit always reflects the overcurrent input signal.
<table>
<tr><td>0</td><td>no overcurrent condition.</td></tr>
<tr><td>1</td><td>overcurrent condition detected.</td></tr>
</table>
(write)ClearSuspendStatus
The HCD writes a ‘1’ to initiate a resume. Writing a ‘0’ has no effect. A resume is initiated only if PortSuspendStatus is set.</td></tr>
<tr><td>2</td><td>R/W</td><td>R/W</td><td>0x0</td><td>(read)PortSuspendStatus
This bit indicates the port is suspended or in the resume sequence. It is set by a SetSuspendState write and cleared when PortSuspendStatusChange is set at the end of the resume interval. This bit cannot be set if CurrentConnectStatus is cleared. This bit is also cleared when PortResetStatusChange is set at the end of the port reset or when the HC is placed in the USBRESUME state. If an upstream resume is in progress, it should propagate to the HC.
<table>
<tr><td>0</td><td>port is not suspended</td></tr>
<tr><td>1</td><td>port is suspended</td></tr>
</table>
(write)SetPortSuspend
The HCD sets the PortSuspendStatus bit by writing a ‘1’ to this bit. Writing a ‘0’ has no effect. If CurrentConnectStatus is cleared, this write does not set PortSuspendStatus; instead it sets ConnectStatusChange. This informs the driver that it attempted to suspend a disconnected port.</td></tr>
<tr><td>1</td><td>R/W</td><td>R/W</td><td>0x0</td><td>(read)PortEnableStatus
This bit indicates whether the port is enabled or disabled. The Root Hub may clear this bit when an overcurrent condition, disconnect event, switched-off power, or operational bus error such as babble is detected. This change also causes PortEnabledStatusChange to be set. HCD sets this bit by writing SetPortEnable and clears it by writing ClearPortEnable. This bit cannot be set when CurrentConnectStatus is cleared. This bit is also set, if not already, at the completion of a port reset when ResetStatusChange is set or port suspend when SuspendStatusChange is set.
<table>
<tr><td>0</td><td>port is disabled</td></tr>
<tr><td>1</td><td>port is enabled</td></tr>
</table></td></tr>
</table>

				(write)**SetPortEnable** The HCD sets **PortEnableStatus** by writing a '1'. Writing a '0' has no effect. If **CurrentConnectStatus** is cleared, this write does not set **PortEnableStatus**, but instead sets **ConnectStatusChange**. This informs the driver that it attempted to enable a disconnected Port.
0	R/W	R/W	0x0	(read)**CurrentConnectStatus** This bit reflects the current state of the downstream port. 0 – No device connected 1 – Device connected (write)**ClearPortEnable** The HCD writes a '1' to clear the PortEnableStatus bit. Writing '0' has no effect. The CurrentConnectStatus is not affected by any write. **Note: This bit is always read '1' when the attached device is nonremovalble(DviceRemoveable[NumberDownstreamPort]).**

9.5.6. HCI Contgroller and PHY Interface Description

9.5.6.1. 0x0800 HCI Interface Register(Default Value:0x1000_0000)

Offset: 0x0800			**Register Name: USB_CTRL**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	Reserved.
28	R	0x1	DMA Transfer Status Enable 0: Disable 1: Enable
27:26	/	/	/
25	R/W	0x0	OHCI count select 1: Simulation mode, the counters will be much shorter then real time 0: Normal mode, the counters will count full time
24	R/W	0x0	Simulation mode 1: Set PHY in a non-driving mode so the EHCI can detect device connection, this is used only for simulation 0: No effect
23:21	/	/	/
20	R/W	0x0	EHCI HS force Set 1 to this field force the ehci enter the high speed mode during bus reset. This field only valid when the bit 1 is set.
19	/	/	/
18	R/W	0x0	1: within 2us of the resume-K to SE0 transition 0: random time value of the resume-K to SE0 transition
17:13	/	/	/
12	R/W	0x0	PP2VBUS

			1: ULPI wrapper interface will automatically set or clear DrvVbus register in ULPI PHY according to the port power status form the root hub 0: ULPI wrapper will ignore the difference between power status of root hub and ULPI PHY
11	R/W	0x0	AHB Master interface INCR16 enable 1: Use INCR16 when appropriate 0: do not use INCR16,use other enabled INCRX or unspecified length burst INCR
10	R/W	0x0	AHB Master interface INCR8 enable 1: Use INCR8 when appropriate 0: do not use INCR8,use other enabled INCRX or unspecified length burst INCR
9	R/W	0x0	AHB Master interface burst type INCR4 enable 1: Use INCR4 when appropriate 0: do not use INCR4,use other enabled INCRX or unspecified length burst INCR
8	R/W	0x0	AHB Master interface INCRX align enable 1: start INCRx burst only on burst x-align address 0: Start burst on any double word boundary Note: This bit must enable if any bit of 11:9 is enabled
7:3	/	/	/
2	R/W	0x0	RC16M CLK enable 0: disable 1:enable
1	/	/	/
0	R/W	0x0	ULPI bypass enable 1: Enable UTMI interface, disable ULPI interface 0: Enable ULPI interface, disable UTMI interface

9.5.6.2. 0x0810 PHY Control Register(Default Value: 0x0000_0008)

Offset: 0x0810			**Register Name: PHY Control Register**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	**LOOPBACKENB**
6	R/W	0x0	**IDPULLUP**
5	R/W	0x0	**VBUSVLDEXT（for phy vbus）** 0: invalid 1: valid
4	R/W	0x0	**VBUSVLDEXTSEL** The internal signal is fixed at 1, the register value is invalid.
3	R/W	0x1	**SIDDQ**
2	R/W	0x0	**COMMONONN**

1:0	R/W	0x0	**VATESTENB**

9.5.6.3. 0x081C HSIC PHY Tune1 Register(Default Value: 0x0000_0010)

Offset: 0x081C			**Register Name: HSIC PHY tune1 Register**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6:5	/	/	TXRPUTUNE
5:4	R/W	0x1	TXRPDTUNE
3:0	R/W	0x0	TXSRTUNE

9.5.6.4. 0x0820 HSIC PHY Tune2 Register(Default Value: 0x0000_0010)

Offset: 0x0820			**Register Name: HSIC PHY tune2 Register**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	bist_en
30	R/W	0x0	TESTBURNIN
29	R/W	0x0	TESTDATAOUTSEL
28	R/W	0x0	TESTCLK
27:24	R/W	0x0	TESTADDR
23:16	R/W	0x0	TESTDATAIN
15:4	R/W	0x1	siddq
3:0	R/W	0x0	refclk div

9.5.6.5. 0x0824 HSIC PHY Tune3 Register(Default Value: 0x0000_0010)

Offset: 0x0824			**Register Name: HSIC PHY tune3 Register**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
5	R/W	0x0	hsic bist_error
4	R/W	0x0	hsic bist_done
3:2	R/W	0x0	hsic testdata out[3:2]
1	R/W	0x1	Non-hsic mode bist_error testdata out[1]
0	R/W	0x0	Non-hsic mode bist_done testdata out[0]

9.5.6.6. 0x0828 HCI SIE Port Disable Control Register(Default Value:0x1000_0000)

Offset: 0x0828	**Register Name: USB_SPDCR**

Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	SE0 Status This bit is set when no-se0 is detected before SOF when bit[1:0] is 10b or 11b.
15:5	/	/	/
4	R/W	0x0	resume_sel When set k-se0 transition 2us, setting this bit to 1, which is cooperated with ss_utmi_backward_enb_i.
3:2	/	/	/
1:0	R/W	0x0	Port Disable Control 00: Port Disable when no-se0 detect before SOF 01: Port Disable when no-se0 detect before SOF 10: No Port Disable when no-se0 detect before SOF 11:Port Disable when no-se0 3 time detect before SOF during 8 frames

9.6. Port Controller

9.6.1. Overview

The Port Controller can be configured with multi-functional input/output pins. All these ports can be configured as GPIO only if multiplexed functions are not used. The total 6 group external PIO interrupt sources are supported and interrupt mode can be configured by software.

The Port Controller has the following features:

- 6 ports(PC,PF,PG,PH,PI,PL)
- Software control for each signal pin
- GPIO peripheral can produce interrupt
- Pull-up/Pull-down/no-Pull register control
- Control the direction of every signal
- 4 drive strengths in each operating mode
- Up to 72 interrupts
- Configurable interrupt edges

9.6.2. Block Diagram

The block diagram of port controller is shown in Figure 9-32.

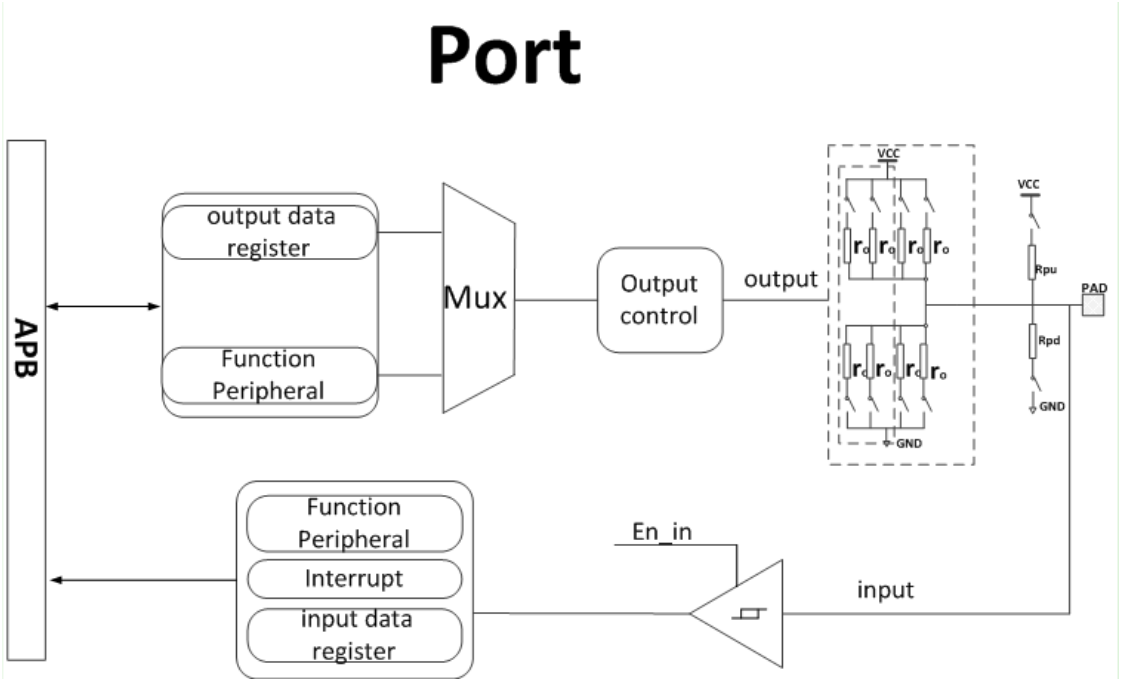

Figure 9- 32. Port Controller Block Diagram

Port controller consists of digital part(GPIO, external interface) and IO analog part(output buffer, dual pull down, pad, etc). Digital part can select output interface by MUX switch; analog part can configure pull up/down, buffer strength.

When executing GPIO read state, the port controller reads the current level of pin into internal register bus. When not executing GPIO read state, external pin and internal register bus is off-status, that is high-impedance.

9.6.3. Operations and Functional Descriptions

9.6.3.1. Multi-function Port Table

The H616 includes 74 multi-functional input/output port pins. There are 6 ports as listed below.

Table 9- 14. H616 Multi-function Port Table

Port Name	Number of Pins	Input Driver	Output Driver	Multiplex Pins	Power
PC	17	Schmitt	CMOS	NAND/SDC/SPI/BOOT/PC-EINT	1.8V/3.3V
PF	7	Schmitt	CMOS	SDC/JTAG/UART/PF-EINT	1.8V/3.3V
PG	20	Schmitt	CMOS	SDC/UART/Audio HUB/PLL/BIST/TWI/PWM/ ADC Digital interface/PG-EINT	1.8V/3.3V
PH	11	Schmitt	CMOS	UART/TCON_TRIG/OWA/Audio HUB/PWM/SPI/ TWI/CIR/PH-EINT	3.3V
PI	17	Schmitt	CMOS	RGMII/DMIC/UART/Audio HUB/TS/TWI/PWM/ HDMI_DDC_CEC/CLK_FANOUT/PI-EINT	1.8V/2.8V/3.3V
PL	2	Schmitt	CMOS	S_TWI	1.8V

The multiplex function pins are shown in Table 9-15 to Table 9-20.

Table 9- 15. PC Multiplex Function Select

GPIO Port	Function2	Function3	Function4	Function5	Function6
PC0	NAND_WE	SDC2_DS	SPI0_CLK		PC_EINT0
PC1	NAND_ALE	SDC2_RST			PC_EINT1
PC2	NAND_CLE		SPI0_MOSI		PC_EINT2
PC3	NAND_CE1		SPI0_CS0	BOOT_SEL1	PC_EINT3
PC4	NAND_CE0		SPI0_MISO	BOOT_SEL2	PC_EINT4
PC5	NAND_RE	SDC2_CLK		BOOT_SEL3	PC_EINT5
PC6	NAND_RB0	SDC2_CMD		BOOT_SEL4	PC_EINT6
PC7	NAND_RB1		SPI0_CS1		PC_EINT7
PC8	NAND_DQ7	SDC2_D3			PC_EINT8
PC9	NAND_DQ6	SDC2_D4			PC_EINT9
PC10	NAND_DQ5	SDC2_D0			PC_EINT10
PC11	NAND_DQ4	SDC2_D5			PC_EINT11
PC12	NAND_DQS				PC_EINT12
PC13	NAND_DQ3	SDC2_D1			PC_EINT13
PC14	NAND_DQ2	SDC2_D6			PC_EINT14
PC15	NAND_DQ1	SDC2_D2	SPI0_WP		PC_EINT15
PC16	NAND_DQ0	SDC2_D7	SPI0_HOLD		PC_EINT16

Table 9- 16. PF Multiplex Function Select

GPIO Port	Function2	Function3	Function4	Function5	Function6
PF0	SDC0_D1	JTAG_MS			PF_EINT0
PF1	SDC0_D0	JTAG_DI			PF_EINT1
PF2	SDC0_CLK	UART0_TX			PF_EINT2
PF3	SDC0_CMD	JTAG_DO			PF_EINT3
PF4	SDC0_D3	UART0_RX			PF_EINT4
PF5	SDC0_D2	JTAG_CK			PF_EINT5
PF6					PF_EINT6

Table 9- 17. PG Multiplex Function Select

GPIO Port	Function2	Function3	Function4	Function5	Function6
PG0	SDC1_CLK				PG_EINT0
PG1	SDC1_CMD				PG_EINT1
PG2	SDC1_D0				PG_EINT2
PG3	SDC1_D1				PG_EINT3
PG4	SDC1_D2				PG_EINT4
PG5	SDC1_D3				PG_EINT5
PG6	UART1_TX		JTAG_MS		PG_EINT6
PG7	UART1_RX		JTAG_CK		PG_EINT7
PG8	UART1_RTS	PLL_LOCK_DBG	JTAG_DO		PG_EINT8
PG9	UART1_CTS		JTAG_DI	AC_ADCY	PG_EINT9
PG10	H_I2S2_MCLK	X32KFOUT		AC_MCLK	PG_EINT10
PG11	H_I2S2_BCLK		BIST_RESULT0	AC_SYNC	PG_EINT11
PG12	H_I2S2_LRCK		BIST_RESULT1	AC_ADCL	PG_EINT12
PG13	H_I2S2_DOUT0	H_I2S2_DIN1	BIST_RESULT2	AC_ADCR	PG_EINT13
PG14	H_I2S2_DIN0	H_I2S2_DOUT1	BIST_RESULT3	AC_ADCX	PG_EINT14
PG15	UART2_TX			TWI4_SCK	PG_EINT15
PG16	UART2_RX			TWI4_SDA	PG_EINT16
PG17	UART2_RTS			TWI3_SCK	PG_EINT17
PG18	UART2_CTS			TWI3_SDA	PG_EINT18
PG19			PWM1		PG_EINT19

Table 9- 18. PH Multiplex Function Select

GPIO Port	Function2	Function3	Function4	Function5	Function6
PH0	UART0_TX		PWM3	TWI1_SCK	PH_EINT0
PH1	UART0_RX		PWM4	TWI1_SDA	PH_EINT1
PH2	UART5_TX	OWA_MCLK	PWM2	TWI2_SCK	PH_EINT2
PH3	UART5_RX		PWM1	TWI2_SDA	PH_EINT3
PH4		OWA_OUT		TWI3_SCK	PH_EINT4

PH5	UART2_TX	H_I2S3_MCLK	SPI1_CS0	TWI3_SDA	PH_EINT5
PH6	UART2_RX	H_I2S3_BCLK	SPI1_CLK	TWI4_SCK	PH_EINT6
PH7	UART2_RTS	H_I2S3_LRCK	SPI1_MOSI	TWI4_SDA	PH_EINT7
PH8	UART2_CTS	H_I2S3_DOUT0	SPI1_MISO	H_I2S3_DIN1	PH_EINT8
PH9		H_I2S3_DIN0	SPI1_CS1	H_I2S3_DOUT1	PH_EINT9
PH10		CIR_RX	TCON_TRIG1		PH_EINT10

Table 9- 19. PI Multiplex Function Select

GPIO Port	Function2	Function3	Function4	Function5	Function6
PI0	RGMII_RXD3/ RMII_NULL	DMIC_CLK	H_I2S0_MCLK	HDMI_SCL	PI_EINT0
PI1	RGMII_RXD2/ RMII_NULL	DMIC_DATA0	H_I2S0_BCLK	HDMI_SDA	PI_EINT1
PI2	RGMII_RXD1/ RMII_RXD1	DMIC_DATA1	H_I2S0_LRCK	HDMI_CEC	PI_EINT2
PI3	RGMII_RXD0/ RMII_RXD0	DMIC_DATA2	H_I2S0_DOUT0	H_I2S0_DIN1	PI_EINT3
PI4	RGMII_RXCK/ RMII_NULL	DMIC_DATA3	H_I2S0_DIN0	H_I2S0_DOUT1	PI_EINT4
PI5	RGMII_RXCTL/ RMII_CRS_DV	UART2_TX	TS0_CLK	TWI0_SCK	PI_EINT5
PI6	RGMII_NULL/ RMII_RXER	UART2_RX	TS0_ERR	TWI0_SDA	PI_EINT6
PI7	RGMII_TXD3/ RMII_NULL	UART2_RTS	TS0_SYNC	TWI1_SCK	PI_EINT7
PI8	RGMII_TXD2/ RMII_NULL	UART2_CTS	TS0_DVLD	TWI1_SDA	PI_EINT8
PI9	RGMII_TXD1/ RMII_TXD1	UART3_TX	TS0_D0	TWI2_SCK	PI_EINT9
PI10	RGMII_TXD0/ RMII_TXD0	UART3_RX	TS0_D1	TWI2_SDA	PI_EINT10
PI11	RGMII_TXCK/ RMII_TXCK	UART3_RTS	TS0_D2	PWM1	PI_EINT11
PI12	RGMII_TXCTL/ RMII_TXEN	UART3_CTS	TS0_D3	PWM2	PI_EINT12
PI13	RGMII_CLKIN/ RMII_NULL	UART4_TX	TS0_D4	PWM3	PI_EINT13
PI14	MDC	UART4_RX	TS0_D5	PWM4	PI_EINT14
PI15	MDIO	UART4_RTS	TS0_D6	CLK_FANOUT0	PI_EINT15
PI16	EPHY_25M	UART4_CTS	TS0_D7	CLK_FANOUT1	PI_EINT16

Table 9- 20. PL Multiplex Function Select

GPIO Port	Function2	Function3	Function4	Function5	Function6
PL0		S_TWI0_SCK			
PL1		S_TWI0_SDA			

9.6.3.2. Port Function

Port Controller supports 6 GPIOs, every GPIO can configure as Input, Output, Functional Peripheral, IO disable or Interrupt function. The configuration instruction of every function is as follows.

Table 9- 21. Port Function

	Function	Buffer Strength	Pull Up	Pull Down
Input	Input function, default input is 0	/	N	Y
	Input function, default input is 1	/	Y	N
Output	Output function	Y	X	X
Disable	Pull-up	/	Y	N
	Pulll-down	/	N	Y
	High-impedance	/	N	N
Interrupt	Trigger mode	/	X	X

/: Non-configure, configuration is invalid

Y: Need configure

X: Select configuration according to actual situation

N: Forbid to configure

9.6.3.3. Pull Up/Down and High-Impedance Logic

Each IO pin can configure the internal pull-up/down function or high-impedance.

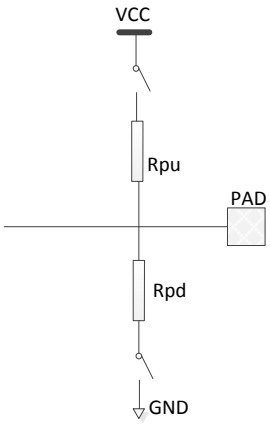

Figure 9- 33. Pull up/down Logic

High-impedance, the output is float state, all buffers are off, the level is decided by external high/low level. When

high-impedance, software configures the switch on Rpu and Rpd as off, and the multiplexing function of IO is set as IO disable or input by software.

Pull-up, an uncertain signal is pulled high by a resistor, the resistor has current-limiting function. When pulling up, the switch on Rpu is breakover by software configuration, IO is pulled up to VCC by Rpu.

Pull-down, an uncertain signal is pulled low by a resistor. When pulling down, the switch on Rpd is breakover by software configuration, IO is pulled down to GND by Rpd.

The pull-up/down of each IO is weak pull-up/down, the pull-up/down resistor contains three kinds of resistance values : 4.7 kΩ, 15 kΩ and 100 kΩ.

The setting of pull-down input, pull-up input, high-impledance input is decided by external circuit.

9.6.3.4. Buffer Strength

Each IO can be set as different buffer strength. The IO buffer diagram is as follows.

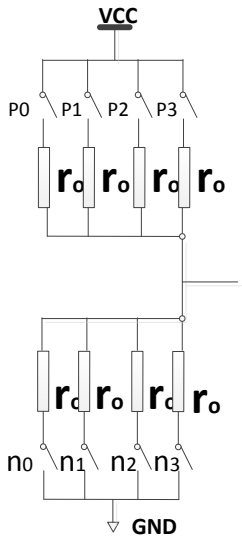

Figure 9- 34. IO Buffer Strength Diagram

When output high level, the n0,n1,n2,n3 of NMOS are off, the p0,p1,p2,p3 of PMOS are on. When buffer strength is set to 0(buffer strength is weakest), only p0 is on, the output impedance is maximum, the impedance value is r0 (on-resistance). When buffer strength is set to 1, only p0 and p1 are on, the output impedance is equivalent to two r0 in parallel, the impedance value is r0/2. When buffer strength is 2, only p0,p1 and p2 are on, the output impedance is equivalent to three r0 in parallel, the impedance value is r0/3. When buffer strength is 3, p0,p1,p2 and p3 are on, the output impedance is equivalent to four r0 in parallel, the impedance value is r0/4.

When output low level, the p0,p1,p2,p3 of PMOS is off, the n0,n1,n2,n3 of NMOS is on. When buffer strength is set to 0(buffer strength is weakest), only n0 is on, the output impedance is maximum, the impedance value is r0. When buffer strength is set to 1, only n0 and n1 are on, the output impedance is equivalent to two r0 in parallel, the impedance value is r0/2. When buffer strength is 2, only n0,n1 and n2 are on, the output impedance is equivalent to three r0 in parallel, the impedance value is r0/3. When buffer strength is 3, n0,n1,n2 and n3 are on, the output

impedance is equivalent to four r0 in parallel, the impedance value is r0/4.

When GPIO is set to input or interrupt function, between output driver circuit and port is unconnected, driver configuration is invalid.

NOTE

The typical value of r0 has 180Ω, 120Ω, 100Ω and 50Ω.

9.6.3.5. Interrupt

Each group IO has independent interrupt number. IO within group uses one interrupt number, when one IO generates interrupt, Port Controller sent interrupt request to GIC. External Interrupt Status Register is used to query which IO generates interrupt.

Interrupt trigger of GPIO supports the following trigger types.

- Positive Edge: When low level changes to high level, the interrupt will generate. No matter how long high level keeps, the interrupt generates only once.
- Negative Edge: When high level changes to low level, the interrupt will generate. No matter how long low level keeps, the interrupt generates only once.
- High Level : Just keep high level and the interrupt will always generate.
- Low Level : Just keep low level and the interrupt will always generate.
- Double Edge : Positive and negative edge.

External Interrupt Configure Register is used to configure trigger type.

GPIO interrupt supports hardware debounce function by setting External Interrupt Debounce Register. Sample trigger signal using lower sample clock, to reach the debounce effect because of the dither frequency of signal is higher than sample frequency.

Set sample clock source by PIO_INT_CLK_SELECT and prescale factor by DEB_CLK_PRE_SCALE.

