R128
User Manual

Revision 0.2

April 7, 2023

Revision History

Revision	Date	Author	Description
0.1	March 31, 2022	KPA0570	Initial release version.
0.2	April 7, 2023	KPA0570	**Chapter 2 Product Description** Update the sections Flashc_enc, I2S/PCM, Bluetooth Subsystem **Chapter 3 System** Update the sections PMU, RISC-V Subsystem, PLIC, TIMER, Watchdog. **Chapter 4 Video & Graphics** Update the features of the section G2D **Chapter 5 Video Input Interfaces** Update the section CSI_JPEG **Chapter 6 Video Output Interfaces** Update the section TCON_LCD **Chapter 7 Audio Interfaces** Update the features of I2S/PCM **Chapter 8 Memory** Update the section Flashc_Enc **Chapter 9 Peripherals** Update the sections USB2.0 DRD, SMHC, SPI, SPI_DBI

1 About This Document

1.1 Purpose and Scope

This document describes the features, logical structures, functions, operating modes, and related registers of each module about R128. For details about the interface timings and related parameters, the pins, pin usages, performance parameters, and package dimension, please refer to the R128_Datasheet.

1.2 Intended Audience

The document is intended for:

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

1.3 Symbol Conventions

1.3.1 Symbol Conventions

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

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

1.3.2 Table Content Conventions

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

Symbol	Description
/	The cell is blank.

1.3.3 Reset Value Conventions

In the register definition tables:

If the column value in a bit or multiple bits row is “/”, that this bit or these multiple bits are unused.

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

1.3.4 Register Attributes

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

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

1.3.5 Numerical System

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

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

The expressions of addresses and data are described as follows.

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

Contents

2 Product Description

2.1 Overview

R128 is a highly integrated wireless audio SoC with high performance. It features a XuanTie RISC-V CPU, a HiFi5 DSP, an 802.11b/g/n WLAN & Dual-mode Bluetooth subsystem, a Voice & Audio CODEC subsystem, and a Power Management Unit (PMU). RISC-V C906 CPU and HiFi5 DSP provide powerful and energy-efficient computing power for applications and audio processing respectively. R128 integrates 3 differential ADCs and 2 differential DACs and can be applied to microphone array-based voice recognition and stereo audio playback solutions. Wi-Fi and bluetooth enable R128 to implement various network applications with an exclusive antenna for each to simultaneously transmit and receive data on 2.4 GHz. R128 also integrates a JPEG encoder, a RGB display engine, and a 2D graphics acceleration system, making it possible to achieve some display applications.

2.2 Device Difference

R128 is configured with different sets of features in different devices. The following table shows the feature differences across different devices.

Table 2-1 Device Feature Differences

Contents	R128-S1	R128-S2	R128-S3
PSRAM	8 MB	16 MB	32 MB
Audio Codec Output	LINEOUTLP/N LINEOUTRP/N	LINEOUTLP/N LINEOUTRP/N	LINEOUTLP/N
GPADC Channels	7	7	8

2.3 Features

2.3.1 CPU Architecture

- XuanTie C906 64-bit RISC-V CPU up to 480 MHz
  - 32 KB I-cache + 32 KB D-cache
- ARM M33 Star@192 MHz MCU
  - 32 KB I-cache + 32 KB D-cache
  - TrustZone

2.3.2 DSP Architecture

- HiFi5 Audio DSP@400 MHz
  - 32 KB I-cache + 32 KB D-cache

2.3.3 Memory Subsystem

2.3.3.1 BROM

- On-chip memory
- Supports system boot from the following devices:
  - SD/eMMC
  - SPI Nor Flash
  - SPI Nand Flash
- Supports secure boot and normal boot
- Secure brom supports load only certified firmware
- Secure brom ensures that the secure boot is a trusted environment
- Supports USB eFEX protocol and UART mboot protocol for firmware upgrade

2.3.3.2 PSRAMC

- Up to 2 PSRAM controllers (HS_PSRAMC, LS_PSRAMC)

HS_PSRAMC:

- R128-S1/R128-S2: SIP 8 MB PSRAM
- R128-S3: SIP 32 MB PSRAM
- Supports AP memory PSRAM