9.6.4. Register List

Module Name	Base Address
GPIO(PC,PF,PG,PH,PI)	0x0300B000

Register Name	Offset	Description
Pn_CFG0	n*0x0024+0x00	Port n Configure Register 0 (n =2,5,6,7,9)
Pn_CFG1	n*0x0024+0x04	Port n Configure Register 1 (n =2,5,6,7,9)
Pn_CFG2	n*0x0024+0x08	Port n Configure Register 2 (n =2,5,6,7,9)
Pn_CFG3	n*0x0024+0x0C	Port n Configure Register 3 (n =2,5,6,7,9)

Pn_DAT	n*0x0024+0x10	Port n Data Register (n =2,5,6,7,9)
Pn_DRV0	n*0x0024+0x14	Port n Multi-Driving Register 0 (n =2,5,6,7,9)
Pn_DRV1	n*0x0024+0x18	Port n Multi-Driving Register 1 (n =2,5,6,7,9)
Pn_PUL0	n*0x0024+0x1C	Port n Pull Register 0 (n =2,5,6,7,9)
Pn_PUL1	n*0x0024+0x20	Port n Pull Register 1 (n =2,5,6,7,9)
Pn_INT_CFG0	0x200+n*0x20+0x00	PIO Interrupt Configure Register 0 (n =2,5,6,7,9)
Pn_INT_CFG1	0x200+n*0x20+0x04	PIO Interrupt Configure Register 1 (n =2,5,6,7,9)
Pn_INT_CFG2	0x200+n*0x20+0x08	PIO Interrupt Configure Register 2 (n =2,5,6,7,9)
Pn_INT_CFG3	0x200+n*0x20+0x0C	PIO Interrupt Configure Register 3 (n =2,5,6,7,9)
Pn_INT_CTL	0x200+n*0x20+0x10	PIO Interrupt Control Register (n =2,5,6,7,9)
Pn_INT_STA	0x200+n*0x20+0x14	PIO Interrupt Status Register (n =2,5,6,7,9)
Pn_INT_DEB	0x200+n*0x20+0x18	PIO Interrupt Debounce Register (n =2,5,6,7,9)
PIO_POW_MOD_SEL	0x0340	PIO Group Withstand Voltage Mode Select Register
PIO_POW_MS_CTL	0x0344	PIO Group Withstand Voltage Mode Select Control Register
PIO_POW_VAL	0x0348	PIO Group Power Value Register

Module Name	**Base Address**
GPIO(PL)	0x07022000

Register Name	**Offset**	**Description**
PL_CFG0	0x0000	Port L Configure Register 0
PL_DAT	0x0010	Port L Data Register
PL_DRV0	0x0014	Port L Multi-Driving Register 0
PL_PUL0	0x001C	Port L Pull Register 0

9.6.5. GPIO(PC,PF,PG,PH,PI) Register Description

9.6.5.1. 0x0048 PC Configure Register 0 (Default Value: 0x7555_5777)

Offset: 0x0048			**Register Name: PC_CFG0**
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30:28	R/W	0x7	PC7_SELECT 000:Input 001:Output 010:NAND_RB1 011:Reserved 100:SPI_CS1 101:Reserved 110:PC_EINT7 111:IO Disable
27	/	/	/
26:24	R/W	0x5	PC6_SELECT 000:Input 001:Output

			010:NAND_RB0 100:Reserved 110:PC_EINT6	011:SDC2_CMD 101:BOOT_SEL4 111:IO Disable
23	/	/	/	
22:20	R/W	0x5	PC5_SELECT 000:Input 010:NAND_RE 100:Reserved 110:PC_EINT5	 001:Output 011:SDC2_CLK 100:BOOT_SEL3 111:IO Disable
19	/	/	/	
18:16	R/W	0x5	PC4_SELECT 000:Input 010:NAND_CE0 100:SPI_MISO 110:PC_EINT4	 001:Output 011:Reserved 101:BOOT_SEL2 111:IO Disable
15	/	/	/	
14:12	R/W	0x5	PC3_SELECT 000:Input 010:NAND_CE1 100:SPI0_CS0 110:PC_EINT3	 001:Output 011:Reserved 101:BOOT_SEL1 111:IO Disable
11	/	/	/	
10:8	R/W	0x7	PC2_SELECT 000:Input 010:NAND_CLE 100:SPI0_MOSI 110:PC_EINT2	 001:Output 011:Reserved 101:Reserved 111:IO Disable
7	/	/	/	
6:4	R/W	0x7	PC1_SELECT 000:Input 010:NAND_ALE 100:Reserved 110:PC_EINT1	 001:Output 011:SDC2_RST 101:Reserved 111:IO Disable
3	/	/	/	
2:0	R/W	0x7	PC0_SELECT 000:Input 010:NAND_WE 100:SPI0_CLK 110:PC_EINT0	 001:Output 011:SDC2_DS 101:Reserved 111:IO Disable

9.6.5.2. 0x004C PC Configure Register 1 (Default Value: 0x7777_7777)

Offset: 0x004C	Register Name: PC_CFG1

Bit	Read/Write	Default/Hex	Description
31	/	/	/
30:28	R/W	0x7	PC15_SELECT 000:Input 001:Output 010:NAND_DQ1 011:SDC2_D2 100:SPI0_WP 101:Reserved 110:PC_EINT15 111:IO Disable
27	/	/	/
26:24	R/W	0x7	PC14_SELECT 000:Input 001:Output 010:NAND_DQ2 011:SDC2_D6 100:Reserved 101:Reserved 110:PC_EINT14 111:IO Disable
23	/	/	/
22:20	R/W	0x7	PC13_SELECT 000:Input 001:Output 010:NAND_DQ3 011:SDC2_D1 100:Reserved 101:Reserved 110:PC_EINT13 111:IO Disable
19	/	/	/
18:16	R/W	0x7	PC12_SELECT 000:Input 001:Output 010:NAND_DQS 011:Reserved 100:Reserved 101:Reserved 110:PC_EINT12 111:IO Disable
15	/	/	/
14:12	R/W	0x7	PC11_SELECT 000:Input 001:Output 010:NAND_DQ4 011:SDC2_D5 100:Reserved 101:Reserved 110:PC_EINT11 111:IO Disable
11	/	/	/
10:8	R/W	0x7	PC10_SELECT 000:Input 001:Output 010:NAND_DQ5 011:SDC2_D0 100:Reserved 101:Reserved 110:PC_EINT10 111:IO Disable
7	/	/	/
6:4	R/W	0x7	PC9_SELECT 000:Input 001:Output 010:NAND_DQ6 011:SDC2_D4 100:Reserved 101:Reserved 110:PC_EINT9 111:IO Disable
3	/	/	/

2:0	R/W	0x7	PC8_SELECT 000:Input 010:NAND_DQ7 100:Reserved 110:PC_EINT8	001:Output 011:SDC2_D3 101:Reserved 111:IO Disable

9.6.5.3. 0x0050 PC Configure Register 2 (Default Value: 0x0000_0007)

Offset: 0x0050			Register Name: PC_CFG2	
Bit	**Read/Write**	**Default/Hex**	**Description**	
31:3	/	/	/	
2:0	R/W	0x7	PC16_SELECT 000:Input 010:NAND_DQ0 100:SPI0_HOLD 110:PC_EINT16	001:Output 011:SDC2_D7 100:Reserved 111:IO Disable

9.6.5.4. 0x0058 PC Data Register (Default Value: 0x0000_0000)

Offset: 0x0058			Register Name: PC_DAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16:0	R/W	0x0	PC_DAT If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read.

9.6.5.5. 0x005C PC Multi-Driving Register 0 (Default Value: 0x5555_5555)

Offset: 0x005C			Register Name: PC_DRV0	
Bit	**Read/Write**	**Default/Hex**	**Description**	
31:30	R/W	0x1	PC15_DRV PC15 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
29:28	R/W	0x1	PC14_DRV PC14 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3

27:26	R/W	0x1	PC13_DRV PC13 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
25:24	R/W	0x1	PC12_DRV PC12 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
23:22	R/W	0x1	PC11_DRV PC11 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
21:20	R/W	0x1	PC10_DRV PC10 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
19:18	R/W	0x1	PC9_DRV PC9 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
17:16	R/W	0x1	PC8_DRV PC8 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
15:14	R/W	0x1	PC7_DRV PC7 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
13:12	R/W	0x1	PC6_DRV PC6 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
11:10	R/W	0x1	PC5_DRV PC5 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
9:8	R/W	0x1	PC4_DRV PC4 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
7:6	R/W	0x1	PC3_DRV PC3 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
5:4	R/W	0x1	PC2_DRV

			PC2 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
3:2	R/W	0x1	PC1_DRV PC1 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
1:0	R/W	0x1	PC0_DRV PC0 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3

9.6.5.6. 0x0060 PC Multi-Driving Register 1 (Default Value: 0x0000_0001)

Offset: 0x0060			Register Name: PC_DRV1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1:0	R/W	0x1	PC16_DRV PC16 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3

9.6.5.7. 0x0064 PC Pull Register 0 (Default Value: 0x0000_5540)

Offset: 0x0064			Register Name: PC_PULL0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	R/W	0x0	PC15_PULL PC15 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
29:28	R/W	0x0	PC14_PULL PC14 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
27:26	R/W	0x0	PC13_PULL PC13 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
25:24	R/W	0x0	PC12_PULL PC12 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved

23:22	R/W	0x0	PC11_PULL PC11 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
21:20	R/W	0x0	PC10_PULL PC10 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
19:18	R/W	0x0	PC9_PULL PC9 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
17:16	R/W	0x0	PC8_PULL PC8 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
15:14	R/W	0x1	PC7_PULL PC7 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
13:12	R/W	0x1	PC6_PULL PC6 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
11:10	R/W	0x1	PC5_PULL PC5 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
9:8	R/W	0x1	PC4_PULL PC4 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
7:6	R/W	0x1	PC3_PULL PC3 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
5:4	R/W	0x0	PC2_PULL PC2 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
3:2	R/W	0x0	PC1_PULL PC Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
1:0	R/W	0x0	PC0_PULL

			PC0 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved

9.6.5.8. 0x0068 PC Pull Register 1 (Default Value: 0x0000_0000)

Offset: 0x0068			Register Name: PC_PULL1	
Bit	**Read/Write**	**Default/Hex**	**Description**	
31:2	/	/	/	
1:0	R/W	0x0	PC16_PULL PC16 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved

9.6.5.9. 0x00B4 PF Configure Register 0 (Default Value: 0x0777_7777)

Offset: 0x00B4			Register Name: PF_CFG0	
Bit	**Read/Write**	**Default/Hex**	**Description**	
31:27	/	/	/	
26:24	R/W	0x7	PF6_SELECT 000:Input 010:Reserved 100:Reserved 110:PF_EINT6	 001:Output 011:Reserved 101:Reserved 111:IO Disable
23	/	/	/	
22:20	R/W	0x7	PF5_SELECT 000:Input 010:SDC0_D2 100:Reserved 110:PF_EINT5	 001:Output 011:JTAG_CK 101:Reserved 111:IO Disable
19	/	/	/	
18:16	R/W	0x7	PF4_SELECT 000:Input 010:SDC0_D3 100:Reserved 110:PF_EINT4	 001:Output 011:UART0_RX 101:Reserved 111:IO Disable
15	/	/	/	
14:12	R/W	0x7	PF3_SELECT 000:Input 010:SDC0_CMD 100:Reserved 110:PF_EINT3	 001:Output 011:JTAG_DO 101:Reserved 111:IO Disable

11	/	/	/
10:8	R/W	0x7	PF2_SELECT 000:Input 001:Output 010:SDC0_CLK 011:UART0_TX 100:Reserved 101:Reserved 110:PF_EINT2 111:IO Disable
7	/	/	/
6:4	R/W	0x7	PF1_SELECT 000:Input 001:Output 010:SDC0_D0 011:JTAG_DI 100:Reserved 101:Reserved 110:PF_EINT1 111:IO Disable
3	/	/	/
2:0	R/W	0x7	PF0_SELECT 000:Input 001:Output 010:SDC0_D1 011:JTAG_MS 100:Reserved 101:Reserved 110:PF_EINT0 111:IO Disable

9.6.5.10. 0x00C4 PF Data Register (Default Value: 0x0000_0000)

Offset: 0x00C4			Register Name: PF_DAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6:0	R/W	0	PF_DAT If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read.

9.6.5.11. 0x00C8 PF Multi-Driving Register 0 (Default Value: 0x0000_1555)

Offset: 0x00C8			Register Name: PF_DRV0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:14	/	/	/
13:12	R/W	0x1	PF6_DRV PF6 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
11:10	R/W	0x1	PF5_DRV PF5 Multi-Driving Select

			00: Level 0 10: Level 2	01: Level 1 11: Level 3
9:8	R/W	0x1	PF4_DRV PF4 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
7:6	R/W	0x1	PF3_DRV PF3 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
5:4	R/W	0x1	PF2_DRV PF2 Multi-Driving Select 00: Level 0--180Ω 10: Level 2--100Ω	 01: Level 1--120Ω 11: Level 3--50Ω
3:2	R/W	0x1	PF1_DRV PF1 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
1:0	R/W	0x1	PF0_DRV PF0 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3

9.6.5.12. 0x00D0 PF Pull Register 0 (Default Value: 0x0000_1040)

Offset: 0x00D0			Register Name: PF_PULL0	
Bit	**Read/Write**	**Default/Hex**	**Description**	
31:14	/	/	/	
13:12	R/W	0x1	PF6_PULL PF6 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
11:10	R/W	0x0	PF5_PULL PF5 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
9:8	R/W	0x0	PF4_PULL PF4 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
7:6	R/W	0x1	PF3_PULL PF3 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved

5:4	R/W	0x0	PF2_PULL PF2 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
3:2	R/W	0x0	PF1_PULL PF1 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
1:0	R/W	0x0	PF0_PULL PF0 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved

9.6.5.13. 0x00D8 PG Configure Register 0 (Default Value: 0x7777_7777)

Offset: 0x00D8			Register Name: PG_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30:28	R/W	0x7	PG7_SELECT 000:Input 001:Output 010:UART1_RX 011:Reserved 100:JTAG_CK 101:Reserved 110:PG_EINT7 111:IO Disable
27	/	/	/
26:24	R/W	0x7	PG6_SELECT 000:Input 001:Output 010:UART1_TX 011:Reserved 100:JTAG_MS 101:Reserved 110:PG_EINT6 111:IO Disable
23	/	/	/
22:20	R/W	0x7	PG5_SELECT 000:Input 001:Output 010:SDC1_D3 011:Reserved 100:Reserved 101:Reserved 110:PG_EINT5 111:IO Disable
19	/	/	/
18:16	R/W	0x7	PG4_SELECT 000:Input 001:Output 010:SDC1_D2 011:Reserved 100:Reserved 101:Reserved 110:PG_EINT4 111:IO Disable
15	/	/	/
14:12	R/W	0x7	PG3_SELECT

			000:Input 010:SDC1_D1 100:Reserved 110:PG_EINT3	001:Output 011:Reserved 101:Reserved 111:IO Disable
11	/	/	/	
10:8	R/W	0x7	PG2_SELECT 000:Input 010:SDC1_D0 100:Reserved 110:PG_EINT2	 001:Output 011:Reserved 101:Reserved 111:IO Disable
7	/	/	/	
6:4	R/W	0x7	PG1_SELECT 000:Input 010:SDC1_CMD 100:Reserved 110:PG_EINT1	 001:Output 011:Reserved 101:Reserved 111:IO Disable
3	/	/	/	
2:0	R/W	0x7	PG0_SELECT 000:Input 010:SDC1_CLK 100:Reserved 110:PG_EINT0	 001:Output 011:Reserved 101:Reserved 111:IO Disable

9.6.5.14. 0x00DC PG Configure Register 1 (Default Value: 0x7777_7777)

Offset: 0x00DC			Register Name: PG_CFG1	
Bit	**Read/Write**	**Default/Hex**	**Description**	
31	/	/	/	
30:28	R/W	0x7	PG15_SELECT 000:Input 010:UART2_TX 100:Reserved 110:PG_EINT15	 001:Output 011:Reserved 101:TWI4_SCK 111:IO Disable
27	/	/	/	
26:24	R/W	0x7	PG14_SELECT 000:Input 010:H_I2S2_DIN0 100:BIST_RESULT3 110:PG_EINT14	 001:Output 011:H_I2S2_DOUT1 101:Reserved 111:IO Disable
23	/	/	/	
22:20	R/W	0x7	PG13_SELECT 000:Input 010:H_I2S2_DOUT0	 001:Output 011:H_I2S2_DIN1

			100:BIST_RESULT2	101:Reserved
			110:PG_EINT13	111:IO Disable
19	/	/	/	
18:16	R/W	0x7	PG12_SELECT 000:Input 010:H_I2S2_LRCK 100:BIST_RESULT1 110:PG_EINT12	 001:Output 011:Reserved 101:Reserved 111:IO Disable
15	/	/	/	
14:12	R/W	0x7	PG11_SELECT 000:Input 010:H_I2S2_BCLK 100:BIST_RESULT0 110:PG_EINT11	 001:Output 011:Reserved 101:Reserved 111:IO Disable
11	/	/	/	
10:8	R/W	0x7	PG10_SELECT 000:Input 010:H_I2S2_MCLK 100:Reserved 110:PG_EINT10	 001:Output 011:X32KFOUT 101:Reserved 111:IO Disable
7	/	/	/	
6:4	R/W	0x7	PG9_SELECT 000:Input 010:UART1_CTS 100:JTAG_DI 110:PG_EINT9	 001:Output 011:Reserved 101:Reserved 111:IO Disable
3	/	/	/	
2:0	R/W	0x7	PG8_SELECT 000:Input 010:UART1_RTS 100:JTAG_DO 110:PG_EINT8	 001:Output 011:PLL_LOCK_DBG 101:Reserved 111:IO Disable

9.6.5.15. 0x00E0 PG Configure Register 2 (Default Value: 0x0000_7777)

Offset: 0x00E0			Register Name: PG_CFG2	
Bit	**Read/Write**	**Default/Hex**	**Description**	
31:15	/	/	/	
14:12	R/W	0x7	PG19_SELECT 000:Input 010:Reserved 100:PWM1 110:PG_EINT19	 001:Output 011:Reserved 101:Reserved 111:IO Disable

11	/	/	/
10:8	R/W	0x7	PG18_SELECT 000:Input 001:Output 010:UART2_CTS 011:Reserved 100:Reserved 101:TWI3_SDA 110:PG_EINT18 111:IO Disable
7	/	/	/
6:4	R/W	0x7	PG17_SELECT 000:Input 001:Output 010:UART2_RTS 011:Reserved 100:Reserved 101:TWI3_SCK 110:PG_EINT17 111:IO Disable
3	/	/	/
2:0	R/W	0x7	PG16_SELECT 000:Input 001:Output 010:UART2_RX 011:Reserved 100:Reserved 101:TWI4_SDA 110:PG_EINT16 111:IO Disable

9.6.5.16. 0x00E8 PG Data Register (Default Value: 0x0000_0000)

Offset: 0x00E8			Register Name: PG_DAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/
19:0	R/W	0	PG_DAT If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read.

9.6.5.17. 0x00EC PG Multi-Driving Register 0 (Default Value: 0x5555_5555)

Offset: 0x00EC			Register Name: PG_DRV0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	R/W	0x1	PG15_DRV PG15 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
29:28	R/W	0x1	PG14_DRV PG14 Multi-Driving Select 00: Level 0 01: Level 1

			10: Level 2	11: Level 3
27:26	R/W	0x1	PG13_DRV PG13 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
25:24	R/W	0x1	PG12_DRV PG12 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
23:22	R/W	0x1	PG11_DRV PG11 Multi-Driving Select 00: Level 0--180Ω 10: Level 2--100Ω	 01: Level 1--120Ω 11: Level 3--50Ω
21:20	R/W	0x1	PG10_DRV PG10 Multi-Driving Select 00: Level 0--180Ω 10: Level 2--100Ω	 01: Level 1--120Ω 11: Level 3--50Ω
19:18	R/W	0x1	PG9_DRV PG9 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
17:16	R/W	0x1	PG8_DRV PG8 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
15:14	R/W	0x1	PG7_DRV PG7 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
13:12	R/W	0x1	PG6_DRV PG6 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
11:10	R/W	0x1	PG5_DRV PG5 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
9:8	R/W	0x1	PG4_DRV PG4 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
7:6	R/W	0x1	PG3_DRV PG3 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3

5:4	R/W	0x1	PG2_DRV PG2 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
3:2	R/W	0x1	PG1_DRV PG1 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
1:0	R/W	0x1	PG0_DRV PG0 Multi-Driving Select 00: Level 0--180Ω 01: Level 1--120Ω 10: Level 2--100Ω 11: Level 3--50Ω

9.6.5.18. 0x00F0 PG Multi-Driving Register 1 (Default Value: 0x0000_0055)

Offset: 0x00F0			Register Name: PG_DRV1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:6	R/W	0x1	PG19_DRV PG19 Multi-Driving Select 00: Level 0--180Ω 01: Level 1--120Ω 10: Level 2--100Ω 11: Level 3--50Ω
5:4	R/W	0x1	PG18_DRV PG18 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
3:2	R/W	0x1	PG17_DRV PG17 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
1:0	R/W	0x1	PG16_DRV PG16 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3

9.6.5.19. 0x00F4 PG Pull Register 0 (Default Value: 0x0000_0554)

Offset: 0x00F4			Register Name: PG_PULL0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	R/W	0x0	PG15_PULL PG15 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up

			10: Pull-down	11: Reserved
29:28	R/W	0x0	PG14_PULL PG14 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
27:26	R/W	0x0	PG13_PULL PG13 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
25:24	R/W	0x0	PG12_PULL PG12 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
23:22	R/W	0x0	PG11_PULL PG11 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
21:20	R/W	0x0	PG10_PULL PG10 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
19:18	R/W	0x0	PG9_PULL PG9 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
17:16	R/W	0x0	PG8_PULL PG8 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
15:14	R/W	0x0	PG7_PULL PG7 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
13:12	R/W	0x0	PG6_PULL PG6 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
11:10	R/W	0x1	PG5_PULL PG5 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
9:8	R/W	0x1	PG4_PULL PG4 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved

7:6	R/W	0x1	PG3_PULL PG3 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
5:4	R/W	0x1	PG2_PULL PG2 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
3:2	R/W	0x1	PG1_PULL PG1 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
1:0	R/W	0x0	PG0_PULL PG0 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved

9.6.5.20. 0x00F8 PG Pull Register 1 (Default Value: 0x0000_0000)

Offset: 0x00F8			Register Name: PG_PULL1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:6	R/W	0x0	PG19_PULL PG19 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
5:4	R/W	0x0	PG18_PULL PG18 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
3:2	R/W	0x0	PG17_PULL PG17 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
1:0	R/W	0x0	PG16_PULL PG16 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved

9.6.5.21. 0x00FC PH Configure Register 0 (Default Value: 0x7777_7777)

Offset: 0x00FC	Register Name: PH_CFG0

Bit	Read/Write	Default/Hex	Description
31	/	/	/
30:28	R/W	0x7	PH7_SELECT 000:Input 001:Output 010:UART2_RTS 011:H_I2S3_LRCK 100:SPI1_MOSI 100:TWI4_SDA 110:PH_EINT7 111:IO Disable
27	/	/	/
26:24	R/W	0x7	PH6_SELECT 000:Input 001:Output 010:UART2_RX 011:H_I2S3_BCLK 100:SPI1_CLK 100:TWI4_SCK 110:PH_EINT6 111:IO Disable
23	/	/	/
22:20	R/W	0x7	PH5_SELECT 000:Input 001:Output 010:UART2_TX 011:H_I2S3_MCLK 100:SPI1_CS0 100:TWI3_SDA 110:PH_EINT5 111:IO Disable
19	/	/	/
18:16	R/W	0x7	PH4_SELECT 000:Input 001:Output 010:Reserved 011:OWA_OUT 100:Reserved 100:TWI3_SCK 110:PH_EINT4 111:IO Disable
15	/	/	/
14:12	R/W	0x7	PH3_SELECT 000:Input 001:Output 010:UART5_RX 011:Reserved 100:PWM1 100:TWI2_SDA 110:PH_EINT3 111:IO Disable
11	/	/	/
10:8	R/W	0x7	PH2_SELECT 000:Input 001:Output 010:UART5_TX 011:OWA_MCLK 100:PWM2 100:TWI2_SCK 110:PH_EINT2 111:IO Disable
7	/	/	/
6:4	R/W	0x7	PH1_SELECT 000:Input 001:Output 010:UART0_RX 011:Reserved 100:PWM4 100:TWI1_SDA 110:PH_EINT1 111:IO Disable
3	/	/	/

2:0	R/W	0x7	PH0_SELECT 000:Input 010:UART0_TX 100:PWM3 110:PH_EINT0	 001:Output 011:Reserved 100:TWI1_SCK 111:IO Disable

9.6.5.22. 0x0100 PH Configure Register 1 (Default Value: 0x0000_0777)

Offset: 0x0100			Register Name: PH_CFG1	
Bit	**Read/Write**	**Default/Hex**	**Description**	
31:11	/	/	/	
10:8	R/W	0x7	PH10_SELECT 000:Input 010:Reserved 100:TCON_TRIG1 110:PH_EINT10	 001:Output 011:IR_RX 100:Reserved 111:IO Disable
7	/	/	/	
6:4	R/W	0x7	PH9_SELECT 000:Input 010:Reserved 100:SPI1_CS1 110:PH_EINT9	 001:Output 011:H_I2S3_DIN0 100:H_I2S3_DOUT1 111:IO Disable
3	/	/	/	
2:0	R/W	0x7	PH8_SELECT 000:Input 010:UART2_CTS 100:SPI1_MISO 110:PH_EINT8	 001:Output 011:H_I2S3_DOUT0 100:H_I2S3_DIN1 111:IO Disable

9.6.5.23. 0x010C PH Data Register (Default Value: 0x0000_0000)

Offset: 0x010C			Register Name: PH_DAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
10:0	R/W	0x0	PH_DAT If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read.

9.6.5.24. 0x0110 PH Multi-Driving Register 0 (Default Value: 0x0015_5555)

Offset: 0x110			Register Name: PH_DRV0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21:20	R/W	0x1	PH10_DRV PH10 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
19:18	R/W	0x1	PH9_DRV PH9 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
17:16	R/W	0x1	PH8_DRV PH8 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
15:14	R/W	0x1	PH7_DRV PH7 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
13:12	R/W	0x1	PH6_DRV PH6 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
11:10	R/W	0x1	PH5_DRV PH5 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
9:8	R/W	0x1	PH4_DRV PH4 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
7:6	R/W	0x1	PH3_DRV PH3 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
5:4	R/W	0x1	PH2_DRV PH2 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
3:2	R/W	0x1	PH1_DRV PH1 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3

1:0	R/W	0x1	PH0_DRV PH0 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3

9.6.5.25. 0x0118 PH Pull Register 0 (Default Value: 0x0000_0050)

Offset: 0x0118			Register Name: PH_PULL0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21:20	R/W	0x0	PH10_PULL PH10 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
19:18	R/W	0x0	PH9_PULL PH9 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
17:16	R/W	0x0	PH8_PULL PH8 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
15:14	R/W	0x0	PH7_PULL PH7 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
13:12	R/W	0x0	PH6_PULL PH6 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
11:10	R/W	0x0	PH5_PULL PH5 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
9:8	R/W	0x0	PH4_PULL PH4 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
7:6	R/W	0x1	PH3_PULL PH3 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
5:4	R/W	0x1	PH2_PULL PH2 Pull-up/down Select

			00: Pull-up/down disable 10: Pull-down	01: Pull-up 11: Reserved
3:2	R/W	0x0	PH1_PULL PH1 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
1:0	R/W	0x0	PH0_PULL PH0 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved

9.6.5.26. 0x0120 PI Configure Register 0 (Default Value: 0x7777_7777)

Offset: 0x0120			Register Name: PI_CFG0	
Bit	**Read/Write**	**Default/Hex**	**Description**	
31	/	/	/	
30:28	R/W	0x7	PI7_SELECT 000:Input 010:RGMII_TXD3/RMII_NULL 100:TS0_SYNC 110:PI_EINT7	 001:Output 011:UART2_RTS 101:TWI1_SCK 111:IO Disable
27	/	/	/	
26:24	R/W	0x7	PI6_SELECT 000:Input 010:RGMII_NULL/RMII_RXER 100:TS0_ERR 110:PI_EINT6	 001:Output 011:UART2_RX 101:TWI0_SDA 111:IO Disable
23	/	/	/	
22:20	R/W	0x7	PI5_SELECT 000:Input 010:RGMII_RXCTL/RMII_CRS_DV 100:TS0_CLK 110:PI_EINT5	 001:Output 011:UART2_TX 101:TWI0_SCK 111:IO Disable
19	/	/	/	
18:16	R/W	0x7	PI4_SELECT 000:Input 010:RGMII_RXCK/RMII_NULL 100:H_I2S0_DIN0 110:PI_EINT4	 001:Output 011:DMIC_DATA3 101:H_I2S0_DOUT1 111:IO Disable
15	/	/	/	
14:12	R/W	0x7	PI3_SELECT 000:Input 010:RGMII_RXD0/RMII_RXD0	 001:Output 011:DMIC_DATA2

			100:H_I2S0_DOUT0 110:PI_EINT3	101:H_I2S0_DIN1 111:IO Disable
11	/	/	/	
10:8	R/W	0x7	PI2_SELECT 000:Input 010:RGMII_RXD1/RMII_RXD1 100:H_I2S0_LRCK 110:PI_EINT2	 001:Output 011:DMIC_DATA1 101:HDMI_CEC 111:IO Disable
7	/	/	/	
6:4	R/W	0x7	PI1_SELECT 000:Input 010:RGMII_RXD2/RMII_NULL 100:H_I2S0_BCLK 110:PI_EINT1	 001:Output 011:DMIC_DATA0 101:HDMI_SDA 111:IO Disable
3	/	/	/	
2:0	R/W	0x7	PI0_SELECT 000:Input 010:RGMII_RXD3/RMII_NULL 100:H_I2S0_MCLK 110:PI_EINT0	 001:Output 011:DMIC_CLK 101:HDMI_SCL 111:IO Disable

9.6.5.27. 0x0124 PI Configure Register 1 (Default Value: 0x7777_7777)

Offset: 0x0124			Register Name: PI_CFG1	
Bit	**Read/Write**	**Default/Hex**	**Description**	
31	/	/	/	
30:28	R/W	0x7	PI15_SELECT 000:Input 010:MDIO 100:TS0_D6 110:PI_EINT15	 001:Output 011:UART4_RTS 101:CLK_FANOUT0 111:IO Disable
27	/	/	/	
26:24	R/W	0x7	PI14_SELECT 000:Input 010:MDC 100:TS0_D5 110:PI_EINT14	 001:Output 011:UART4_RX 101:PWM4 111:IO Disable
23	/	/	/	
22:20	R/W	0x7	PI13_SELECT 000:Input 010:RGMII_CLKIN/RMII_NULL 100:TS0_D4 110:PI_EINT13	 001:Output 011:UART4_TX 101:PWM3 111:IO Disable

19	/	/	/	
18:16	R/W	0x7	PI12_SELECT 000:Input 010:RGMII_TXCTL/RMII_TXEN 100:TS0_D3 110:PI_EINT12	 001:Output 011:UART3_CTS 101:PWM2 111:IO Disable
15	/	/	/	
14:12	R/W	0x7	PI11_SELECT 000:Input 010:RGMII_TXCK/RMII_TXCK 100:TS0_D2 110:PI_EINT11	 001:Output 011:UART3_RTS 101:PWM1 111:IO Disable
11	/	/	/	
10:8	R/W	0x7	PI10_SELECT 000:Input 010:RGMII_TXD0/RMII_TXD0 100:TS0_D1 110:PI_EINT10	 001:Output 011:UART3_RX 101:TWI2_SDA 111:IO Disable
7	/	/	/	
6:4	R/W	0x7	PI9_SELECT 000:Input 010:RGMII_TXD1/RMII_TXD1 100:TS0_D0 110:PI_EINT9	 001:Output 011:UART3_TX 101:TWI2_SCK 111:IO Disable
3	/	/	/	
2:0	R/W	0x7	PI8_SELECT 000:Input 010:RGMII_TXD2/RMII_NULL 100:TS0_DVLD 110:PI_EINT8	 001:Output 011:UART2_RTS 101:TWI1_SDA 111:IO Disable

9.6.5.28. 0x0128 PI Configure Register 2 (Default Value: 0x0000_0007)

Offset: 0x0128			Register Name: PI_CFG2	
Bit	**Read/Write**	**Default/Hex**	**Description**	
31:3	/	/	/	
2:0	R/W	0x7	PI16_SELECT 000:Input 010:EPHY_25M 100:TS0_D7 110:PI_EINT16	 001:Output 011:UART4_CTS 101:CLK_FANOUT1 111:IO Disable

9.6.5.29. 0x0130 PI Data Register (Default Value: 0x0000_0000)

Offset: 0x0130			Register Name: PI_DAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
16:0	R/W	0x0	PI_DAT If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read.