- Supports 64 Mbit/256 Mbit PSRAM
- Supports OPI as the interface of PSRAM
- Supports the auto-refreshing and self-refreshing of PSRAM
- Supports up to 800 MHz PSRAM. The ratio of PSRAM controller and PSRAM clock is 1:4.
- Supports indirectly accessing the registers of PSRAM through interface configuration
- Supports caching reading/writing commands through CAM
- Supports out-of-order execution of commands
- Supports prefetching read channel

LS_PSRAMC:

- R128-S2: SIP 8 MB PSRAM
- Supports any frequency ratio of AHB and OPI clock
- Supports CPU/DMA to operate PSRAM through SBUS
- Supports PSRAM Wrap Mode (enter/exit)
- Supports 4 offset address ranges
- Supports CBUS out-of-order reading/writing and XIP code execution

2.3.3.3 SMHC

- Compatible with Secure Digital Memory (SD mem-version 2.0)
- Compatible with Secure Digital I/O (SDIO-version 3.0)
- Compatible with embedded MultiMediaCard (eMMC-version 5.0)
- Supports Card insertion and removal interrupt
- Supports hardware CRC generation and error detection
- Supports programmable baud rate
- Supports SDIO interrupts in 1-bit and 4-bit modes
- Supports block size of 1 to 65535 bytes
- Supports descriptor-based internal DMA controller
- Internal 1024-Bytes RX FIFO and 1024-Bytes TX FIFO
- Supports 1-bit, 4-bit SD and SDIO data bus width
- Supports 1-bit, 4-bit eMMC data bus width

2.3.3.4 Flashc_Enc

- Supports arbitrary frequency ratio of AHB clock and SPI clock

- Supports 4 segments of offset address range
- Supports receiving and transmitting in 1/2/4-wire SPI
- Supports flash programing and reading by configuring registers (SBUS)
- Supports out-of-order reading CBUS and running codes through XIP
- Supports continuous reading mode (enter/exit) and wrap mode (enter/exit)
- Supports the basic operation of SPI flash
- Supports 8 MB SIP Nor Flash (for R128-S1) and 16 MB SIP Nor Flash (for R128-S2)
- Supports real-time AES encoding and decoding when reading and writing data through SBUS data and reading data through CBUS

2.3.4 System

2.3.4.1 PMU

- Supports 3.0 V-5.5 V external single supply
- Integrates DCDC/LDO and other power modules, and power all circuits within the IC
- The internal digital circuit is divided into power domains. Each of them has independent power switch, which is determined by system low-power status.
- Supports standby, hibernation and other low-power modes, which can be switched over by PMU.
- Manages the opening and close of analog modules like DCDC/LDO, DCXO/DPLL. The starting duration is configured by the software.

2.3.4.2 GPRCM

- Manages the power of this system
- Manages the reset of each system
- Manages the OSC clock

2.3.4.3 CCU

- Up to 2 CCU controllers (CCU, CCU_AON)

CCU:

- Supports configuring module clock
- Supports clock output control
- Supports bus clock gating
- Supports bus software reset

CCU_AON:

- Supports managing the OSC clock
- Supports bus source and divisions
- Supports configuring modules clock
- Supports clock output control
- Supports bus clock gating
- Supports bus software reset

2.3.4.4 DMAC

- Up to 2 DMACs
- Up to 16 DMA channels for each DMAC
- Provides 32 peripheral DMA requests for data reading and 32 peripheral DMA requests for data writing
- Supports transferring data with a linked list
- Supports programmable 8-bit, 16-bit, 32-bit, and 64-bit data width
- Supports programmable DMA burst length
- DRQ response includes the waiting mode and handshake mode
- DMA channel support pause function
- Memory devices support non-aligned transfer

2.3.4.5 Timer

- 8 timers: 5 for SW domain and 3 for AON domain
- Configurable counting clock: LOSC and OSC40M. Whether LOSC is internal low-frequency clock or external low-frequency clock (with greater accuracy) depends on LOSC_SRC_SEL.
- Supports 8 prescale factors
- Programmable 32-bit down timer
- Supports two timing modes: periodic mode and single counting mode
- Generates an interrupt when the count is decreased to 0