9.6.5.30. 0x0134 PI Multi-Driving Register 0 (Default Value: 0x5555_5555)

Offset: 0x0134			Register Name: PI_DRV0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	R/W	0x1	PI15_DRV PI15 Multi-Driving Select 00: Level 0--180Ω 01: Level 1--120Ω 10: Level 2--100Ω 11: Level 3--50Ω
29:28	R/W	0x1	PI14_DRV PI14 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
27:26	R/W	0x1	PI13_DRV PI13 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
25:24	R/W	0x1	PI12_DRV PI12 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
23:22	R/W	0x1	PI11_DRV PI11 Multi-Driving Select 00: Level 0--180Ω 01: Level 1--120Ω 10: Level 2--100Ω 11: Level 3--50Ω
21:20	R/W	0x1	PI10_DRV PI10 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
19:18	R/W	0x1	PI9_DRV PI9 Multi-Driving Select 00: Level 0 01: Level 1

			10: Level 2	11: Level 3
17:16	R/W	0x1	PI8_DRV PI8 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
15:14	R/W	0x1	PI7_DRV PI7 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
13:12	R/W	0x1	PI6_DRV PI6 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
11:10	R/W	0x1	PI5_DRV PI5 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
9:8	R/W	0x1	PI4_DRV PI4 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
7:6	R/W	0x1	PI3_DRV PI3 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
5:4	R/W	0x1	PI2_DRV PI2 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
3:2	R/W	0x1	PI1_DRV PI1 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3
1:0	R/W	0x1	PI0_DRV PI0 Multi-Driving Select 00: Level 0 10: Level 2	 01: Level 1 11: Level 3

9.6.5.31. 0x0138 PI Multi-Driving Register 1 (Default Value: 0x0000_0001)

Offset: 0x0138			Register Name: PI_DRV1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/

1:0	R/W	0x1	PI16_DRV PI16 Multi-Driving Select 00: Level 0--180Ω 01: Level 1--120Ω 10: Level 2--100Ω 11: Level 3--50Ω

9.6.5.32. 0x013C PI Pull Register 0 (Default Value: 0x0000_0000)

Offset: 0x013C			Register Name: PI_PULL0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	R/W	0x0	PI15_PULL PI15 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
29:28	R/W	0x0	PI14_PULL PI14 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
27:26	R/W	0x0	PI13_PULL PI13 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
25:24	R/W	0x0	PI12_PULL PI12 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
23:22	R/W	0x0	PI11_PULL PI11 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
21:20	R/W	0x0	PI10_PULL PI10 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
19:18	R/W	0x0	PI9_PULL PI9 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
17:16	R/W	0x0	PI8_PULL PI8 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
15:14	R/W	0x0	PI7_PULL PI7 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up

			10: Pull-down	11: Reserved
13:12	R/W	0x0	PI6_PULL PI6 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
11:10	R/W	0x0	PI5_PULL PI5 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
9:8	R/W	0x0	PI4_PULL PI4 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
7:6	R/W	0x0	PI3_PULL PI3 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
5:4	R/W	0x0	PI2_PULL PI2 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
3:2	R/W	0x0	PI1_PULL PI1 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved
1:0	R/W	0x0	PI0_PULL PI0 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved

9.6.5.33. 0x0140 PI Pull Register 1 (Default Value: 0x0000_0000)

Offset: 0x0140			**Register Name: PI_PULL1**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1:0	R/W	0x0	PI16_PULL PI16 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	 01: Pull-up 11: Reserved

9.6.5.34. 0x0240 PC External Interrupt Configure Register 0 (Default Value: 0x0000_0000)

Offset: 0x0240			Register Name: PC_EINT_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	EINT7_CFG External INT7 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
27:24	R/W	0x0	EINT6_CFG External INT6 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
23:20	R/W	0x0	EINT5_CFG External INT5 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
19:16	R/W	0x0	EINT4_CFG External INT4 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
15:12	R/W	0x0	EINT3_CFG External INT3 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
11:8	R/W	0x0	EINT2_CFG

			External INT2 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
7:4	R/W	0x0	EINT1_CFG External INT1 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
3:0	R/W	0x0	EINT0_CFG External INT0 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

9.6.5.35. 0x0244 PC External Interrupt Configure Register 1 (Default Value: 0x0000_0000)

Offset: 0x0244			Register Name: PC_EINT_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	EINT15_CFG External INT15 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
27:24	R/W	0x0	EINT14_CFG External INT14 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

23:20	R/W	0x0	EINT13_CFG External INT13 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
19:16	R/W	0x0	EINT12_CFG External INT12 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
15:12	R/W	0x0	EINT11_CFG External INT11 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
11:8	R/W	0x0	EINT10_CFG External INT10 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
7:4	R/W	0x0	EINT9_CFG External INT9 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
3:0	R/W	0x0	EINT8_CFG External INT8 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level

			0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

9.6.5.36. 0x0248 PC External Interrupt Configure Register 2 (Default Value: 0x0000_0000)

Offset: 0x0248			Register Name: PC_EINT_CFG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3:0	R/W	0x0	EINT16_CFG External INT16 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

9.6.5.37. 0x0250 PC External Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0250			Register Name: PC_EINT_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	EINT16_CTL External INT16 Enable 0: Disable 1: Enable
15	R/W	0x0	EINT15_CTL External INT15 Enable 0: Disable 1: Enable
14	R/W	0x0	EINT14_CTL External INT14 Enable 0: Disable 1: Enable
13	R/W	0x0	EINT13_CTL External INT13 Enable 0: Disable 1: Enable
12	R/W	0x0	EINT12_CTL External INT12 Enable 0: Disable

			1: Enable
11	R/W	0x0	EINT11_CTL External INT11 Enable 0: Disable 1: Enable
10	R/W	0x0	EINT10_CTL External INT10 Enable 0: Disable 1: Enable
9	R/W	0x0	EINT9_CTL External INT9 Enable 0: Disable 1: Enable
8	R/W	0x0	EINT8_CTL External INT8 Enable 0: Disable 1: Enable
7	R/W	0x0	EINT7_CTL External INT7 Enable 0: Disable 1: Enable
6	R/W	0x0	EINT6_CTL External INT6 Enable 0: Disable 1: Enable
5	R/W	0x0	EINT5_CTL External INT5 Enable 0: Disable 1: Enable
4	R/W	0x0	EINT4_CTL External INT4 Enable 0: Disable 1: Enable
3	R/W	0x0	EINT3_CTL External INT3 Enable 0: Disable 1: Enable
2	R/W	0x0	EINT2_CTL External INT2 Enable 0: Disable 1: Enable
1	R/W	0x0	EINT1_CTL External INT1 Enable 0: Disable 1: Enable

0	R/W	0x0	EINT0_CTL External INT0 Enable 0: Disable 1: Enable

9.6.5.38. 0x0254 PC External Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0254			Register Name: PC_EINT_STATUS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	EINT16_STATUS External INT16 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
15	R/W	0x0	EINT15_STATUS External INT15 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
14	R/W	0x0	EINT14_STATUS External INT14 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
13	R/W	0x0	EINT13_STATUS External INT13 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
12	R/W	0x0	EINT12_STATUS External INT12 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
11	R/W	0x0	EINT11_STATUS External INT11 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
10	R/W	0x0	EINT10_STATUS External INT10 Pending Bit 0: No IRQ pending 1: IRQ pending

			Write ‘1’ to clear
9	R/W	0x0	EINT9_STATUS External INT9 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
8	R/W	0x0	EINT8_STATUS External INT8 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
7	R/W	0x0	EINT7_STATUS External INT7 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
6	R/W	0x0	EINT6_STATUS External INT6 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
5	R/W	0x0	EINT5_STATUS External INT5 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
4	R/W	0x0	EINT4_STATUS External INT4 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
3	R/W	0x0	EINT3_STATUS External INT3 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
2	R/W	0x0	EINT2_STATUS External INT2 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
1	R/W	0x0	EINT1_STATUS External INT1 Pending Bit 0: No IRQ pending 1: IRQ pending

			Write ‘1’ to clear
0	R/W	0x0	EINT0_STATUS External INT0 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear

9.6.5.39. 0x0258 PC External Interrupt Debounce Register (Default Value: 0x0000_0000)

Offset: 0x0258			Register Name: PC_EINT_DEB
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6:4	R/W	0x0	DEB_CLK_PRE_SCALE Debounce Clock Pre-scale n The selected clock source is prescaled by 2^n.
3:1	/	/	/
0	R/W	0x0	PIO_INT_CLK_SELECT PIO Interrupt Clock Select 0: LOSC 32KHz 1: HOSC 24MHz

9.6.5.40. 0x02A0 PF External Interrupt Configure Register 0 (Default Value: 0x0000_0000)

Offset: 0x02A0			Register Name: PF_EINT_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0x0	EINT6_CFG External INT6 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
23:20	R/W	0x0	EINT5_CFG External INT5 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

19:16	R/W	0x0	EINT4_CFG External INT4 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
15:12	R/W	0x0	EINT3_CFG External INT3 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
11:8	R/W	0x0	EINT2_CFG External INT2 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
7:4	R/W	0x0	EINT1_CFG External INT1 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
3:0	R/W	0x0	EINT0_CFG External INT0 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

9.6.5.41. 0x02B0 PF External Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x02B0	Register Name: PF_EINT_CTL

Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6	R/W	0x0	EINT6_CTL External INT6 Enable 0: Disable 1: Enable
5	R/W	0x0	EINT5_CTL External INT5 Enable 0: Disable 1: Enable
4	R/W	0x0	EINT4_CTL External INT4 Enable 0: Disable 1: Enable
3	R/W	0x0	EINT3_CTL External INT3 Enable 0: Disable 1: Enable
2	R/W	0x0	EINT2_CTL External INT2 Enable 0: Disable 1: Enable
1	R/W	0x0	EINT1_CTL External INT1 Enable 0: Disable 1: Enable
0	R/W	0x0	EINT0_CTL External INT0 Enable 0: Disable 1: Enable

9.6.5.42. 0x02B4 PF External Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x02B4			Register Name: PF_EINT_STATUS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6	R/W	0x0	EINT6_STATUS External INT6 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
5	R/W	0x0	EINT5_STATUS External INT5 Pending Bit 0: No IRQ pending

			1: IRQ pending Write '1' to clear
4	R/W	0x0	EINT4_STATUS External INT4 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
3	R/W	0x0	EINT3_STATUS External INT3 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
2	R/W	0x0	EINT2_STATUS External INT2 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
1	R/W	0x0	EINT1_STATUS External INT1 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
0	R/W	0x0	EINT0_STATUS External INT0 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

9.6.5.43. 0x02B8 PF External Interrupt Debounce Register (Default Value: 0x0000_0000)

Offset: 0x02B8			Register Name: PF_EINT_DEB
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6:4	R/W	0x0	DEB_CLK_PRE_SCALE Debounce Clock Pre-scale n The selected clock source is prescaled by 2^n.
3:1	/	/	/
0	R/W	0x0	PIO_INT_CLK_SELECT PIO Interrupt Clock Select 0: LOSC 32KHz 1: HOSC 24MHz

9.6.5.44. 0x02C0 PG External Interrupt Configure Register 0 (Default Value: 0x0000_0000)

Offset: 0x02C0			Register Name:PG_EINT_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	EINT7_CFG External INT7 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
27:24	R/W	0x0	EINT6_CFG External INT6 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
23:20	R/W	0x0	EINT5_CFG External INT5 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
19:16	R/W	0x0	EINT4_CFG External INT4 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
15:12	R/W	0x0	EINT3_CFG External INT3 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
11:8	R/W	0x0	EINT2_CFG

			External INT2 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
7:4	R/W	0x0	EINT1_CFG External INT1 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
3:0	R/W	0x0	EINT0_CFG External INT0 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

9.6.5.45. 0x02C4 PG External Interrupt Configure Register 1 (Default Value: 0x0000_0000)

Offset: 0x02C4			Register Name: PG_EINT_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	EINT15_CFG External INT15 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
27:24	R/W	0x0	EINT14_CFG External INT14 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

23:20	R/W	0x0	EINT13_CFG External INT13 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
19:16	R/W	0x0	EINT12_CFG External INT12 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
15:12	R/W	0x0	EINT11_CFG External INT11 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
11:8	R/W	0x0	EINT10_CFG External INT10 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
7:4	R/W	0x0	EINT9_CFG External INT9 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
3:0	R/W	0x0	EINT8_CFG External INT8 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level

			0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

9.6.5.46. 0x02C8 PG External Interrupt Configure Register 2 (Default Value: 0x0000_0000)

Offset: 0x02C8			Register Name: PG_EINT_CFG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:12	R/W	0x0	EINT19_CFG External INT19 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
11:8	R/W	0x0	EINT18_CFG External INT18 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
7:4	R/W	0x0	EINT17_CFG External INT17 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
3:0	R/W	0x0	EINT16_CFG External INT16 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

9.6.5.47. 0x02D0 PG External Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x02D0			Register Name: PG_EINT_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/
19	R/W	0x0	EINT19_CTL External INT19 Enable 0: Disable 1: Enable
18	R/W	0x0	EINT18_CTL External INT18 Enable 0: Disable 1: Enable
17	R/W	0x0	EINT17_CTL External INT17 Enable 0: Disable 1: Enable
16	R/W	0x0	EINT16_CTL External INT16 Enable 0: Disable 1: Enable
15	R/W	0x0	EINT15_CTL External INT15 Enable 0: Disable 1: Enable
14	R/W	0x0	EINT14_CTL External INT14 Enable 0: Disable 1: Enable
13	R/W	0x0	EINT13_CTL External INT13 Enable 0: Disable 1: Enable
12	R/W	0x0	EINT12_CTL External INT12 Enable 0: Disable 1: Enable
11	R/W	0x0	EINT11_CTL External INT11 Enable 0: Disable 1: Enable
10	R/W	0x0	EINT10_CTL External INT10 Enable 0: Disable 1: Enable

9	R/W	0x0	EINT9_CTL External INT9 Enable 0: Disable 1: Enable
8	R/W	0x0	EINT8_CTL External INT8 Enable 0: Disable 1: Enable
7	R/W	0x0	EINT7_CTL External INT7 Enable 0: Disable 1: Enable
6	R/W	0x0	EINT6_CTL External INT6 Enable 0: Disable 1: Enable
5	R/W	0x0	EINT5_CTL External INT5 Enable 0: Disable 1: Enable
4	R/W	0x0	EINT4_CTL External INT4 Enable 0: Disable 1: Enable
3	R/W	0x0	EINT3_CTL External INT3 Enable 0: Disable 1: Enable
2	R/W	0x0	EINT2_CTL External INT2 Enable 0: Disable 1: Enable
1	R/W	0x0	EINT1_CTL External INT1 Enable 0: Disable 1: Enable
0	R/W	0x0	EINT0_CTL External INT0 Enable 0: Disable 1: Enable

9.6.5.48. 0x02D4 PG External Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x02D4	Register Name: PG_EINT_STATUS

Bit	Read/Write	Default/Hex	Description
31:20	/	/	/
19	R/W	0x0	EINT19_STATUS External INT19 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
18	R/W	0x0	EINT18_STATUS External INT18 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
17	R/W	0x0	EINT17_STATUS External INT17 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
16	R/W	0x0	EINT16_STATUS External INT16 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
15	R/W	0x0	EINT15_STATUS External INT15 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
14	R/W	0x0	EINT14_STATUS External INT14 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
13	R/W	0x0	EINT13_STATUS External INT13 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
12	R/W	0x0	EINT12_STATUS External INT12 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
11	R/W	0x0	EINT11_STATUS External INT11 Pending Bit 0: No IRQ pending

			1: IRQ pending Write ‘1’ to clear
10	R/W	0x0	EINT10_STATUS External INT10 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
9	R/W	0x0	EINT9_STATUS External INT9 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
8	R/W	0x0	EINT8_STATUS External INT8 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
7	R/W	0x0	EINT7_STATUS External INT7 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
6	R/W	0x0	EINT6_STATUS External INT6 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
5	R/W	0x0	EINT5_STATUS External INT5 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
4	R/W	0x0	EINT4_STATUS External INT4 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
3	R/W	0x0	EINT3_STATUS External INT3 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
2	R/W	0x0	EINT2_STATUS External INT2 Pending Bit 0: No IRQ pending

			1: IRQ pending Write '1' to clear
1	R/W	0x0	EINT1_STATUS External INT1 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
0	R/W	0x0	EINT0_STATUS External INT0 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

9.6.5.49. 0x02D8 PG External Interrupt Debounce Register (Default Value: 0x0000_0000)

Offset: 0x02D8			Register Name: PG_EINT_DEB
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6:4	R/W	0x0	DEB_CLK_PRE_SCALE Debounce Clock Pre-scale n The selected clock source is prescaled by 2^n.
3:1	/	/	/
0	R/W	0x0	PIO_INT_CLK_SELECT PIO Interrupt Clock Select 0: LOSC 32KHz 1: HOSC 24MHz

9.6.5.50. 0x02E0 PH External Interrupt Configure Register 0 (Default Value: 0x0000_0000)

Offset: 0x02E0			Register Name:PH_EINT_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	EINT7_CFG External INT7 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
27:24	R/W	0x0	EINT6_CFG External INT6 Mode 0x0: Positive Edge

			0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
23:20	R/W	0x0	EINT5_CFG External INT5 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
19:16	R/W	0x0	EINT4_CFG External INT4 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
15:12	R/W	0x0	EINT3_CFG External INT3 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
11:8	R/W	0x0	EINT2_CFG External INT2 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
7:4	R/W	0x0	EINT1_CFG External INT1 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

3:0	R/W	0x0	EINT0_CFG External INT0 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

9.6.5.51. 0x02E4 PH External Interrupt Configure Register 1 (Default Value: 0x0000_0000)

Offset: 0x02E4			Register Name: PH_EINT_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11:8	R/W	0x0	EINT10_CFG External INT10 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
7:4	R/W	0x0	EINT9_CFG External INT9 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
3:0	R/W	0x0	EINT8_CFG External INT8 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

9.6.5.52. 0x02F0 PH External Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x02F0			Register Name: PH_EINT_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**

31:11	/	/	/
10	R/W	0x0	EINT10_CTL External INT10 Enable 0: Disable 1: Enable
9	R/W	0x0	EINT9_CTL External INT9 Enable 0: Disable 1: Enable
8	R/W	0x0	EINT8_CTL External INT8 Enable 0: Disable 1: Enable
7	R/W	0x0	EINT7_CTL External INT7 Enable 0: Disable 1: Enable
6	R/W	0x0	EINT6_CTL External INT6 Enable 0: Disable 1: Enable
5	R/W	0x0	EINT5_CTL External INT5 Enable 0: Disable 1: Enable
4	R/W	0x0	EINT4_CTL External INT4 Enable 0: Disable 1: Enable
3	R/W	0x0	EINT3_CTL External INT3 Enable 0: Disable 1: Enable
2	R/W	0x0	EINT2_CTL External INT2 Enable 0: Disable 1: Enable
1	R/W	0x0	EINT1_CTL External INT1 Enable 0: Disable 1: Enable
0	R/W	0x0	EINT0_CTL External INT0 Enable 0: Disable 1: Enable

9.6.5.53. 0x02F4 PH External Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x02F4			Register Name: PH_EINT_STATUS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10	R/W	0x0	EINT10_STATUS External INT10 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
9	R/W	0x0	EINT9_STATUS External INT9 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
8	R/W	0x0	EINT8_STATUS External INT8 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
7	R/W	0x0	EINT7_STATUS External INT7 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
6	R/W	0x0	EINT6_STATUS External INT6 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
5	R/W	0x0	EINT5_STATUS External INT5 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
4	R/W	0x0	EINT4_STATUS External INT4 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
3	R/W	0x0	EINT3_STATUS External INT3 Pending Bit 0: No IRQ pending

			1: IRQ pending Write '1' to clear
2	R/W	0x0	EINT2_STATUS External INT2 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
1	R/W	0x0	EINT1_STATUS External INT1 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
0	R/W	0x0	EINT0_STATUS External INT0 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

9.6.5.54. 0x02F8 PH External Interrupt Debounce Register (Default Value: 0x0000_0000)

Offset: 0x02F8			Register Name: PH_EINT_DEB
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6:4	R/W	0x0	DEB_CLK_PRE_SCALE Debounce Clock Pre-scale n The selected clock source is prescaled by 2^n.
3:1	/	/	/
0	R/W	0x0	PIO_INT_CLK_SELECT PIO Interrupt Clock Select 0: LOSC 32KHz 1: HOSC 24MHz

9.6.5.55. 0x0300 PI External Interrupt Configure Register 0 (Default Value: 0x0000_0000)

Offset: 0x0300			Register Name: PI_EINT_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	EINT7_CFG External INT7 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level

			0x4: Double Edge (Positive/ Negative) Others: Reserved
27:24	R/W	0x0	EINT6_CFG External INT6 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
23:20	R/W	0x0	EINT5_CFG External INT5 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
19:16	R/W	0x0	EINT4_CFG External INT4 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
15:12	R/W	0x0	EINT3_CFG External INT3 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
11:8	R/W	0x0	EINT2_CFG External INT2 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
7:4	R/W	0x0	EINT1_CFG External INT1 Mode 0x0: Positive Edge

			0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
3:0	R/W	0x0	EINT0_CFG External INT0 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

9.6.5.56. 0x0304 PI External Interrupt Configure Register 1 (Default Value: 0x0000_0000)

Offset: 0x0304			Register Name: PI_EINT_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	EINT15_CFG External INT15 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
27:24	R/W	0x0	EINT14_CFG External INT14 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
23:20	R/W	0x0	EINT13_CFG External INT13 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
19:16	R/W	0x0	EINT12_CFG External INT12 Mode

			0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
15:12	R/W	0x0	EINT11_CFG External INT11 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
11:8	R/W	0x0	EINT10_CFG External INT10 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
7:4	R/W	0x0	EINT9_CFG External INT9 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
3:0	R/W	0x0	EINT8_CFG External INT8 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

9.6.5.57. 0x0308 PI External Interrupt Configure Register 2 (Default Value: 0x0000_0000)

Offset: 0x0308			Register Name: PI_EINT_CFG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/

3:0	R/W	0x0	EINT16_CFG External INT16 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved

9.6.5.58. 0x0310 PI External Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0310			Register Name: PI_EINT_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	EINT16_CTL External INT16 Enable 0: Disable 1: Enable
15	R/W	0x0	EINT15_CTL External INT15 Enable 0: Disable 1: Enable
14	R/W	0x0	EINT14_CTL External INT14 Enable 0: Disable 1: Enable
13	R/W	0x0	EINT13_CTL External INT13 Enable 0: Disable 1: Enable
12	R/W	0x0	EINT12_CTL External INT12 Enable 0: Disable 1: Enable
11	R/W	0x0	EINT11_CTL External INT11 Enable 0: Disable 1: Enable
10	R/W	0x0	EINT10_CTL External INT10 Enable 0: Disable 1: Enable
9	R/W	0x0	EINT9_CTL External INT9 Enable

			0: Disable 1: Enable
8	R/W	0x0	EINT8_CTL External INT8 Enable 0: Disable 1: Enable
7	R/W	0x0	EINT7_CTL External INT7 Enable 0: Disable 1: Enable
6	R/W	0x0	EINT6_CTL External INT6 Enable 0: Disable 1: Enable
5	R/W	0x0	EINT5_CTL External INT5 Enable 0: Disable 1: Enable
4	R/W	0x0	EINT4_CTL External INT4 Enable 0: Disable 1: Enable
3	R/W	0x0	EINT3_CTL External INT3 Enable 0: Disable 1: Enable
2	R/W	0x0	EINT2_CTL External INT2 Enable 0: Disable 1: Enable
1	R/W	0x0	EINT1_CTL External INT1 Enable 0: Disable 1: Enable
0	R/W	0x0	EINT0_CTL External INT0 Enable 0: Disable 1: Enable

9.6.5.59. 0x0314 PI External Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0314			Register Name: PI_EINT_STATUS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/

16	R/W	0x0	EINT16_STATUS External INT16 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
15	R/W	0x0	EINT15_STATUS External INT15 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
14	R/W	0x0	EINT14_STATUS External INT14 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
13	R/W	0x0	EINT13_STATUS External INT13 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
12	R/W	0x0	EINT12_STATUS External INT12 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
11	R/W	0x0	EINT11_STATUS External INT11 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
10	R/W	0x0	EINT10_STATUS External INT10 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
9	R/W	0x0	EINT9_STATUS External INT9 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
8	R/W	0x0	EINT8_STATUS External INT8 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear

7	R/W	0x0	EINT7_STATUS External INT7 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
6	R/W	0x0	EINT6_STATUS External INT6 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
5	R/W	0x0	EINT5_STATUS External INT5 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
4	R/W	0x0	EINT4_STATUS External INT4 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
3	R/W	0x0	EINT3_STATUS External INT3 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
2	R/W	0x0	EINT2_STATUS External INT2 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
1	R/W	0x0	EINT1_STATUS External INT1 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear
0	R/W	0x0	EINT0_STATUS External INT0 Pending Bit 0: No IRQ pending 1: IRQ pending Write ‘1’ to clear

9.6.5.60. 0x0318 PI External Interrupt Debounce Register (Default Value: 0x0000_0000)

Offset: 0x0318	Register Name: PI_EINT_DEB

Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:4	R/W	0x0	DEB_CLK_PRE_SCALE Debounce Clock Pre-scale n The selected clock source is prescaled by 2^n.
3:1	/	/	/
0	R/W	0x0	PIO_INT_CLK_SELECT PIO Interrupt Clock Select 0: LOSC 32KHz 1: HOSC 24MHz

9.6.5.61. 0x0340 PIO Group Withstand Voltage Mode Select Register (Default Value: 0x0000_0000)

Offset: 0x0340			Register Name: PIO_POW_MOD_SEL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
12	R/W	0x0	VCC-IO&PH_Port&PF_Port POWER MODE Select 0: 3.3 V 1: 1.8 V
11:9	/	/	/
8	R/W	0x0	PI_POWER MODE Select 0: 3.3 V 1: 1.8 V
7	/	/	/
6	R/W	0x0	PG_POWER MODE Select 0: 3.3 V 1: 1.8 V
5	/	/	/
4	/	/	/
3	/	/	/
2	R/W	0x0	PC_POWER MODE Select 0: 3.3 V 1: 1.8 V
1	/	/	/
0	/	/	/

NOTE

When the power domain of GPIO is larger than 1.8 V, the withstand voltage is set to 3.3 V mode, the corresponding value in 0x0340 register is set to 0.

When the power domain of GPIO is 1.8 V, the withstand voltage is set to 1.8 V mode, the corresponding value in 0x0340 register is set to 1.

9.6.5.62. 0x0344 PIO Group Withstand Voltage Mode Select Control Register (Default Value: 0x0000_0000)

Offset: 0x0344			Register Name: PIO_POW_MS_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
12	R/W	0x0	VCC-IO&PH_Port&PF_Port Withstand Voltage Mode Select Control 0: Enable 1: Disable
11:9	/	/	/
8	R/W	0x0	VCC-PI Withstand Voltage Mode Select Control 0: Enable 1: Disable
7	/	/	/
6	R/W	0x0	VCC-PG Withstand Voltage Mode Select Control 0: Enable 1: Disable
5	/	/	/
4	/	/	/
3	/	/	/
2	R/W	0x0	VCC-PC Withstand Voltage Mode Select Control 0: Enable 1: Disable
1	/	/	/
0	/	/	/

NOTE

For 1.8 V and 3.3 V power, the withstand function is enabled by default, the corresponding bit in 0x0344 register is set to 0.

For 2.5 V power, the withstand function is disabled, the corresponding bit in 0x0344 register is set to 1, and the withstand mode in 0x0340 register needs be set to 3.3 V.

9.6.5.63. 0x0348 PIO Group Power Value Register

Offset: 0x0348		Register Name: PIO_POW_VAL
Bit	**Read/Write**	**Description**
31:17	/	/
16	R	VCC-IO&PH_Port&PF_Port Power Value
15:9	/	/
8	R	PI_Port Power Value
7	/	/
6	R	PG_Port Power Value
5	/	/

4	/	/
3	/	/
2	R	PC_Port Power Value
1	/	/
0	/	/

NOTE

When the reading value of the 0x0348 register is 0, it indicates that IO power voltage is greater than 2.5 V.

When the reading value of the 0x0348 register is 1, it indicates that IO power voltage is less than 2.0 V.

9.6.5.64. 0x0350 PIO Group Power Voltage Select Control Register (Default Value: 0x0000_0001)

Offset: 0x0350			Register Name: PIO_PV_SEL_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x1	PF Port Power Voltage Select Control 0: 1.8 V 1: 3.3 V

9.6.6. GPIO(PL) Register Description

9.6.6.1. 0x0000 PL Configure Register 0 (Default Value: 0x0000_0077)

Offset: 0x0000			Register Name: PL_CFG0	
Bit	**Read/Write**	**Default/Hex**	**Description**	
31:7	/	/	/	
6:4	R/W	0x7	PL1_SELECT 000:Input 010:Reserved 100:Reserved 110:Reserved	 001:Output 011:S_TWI0_SDA 101:Reserved 111:IO Disable
3	/	/	/	
2:0	R/W	0x7	PL0_SELECT 000:Input 010:Reserved 100:Reserved 110:Reserved	 001:Output 011:S_TWI0_SCK 101:Reserved 111:IO Disable

9.6.6.2. 0x0010 PL Data Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: PL_DAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:6	/	/	/
5:0	R/W	0x0	PL_DAT If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read.

9.6.6.3. 0x0014 PL Multi-Driving Register 0 (Default Value: 0x0000_0005)

Offset: 0x0014			Register Name: PL_DRV0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3:2	R/W	0x1	PL1_DRV PL1 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
1:0	R/W	0x1	PL0_DRV PL0 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3

9.6.6.4. 0x001C PL Pull Register 0 (Default Value: 0x0000_0005)

Offset: 0x001C			Register Name: PL_PULL0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3:2	R/W	0x1	PL1_PULL PL1 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
1:0	R/W	0x1	PL0_PULL PL0 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved

9.7. LRADC

9.7.1. Overview

The Low Rate ADC(LRADC) is 6-bit resolution for Key application. The LRADC can work up to maximum conversion rate of 2 kHz.

- Power supply voltage:1.8 V; reference voltage:1.35 V
- Interrupt support
- Support Hold Key and General Key
- Support normal, continue and single work mode
- 6-bits resolution, and sample rate up to 2 kHz
- Voltage input range between 0 to 1.35 V

9.7.2. Block Diagram

Figure 9-35 shows a block diagram of the LRADC.

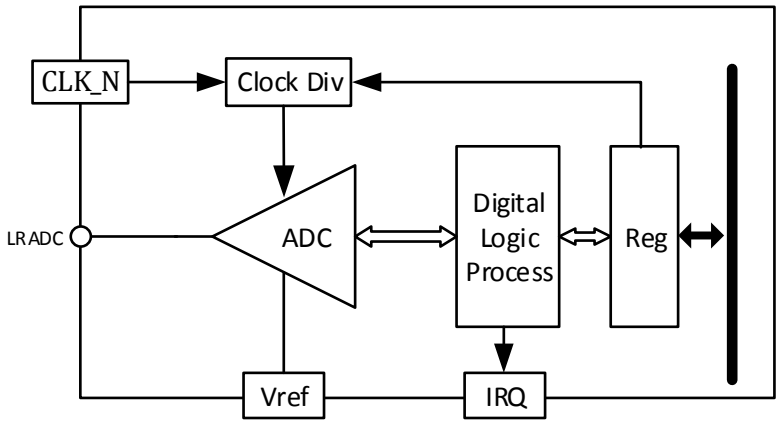

Figure 9- 35. LRADC Block Diagram

9.7.3. Operations and Functional Descriptions

9.7.3.1. External Signals

Table 9-22 describes the external signal of LRADC.

Table 9- 22. LRADC External Signals

Signal	Description	Type
LRADC	ADC Input	AI

9.7.3.2. Clock Sources

Table 9-23 describes the clock source for LRADC.

Table 9- 23. LRADC Clock Sources

Clock Sources	Description
LOSC	32.768 kHz LOSC

9.7.3.3. LRADC Work Mode

(1). Normal Mode

ADC gathers 8 samples,the average of the 8 samples is updated in data register,and the data interrupt sign is enabled.It is sampled repeatedly according to this mode until ADC stop.

(2). Continue Mode

ADC gathers 8 samples every other 8*(N+1) sample cycle. The average of every 8 samples is updated in the data register, and the data interrupt sign is enabled. (N is defined in the bit[19:16] of **LRADC_CTRL_REG**).

(3).Single Mode

ADC gathers 8 samples, the average of the 8 samples is updated in data register, and the data interrupt sign is enabled, since then ADC stops sample.

9.7.3.4. Interrupt

Each LRADC channel has five interrupt sources and five interrupt enable controls.

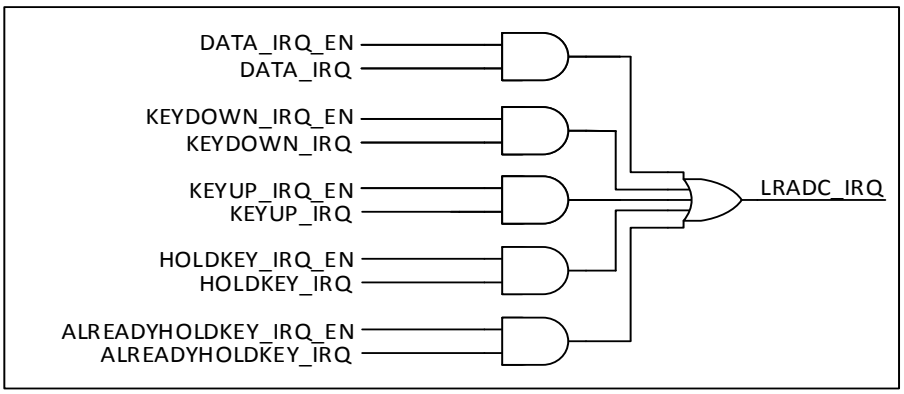

Figure 9- 36. LRADC Interrupt

When input voltage is between LEVELA(1.35V) and LEVELB(control by the bit[5:4] of LRADC_CTRL), IRQ1 can be generated. When input voltage is lower than LEVELB, IRQ2 can be generated.

If the controller receives IRQ1, and does not receive IRQ2 at some time, then the controller will generate Hold KEY Interrupt, otherwise DATA_IRQ Interrupt.

Hold KEY usually is used for self-locking key. When self-locking key holds locking status, the controller receives IRQ2, then the controller will generate Already Hold Key Interrupt.

9.7.4. Programming Guidelines

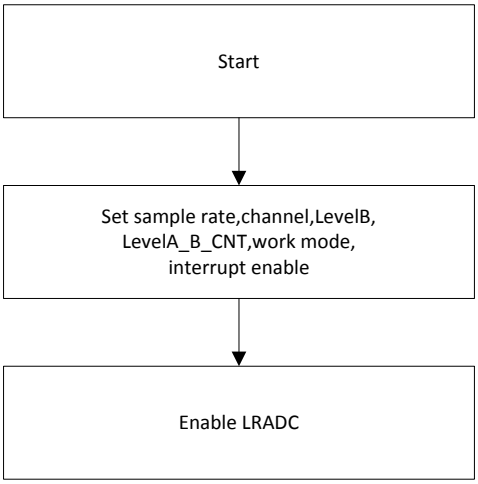

Figure 9- 37. LRADC Initial Process

(1) Set CONTINUE_TIME_SELECT when LRADC works in continue mode.

(2) The range of input voltage is from 0 to LEVELB.

(3) Calculation formula: LRADC_DATA = Vin/V_{REF}*64, V_{REF}=1.35 V

(4) LRADC has 6-bit resolution,1-bit offset error, 1-bit quantizing error. After LRADC calibrates 1-bit offset error, LRADC has 5-bit resolution.

9.7.5. Register List

Module Name	Base Address
LRADC	0x05070800

Register Name	Offset	Description
LRADC_CTRL	0x0000	LRADC Control Register
LRADC_INTC	0x0004	LRADC Interrupt Control Register
LRADC_INTS	0x0008	LRADC Interrupt Status Register
LRADC_DATA0	0x000C	LRADC Data Register0

9.7.6. Register Description

9.7.6.1. 0x0000 LRADC Control Register (Default Value: 0x0100_0168)

Offset: 0x0000			Register Name: LRADC_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 24	R/W	0x1	FIRST_CONVERT_DLY ADC First Convert Delay Setting ADC conversion is delayed by n samples.
23:20	/	/	/
19:16	R/W	0x0	CONTINUE_TIME_SELECT Continue Mode Time Select One of 8*(N+1) sample as a valuable sample data.
15:14	/	/	/
13:12	R/W	0x0	KEY_MODE_SELECT Key Mode Select 00: Normal Mode 01: Single Mode 10: Continue Mode
11:8	R/W	0x1	LEVELA_B_CNT Level A to Level B time threshold select, judge ADC convert value in level A to level B in n+1 samples.
7	R/W	0x0	LRADC_HOLD_KEY_EN LRADC Hold KEY Enable 0: Disable 1: Enable
6	R/W	0x1	LRADC_CHANNEL_EN LRADC Channel Enable 0: Disable 1: Enable
5: 4	R/W	0x2	LEVELB_VOL. Level B Corresponding Data Value Setting (the real voltage value) 00: 0x3C (1.266 V) 01: 0x39 (1.202 V) 10: 0x36 (1.139 V) 11: 0x33 (1.076 V)
3: 2	R/W	0x2	LRADC_SAMPLE_RATE LRADC Sample Rate 00: 2 kHz 01: 1 kHz 10: 500 Hz 11: 250 Hz
1	/	/	/
0	R/W	0x0	LRADC_EN LRADC Enable

			0: Disable 1: Enable

9.7.6.2. 0x0004 LRADC Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: LRADC_INTC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4	R/W	0x0	ADC0_KEYUP_IRQ_EN ADC0 Key Up IRQ Enable 0: Disable 1: Enable
3	R/W	0x0	ADC0_ALRDY_HOLD_IRQ_EN ADC0 Already Hold Key IRQ Enable 0: Disable 1: Enable
2	R/W	0x0	ADC0_HOLD_IRQ_EN ADC0 Hold Key IRQ Enable 0: Disable 1: Enable
1	R/W	0x0	ADC0_KEYDOWN_EN ADC0 Key Down Enable 0: Disable 1: Enable
0	R/W	0x0	ADC0_DATA_IRQ_EN ADC0 Data IRQ Enable 0: Disable 1: Enable

9.7.6.3. 0x0008 LRADC Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: LRADC_INTS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4	R/W1C	0x0	ADC0_KEYUP_PENDING ADC0 Key up Pending Bit When general key is pulled up, and the corresponding interrupt is enabled, the status bit is set. 0: No IRQ 1: IRQ Pending Writing 1 to the bit will clear it and its corresponding interrupt if the interrupt is enabled.

3	R/W1C	0x0	ADC0_ALRDY_HOLD_PENDING ADC0 Already Hold Pending Bit When hold key is pulled down and the general key is pulled down, and the corresponding interrupt is enabled. 0: No IRQ 1: IRQ Pending Writing 1 to the bit will clear it and its corresponding interrupt if the interrupt is enabled.
2	R/W1C	0x0	ADC0_HOLDKEY_PENDING ADC0 Hold Key Pending Bit When hold key is pulled down, and the corresponding interrupt is enabled, the status bit is set and the interrupt line is set. 0: NO IRQ 1: IRQ Pending Writing 1 to the bit will clear it and its corresponding interrupt if the interrupt is enabled.
1	R/W1C	0x0	ADC0_KEYDOWN_PENDING ADC0 Key Down IRQ Pending Bit When general key is pulled down, and the corresponding interrupt is enabled, the status bit is set and the interrupt line is set. 0: No IRQ 1: IRQ Pending Writing 1 to the bit will clear it and its corresponding interrupt if the interrupt is enabled.
0	R/W1C	0x0	ADC0_DATA_PENDING ADC0 Data IRQ Pending Bit 0: No IRQ 1: IRQ Pending Writing 1 to the bit will clear it and its corresponding interrupt if the interrupt is enabled.

9.7.6.4. 0x000C LRADC Data Register0 (Default Value: 0x0000_003F)

Offset: 0x000C			**Register Name: LRADC_DATA0**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:6	/	/	/
5:0	R	0x3F	LRADC0_DATA LRADC0 Data

9.8. CIR Receiver

9.8.1. Overview

The CIR (Consumer Infrared) receiver is a capturer of the pulse from IR Receiver module and uses Run-Length Code (RLC) to encode the pulse. The CIR receiver samples the input signal on the programmable frequency and records these samples into RX FIFO when one CIR signal is found on the air. The CIR receiver uses Run-Length Code (RLC) to encode pulse width. The encoded data is buffered in a 64 levels and 8-bit width RX FIFO; the MSB bit is used to record the polarity of the receiving CIR signal. The high level is represented as '1' and the low level is represented as '0'. The rest 7 bits are used for the length of RLC. The maximum length is 128. If the duration of one level (high or low level) is more than 128, another byte is used.

In the air, there is always some noise. One threshold can be set to filter the noise to reduce system loading and improve the system stability.

The CIR receiver has the following features:

- Full physical layer implementation
- Supports CIR for remote control or wireless keyboard
- 64x8 bits FIFO for data buffer
- Programmable FIFO thresholds
- Interrupt support
- Sample clock up to 1 MHz

9.8.2. Block Diagram

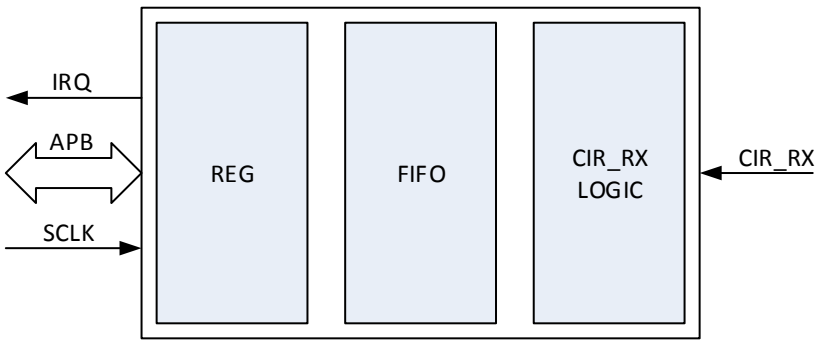

Figure 9- 38. CIR Receiver Block Diagram

9.8.3. Operations and Functional Descriptions

9.8.3.1. External Signals

Table 9-24 describes the external signals of CIR Receiver.

Table 9- 24. CIR Receiver External Signals

Signal	Description	Type
IR_RX	CIR input signal	I

9.8.3.2. Clock Sources

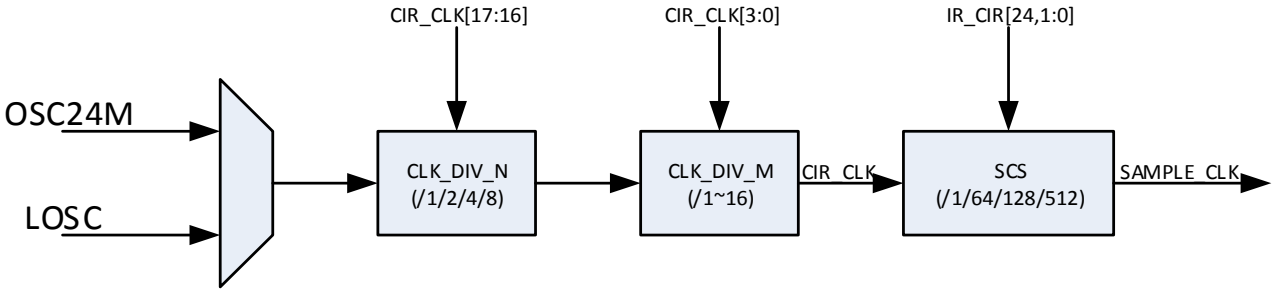

Figure 9- 39. CIR Receiver Clock

9.8.3.3. Typical Application

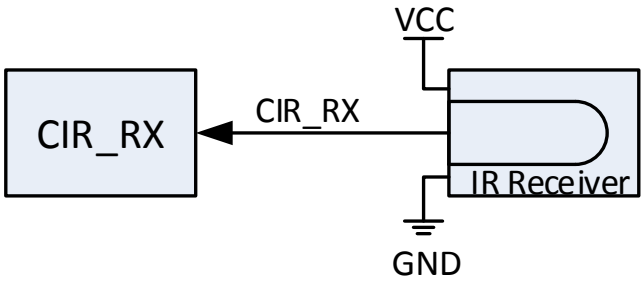

Figure 9- 40. CIR Receiver Application Diagram

9.8.3.4. Function Implementation

Figure 9- 41. NEC Protocol

In fact, CIR receiver module is a timer with capture function.

When CIR_RX signals satisfy ATHR (Active Threshold), CIR receiver can start to capture. In the process, the signal is ignored if the pulse width of the signal is less than NTHR. When CIR_RX signals satisfy ITHR (Idle Threshold), the capture process is stopped and the Receiver Packet End interrupt is generated, then Receiver Packet End Flag is asserted.

In a capture process, every effective pulse is buffered to FIFO in bytes according to the form of Run-Length Code. The MSB bit of a byte is polarity of pulse, and the rest 7 bits is pulse width by taking Sample Clock as basic unit. This is the code form of RLC-Byte. When the level changes or the pulse width counting overflows, RLC-Byte is buffered to FIFO. The CIR_RX module receives infrared signals transmitted by the infrared remote control, the software decodes the signals.

9.8.3.5. Operating Mode

- **Sample Clock**

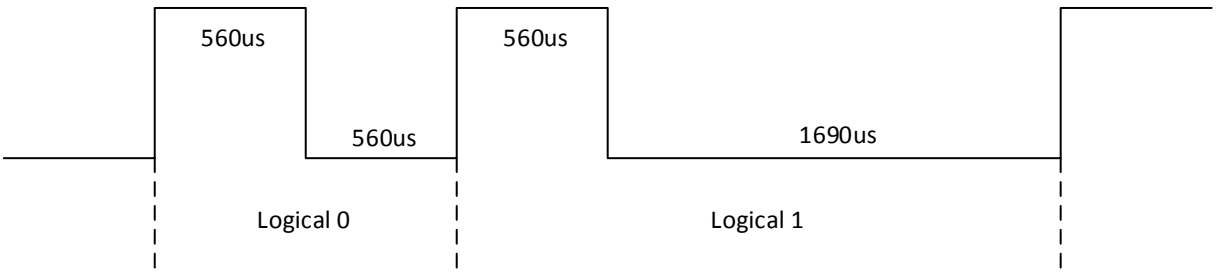

Figure 9- 42. Logical '0' and Logical '1' of NEC Protocol

For NEC protocol, a logical "1" takes 2.25ms(560us+1680us) to transmit, while a logical "0" is only half of that, being 1.12ms(560us+560us). For example, if sample clock is 31.25 kHz, sample cycle is 32us, then 18 sample cycles is 560us. So the RLC of 560us low level is 0x12, the RLC of 560us high level is 0x92. Then a logical "1" takes code 0x12 and code 0xb5 to transmit, a logical "0" takes code 0x12 and code 0x92 to transmit.

- **ATHR(Active Threshold)**

When CIR receiver is in Idle state, if electrical level of CIR_RX signal changes (positive jump or negative jump), and the duration reaches this threshold, then CIR takes the starting of the signal as a lead code, turns into active state and starts to capture CIR_RX signals.

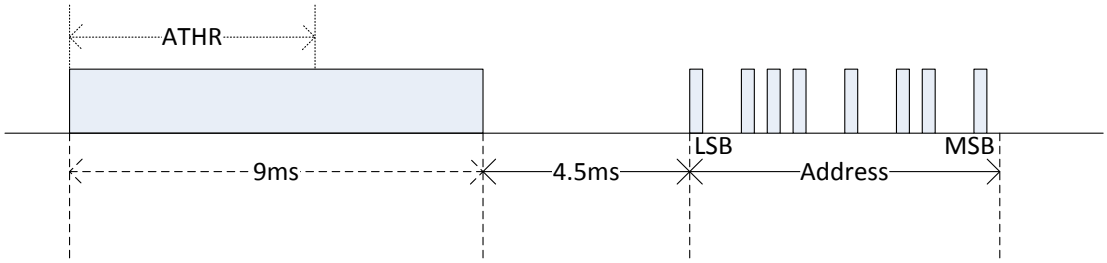

Figure 9- 43. ATHR Definition

- **ITHR(Idle Threshold)**

If electrical level of CIR_RX signals has no change, and the duration reaches this threshold, then CIR receiver enters into Idle state and ends this capture.

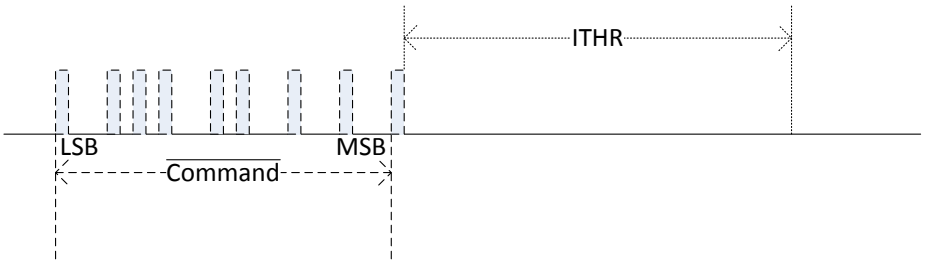

Figure 9- 44. ITHR Definition

- **NTHR(Noise Threshold)**

In capture process, the pulse is ignored if the pulse width is less than Noise Threshold.

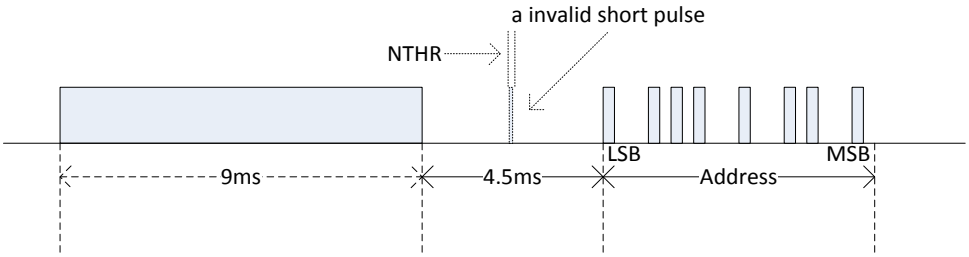

Figure 9- 45. NTHR Definition

- **APAM(Active Pulse Accept Mode)**

APAM is used to fit the type of lead code. If a pulse does not fit the type of lead code, it is not regarded as a lead code even if the pulse width reaches ATHR.