2.3.4.6 Watchdog

- Up to 4 watchdogs, and one of them is secure world watchdog
- HOSC_32K clock sources and 32K system
- Supports 12 configurable initial count value
- Supports generating timeout interrupt
- Supports outputting reset signal

- Supports restarting timer

2.3.4.7 RTC

- Implements time counter and timing wakeup
- Provides counters for counting year, month, day, hour, minute, and second
- 4-channel clock sources: hosc_32K, rccal_32K, rcosc, losc_clk
- Configures initial value by software anytime
- Supports timing alarm, and generates interrupt and wakes up the PMU system

2.3.4.8 Message Box

- Supports 3 CPU to transmit information through channels. Each CPU has a MSGBOX.
  - CPU0: ARM CPU
  - CPU1: RISC-V
  - CPU2: DSP
- There are four channels every two CPU, and the FIFO depth of one channel is 8 x 32 bits.
- Supports interrupts

2.3.4.9 Spinlock

- Supports 32 lock units
- Two kinds of lock status: locked and unlocked
- Lock time of the processor is predictable (less than 200 cycles)

2.3.5 Video and Graphics

2.3.5.1 DE

- Supports output size up to 1024x1024 pixels
- Supports two alpha blending UI channels for main display
- Supports four overlay layers in each channel, and channel0 has an independent scaler
- Supports potter-duff compatible blending operation
- Supports input format ARGB8888/XRGB8888/RGB888/ARGB4444/ARGB1555/RGB565
- Supports SmartColor2.0 for excellent display experience
  - Fully programmable color matrix
  - Dynamic gamma

- Supports write back only for verification

2.3.5.2 G2D

- Supports layer size up to 1024x1024 pixels
- Supports pre-multiply alpha image data
- Supports color key
- Supports two pipes Porter-Duff alpha blending
- Supports multiple video formats 4:2:0, 4:2:2, 4:1:1 and multiple pixel formats (8/16/24/32 bits graphics layer)
- Supports any format convert function above
- Supports 1/16× to 32× resize ratio
- Supports 32-phase 8-tap horizontal anti-alias filter and 32-phase 4-tap vertical anti-alias filter.
- Supports window clip
- Supports FillRectangle, BitBlit, StretchBlit and MaskBlit
- Supports horizontal and vertical flip, clockwise 0/90/180/270 degree rotate for normal buffer

2.3.6 Video Output Interfaces

2.3.6.1 TCON_LCD

- Supports RGB interface with DE/SYNC mode, up to 1024x768@60fps
- Supports serial RGB/dummy RGB interface, up to 800x480@60fps
- Supports i8080 interface, up to 800x480@60fps
- Supports BT656 interface for NTSC and PAL
- Supports RGB666 and RGB565 with dither function

2.3.7 Video Input Interfaces

2.3.7.1 CSI_JPEG

CSI:

- Supports YUV422 format input and YUV420 format NV12 output
- Supports scaling image down 1/2 and cropwin
- Supports receiving JPEG streams directly output by sensor
- Supports receiving the images with unconventional resolutions (X and Y can be an integer multiple of 16, such as: 192*192, 304*224)

- Supports receiving the images with conventional resolutions (such as 128*128, 256*256, 320*240, 640*480, 1280*720, 1920*1088)

JPEG:

- Supports 640*480@60fps in the online mode, and 640*480@30fps in the offline mode
- Supports 1280*720@40fps in the online mode, and 1280*720@20fps in the offline mode
- Supports up to 1920*1088 online/offline encoding
- Supports encoding after scaling images down 1/2 and cropwin
- Supports block output in the online mode to reduce SRAM usage and improve bandwidth utilization ratio
- Supports online/offline encoding (such as 192*192 and 304*224) the images with non-conventional resolution
- Supports the online/offline encoding the images with conventional resolution images