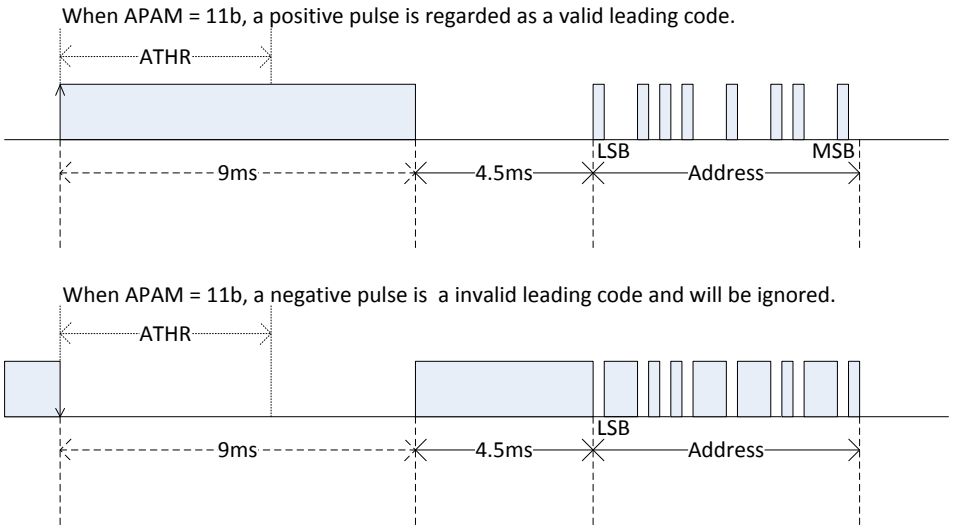

Figure 9- 46. APAM Definition

9.8.4. Programming Guidelines

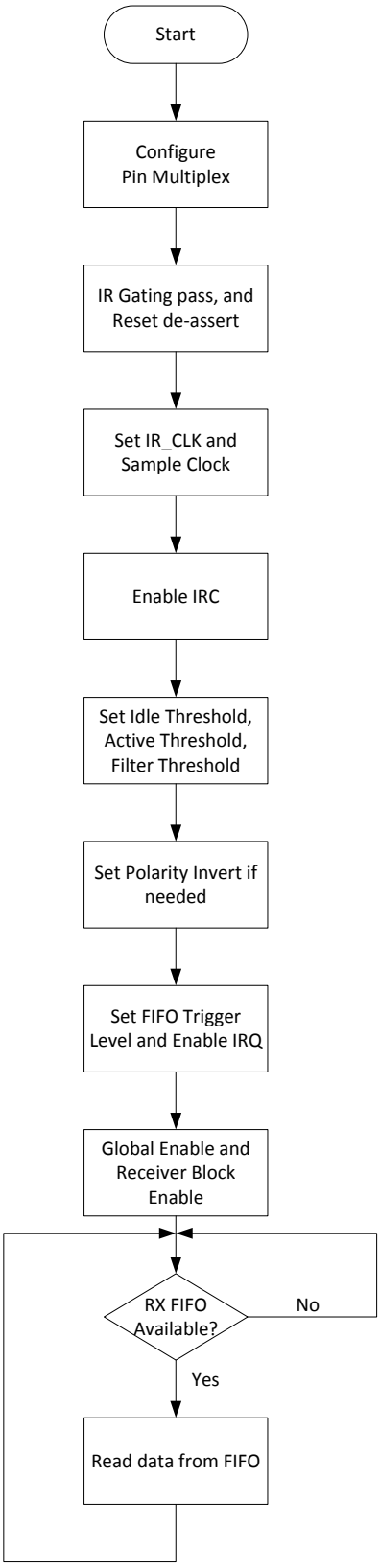

Figure 9- 47. CIR Receiver Process

9.8.5. Register List

Module Name	Base Address
CIR_RX	0x07040000

Register Name	Offset	Description
CIR_CTL	0x0000	CIR Control Register
CIR_RXPCFG	0x0010	CIR Receiver Pulse Configure Register
CIR_RXFIFO	0x0020	CIR Receiver FIFO Register
CIR_RXINT	0x002C	CIR Receiver Interrupt Control Register
CIR_RXSTA	0x0030	CIR Receiver Status Register
CIR_RXCFG	0x0034	CIR Receiver Configure Register

9.8.6. Register Description

9.8.6.1. 0x0000 CIR Receiver Control Register(Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: CIR_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:6	R/W	0x0	Active Pulse Accept Mode 00, 01: Both positive and negative pulses are valid as a leading code 10: Only negative pulse is valid as a leading code 11: Only positive pulse is valid as a leading code
5:4	R/W	0x0	CIR ENABLE 00~10: Reserved 11: CIR mode enable
3:2	/	/	/
1	R/W	0x0	RXEN Receiver Block Enable 0: Disable 1: Enable
0	R/W	0x0	GEN Global Enable A disable on this bit overrides any other block or channel enables and flushes all FIFOs. 0: Disable 1: Enable

9.8.6.2. 0x0010 CIR Receiver Pulse Configure Register(Default Value: 0x0000_0004)

Offset: 0x0010			Register Name: CIR_RXPCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x1	RPPI Receiver Pulse Polarity Invert 0: Do not invert receiver signal 1: Invert receiver signal
1:0	/	/	/

9.8.6.3. 0x0020 CIR Receiver FIFO Register(Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: CIR_RXFIFO
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R	0x0	Receiver Byte FIFO

9.8.6.4. 0x002C CIR Receiver Interrupt Control Register(Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: CIR_RXINT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:14	/	/	/
13:8	R/W	0x0	RAL RX FIFO available received byte level for interrupt and DMA request TRIGGER_LEVEL = RAL + 1
5	R/W	0x0	DRQ_EN RX FIFO DMA Enable 0: Disable 1: Enable When set to '1', the Receiver FIFO DRQ is asserted if reaching RAL. The DRQ is de-asserted when condition fails.
4	R/W	0x0	RAI_EN RX FIFO Available Interrupt Enable 0: Disable 1: Enable When set to '1', the Receiver FIFO IRQ is asserted if reaching RAL. The IRQ is de-asserted when condition fails.
3:2	/	/	/
1	R/W	0x0	RPEI_EN Receiver Packet End Interrupt Enable 0: Disable

			1: Enable
0	R/W	0x0	ROI_EN Receiver FIFO Overrun Interrupt Enable 0: Disable 1: Enable

9.8.6.5. 0x0030 CIR Receiver Status Register(Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: CIR_RXSTA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:14	/	/	/
13:8	R	0x0	RAC RX FIFO Available Counter 0: No available data in RX FIFO 1: 1 byte available data in RX FIFO 2: 2 bytes available data in RX FIFO … 64: 64 bytes available data in RX FIFO
7	R	0x0	STAT Status of CIR 0: Idle 1: Busy
6:5	/	/	/
4	R/W1C	0x0	RA RX FIFO Available 0: RX FIFO not available according its level 1: RX FIFO available according its level This bit is cleared by writing a ‘1’.
3:2	/	/	/
1	R/W1C	0x0	RPE Receiver Packet End Flag 0: STO was not detected. In CIR mode, one CIR symbol is receiving or not detected. 1: STO field or packet abort symbol (7’b0000,000 and 8’b0000,0000 for MIR and FIR) is detected. In CIR mode, one CIR symbol is received. This bit is cleared by writing a ‘1’.
0	R/W1C	0x0	ROI Receiver FIFO Overrun 0: Receiver FIFO not overrun 1: Receiver FIFO overrun This bit is cleared by writing a ‘1’.

9.8.6.6. 0x0034 CIR Receiver Configure Register(Default Value: 0x0000_1828)

<table>
<tr><td colspan="3">Offset: 0x0034</td><td>Register Name: CIR_RXCFG</td></tr>
<tr><td>Bit</td><td>Read/Write</td><td>Default/Hex</td><td>Description</td></tr>
<tr><td>31:25</td><td>/</td><td>/</td><td>/</td></tr>
<tr><td>24</td><td>R/W</td><td>0x0</td><td>SCS2
Bit2 of Sample Clock Select for CIR
This bit is defined by SCS bits below.</td></tr>
<tr><td>23</td><td>R/W</td><td>0x0</td><td>ATHC
Active Threshold Control for CIR
0: ATHR in unit of (Sample Clock)
1: ATHR in unit of (128*Sample Clocks)</td></tr>
<tr><td>22:16</td><td>R/W</td><td>0x0</td><td>ATHR
Active Threshold for CIR
These bits control the duration of CIR from idle to active state. The duration can be calculated by ((ATHR + 1)*(ATHC? Sample Clock: 128*Sample Clock)).</td></tr>
<tr><td>15:8</td><td>R/W</td><td>0x18</td><td>ITHR
Idle Threshold for CIR
The Receiver uses it to decide whether the CIR command has been received. If there is no CIR signal on the air, the receiver is staying in IDLE status. One active pulse will bring the receiver from IDLE status to Receiving status. After the CIR is end, the inputting signal will keep the specified level (high or low level) for a long time. The receiver can use this idle signal duration to decide that it has received the CIR command. The corresponding flag is asserted. If the corresponding interrupt is enabled, the interrupt line is asserted to CPU.
When the duration of signal keeps one status (high or low level) for the specified duration ((ITHR + 1)*128 sample_clk), this means that the previous CIR command has been finished.</td></tr>
<tr><td>7:2</td><td>R/W</td><td>0xA</td><td>NTHR
Noise Threshold for CIR
When the duration of signal pulse (high or low level) is less than NTHR, the pulse is taken as noise and should be discarded by hardware.
0: All samples are recorded into RX FIFO
1: If the signal is only one sample duration, it is taken as noise and discarded.
2: If the signal is less than (<=) two sample duration, it is taken as noise and discarded.
…
61: If the signal is less than (<=) sixty-one sample duration, it is taken as noise and discarded.</td></tr>
<tr><td>1:0</td><td>R/W</td><td>0x0</td><td>SCS
Sample Clock Select for CIR
<table>
<tr><td>SCS2</td><td>SCS[1]</td><td>SCS[0]</td><td>Sample Clock</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>CIR_CLK/64</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>CIR_CLK /128</td></tr>
</table></td></tr>
</table>

			0	1	0	CIR_CLK /256	
			0	1	1	CIR_CLK /512	
			1	0	0	CIR_CLK	
			1	0	1	Reserved	
			1	1	0	Reserved	
			1	1	1	Reserved	

9.9. PWM

9.9.1. Overview

The PWM controller has 4 PWM channels(PWM1,PWM2,PWM3,PWM4), and divides to 3 groups: PWM1, PWM23 pair, PWM4. PWM23 pair consists of PWM2 and PWM3. PWM23 pair supports deadzone function.

The PWM has the following features:

- 4 PWM channels(PWM1, PWM2, PWM3, PWM4)
- PWM1/PWM4 has the single channel characteristics of PWM module, and has no pair function
- Supports pulse(configurable pulse number), cycle and complementary pair output
- Supports capture input
- Programming deadzone output
- Build-in the programmable dead-time generator, controllable dead-time
- Three kinds of output waveform: continuous waveform, pulse waveform and complementary pair
- Output frequency range: 0~ 24 MHz/100 MHz
- Various duty-cycle: 0% ~100%
- Minimum resolution: 1/65536
- Interrupt generation of PWM output and capture input

9.9.2. Block Diagram

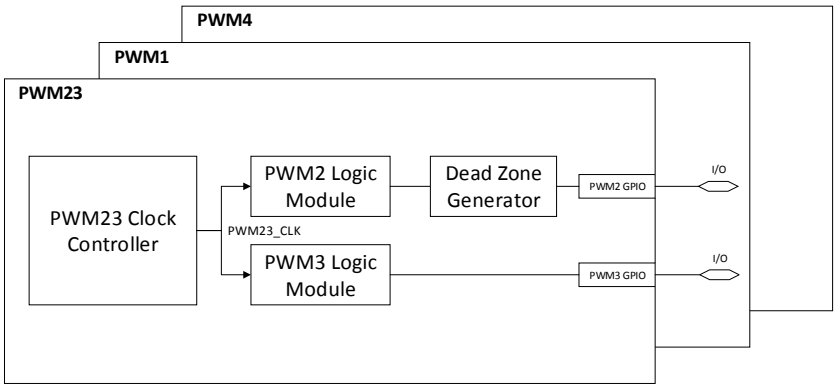

Figure 9- 48. PWM Block Diagram

The PWM23 pair consists of 1 clock module, 2 timer logic module and 1 programmable dead-time generator.

The PWM1 consists of 1 clock module, 1 timer logic module.

The PWM4 consists of 1 clock module, 1 timer logic module.

The PWM1 and PWM4 do not support deadzone function.

9.9.3. Operations and Functional Descriptions

9.9.3.1. External Signals

Table 9-25 describes the external signals of the PWM.

Table 9- 25. PWM External Signals

Signal	Description	Type
PWM1	Pulse Width Module Channel1	I/O
PWM2	Pulse Width Module Channel2	I/O
PWM3	Pulse Width Module Channel3	I/O
PWM4	Pulse Width Module Channel4	I/O

9.9.3.2. Typical Application

- Suitable for display device, such as LCD
- Suitable for electric motor control

9.9.3.3. Clock Controller

Using PWM23 pair as an example, the clock controller diagram is as follows. The clock diagram of PWM1 is the same as PWM3 of the PWM23 pair. The clock diagram of PWM4 is the same as PWM2 of the PWM23 pair.

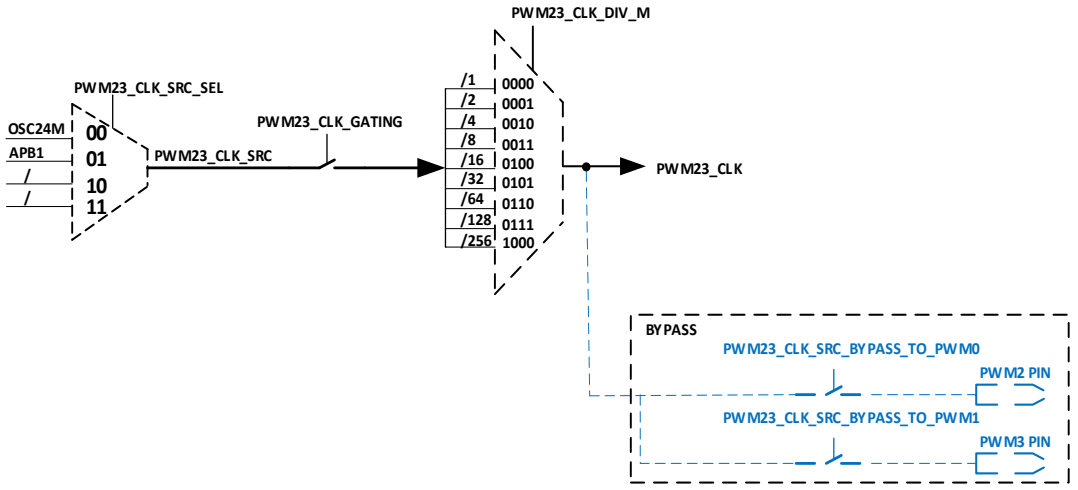

Figure 9- 49. PWM23 Pair Clock Controller Diagram

The clock sources of PWM have OSC24M and APB1 Bus. OSC24M comes from external high frequency oscillator, APB1 is APB1 bus clock, usually is 100 MHz.

PWM23 pair includes clock source select(PWM23_CLK_SRC_SEL), the first-level exponent divider (PWM23_CLK_DIV_M), the second-level count divider(PRESCAL_K), clock source bypass(CLK_SRC_BYPASS) and clock switch(PWM23_CLK_GATING).

The clock source bypass function is that clock source directly accesses PWM output, the PWM output waveform is the waveform of clock controller output. The BYPASS gridlines in the above figure indicates clock source bypass function, the details about implement, please see Figure 9-50.

9.9.3.4. PWM Output

Using PWM23 pair as an example, Figure 9-50 indicates the output logic module diagram of PWM23 pair. The logic module diagram of PWM1 is the same as PWM3 of the PWM23 pair. The logic module diagram of PWM4 is the same as PWM2 of the PWM23 pair.

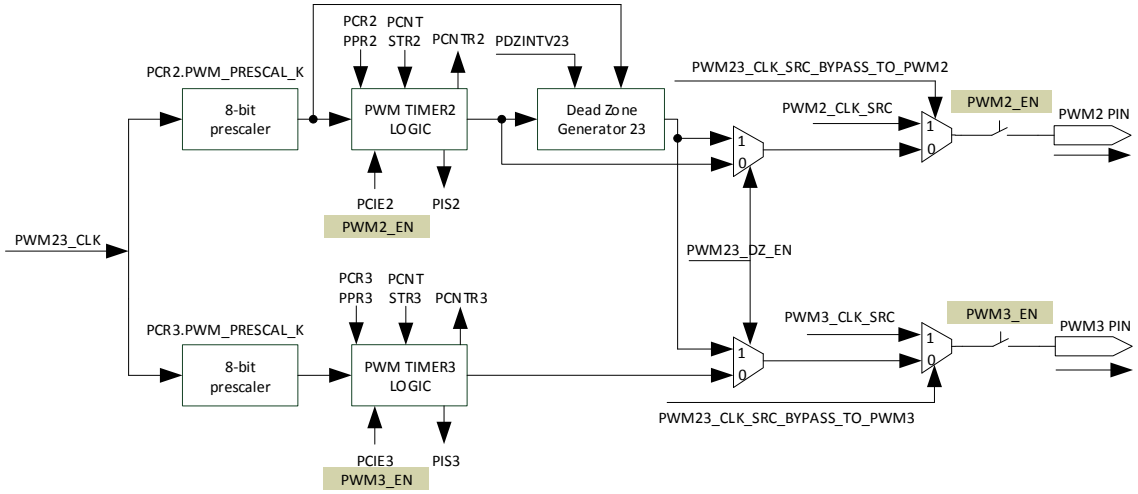

Figure 9- 50. PWM23 Pair Output Logic Module Diagram

PWM Timer Logic module(PWM_TIMER_LOGIC) consists of one 16-bit up-counter and one 16-bit comparator. The up-counter is used to control PWM cycle, the comparator is used to control duty-cycle.

The up-counter and the comparator support cache loading, after PWM output is enabled, the register values of the up-counter and the comparator(**PPR[PWM_ENTIRE_CYCLE]** and **PPR[PWM_ACTIVE_CYCLE]**) can be changed anytime, the changed value caches into the cache register. When the value of the up-counter reaches the value of **PPR[PWM_ENTIRE_CYCLE]**, the value of the cache register can be loaded to the up-counter and the comparator. Cache-loading is good to avoid unstable PWM output waveform with burred feature when updating the values of the up-counter and the comparator.

PWM supports cycle and pulse waveform output.

Cycle mode: When the value of the up-counter reaches **PPR.[PWM_ENTIRE_CYCLE]**, the value of the up-counter is loaded automatically to 0 and the up-counter continues to count, then the output waveform is a continuous waveform.

Pulse mode: When the value of the up-counter reaches **PPR.[PWM_ENTIRE_CYCLE]**, the value of the up-counter is loaded automatically to 0 and the up-counter stops counting, then the output waveform is a pulse waveform.

9.9.3.5. Period and Duty-cycle

The period, duty-cycle and active state of PWM output waveform are decided by the up-counter and comparator. The rule of the comparator is as follows.

PCNTR >= (PWM_ENTIRE_CYCLE - PWM_ACT_CYCLE), output “active state”
PCNTR < (PWM_ENTIRE_CYCLE - PWM_ACT_CYCLE), output “~ (active state)”

(1) Active state of PWM2 channel is high level (PCR2. PWM_ACT_STA = 1)

When PCNTR2 >= (PPR2. PWM_ENTIRE_CYCLE - PPR2.PWM_ACT_CYCLE), then PWM2 outputs 1(high level).
When PCNTR2 < (PPR2. PWM_ENTIRE_CYCLE - PPR2.PWM_ACT_CYCLE), then PWM2 outputs 0(low level).

The formula of PWM output period and duty-cycle is as follows.
T_{period} = (PWM23_CLK / PWM2_PRESCALE_K)$^{-1}$ * PPR2. PWM_ENTIRE_CYCLE
$T_{high\text{-}level}$ = (PWM23_CLK / PWM2_PRESCALE_K)$^{-1}$ * PPR2.PWM_ACT_CYCLE
$T_{low\text{-}level}$ = (PWM23_CLK / PWM2_PRESCALE_K)$^{-1}$ * (PPR2.PWM_ENTIRE_CYCLE - PPR2.PWM_ACT_CYCLE)
Duty-cycle = (high level time) / (1 period time) = $T_{high\text{-}level}$ / T_{period}

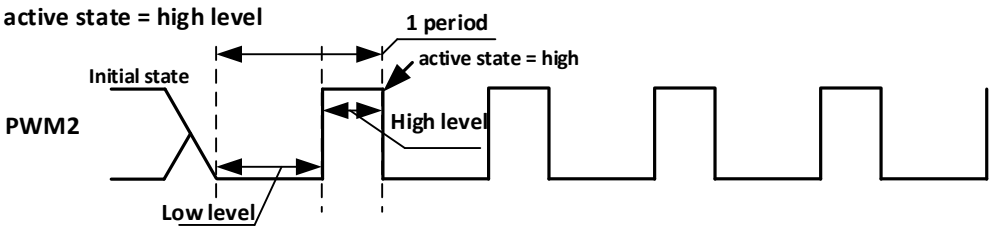

Figure 9- 51. The Period and Duty-cycle of PWM2 High Level Active State

(2) Active state of PWM2 channel is low level (PCR2. PWM_ACT_STA = 0)

When PCNTR2 >= (PPR2. PWM_ENTIRE_CYCLE – PPR2.PWM_ACT_CYCLE), then PWM2 outputs 0.
When PCNTR2 < (PPR2. PWM_ENTIRE_CYCLE – PPR2.PWM_ACT_CYCLE), then PWM2 outputs 1.

The formula of PWM output period and duty-cycle is as follows.
T_{period} = (PWM23_CLK / PWM2_PRESCALE_K)$^{-1}$ * PPR2. PWM_ENTIRE_CYCLE
$T_{high\text{-}level}$ = (PWM23_CLK / PWM2_PRESCALE_K)$^{-1}$ * (PPR2.PWM_ENTIRE_CYCLE – PPR2.PWM_ACT_CYCLE)
$T_{low\text{-}level}$ = (PWM23_CLK / PWM2_PRESCALE_K)$^{-1}$ * PPR2.PWM_ACT_CYCLE
Duty-cycle = (low level time) / (1 period time) = $T_{low\text{-}level}$ / T_{period}

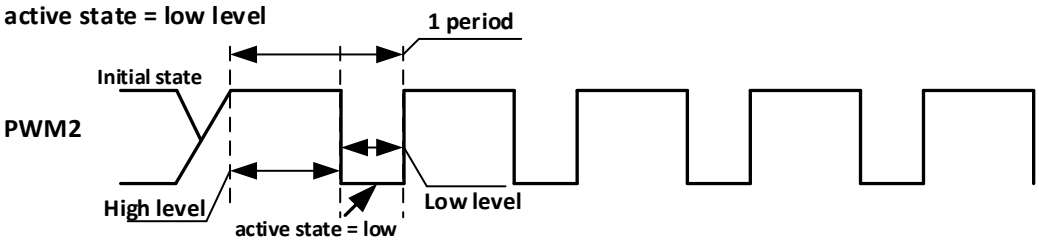

Figure 9- 52. The Period and Duty-cycle of PWM2 Low Level Active State

9.9.3.6. Pulse Mode and Cycle Mode

PWM output supports pulse mode and cycle mode. PWM in pulse mode outputs one pulse waveform, but PWM in cycle mode outputs continuous waveform. Figure 9-53 shows the PWM(take PWM2 as an example) output waveform in pulse mode and cycle mode.

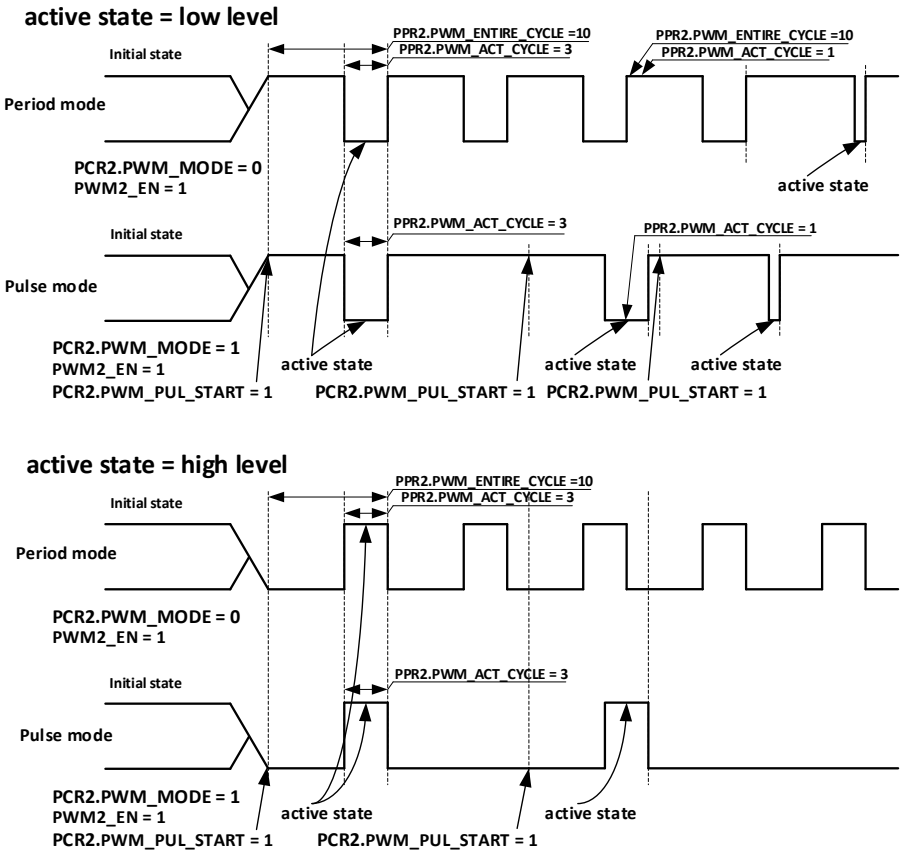

Figure 9- 53. PWM2 Output Waveform in Pulse Mode and Cycle Mode

When PCR2.PWM_MODE is 0, PWM2 outputs cycle waveform. The calculating formula of T_{period} and $T_{active\text{-}state}$ is as follows.

T_{period} = (PWM23_CLK / PWM2_PRESCALE_K)$^{-1}$ * PPR2. PWM_ENTIRE_CYCLE

$T_{active\ state}$ = (PWM23_CLK / PWM2_PRESCALE_K)$^{-1}$ * PPR2.PWM_ACT_CYCLE

When PCR2.PWM_ACT_STA is 0, the active state of cycle waveform is low level.

When PCR2.PWM_ACT_STA is 1, the active state of cycle waveform is high level.

When PCR2.PWM_MODE is 1, PWM2 outputs pulse waveform. The calculating formula of pulse length is as follows.

Pulse length = PWM23_CLK / PWM2_PRESCALE_K * PPR2. PWM _ACT_CYCLE

When PCR2.PWM_ACT_STA is 0, the pulse level is low level, PWM0 channel outputs low pulse.

When PCR2.PWM_ACT_STA is 1, the pulse level is high level, PWM0 channel outputs high pulse.

After PWM2 channel enabled, PCR2. PWM_PUL_START needs be set to 1 when PWM2 needs output pulse waveform, after completed output, PCR2. PWM_PUL_START can be cleared to 0 by hardware.

The up-counter and comparator for PWM2 channel support cache loading, after PWM2 channel is enabled, whether cycle mode or pulse mode, PPR2 value is modified and cached to the buffer register of PPR2 , when the up-counter

value reaches PPR2. PWM_ENTIRE_CYCLE, the value in the buffer register will be loaded to up-counter and comparator, namely the value of up-counter and comparator will be overloaded in the next cycle.

Take Figure 9-53(active state =low level) as an example.

In cycle mode, the initial value of PPR2.PWM_ENTIRE_CYCLE is 10, the initial value of PPR2. PWM _ACT_CYCLE is 3. At some time, the value of PPR2. PWM _ACT_CYCLE is modified to 1, during the current cycle, the modified PPR2 values is cached to PPR2 buffer register, at the beginning of the next cycle, the value of PPR2 buffer register is loaded into up-counter and comparator, then up-counter starts to work.

In pulse mode, the initial value of PPR2. PWM _ACT_CYCLE is 3, in the generation process of a single pulse , the value of PPR2. PWM _ACT_CYCLE is modified to 1, during the current cycle, the modified PPR2 values is cached to PPR2 buffer register, when the value of up-counter reaches PPR2. PWM_ENTIRE_CYCLE, then the pulse waveform output ends, the value of PPR2 buffer register is loaded into up-counter and comparator, at the next time, after PCR2. PWM_PUL_START is set to 1, the modified value of PPR2 has taken effect.

NOTE

The time that loading PPR2 buffer register value into up-counter and comparator is very short, which can be ignored, and does not affect the PWM output.

9.9.3.7. Complementary Pair Output

The PWM23 pair supports complementary pair output and PWM pair with dead-time. Figure 9-54 shows the complementary pair output of PWM23 pair.

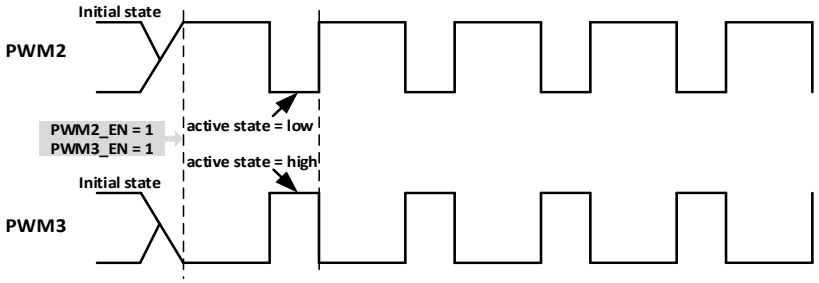

Figure 9- 54. PWM23 Complementary Pair Output

The complementary pair output need satisfy the following three conditions:

- The same frequency,the same duty-cycle
- Opposite active state
- Enable two channels of PWM pair at the same time

9.9.3.8. Dead-time Generator

Every PWM pair has a programmable dead-time generator. When the dead-time function of PWM pair enabled,

PWM23 pair outputs a pair of PWM waveforms that insert dead-time, PWM23 pair output waveform is decided by PWM2 timer logic module and DeadZone Generator23. Figure 9-55 shows the output waveform.

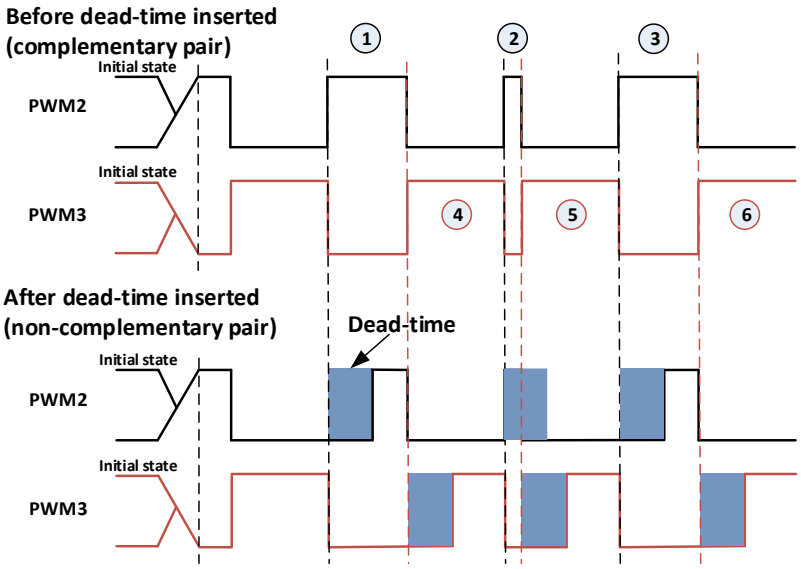

Figure 9- 55. PWM23 Pair Waveform Before/After Insert Dead-time

The PWM waveform before the insertion of dead-time indicates a complementary waveform pair of non-inserted dead-time in Dead Zone Generator 23.

The PWM waveform after the insertion of dead-time indicates a non-complementary PWM waveform pair inserted dead-time in a complementary waveform pair of Dead Zone Generator 23. The PWM waveform pair at last outputs to PWM2 pin and PWM3 pin.

For complementary pair of Dead Zone Generator 23, the principle of inserting dead-time is that to insert dead-time as soon as the rising edge came. If high level time for mark② in the above figure is less than dead-time, then dead-time will override the high level. The setting of dead-time need consider the period and duty-cycle of output waveform. Dead-time formula is defined as follows:

Dead-time = (PWM23_CLK / PWM0_PRESCALE_K)$^{-1}$ * PDZINTV23

9.9.3.9. Capture Input

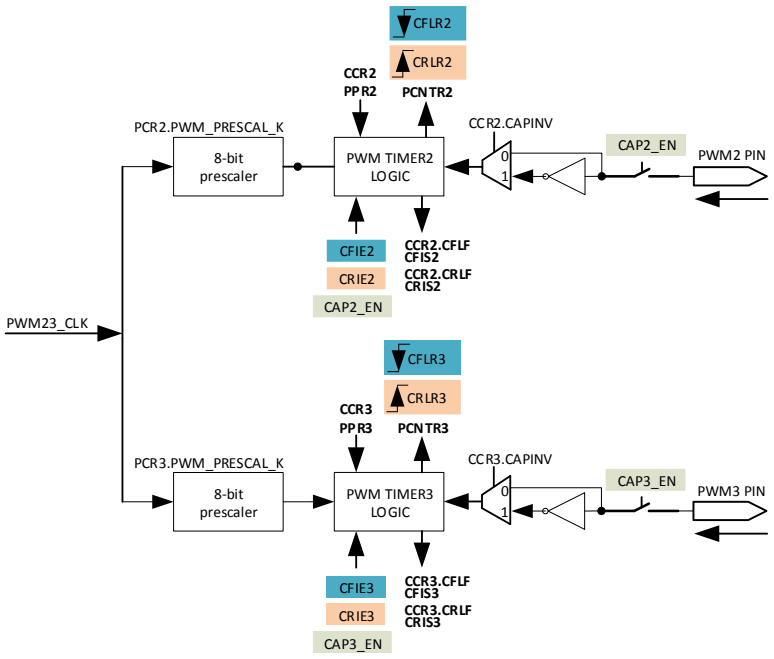

Figure 9- 56. PWM23 Capture Logic Module Diagram

Besides the timer logic module of every PWM channel generates PWM output, it can be used to capture rising edge and falling edge of the external clock. Using PWM2 channel as an example, PWM2 channel has one **CFLR2** and one **CRLR2** for capturing up-counter value in falling edge and rising edge, respectively. You can calculate the period of external clock by **CFLR2** and **CRLR2**.

$T_{high\text{-}level}$ = (PWM23_CLK / PWM2_PRESCALE_K)$^{-1}$ * CRLR2

$T_{low\text{-}level}$ = (PWM23_CLK / PWM2_PRESCALE_K)$^{-1}$ * CFLR2

T_{period} = $T_{high\text{-}level}$ + $T_{low\text{-}level}$

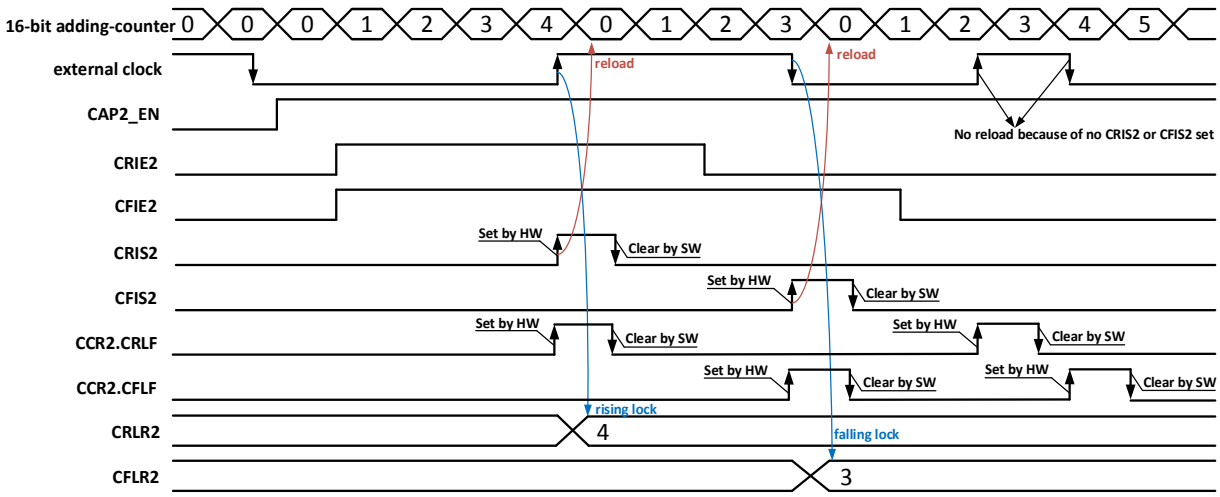

Figure 9- 57. PWM2 Channel Capture Timing

When the capture input function of PWM2 channel is enabled, the PCNTR of PWM2 channel starts to work.

When the timer logic module of PWM2 captures one rising edge, the current value of up-counter is locked to **CRLR2**, and **CCR2.CRLF** is set to 1. If **CRIE2** is 1, then **CRIS2** is set to 1, PWM2 channel sends interrupt request, and the up-counter is loaded to 0 and continues to count. If **CRIE2** is 0, the timer logic module of PWM2 captures rising edge, **CRIS2** can not be set to 1, the up-counter is not loaded to 0.

When the timer logic module of PWM2 captures one falling edge, the current value of **PCNTR** is locked to **CFLR2**, and **CCR2.CFLF** is set to 1 . If **CFIE2** is 1, then **CFIS2** is set to 1, PWM2 channel sends interrupt request, and the up-counter is loaded to 0 and continues to count. If **CFIE2** is 0, the timer logic module of PWM2 captures falling edge, **CFIS2** can not be set to 1, the up-counter is not loaded to 0.

9.9.3.10. Interrupt

PWM supports interrupt generation when PWM channel is configured to PWM output or capture input .

For PWM output function, when one period of PWM waveform is output in cycle mode, the PIS of the corresponding PWM channel is set to 1; when (PWM_PULNUM+1) periods of PWM waveform is output in pulse mode, the PIS of the corresponding PWM channel is set to 1.

NOTE

The PIS bit is set to 1 automatically by hardware and cleared by software.

For capture input function, when the timer logic module of the capture channel2 captures rising edge, and **CRIE2** is 1, then **CRIS2** is set to 1; when the timer logic module of the capture channel2 captures falling edge, and **CFIE2** is 1, then **CFIS2** is set to 1.

9.9.4. Working Mode

The following working mode takes PWM23 as an example.

9.9.4.1. Clock Configuration

(1) PWM gating: When using PWM, write 1 to **PCGR[PWMx_CLK_GATING]**.
(2) PWM clock source select: Set **PCCR23[PWM23_CLK_SRC]** to select OSC24M or APB1 clock.
(3) PWM clock divider: Set **PCCR23[PWM23_CLK_DIV_M]** to select different frequency division coefficient (1/2/4/8/16/32/64/128/256).
(4) PWM clock bypass: Set **PCGR[PWM_CLK_SRC_BYPASS_TO_PWM]** to 1, output the PWM clock after the secondary frequency division to the corresponding PWM output pin.
(5) PWM internal clock configuration: Set **PCR[PWM_PRESCAL_K]** to select any frequency division coefficient from 1 to 256.

9.9.4.2. PWM Configuration

(1) PWM mode: Set **PCR[PWM_MODE]** to select cycle mode or pulse mode, if pulse mode, PWM_PUL_NUM needs be configured.
(2) PWM valid level: Set **PCR[PWM_ACT_STA]** to select low level or high level.
(3) PWM duty-cycle: Configure **PPR[PWM_ENTIRE_CYCLE]** and **PPR[PWM _ACT_CYCLE]** after clock gating is opened.
(4) Enable PWM: Configure PER to select the corresponding PWM enable bit; when selecting pulse mode, **PCR[PWM_PUL_START]** needs be enabled.

9.9.4.3. Deadzone Control

(1) Deadzone initial value: Set **PDZCR23[PDZINTV23]**.
(2) Deaszone enable: Set **PDZCR23[PWM23_DZ_CN]**.

9.9.4.4. Capture Input

(1) Capture enable: Configure CER to enable the corresponding channel.
(2) Capture mode: Configure **CCR[CRLF]** and **CCR[CFLF]** to select rising edge capture or falling edge capture, configure **CCR[CAPINV]** to select whether the input signal does reverse processing.

9.9.5. Register List

Module Name	Base Address
PWM	0x0300A000

Register Name	Offset	Description
PIER	0x0000	PWM IRQ Enable Register
PISR	0x0004	PWM IRQ Status Register
CIER	0x0010	Capture IRQ Enable Register
CISR	0x0014	Capture IRQ Status Register
PCCR01	0x0020	PWM01 Clock Configuration Register
PCCR23	0x0024	PWM23 Clock Configuration Register
PCCR45	0x0028	PWM45 Clock Configuration Register
PDZCR01	0x0030	PWM01 Dead Zone Control Register
PDZCR23	0x0034	PWM23 Dead Zone Control Register
PDZCR45	0x0038	PWM45 Dead Zone Control Register
PER	0x0040	PWM Enable Register
CER	0x0044	Capture Enable Register
PCR	0x0060+0x0000+N*0x0020(N=1~4)	PWM Control Register
PPR	0x0060+0x0004+N*0x0020(N=1~4)	PWM Period Register
PCNTR	0x0060+0x0008+N*0x0020(N=1~4)	PWM Count Register

CCR	0x0060+0x000C+N*0x0020(N=1~4)	Capture Control Register
CRLR	0x0060+0x0010+N*0x0020(N=1~4)	Capture Rise Lock Register
CFLR	0x0060+0x0014+N*0x0020(N=1~4)	Capture Fall Lock Register

9.9.6. Register Description

9.9.6.1. 0x0000 PWM IRQ Enable Register (Default Value: 0x0000_0000)

Offset:0x0000			Register Name: PIER
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4	R/W	0x0	PCIE4 PWM Channel 4 Interrupt Enable 0: PWM channel 4 interrupt disable 1: PWM channel 4 interrupt enable
3	R/W	0x0	PCIE3 PWM Channel 3 Interrupt Enable 0: PWM channel 3 interrupt disable 1: PWM channel 3 interrupt enable
2	R/W	0x0	PCIE2 PWM Channel 2 Interrupt Enable 0: PWM channel 2 interrupt disable 1: PWM channel 2 interrupt enable
1	R/W	0x0	PCIE1 PWM Channel 1 Interrupt Enable 0: PWM channel 1 interrupt disable 1: PWM channel 1 interrupt enable
0	/	/	/

9.9.6.2. 0x0004 PWM IRQ Status Register (Default Value: 0x0000_0000)

Offset:0x0004			Register Name: PISR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4	R/W1C	0x0	PIS4 PWM Channel 4 Interrupt Status When PWM channel 4 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 4 interrupt is not pending. Reads 1: PWM channel 4 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 4 interrupt status.

3	R/W1C	0x0	PIS3 PWM Channel 3 Interrupt Status When PWM channel 3 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 3 interrupt is not pending. Reads 1: PWM channel 3 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 3 interrupt status.
2	R/W1C	0x0	PIS2 PWM Channel 2 Interrupt Status When PWM channel 2 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 2 interrupt is not pending. Reads 1: PWM channel 2 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 2 interrupt status.
1	R/W1C	0x0	PIS1 PWM Channel 1 Interrupt Status When PWM channel 1 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 1 interrupt is not pending. Reads 1: PWM channel 1 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 1 interrupt status.
0	/	/	/

9.9.6.3. 0x0010 PWM Capture IRQ Enable Register (Default Value: 0x0000_0000)

Offset:0x0010			**Register Name: CIER**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9	R/W	0x0	CFIE4 If the bit is set 1, when capturing channel 4 captures falling edge, it generates a capturing channel 4 pending. 0: Capture channel 4 fall lock interrupt disable 1: Capture channel 4 fall lock interrupt enable
8	R/W	0x0	CRIE4 If the bit is set 1, when capturing channel 4 captures rising edge, it generates a capturing channel 4 pending. 0: Capturing channel 4 rise lock interrupt disable 1: Capturing channel 4 rise lock interrupt enable
7	R/W	0x0	CFIE3 If the bit is set 1, when capturing channel 3 captures falling edge, it generates a capturing channel 3 pending.

			0: Capturing channel 3 fall lock interrupt disable 1: Capturing channel 3 fall lock interrupt enable
6	R/W	0x0	CRIE3 If the bit is set 1, when capturing channel 3 captures rising edge, it generates a capturing channel 3 pending. 0: Capturing channel 3 rise lock interrupt disable 1: Capturing channel 3 rise lock interrupt enable
5	R/W	0x0	CFIE2 If the bit is set 1, when capturing channel 2 captures falling edge, it generates a capturing channel 2 pending. 0: Capturing channel 2 fall lock interrupt disable 1: Capturing channel 2 fall lock interrupt enable
4	R/W	0x0	CRIE2 If the bit is set 1, when capturing channel 2 captures rising edge, it generates a capturing channel 2 pending. 0: Capturing channel 2 rise lock interrupt disable 1: Capturing channel 2 rise lock interrupt enable
3	R/W	0x0	CFIE1 If the bit is set 1, when capturing channel 1 captures falling edge, it generates a capturing channel 1 pending. 0: Capturing channel 1 fall lock interrupt disable 1: Capturing channel 1 fall lock interrupt enable
2	R/W	0x0	CRIE1 If the bit is set 1, when capturing channel 1 captures rising edge, it generates a capturing channel 1 pending. 0: Capturing channel 1 rise lock interrupt disable 1: Capturing channel 1 rise lock interrupt enable
1:0	/	/	/

9.9.6.4. 0x0014 PWM Capture IRQ Status Register (Default Value: 0x0000_0000)

Offset:0x0014			**Register Name: CISR**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9	R/W1C	0x0	CFIS4 Capturing channel 4 falling lock interrupt status. When capturing channel 4 captures falling edge, if capturing channel 4 fall lock interrupt (**CFIE4**) is enabled, this bit is set 1 by hardware.Writing 1 to clear this bit. Reads 0: Capturing channel 4 interrupt is not pending. Reads 1: Capturing channel 4 interrupt is pending. Writes 0: No effect. Writes 1: Clear capturing channel 4 interrupt status.
8	R/W1C	0x0	CRIS4

			Capturing channel 4 rising lock interrupt status. When capturing channel 4 captures rising edge, if capturing channel 4 rise lock interrupt (**CRIE4**) is enabled, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: Capturing channel 4 interrupt is not pending. Reads 1: Capturing channel 4 interrupt is pending. Writes 0: No effect. Writes 1: Clear capturing channel 4 interrupt status.
7	R/W1C	0x0	CFIS3 Capture channel 3 falling lock interrupt status. When capture channel 3 captures falling edge, if capture channel 3 fall lock interrupt (**CFIE3**) is enabled, this bit is set 1 by hardware.Writing 1 to clear this bit. Reads 0: Capture channel 3 interrupt is not pending. Reads 1: Capture channel 3 interrupt is pending. Writes 0: no effect. Writes 1: Clear capture channel 3 interrupt status.
6	R/W1C	0x0	CRIS3 Capture channel 3 rising lock interrupt status. When capture channel 3 captures rising edge, if capture channel 3 rise lock interrupt (**CRIE3**) is enabled, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: Capture channel 3 interrupt is not pending. Reads 1: Capture channel 3 interrupt is pending. Writes 0: no effect. Writes 1: Clear capture channel 3 interrupt status.
5	R/W1C	0x0	CFIS2 Capture channel 2 falling lock interrupt status. When capture channel 2 captures falling edge, if capture channel 2 fall lock interrupt (**CFIE2**) is enabled, this bit is set 1 by hardware.Writing 1 to clear this bit. Reads 0: Capture channel 2 interrupt is not pending. Reads 1: Capture channel 2 interrupt is pending. Writes 0: no effect. Writes 1: Clear capture channel 2 interrupt status.
4	R/W1C	0x0	CRIS2 Capture channel 2 rising lock interrupt status. When capture channel 2 captures rising edge, if capture channel 2 rise lock interrupt (**CRIE2**) is enabled, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: Capture channel 2 interrupt is not pending. Reads 1: Capture channel 2 interrupt is pending. Writes 0: no effect. Writes 1: Clear capture channel 2 interrupt status.
3	R/W1C	0x0	CFIS1

			Capture channel 1 falling lock interrupt status. When capture channel 1 captures falling edge, if capture channel 1 fall lock interrupt (**CFIE1**) is enabled, this bit is set 1 by hardware.Writing 1 to clear this bit. Reads 0: Capture channel 1 interrupt is not pending. Reads 1: Capture channel 1 interrupt is pending. Writes 0: no effect. Writes 1: Clear capture channel 1 interrupt status.
2	R/W1C	0x0	CRIS1 Capture channel 1 rising lock interrupt status. When capture channel 1 captures rising edge, if capture channel 1 rise lock interrupt (**CRIE1**) is enabled, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: Capture channel 1 interrupt is not pending. Reads 1: Capture channel 1 interrupt is pending. Writes 0: no effect. Writes 1: Clear capture channel 1 interrupt status.
1:0	/	/	/

9.9.6.5. 0x0020 PWM01 Clock Configuration Register (Default Value: 0x0000_0000)

Offset:0x0020			**Register Name: PCCR01**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8:7	R/W	0x0	PWM01_CLK_SRC Select PWM01 clock source 00: OSC24M 01: APB1 Others: Reserved
6	R/W	0x0	PWM01_CLK_SRC_BYPASS_TO_PWM1 Bypass PWM01 clock source to PWM1 output 0: not bypass 1: bypass
5	/	/	/
4	R/W	0x0	PWM01_CLK_GATING Gating clock for PWM01 0: Mask 1: Pass
3:0	R/W	0x0	PWM01_CLK_DIV_M PWM01 clock divide M 0000: /1 0001: /2 0010: /4 0011: /8

			0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 others: Reserved

9.9.6.6. 0x0024 PWM23 Clock Configuration Register (Default Value: 0x0000_0000)

Offset:0x0024			Register Name: PCCR23
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8:7	R/W	0x0	PWM23_CLK_SRC_SEL Select PWM23 clock source 00: OSC24M 01: APB1 Others: Reserved
6	R/W	0x0	PWM23_CLK_SRC_BYPASS_TO_PWM3 Bypass PWM23 clock source to PWM3 output 0: not bypass 1: bypass
5	R/W	0x0	PWM23_CLK_SRC_BYPASS_TO_PWM2 Bypass PWM23 clock source to PWM2 output 0: not bypass 1: bypass
4	R/W	0x0	PWM23_CLK_GATING Gating clock for PWM23 0: Mask 1: Pass
3:0	R/W	0x0	PWM23_CLK_DIV_M PWM23 clock divide M 0000: /1 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 others: /

9.9.6.7. 0x0028 PWM45 Clock Configuration Register (Default Value: 0x0000_0000)

Offset:0x0028			Register Name: PCCR45
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8:7	R/W	0x0	PWM45_CLK_SRC_SEL Select PWM45 clock source 00: OSC24M 01: APB1 Others: Reserved
6	/	/	/
5	R/W	0x0	PWM45_CLK_SRC_BYPASS_TO_PWM4 Bypass PWM45 clock source to PWM4 output 0: not bypass 1: bypass
4	R/W	0x0	PWM45_CLK_GATING Gating clock for PWM45 0: Mask 1: Pass
3:0	R/W	0x0	PWM45_CLK_DIV_M PWM45 clock divide M 0000: /1 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 others: /

9.9.6.8. 0x0030 PWM01 Dead Zone Control Register (Default Value: 0x0000_0000)

Offset:0x0030			Register Name: PDZCR01
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:8	R/W	0x0	PDZINTV01 PWM01 Dead Zone interval value
7:1	/	/	/
0	R/W	0x0	PWM01_DZ_EN PWM01 Dead Zone enable 0: Dead Zone disable 1: Dead Zone enable

9.9.6.9. 0x0034 PWM23 Dead Zone Control Register (Default Value: 0x0000_0000)

Offset:0x0034			Register Name: PDZCR23
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:8	R/W	0x0	PWM23_DZ_INTV PWM23 Dead Zone interval value
7:1	/	/	/
0	R/W	0x0	PWM23_DZ_EN PWM23 Dead Zone enable 0: Dead Zone disable 1: Dead Zone enable

9.9.6.10. 0x0038 PWM45 Dead Zone Control Register (Default Value: 0x0000_0000)

Offset:0x0038			Register Name: PDZCR45
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:8	R/W	0x0	PWM45_DZ_INTV PWM45 Dead Zone interval value
7:1	/	/	/
0	R/W	0x0	PWM45_DZ_EN PWM45 Dead Zone enable 0: Dead Zone disable 1: Dead Zone enable

9.9.6.11. 0x0040 PWM Enable Register (Default Value: 0x0000_0000)

Offset:0x0040			Register Name: PER
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4	R/W	0x0	PWM4_EN When enable PWM, the 16-bit up-counter starts working and PWM channel4 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable
3	R/W	0x0	PWM3_EN When enable PWM, the 16-bit up-counter starts working and PWM channel3 is permitted to output PWM waveform. 0: PWM disable

			1: PWM enable
2	R/W	0x0	PWM2_EN When enable PWM, the 16-bit up-counter starts working and PWM channel2 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable
1	R/W	0x0	PWM1_EN When enable PWM, the 16-bit up-counter starts working and PWM channel1 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable
0	/	/	/

9.9.6.12. 0x0044 Capture Enable Register (Default Value: 0x0000_0000)

Offset: 0x0044			**Register Name: CER**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4	R/W	0x0	CAP4_EN When enable capture function, the 16-bit up-counter starts working and capture channel4 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable
3	R/W	0x0	CAP3_EN When enable capture function, the 16-bit up-counter starts working and capture channel3 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable
2	R/W	0x0	CAP2_EN When enable capture function, the 16-bit up-counter starts working and capture channel2 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable
1	R/W	0x0	CAP1_EN When enable capture function, the 16-bit up-counter starts working and capture channel1 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable
0	/	/	/

9.9.6.13. 0x0060 + N*0x20 PWM Control Register (Default Value: 0x0000_0000)

Offset:0x0060+0x0+N*0x20(N=1~4)	**Register Name: PCR**

Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11	R	0x0	PWM_PERIOD_RDY PWM Period Register Ready 0: PWM period register is ready to write 1: PWM period register is busy
10	R/W1S	0x0	PWM_PUL_START PWM Pulse Output Start 0: No effect 1: Output pulse for PWM_CYCLE_NUM+1 After finishing configuration for outputting pulse, set this bit once and then PWM would output waveform. After the waveform is finished, the bit will be cleared automatically.
9	R/W	0x0	PWM_MODE PWM Output Mode Select 0: Cycle mode 1: Pulse mode
8	R/W	0x0	PWM_ACT_STA PWM Active State 0: Low Level 1: High Level
7:0	R/W	0x0	PWM_PRESCAL_K PWM pre-scale K, actual pre-scale is (K+1) K = 0, actual pre-scale: 1 K = 1, actual pre-scale: 2 K = 2, actual pre-scale: 3 K = 3, actual pre-scale: 4 ..... K = 255, actual pre-scale: 256

9.9.6.14. 0x0064 + N*0x20 PWM Period Register (Default Value: 0x0000_0000)

Offset:0x0060+0x04+N*0x20(N=1~4)			Register Name: PPR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	PWM_ENTIRE_CYCLE Number of the entire cycles in the PWM clock 0: 1 cycle 1: 2 cycles … N: N+1 cycles If the register need to be modified dynamically, the PCLK should be faster than the PWM CLK.
15:0	R/W	0x0	PWM_ACT_CYCLE Number of the active cycles in the PWM clock

			0: 0 cycle 1: 1 cycle … N: N cycles

9.9.6.15. 0x0068 + N*0x20 PWM Counter Register (Default Value: 0x0000_0000)

Offset:0x0060+0x08+N*0x20(N=1~4)			Register Name: PCNTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R	0x0	PWM_COUNTER_STATUS On PWM output or capture input, reading this register could get the current value of the PWM 16-bit up-counter.

9.9.6.16. 0x006C + N*0x20 PWM Capture Control Register (Default Value: 0x0000_0000)

Offset:0x0060+0x0C+N*0x20(N=1~4)			Register Name: CCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W1C	0x0	CRLF When capture channel captures rising edge, the 16-bit up-counter’s current value is latched to CRLR and then this bit is set to 1 by hardware. Write 1 to clear this bit.
1	R/W1C	0x0	CFLF When capture channel captures falling edge, the 16-bit up-counter’s current value is latched to CFLR and then this bit is set to 1 by hardware. Write 1 to clear this bit.
0	R/W	0x0	CAPINV Inversing the signal inputted from capture channel before capture channel's 16-bit counter. 0: not inverse 1: inverse

9.9.6.17. 0x0070 + N*0x20 PWM Capture Rise Lock Register (Default Value: 0x0000_0000)

Offset:0x0060+0x10+N*0x20(N=1~4)			Register Name: CRLR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R	0x0	When capture channel captures rising edge, the 16-bit up-counter’s current value is latched to this register.

9.9.6.18. 0x0074 + N*0x20 PWM Capture Fall Lock Register (Default Value: 0x0000_0000)

Offset:0x0060+0x14+N*0x20(N=1~4)			Register Name: CFLR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R	0x0	When the capturing channel captures the falling edge, the current value of the 16-bit up-counter is latched to the register.

9.10. TSC

9.10.1. Overview

The transport stream controller(TSC) is responsible for de-multiplexing and pre-processing the input multimedia data defined in ISO/IEC 13818-1.

The transport stream controller receives multimedia data stream from SSI(Synchronous Serial Port)/SPI(Synchronous Parallel Port) inputs and de-multiplexing the data into Packets by PID (Packet Identify). Before the Packet to be store to memory by DMA, it can be pre-processed by the Transport Stream Descrambler.

The transport stream controller can be used for almost all multi-media application cases, for example: DVB Set top Box, IPTV, Streaming-media Box, multi-media players and so on.

Features:

- Supports SPI/SSI interface,interface timing parameters are configurable
- 32 channels PID filter for each TSF
- Supports multiple transport stream packet (188, 192, 204) format
- Hardware packet synchronous byte error detecting
- Hardware PCR packet detecting
- 64x16-bits FIFO for TSG, 64x32-bits FIFO for TSF
- Configurable SPI transport stream generator for streams in DRAM memory
- Supports DVB-CSA V1.1, DVB-CSA V2.1 Descrambler

9.10.2. Block Diagram

Figure 9-58 shows a block diagram of the TSC.

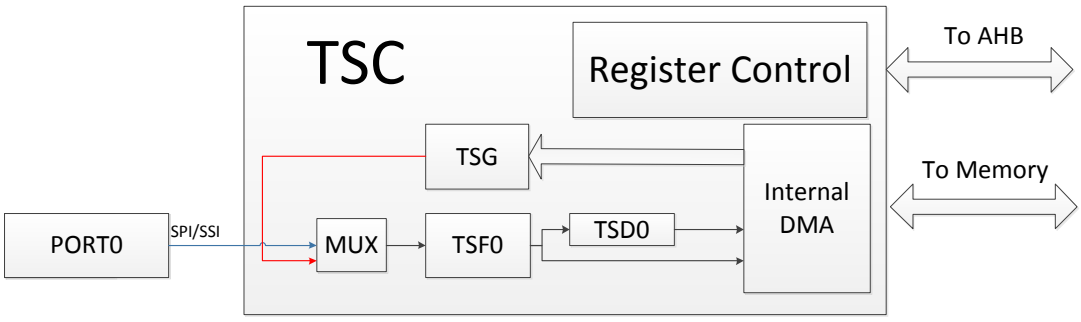

Figure 9- 58. TSC Block Diagram

TSC – TS Controller; TSF – TS Filter; TSD – TS Descrambler; TSG – TS Generator

9.10.3. Operations and Functional Descriptions

9.10.3.1. External Signals

Table 9-26 describes the external signals of TSC.

Table 9- 26. TSC External Signals

Signal	Description	Type
TS_CLK	Clock of SPI/SSI data input	I
TS_ERR	Error indicate	I
TS_SYNC	Packet sync (or Start flag) for TS packet	I
TS_DVLD	Data valid flag for TS data input	I
TS_D[7:0]	TS data input Data[7:0] are used in SPI mode; only Data[0] is used in SSI mode.	I

9.10.3.2. Clock Sources

The following table describes the clock sources of TSC.

Table 9- 27. TSC Clock Sources

Clock Sources	Description
OSC24M	24 MHz Crystal
PLL_PERIPH0(1X)	Peripheral Clock, the default value is 600 MHz

9.10.3.3. Timing Diagram

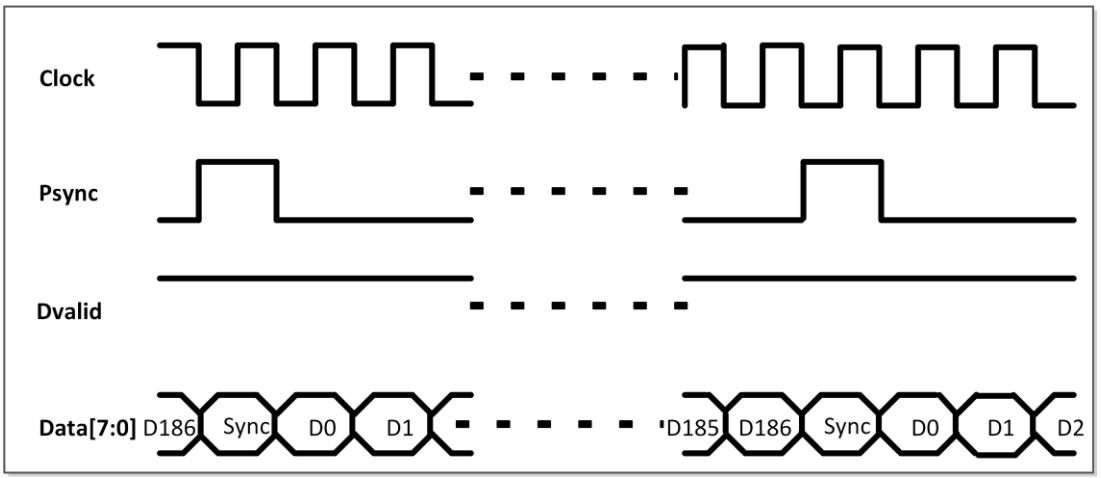

Figure 9- 59. Input Timing for SPI Mode

(CLOCK = Rising Edge, PSYNC = High Active, DVALID = High Active, Packet Size = 188 Bytes)

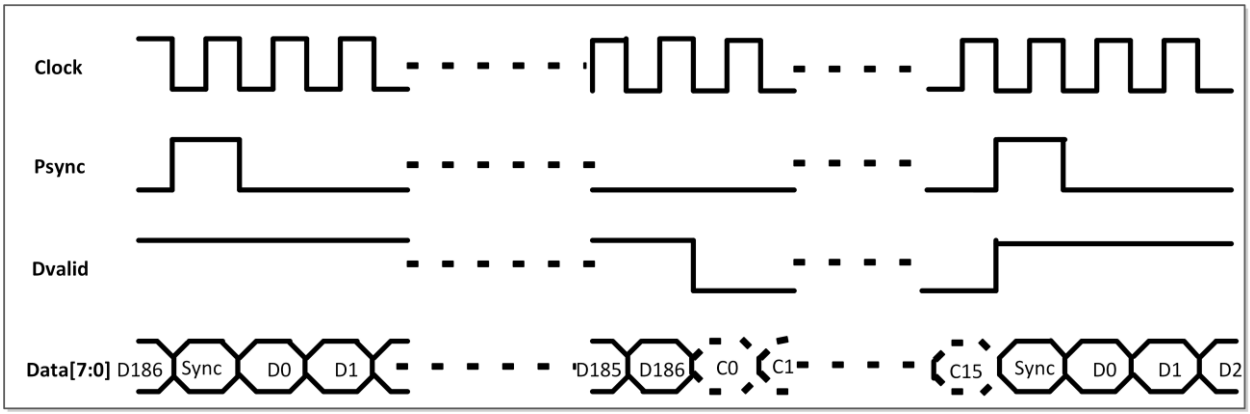

Figure 9- 60. Alternative Input Timing for SPI Mode

(CLOCK = Rising Edge, PSYNC = High Active, DVALID = High Active, Packet Size = 188 Bytes)

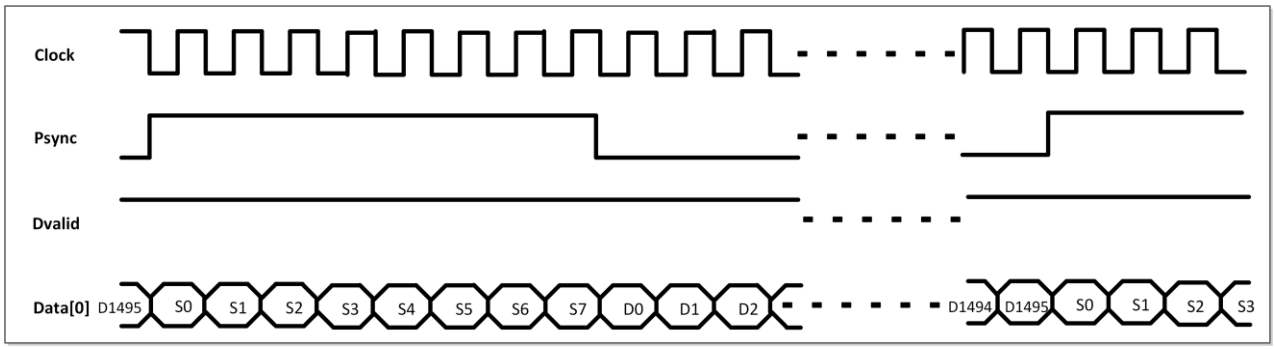

Figure 9- 61. Alternative Input Timing for SSI Mode

(CLOCK = Rising Edge, PSYNC = High Active, DVALID = High Active, Packet Size = 188 Bytes)

9.10.4. Programming Guidelines

9.10.4.1. Initialization

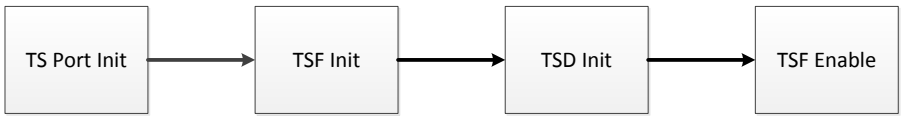

Figure 9- 62. TSC Initialization

The PID,DMA ADDR,DMA SIZE,Write Pointer,Read Pointer Register for TSF must clear to 0 first after power-up.

9.10.4.2. PID Changing

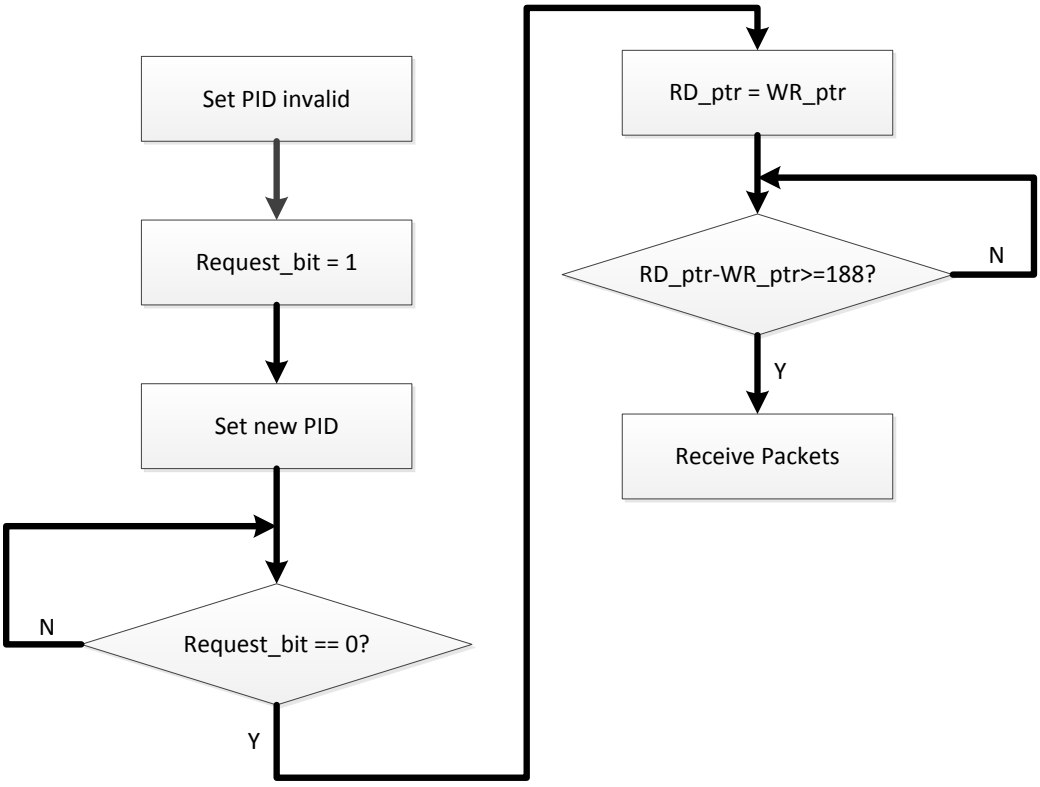

Figure 9- 63. PID Changing

NOTE

Request_bit is the bit8 of the TSF Control and Status Register.

9.10.5. Register List

Module Name	Base Address
TSC	0x05060000
TSG	0x05060040
TSF	0x05060100
TSD	0x05060180

Register Name	Offset	Description
TSC		
TSC_PCTLR	TSC + 0x10	TSC Port Control Register
TSC_PPARR	TSC + 0x14	TSC Port Parameter Register
TSC_TSFMUXR	TSC + 0x20	TSC TSF Input Multiplex Control Register
TSC_OUTMUXR	TSC + 0x28	TSC Port Output Multiplex Control Register
TSG		

TSG_CTLR	TSG + 0x00	TSG Control Register
TSG_PPR	TSG + 0x04	TSG Packet Parameter Register
TSG_STAR	TSG + 0x08	TSG Status Register
TSG_CCR	TSG + 0x0C	TSG Clock Control Register
TSG_BBAR	TSG + 0x10	TSG Buffer Base Address Register
TSG_BSZR	TSG + 0x14	TSG Buffer Size Register
TSG_BPR	TSG + 0x18	TSG Buffer Pointer Register
TSF		
TSF_CTLR	TSF + 0x00	TSF Control Register
TSF_PPR	TSF + 0x04	TSF Packet Parameter Register
TSF_STAR	TSF + 0x08	TSF Status Register
TSF_DIER	TSF + 0x10	TSF DMA Interrupt Enable Register
TSF_OIER	TSF + 0x14	TSF Overlap Interrupt Enable Register
TSF_DISR	TSF + 0x18	TSF DMA Interrupt Status Register
TSF_OISR	TSF + 0x1C	TSF Overlap Interrupt Status Register
TSF_PCRCR	TSF + 0x20	TSF PCR Control Register
TSF_PCRDR	TSF + 0x24	TSF PCR Data Register
TSF_CENR	TSF + 0x30	TSF Channel Enable Register
TSF_CPER	TSF + 0x34	TSF Channel PES Enable Register
TSF_CDER	TSF + 0x38	TSF Channel Descramble Enable Register
TSF_CINDR	TSF + 0x3C	TSF Channel Index Register
TSF_CCTLR	TSF + 0x40	TSF Channel Control Register
TSF_CSTAR	TSF + 0x44	TSF Channel Status Register
TSF_CCWIR	TSF + 0x48	TSF Channel CW Index Register
TSF_CPIDR	TSF + 0x4C	TSF Channel PID Register
TSF_CBBAR	TSF + 0x50	TSF Channel Buffer Base Address Register
TSF_CBSZR	TSF + 0x54	TSF Channel Buffer Size Register
TSF_CBWPR	TSF + 0x58	TSF Channel Buffer Write Pointer Register
TSF_CBRPR	TSF + 0x5C	TSF Channel Buffer Read Pointer Register
TSD		
TSD_CTLR	TSD + 0x00	TSD Control Register
TSD_STAR	TSD + 0x04	TSD Status Register
TSD_CWIR	TSD + 0x1C	TSD Control Word Index Register
TSD_CWR	TSD + 0x20	TSD Control Word Register

9.10.6. Register Description

9.10.6.1. 0x0010 TSC Port Control Register(Default Value: 0x0000_0000)

Offset: TSC+0x10			**Register Name: TSC_PCTLR**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	TSInPort0Ctrl

			TS Input Port0 Control 0 : SPI 1 : SSI

9.10.6.2. 0x0014 TSC Port Parameter Register(Default Value: 0x0000_0000)

Offset: TSC+0x14			Register Name:TSC_PPARR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4	R/W	0x0	TS Input Port0 SSI Data Order 0: MSB first for one byte data 1: LSB first for one byte data
3	R/W	0x0	TS Input Port0 CLOCK Signal Polarity 0: Rise edge capturing 1: Fall edge capturing
2	R/W	0x0	TS Input Port0 ERROR Signal Polarity 0: High level active 1: Low level active
1	R/W	0x0	TS Input Port0 DVALID Signal Polarity 0: High level active 1: Low level active
0	R/W	0x0	TS Input Port0 PSYNC Signal Polarity 0: High level active 1: Low level active

9.10.6.3. 0x0020 TSC TSF Input Multiplex Control Register(Default Value: 0x0000_0000)

Offset: TSC+0x20			Register Name: TSC_TSFMUXR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3:0	R/W	0x0	TSF0InputMuxCtrl TSF0 Input Multiplex Control 0000: Data from TSG 0001: Data from TS IN Port0 Others : Reserved

9.10.6.4. 0x0030 TSC Interrupt Status Register(Default Value: 0x0000_0000)

Offset: TSC+0x30			Register Name:TSC_INT_STATUS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/

16	R	0x0	TSG Interrupt Global Status When all TSG interrupt status bits are cleared ,this bit will be cleared by hardware.
15:1	/	/	/
0	R	0x0	TSF0 Interrupt Global Status When all TSF0 interrupt status bits are cleared ,this bit will be cleared by hardware.

9.10.6.5. 0x0000 TSG Control and Status Register(Default Value: 0x0000_0000)

Offset: TSG+0x00			Register Name: TSG_CSR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:24	R	0x0	TSGSts Status for TS Generator 00: IDLE state 01: Running state 10: PAUSE state Others: Reserved
23:10	/	/	/
9	R/W	0x0	TSGLBufMode Loop Buffer Mode When set to '1', the TSG external buffer is in loop mode.
8	R/W	0x0	TSGSyncByteChkEn Sync Byte Check Enable Enable/Disable check SYNC byte for receiving new packet 0: Disable 1: Enable If enable check SYNC byte and an error SYNC byte is receiver, TS Generator would come into PAUSE state. If the correspond interrupt is enabled, the interrupt would happen.
7:3	/	/	/
2	R/W	0x0	TSGPauseBit Pause Bit for TS Generator Write '1' to pause TS Generator. TS Generator would stop fetch new data from DRAM. After finished this operation, this bit will clear to zero by hardware. In PAUSE state, write '1' to resume this state.
1	R/W	0x0	TSGStopBit Stop Bit for TS Generator Write '1' to stop TS Generator. TS Generator would stop fetch new data from DRAM. The data already in its FIFO should be sent to TS filter. After finished this operation, this bit will clear to zero by hardware.
0	R/W	0x0	TSGStartBit Start Bit for TS Generator

			Write '1' to start TS Generator. TS Generator would fetch data from DRAM and generate SPI stream to TS filter. This bit will clear to zero by hardware after TS Generator is running.

9.10.6.6. 0x0004 TSG Packet Parameter Register(Default Value: 0x0047_0000)

Offset: TSG+0x04			Register Name: TSG_PPR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:16	R/W	0x47	SyncByteVal Sync Byte Value This is the value of sync byte used in the TS Packet.
15:8	/	/	/
7	R/W	0x0	SyncBytePos Sync Byte Position 0: The 1st byte position 1: The 5th byte position This bit is only used for 192 bytes packet size.
6:2	/	/	/
1:0	R/W	0x0	PktSize Packet Size Byte Size for one TS packet 0: 188 bytes Others: Reserved

9.10.6.7. 0x0008 TSG Interrupt Enable and Status Register(Default Value: 0x0000_0000)

Offset: TSG+0x08			Register Name: TSG_IESR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/
19	R/W	0x0	TSGEndIE TS Generator (TSG) End Interrupt Enable 0: Disable 1: Enable If set this bit, the interrupt would assert to CPU when all data in external DRAM are sent to TS PID filter.
18	R/W	0x0	TSGFFIE TS Generator (TSG) Full Finish Interrupt Enable 0: Disable 1: Enable
17	R/W	0x0	TSGHFIE TS Generator (TSG) Half Finish Interrupt Enable

			0: Disable 1: Enable
16	R/W	0x0	TSGErrSyncByteIE TS Generator (TSG) Error Sync Byte Interrupt Enable 0: Disable 1: Enable
15:4	/	/	/
3	R/W1C	0x0	TSGEndSts TS Generator (TSG) End Status Write '1' to clear.
2	R/W1C	0x0	TSGFFSts TS Generator (TSG) Full Finish Status Write '1' to clear.
1	R/W1C	0x0	TSGHFSts TS Generator (TSG) Half Finish Status Write '1' to clear.
0	R/W1C	0x0	TSGErrSyncByteSts TS Generator (TSG) Error Sync Byte Status Write '1' to clear.

9.10.6.8. 0x000C TSG Clock Control Register(Default Value: 0x0000_0000)

Offset: TSG+0x0C			Register Name: TSG_CCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	TSGCDF_N TSG Clock Divide Factor (N) The Numerator part of TSG Clock Divisor Factor.
15:0	R/W	0x0	TSGCDF_D TSG Clock Divide Factor (D) The Denominator part of TSG Clock Divisor Factor. Frequency of output clock: Fo = (Fi*(N+1))/(8*(D+1)). Fi is the input special clock of TSC, and D must not less than N.

9.10.6.9. 0x0010 TSG Buffer Base Address Register(Default Value: 0x0000_0000)

Offset: TSG+0x10			Register Name: TSG_BBAR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	RW	0x0	TSGBufBase Buffer Base Address This value is a start address of TSG buffer.

			NOTE **This value should be 4-word (16 bytes) align, and the lowest 4-bit of this value should be zero.**

9.10.6.10. 0x0014 TSG Buffer Size Register(Default Value: 0x0000_0000)

Offset: TSG+0x14			Register Name:TSG_BSZR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:0	R/W	0x0	TSGBufSize Data Buffer Size for TS Generator It is in byte unit. The size should be 4-word (16 bytes) align, and the lowest 4 bits should be zero.

9.10.6.11. 0x0018 TSG Buffer Pointer Register(Default Value: 0x0000_0000)

Offset: TSG+0x18			Register Name: TSG_BPR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:0	R	0x0	TSGBufPtr Data Buffer Pointer for TS Generator Current TS generator data buffer read pointer (in byte unit)

9.10.6.12. 0x0000 TSF Control and Status Register(Default Value: 0x0000_0000)

Offset: TSF+0x00			Register Name: TSF_CSR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8	R/WAC	0x0	Channel Change PID Request This bit is used to send a request to hardware for changing the PID of the channel.It will be cleared by hardware when the channel changing finish. Writing '0' has no effect.
7:3	/	/	/
2	R/W	0x0	TSF Enable 00: Disable TSF Input 01: Enable TSF Input
1	/	/	/
0	R/WAC	0x0	TSFGSRF TSF Global Soft Reset

			Writing '1' by software will reset all status and state machine of TSF. And it is cleared by hardware after finish reset. Writing '0' by software has no effect.

9.10.6.13. 0x0004 TSF Packet Parameter Register(Default Value: 0x0047_0000)

Offset: TSF+0x04			Register Name: TSF_PPR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	LostSyncThd Lost Sync Packet Threshold It is used for packet sync lost by checking the value of sync byte.
27:24	R/W	0x0	SyncThd Sync Packet Threshold It is used for packet sync by checking the value of sync byte.
23:16	R/W	0x47	SyncByteVal Sync Byte Value This is the value of sync byte used in the TS Packet.
15:10	/	/	/
9:8	R/W	0x0	SyncMthd Packet Sync Method 00: By PSYNC signal 01: By sync byte 10: By both PSYNC and Sync Byte 11: Reserved
7:2	/	/	/
1:0	R/W	0x0	PktSize Packet Size Byte size for one TS packet 00: 188 bytes 01: 192 bytes 10: 204 bytes 11: Reserved

9.10.6.14. 0x0008 TSF Interrupt Enable and Status Register(Default Value: 0x0000_0000)

Offset: TSF+0x08			Register Name: TSF_IESR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/
19	R/W	0x0	TSFFOIE TS PID Filter (TSF) Internal FIFO Overrun Interrupt Enable 0: Disable 1: Enable

18	R/W	0x0	TSFPPDIE TS PCR Packet Detect Interrupt Enable 0: Disable 1: Enable
17	R/W	0x0	TSFCOIE TS PID Filter (TSF) Channel Overlap Interrupt Global Enable 0: Disable 1: Enable
16	R/W	0x0	TSFCDIE TS PID Filter (TSF) Channel DMA Interrupt Global Enable 0: Disable 1: Enable
15:4	/	/	/
3	R/W1C	0x0	TSFFOIS TS PID Filter (TSF) Internal FIFO Overrun Status Write '1' to clear.
2	R/W1C	0x0	TSFPPDIS TS PCR Packet Found Status When it is '1', one TS PCR Packet is found. Write '1' to clear.
1	R	0x0	TSFCOIS TS PID Filter (TSF) Channel Overlap Status It is global status for 32 channel. It would clear to zero after all channels status bits are cleared.
0	R	0x0	TSFCDIS TS PID Filter (TSF) Channel DMA Status It is global status for 32 channel. It would clear to zero after all channels status bits are cleared.

9.10.6.15. 0x0010 TSF DMA Interrupt Enable Register(Default Value: 0x0000_0000)

Offset: TSF+0x10			Register Name: TSF_DIER
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	DMAIE DMA Interrupt Enable DMA interrupt enable bits for channel 0~31.

9.10.6.16. 0x0014 TSF Overlap Interrupt Enable Register(Default Value: 0x0000_0000)

Offset: TSF+0x14			Register Name: TSF_OIER
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	OLPIE Overlap Interrupt Enable

			Overlap interrupt enable bits for channel 0~31.

9.10.6.17. 0x0018 TSF DMA Interrupt Status Register(Default Value: 0x0000_0000)

Offset: TSF+0x18			Register Name: TSF_DISR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W1C	0x0	DMAIS DMA Interrupt Status DMA interrupt Status bits for channel 0~31. Set by hardware, and can be cleared by software writing '1'. When both these bits and the corresponding DMA Interrupt Enable bits set, the TSF interrupt will generate.

9.10.6.18. 0x001C TSF Overlap Interrupt Status Register(Default Value: 0x0000_0000)

Offset: TSF+0x1C			Register Name: TSF_OISR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W1C	0x0	OLPIS Overlap Interrupt Status Overlap interrupt Status bits for channel 0~31. Set by hardware, and can be cleared by software writing '1'. When both these bits and the corresponding Overlap Interrupt Enable bits set, the TSF interrupt will generate.

9.10.6.19. 0x0020 TSF PCR Control Register(Default Value: 0x0000_0000)

Offset: TSF+0x20			Register Name: TSF_PCRCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	PCRDE PCR Detecting Enable 0: Disable 1: Enable
15:13	/	/	/
12:8	R/W	0x0	PCRCIND Channel Index m for Detecting PCR packet (m from 0 to 31)
7:1	/	/	/
0	R	0x0	PCRLSB PCR Contest LSB 1 bit--PCR[0].

9.10.6.20. 0x0024 TSF PCR Data Register(Default Value: 0x0000_0000)

Offset: TSF+0x24			Register Name: TSF_PCRDR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	PCRMSB PCR Data High 32 bits--PCR[33:1].

9.10.6.21. 0x0030 TSF Channel Enable Register(Default Value: 0x0000_0000)

Offset: TSF+0x30			Register Name: TSF_CENR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	FILTEREN Filter Enable for Channel 0~31 0: Disable 1: Enable From Disable to Enable, internal status of the corresponding filter channel will be reset.

9.10.6.22. 0x0034 TSF Channel PES Enable Register(Default Value: 0x0000_0000)

Offset: TSF+0x34			Register Name: TSF_CPER
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	PESEN PES Packet Enable for Channel 0~31 0: Disable 1: Enable These bits should not be changed during the corresponding channel enable.

9.10.6.23. 0x0038 TSF Channel Descramble Enable Register(Default Value: 0x0000_0000)

Offset: TSF+0x38			Register Name: TSF_CDER
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	DESCEN Descramble Enable for Channel 0~31 0: Disable 1: Enable These bits should not be changed during the corresponding channel enable.

9.10.6.24. 0x003C TSF Channel Index Register(Default Value: 0x0000_0000)

Offset: TSF+0x3C			Register Name: TSF_CINDR
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PES_DESCRAM_ENABLE 1: enable 0: disable
30:5	/	/	/
4:0	R/W	0x0	CHIND Channel Index This value is the channel index for channel private registers access. Range is from 0x00 to 0x1f. Address range of channel private registers is 0x40~0x7f.

9.10.6.25. 0x0048 TSF Channel CW Index Register(Default Value: 0x0000_0000)

Offset: TSF+0x48			Register Name: TSF_CCWIR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2:0	R/W	0x0	CWIND Related Control Word Index Index to the control word used by this channel when Descramble Enable of this channel enable. This value is useless when the corresponding Descramble Enable is '0'.

9.10.6.26. 0x004C TSF Channel PID Register(Default Value: 0x1FFF_0000)

Offset: TSF+0x4C			Register Name: TSF_CPIDR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x1fff	PIDMSK Filter PID Mask for Channel
15:0	R/W	0x0	PIDVAL Filter PID value for Channel

9.10.6.27. 0x0050 TSF Channel Buffer Base Address Register(Default Value: 0x0000_0000)

Offset: TSF+0x50			Register Name: TSF_CBBAR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	TSFBufBAddr Data Buffer Base Address for Channel It is 4-word (16 bytes) align address. The LSB four bits should be zero.

9.10.6.28. 0x0054 TSF Channel Buffer Size Register(Default Value: 0x0000_0000)

Offset: TSF+0x54			Register Name: TSF_CBSZR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:24	R/W	0x0	CHDMAIntThd DMA Interrupt Threshold for Channel The unit is TS packet size. When received packet (has also stored in DRAM) size is beyond (>=) threshold value, the corresponding channel interrupt is generated to CPU. TSC should count the new received packet again, when exceed the specified threshold value, one new interrupt is generated again. 00: 1/2 data buffer packet size 01: 1/4 data buffer packet size 10: 1/8 data buffer packet size 11: 1/16 data buffer packet size
23:21	/	/	/
20:0	R/W	0x0	CHBufPktSz Data Buffer Packet Size for Channel The exact buffer size of buffer is N+1 bytes. The maximum buffer size is 2 MB. This size should be 4-word (16 bytes) aligned. The LSB four bits should be zero.

9.10.6.29. 0x0058 TSF Channel Buffer Write Pointer Register(Default Value: 0x0000_0000)

Offset: TSF+0x58			Register Name: TSF_CBWPR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20:0	R/W	0x0	BufWrPtr Data Buffer Write Pointer (in Bytes) This value is changed by hardware, when data is filled into buffer, this pointer is increased. And this pointer can be set by software, but it should not be changed by software during the corresponding channel is enabled.

9.10.6.30. 0x005C TSF Channel Buffer Read Pointer Register(Default Value: 0x0000_0000)

Offset: TSF+0x5C			Register Name: TSF_CBRPR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20:0	R/W	0x0	BufRdPtr

			Data Buffer Read Pointer (in Bytes) This pointer should be changed by software after the data of buffer is read.

9.10.6.31. 0x0000 TSD Control Register(Default Value: 0x0000_0000)

Offset: TSD+0x00			Register Name: TSD_CTLR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	TS Descramble Flag Clear 0: Clear 1: Not clear
15:2	/	/	/
1:0	R/W	0x0	DescArith Descramble Arithmetic 00: DVB CSA V1.1 01: DVB CSA V2.1 Others: Reserved

9.10.6.32. 0x001C TSD Control Word Index Register(Default Value: 0x0000_0000)

Offset: TSD+0x1C			Register Name: TSD_CWIR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6:4	R/W	0x0	CWI Control Word Index This value is the control index for control word access. Range is from 0x0 to 0x7.
3:0	R/W	0x0	CWII Control Word Internal Index 0000: Odd Control Word 1ST 32-bit, OCW[31:0]; 0001: Odd Control Word 2ND 32-bit, OCW[63:32]; 0100: Even Control Word 1ST 32-bit, ECW[31:0]; 0101: Even Control Word 2ND 32-bit, ECW[63:32]; Others: Reserved

9.10.6.33. 0x0020 TSD Control Word Register(Default Value: 0x0000_0000)

Offset: TSD+0x20			Register Name: TSD_CWR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	CWD Content of Control Word corresponding to the TSD_CWIR value

Contents

Figures

Chapter 10 Security System

10.1. Crypto Engine

10.1.1. Overview

The Crypto Engine(CE) module is one encryption/decryption algorithm accelerator. It supports kinds of symmetric, asymmetric, Hash, and RNG algorithms. There are two software interfaces for secure and non-secure world each. The software interface is simple for configuration, only setting interrupt control, task description address and load tag. Algorithm control information is written in memory by task descriptor, then CE automatically reads it when executing request. It supports parallel requests from 4 channels, and has an internal DMA controller to transfer data between CE and memory.

The CE has the following features:

- Symmetrical algorithm: AES, DES, 3DES, XTS
  - 128-, 192-, 256-bit key size for AES
  - ECB, CBC, CTR, CTS, OFB, CFB, CBC-MAC modes for AES
  - AES-CFB mode support CFB1, CFB8, CFB64, CFB128
  - AES-CTR supports CTR16, CTR32, CTR64, CTR128
  - ECB, CBC, CTR, CBC-MAC modes for DES/3DES
  - DES-CTR mode supports CTR16, CTR32, CTR64
  - 256-bit, 512-bit key for XTS
- Hash algorithm: MD5, SHA1, SHA224, SHA256, SHA384, SHA512, HMAC-SHA1, HMAC-SHA256
  - MD5, SHA, HMAC are padded using hardware, if not last package, input should aligned with computation block, namely 512bits or 1024bits
- Asymmetrical algorithm: RSA512/1024/2048/4096-bit, ECC160/224/256/384/521-bit
- 160-bit hardware PRNG with 175-bit seed. Output aligns with 5 words
- 256-bit hardware TRNG. Output aligns with 8 words
- Supports secure and non-secure interfaces respectively, each world issues task request through its own interface, does not know each other's existence
- Supports task chain mode for each request. Task or task chain are executed at request order
- Symmetric, asymmetric, HASH ctrl logics are separate, can handle task simultaneously. Symmetric logic can select instantiate 2 suits at implementation time
- 8 scatter group(sg) are supported for both input and output data. sg size is in units of word. DMA reads and writes data at word aligned
- DMA has multiple channel, each channel corresponds one suit of algorithms

10.1.2. Block Diagram

The following figure shows the block diagram of Crypto Engine.

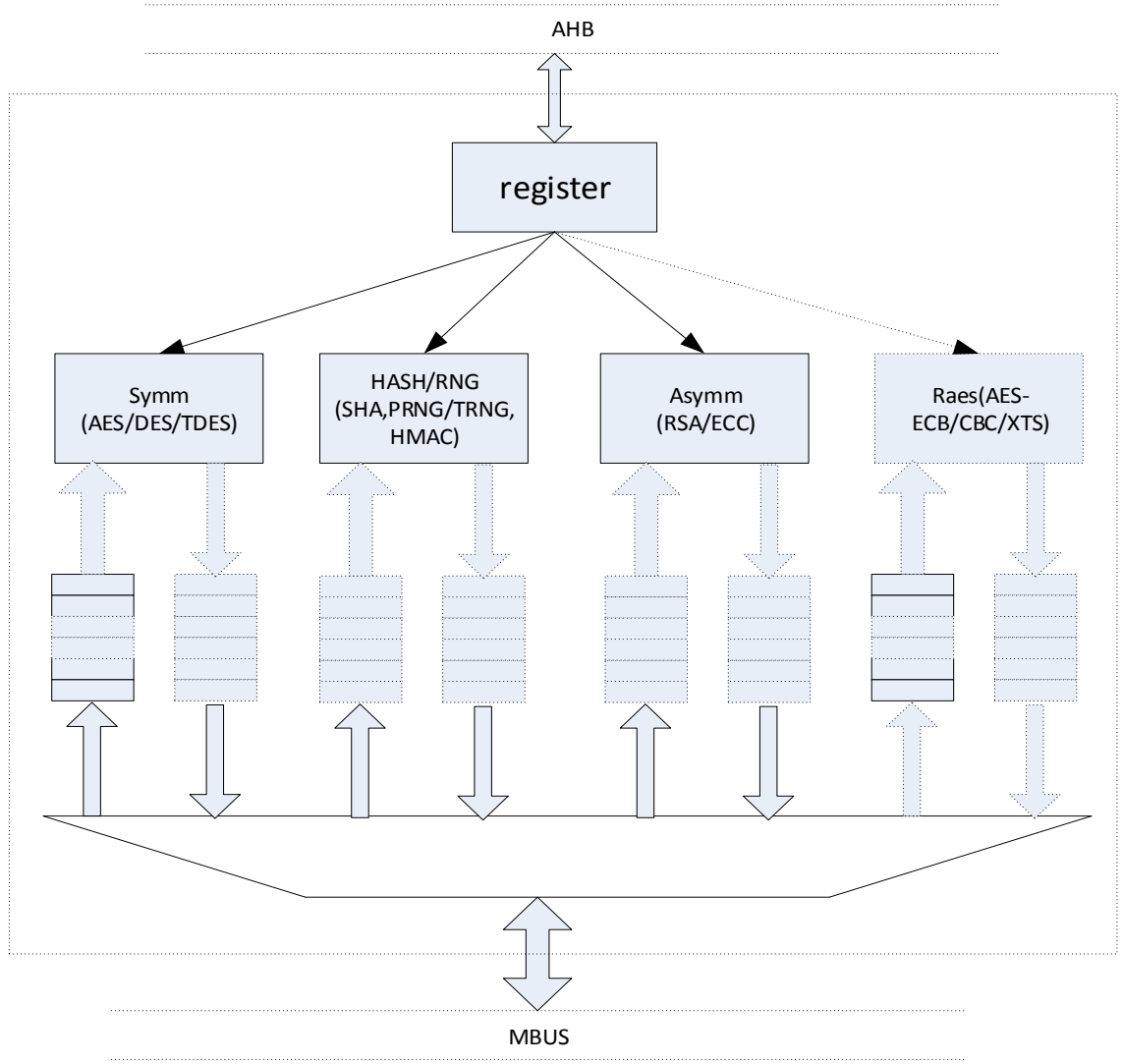

Figure 10- 1. CE Block Diagram

10.1.3. Operations and Functional Descriptions

10.1.3.1. Task Descriptor

Software make request through task descriptor, providing algorithm type, mode, key address, source/destination address and size, etc. The task descriptor is as follows.

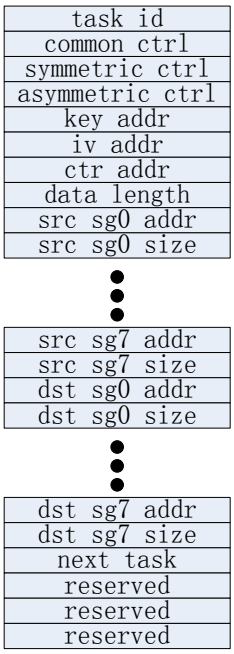

Figure 10- 2. Task Chaining

Task chaining id supports 0~3.

- The **key addr** field is address for the key of each algorithm, and for the total length address of HASH when last package, also for extension feature micro codes address. **Must be word address**.
- The **iv addr** field is address for the IV or modulus, or tweak value address for XTS. **Must be word address**.
- The **ctr addr** is address for the IV of next block, and for HMAC K1 address. **Must be word address**.
- The **src/dst sgX adr** field indicates 32-bit address for source and destination data. **Must be word address**.
- The **src/dst sgX size** field indicates the size for each sg respectively.
- The **next task** field should be set to 0 when no next task, else set to the descriptor of next task. **Must be word address**.
- **Reserved** is used for CSA CW address. **Must be word address**.

10.1.3.2. Task Descriptor Queue Common Control

Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	Interrupt enable for current task 0: disable interrupt 1: enable interrupt
30:25	/	/	/
24:17	R/W	0x0	cbc_mac_len The outcome bit length of CBC-MAC when in CBC-MAC mode.
16	R/W	0x0	IV mode IV mode for SHA1/SHA224/SHA256/SHA384/SHA512/MD5 or constants 0: use initial constants defined in FIPS-180 1: use input iv
15	R/W	0x0	Last HMAC plaintext 0: not the last HMAC plaintext package. Padding is not required. 1: the last HMAC plaintext package.Padding is required

14:9	/	/	/
8	R/W	0x0	OP DIR Algorithm Operation Direction 0: Encryption 1: Decryption
7	/	/	/
6:0	R/W	0x0	Algorithm Type 0x0: AES 0x1: DES 0x2: Triple DES (3DES) 0x10: MD5 0x11: SHA-1 0x12: SHA-224 0x13: SHA-256 0x14: SHA-384 0x15: SHA-512 0x16: HMAC-SHA1 0x17: HMAC-SHA256 0x1C: TRNG 0x1D: PRNG 0x20: RSA 0x21: ECC 0x30: RAES Others: Reserved

10.1.3.3. Task Descriptor Queue Symmetric Control

Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26	R/W	0x0	no_modk 0: have module key derivation function 1: no module key derivation function
25:24	/	/	/
23:20	R/W	0x0	KEY_SELECT key select for AES 0000: Select input CE_KEYx (Normal Mode) 0001: Select {SSK} 0010: Select {HUK} 0011: Select {RSSK}, used for decrypt HDCP key, EK, BSSK 0100-0111: Reserved 1000-1111: Select internal Key n (n from 0 to 7)
19:18	R/W	0x0	CFB_WIDTH For AES-CFB width 00: CFB1

			01: CFB8 10: CFB64 11: CFB128
17	R/W	0x0	PRNG_LD Load new 15bits key into lfsr for PRNG
16	R/W	0x0	AES CTS last package flag When setting to '1', it means this is the last package for AES-CTS mode(the size of the last package >128bit).
15:14	/	/	/
13	R/W	0x0	xts_last 0: not last block for XTS 1: last block for XTS
12	R/W	0x0	xts_first 0: not first block for XTS 1: first block for XTS
11:8	R/W	0x0	Operation Mode for Symmetric AES/DES/3DES/RAES Modes DES/3DES only supports ECB/CBC/CTR. RAES only supports ECB/CBC/XTS. 0000: Electronic Code Book (ECB) mode 0001: Cipher Block Chaining (CBC) mode 0010: Counter (CTR) mode 0011: CipherText Stealing (CTS) mode 0100: Output feedback (OFB)mode 0101: Cipher feedback (CFB)mode 0110: CBC-MAC mode 1001: XTS mode Other: Reserved
7:4	/	/	/
3:2	R/W	0x0	CTR WIDTH Counter width for CTR mode 00: 16-bit Counter 01: 32-bit Counter 10: 64-bit Counter 11: 128-bit Counter
1:0	R/W	0x0	AES KEY SIZE 00: 128-bit 01: 192-bit 10: 256-bit 11: Reserved

10.1.3.4. Task Descriptor Queue Asymmetric Control

Bit	Read/Write	Default/Hex	Description
31:21	/	/	/

20:16	R/W	0x0	PKC algorithm mode For modular computation: 00000: modular exponent(RSA) 00001: modular add 00010: modular minus 00011: modular multiplication others: reserved For ECC: 00000: point add 00001: point double 00010: point multiplication 00011: point verification 00100: encryption 00101: decryption 00110: sign 00111: sign verify others: reserved
15:8	/	/	/
7:0	R/W	0x0	Asymmetric algorithm operation width field It indicates how much width this request apply, as words.

10.1.3.5. Task Request

Basically, there are 4 steps for one task handling from software.

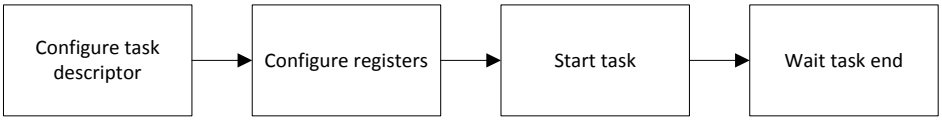

Figure 10- 3. Task Request Process

Step1: Software should configure task descriptor in memory, including all fields in descriptor. Channel id corresponds to one channel in CE. According to algorithm type, software should set the fields in common control, symmetric control, asymmetric control, then provide key/iv/ctr address and the data length of this task. Source and destination sg address and size are set based on upper application. If there is another task concatenating after this task, then set its descriptor address at next descriptor field.

Step 2: Software should set registers, including task descriptor address, interrupt control.

Step 3: Software reads load register to ensure that the bit0 is zero, then starts request by pulling up the bit0 of the load register.

Step 4: Wait task end.

10.1.3.6. Data Length Setting

Data length field in task descriptor has different meaning for different algorithms.
For HASH algorithm, data length field indicates valid source data bit number, for others indicates source data byte number. The data length of HASH should be 512/1024-bit aligned if current request is not the last data block, because of hardware padding.
For PRNG, data length should be 5 words aligned.
For TRNG, data length should be 8 words aligned.
Data size in source and destination sg is as words, whose value should corresponds with data length field, or else CE will report error and stop execution.

10.1.3.7. Security Operation

When CPU issues request to CE module, CE module will save the secure mode of CPU. When executing this request, this state bit works as access tag for inner and system resource. For HUK/RSSK/SSK from SID, only secure mode can access, or else these keys will be used as 0. For access to SID and keysram module through AHB bus, only secure mode can success, or else will read 0 or can not write. When issuing MBUS read and write requests, CE will use send this secure mode bit to BUS, so secure request can access secure and non-secure space, but non-secure request only can access non-secure space.

10.1.3.8. Task Parallel

Algorithms are divided into 3 types: symmetric, HASH/RNG, asymmetric. Each type has a task queue with 8 elements for requests. Tasks in each queue are handled in sequence. Among these 3 types, task request and complete time are not sure. If one type uses the outcome of another type, software should make sure that start one type after another type is finished.

CE supports 4 channels in each world, and 3 suits algorithm type which can run in parallel. When software issues request, it first checks if load bit is low, which means software can request. If load bit is high, which means last request is not registered by CE, software should wait until load bit is low. If software makes several requests with the same type, these tasks will be executed in request sequence. If software makes several requests with different types, these tasks will be executed in parallel. Because parallel tasks would finish out of order, software should make different type request with different channel id, which results in generating different interrupt status bit.

10.1.3.9. PKC Microcode

PKC module supports RSA, ECC asymmetric algorithms in the form of microcode. It implements basic modular add, minus, multiplication, point add, point double, and logic computing, etc. Complete RSA/ECC encryption, decryption, sign, verify are implemented with these microcode.

Asymmetric algorithms RSA/ECC are implemented as microcode in PKC module. Asymmetric encryption, decryption, sign, verify operations are composed with certain fixed microcode with hardware.

10.1.3.10. PKC Configuration

Before starting PKC, task description must be configured. Parameters to PKC are assigned to source sg, outcome is put to destination sg.
For RSA, parameters should be at the order of key, modulus, plaintext.
For ECC point add P2 = P0 + P1, parameters should be at the order of p, P0x, P0y, P1x, P1y. Output is at the order of P2x, P2y.
For ECC point double P2 = 2*P0, parameters should be at the order of p, a, P0x, P0y. Output is at the order of P2x, P2y.
For ECC point multiplication P2 = k*P0, parameters should be at the order of p, k, a, P0x, P0y. Output is at the order of P2x, P2y.
For ECC point verification, parameters should be at the order of p, a, P0x, P0y, b. Output is 1 or 0.
For ECC encryption, parameters should be at the order of random k, p, a, Gx, Gy, Qx, Qy, m. Output is at the order of Rx, Ry, c.
For ECC decryption, parameters should be at the order of random k, p, a, Rx, Ry, c. Output is m.
For ECC signature, parameters should be at the order of random k, p, a, Gx, Gy, n, d, e. Output is at the order of r, s.
For ECC signature verification, parameters should be at the order of n, s, e, r, p, a, Gx, Gy, Qx, Qy, n, r. Output is 1 or 0.

10.1.3.11. Error Check

CE module includes error detection for task configuration, data computing error, and authentication invalid. When algorithm type in task description is read into module, CE will check if this type is supported through checking algorithm type field in common control. If type value is out of scope, CE will issue interrupt signal and set error state. Each type has certain input and output data size. After getting task descriptor, input size and output size configuration will be checked to avoid size error. If size configuration is wrong, CE will issue interrupt signal and set error state. To protect keys would be put into keysram from disclose, if request using RSSK is for AES decryption and destination address is not in keysram space, CE would not execute this task. It will issue interrupt signal and set error state.

10.1.3.12. Clock Requirement

Clock Name	Description	Requirement
ahb_clk	AHB bus clock	24 MHz ~ 200 MHz
m_clk	MBUS clk	24 MHz ~ 400 MHz
ce_clk	CE work clock	24 MHz ~ 300 MHz

10.1.4. Register List

Module Name	Base Address
CE_NS	0x01904000
CE_S	0x01904800

Register Name	Offset	Description
CE_TDA	0x0000	Task Descriptor Address
CE_ICR	0x0008	Interrupt Control Register
CE_ISR	0x000C	Interrupt Status Register
CE_TLR	0x0010	Task Load Register
CE_TSR	0x0014	Task Status Register
CE_ESR	0x0018	Error Status Register
CE_SCSA	0x0024	Symmetric Algorithm DMA Current Source Address
CE_SCDA	0x0028	Symmetric Algorithm DMA Current Destination Address
CE_HCSA	0x0034	HASH Algorithm DMA Current Source Address
CE_HCDA	0x0038	HASH Algorithm DMA Current Destination Address
CE_ACSA	0x0044	Asymmetric Algorithm DMA Current Source Address
CE_ACDA	0x0048	Asymmetric Algorithm DMA Current Destination Address
CE_XCSA	0x0054	XTS Algorithm DMA Current Source Address
CE_XCDA	0x0058	XTS Algorithm DMA Current Destination Address

10.1.5. Register Description

10.1.5.1. 0x0000 CE Task Descriptor Address Register(Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: CE_TDA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	Task Descriptor Address Must be word address.

10.1.5.2. 0x0008 CE Interrupt Control Register(Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: CE_ICR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3:0	R/W	0x0	Task Channel3~0 Interrupt Enable 0: Disable 1: Enable

10.1.5.3. 0x000C CE Interrupt Status Register(Default Value: 0x0000_0000)

Offset: 0x000C			Register Name: CE_ISR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3:0	R/W1C	0x0	Task Channel3~0 End Pending 0: Not finished

			1: Finished It indicates if task has been completed . Write '1' to clear it.

10.1.5.4. 0x0010 CE Task Load Register(Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: CE_TLR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:15	/	/	/
14:8	R/W	0x0	Algorithm type, the same with type field in description common control.
7:1	/	/	/
0	R/W	0x0	Task Load When setting, CE can load the descriptor of task if task FIFO is not full.

10.1.5.5. 0x0014 CE Task Status Register(Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: CE_TSR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:6	R	0x0	indicate which channel in run for XTS 00: task channel0 01: task channel1 10: task channel2 11: task channel3
5:4	R	0x0	indicate which channel in run for asymmetric 00: task channel0 01: task channel1 10: task channel2 11: task channel3
3:2	R	0x0	indicate which channel in run for digest 00: task channel0 01: task channel1 10: task channel2 11: task channel3
1:0	R	0x0	indicate which channel in run for symmetric 00: task channel0 01: task channel1 10: task channel2 11: task channel3

10.1.5.6. 0x0018 CE Error Status Register(Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: CE_ESR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W1C	0x0	Task channel 3 error type. (the same for other channels) Bit 24: algorithm not support Bit 25: data length error Bit 26: keysram access error. Write '1' to clear. Bit 29: address invalid other: reserved
23:16	R/W1C	0x0	Task channel 2 error type. Bit 16: algorithm not support Bit 17: data length error Bit 18: keysram access error. Write '1' to clear. Bit 21: address invalid other: reserved
15:8	R/W1C	0x0	Task channel 1 error type. Bit 8: algorithm not support Bit 9: data length error Bit 10: keysram access error. Write '1' to clear. Bit 13: address invalid other: reserved
7:0	R/W1C	0x0	Task channel 0 error type. Bit 0: algorithm not support Bit 1: data length error Bit 2: keysram access error. Write '1' to clear. Bit 5: address invalid other: reserved

10.1.5.7. 0x0024 CE Symmetric Current Source Address Register(Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: CE_SCSA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	Symmetric algorithm current source address read by DMA.

10.1.5.8. 0x0028 CE Symmetric Current Destination Address Register(Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: CE_SCDA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	Symmetric algorithm current destination address written by DMA.

10.1.5.9. 0x0034 CE HASH Current Source Address Register(Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: CE_HCSA
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	HASH algorithm current source address read by DMA.

10.1.5.10. 0x0038 CE HASH Current Destination Address Register(Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: CE_HCDA
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	HASH algorithm current destination address written by DMA.

10.1.5.11. 0x0044 CE Asymmetric Current Source Address Register(Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: CE_ACSA
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	Asymmetric algorithm current source address read by DMA.

10.1.5.12. 0x0048 CE Asymmetric Current Destination Address Register(Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: CE_ACDA
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	Asymmetric algorithm current destination address written by DMA.

10.1.5.13. 0x0054 CE XTS Current Source Address Register(Default Value: 0x0000_0000)

Offset: 0x0054			Register Name: CE_XCSA
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	XTS algorithm current source address read by DMA.

10.1.5.14. 0x0058 CE XTS Current Destination Address Register(Default Value: 0x0000_0000)

Offset: 0x0058			Register Name: CE_XCDA
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	XTS algorithm current destination address written by DMA.

10.2. Security ID

The Security ID(SID) is one electrical efuse for saving key, which includes chip ID, thermal sensor, and security key, etc.

The SID module has the following features:

- The module register is non-secure forever, efuse has secure zone and non-secure zone
- A fuse only can program one time