2.3.8 Audio Interfaces

2.3.8.1 Audio Codec

- HiFi Audio ADC
  - 3-channel ADCs @ 24-bit
  - Up to 98 dB SNR during ADC recording path (signal through PGA and ADC with A-weighted filter)
  - 3 fully-differential analog microphone inputs with 0 dB~30 dB amplifier gain
  - Supports sample rates ranging from 8 kHz to 96 kHz
  - Digital volume control with 0.5 dB step
  - Digital high-pass filter
  - 128x24-bit FIFO for recording received data
- HiFi Audio DAC
  - 2-channel DACs @ 24-bit (for R128-S1 and R128-S2)
  - 1-channel DAC @ 24-bit (for R128-S3)
  - Up to 119 dB SNR in the DAC playback path (signal through DAC and lineout with A-weighted filter)
  - Supports sample rates ranging from 8 kHz to 384 kHz
  - Digital volume control with 0.5 dB step
  - 20-band Biquads filter for EQ
  - 3-band dynamic range control
  - 128x24-bit FIFO for playing transmitted data

- Three differential microphone inputs: MICIN1P/N, MICIN2P/N, MICIN3P/N.
- Two stereo LINEOUT outputs: LINEOUTLP/N and LINEOUTRP/N (for R128 S1 and R128 S2)
- One differential LINEOUT output: LINEOUTLP/N (for R128 S3)
- Built-in audio PLL with flexible clocking scheme
- DMA and interrupt support both receiving and transmitting
- Integrated ALDO for analog part
- One low-noise analog microphone bias output three audio inputs

2.3.8.2 I2S/PCM

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

2.3.8.3 DMIC

- Supports up to 8 channels
- Sample rate from 8 kHz to 48 kHz

2.3.8.4 OWA

- In compliance with S/PDIF Interface
- Compatible with standard IEC-60958 and IEC-61937
  - IEC-60958 supports 16-bit, 20-bit and 24-bit data formats
  - IEC-61937 uses the IEC-60958 series for the conveying of non-linear PCM bit streams, each sub-frame transmits 16-bit
- TXFIFO and RXFIFO
  - One 128×24bits TXFIFO and one 64×24bits RXFIFO for audio data transmission
  - Programmable FIFO thresholds
- Supports TX/RX DMA slave interface
- Supports multiple function clock
  - Separate clock for OWA TX and OWA RX
  - The TX function clock supports the frequency of 24.576 MHz and 22.579 MHz
  - The RX function clock supports the frequency of 60MHz and 240MHz, which can realize the sampling rate of 8 kHz to 96 kHz and 32 kHz to 192 kHz respectively
- Supports hardware parity on TX/RX
  - Hardware parity generation on the transmitter
  - Hardware parity checking on the receiver
- Supports channel status capture for the receiver
- Supports channel sample rate capture on the receiver
- Supports insertion detection for the receiver
- Supports channel status insertion for the transmitter
- Supports interrupts and DMA

2.3.9 Security Subsystem

2.3.9.1 Crypto Engine

- Supports Symmetrical Algorithm: AES, DES, 3DES
- Supports 128-bits, 192-bits and 256-bits key size for AES
- Supports ECB, CBC, CTR, CTS, OFB, CFB modes for AES
- Supports 1, 8, 64, 128bit width for AES-CFB
- Supports 16bit, 32bit, 64bit, 128bit wide size for AES CTR

- Supports ECB, CBC, CTR, CBC_MAC modes for DES/3DES
- Supports 16bit, 32bit, 64bit wide size for DES/3DES CTR
- Supports Hash Algorithms: MD5, SHA1, SHA224, SHA256, SHA384, SHA512, HMAC
- Supports multi-package mode for MD5, SHA1, SHA224, SHA256, SHA384, SHA512
- Supports Asymmetrical Algorithm: RSA512/1024/2048bit
- Supports internal DMA Controller for data transmission with memory
- Supports secure and non-secure interfaces respectively
- Supports accessing Secure and non-secure interfaces by non-secure host when secure_mode is 0

2.3.9.2 SMC

- The SMC (GSecure Memory Control) is always secure, only secure CPU can access the SMC
- Sets secure area of HSPSRAM
- Sets secure property that Master accesses to HSPSRAM

2.3.9.3 SPC

- The SPC (Secure Peripherals Control) is always secure, only secure CPU can access the SPC
- Sets secure property of peripherals
- Supports safety access of flash controller

2.3.10 Peripherals

2.3.10.1 USB2.0 DRD

- Complies with USB 2.0 Specification
- Supports High-Speed (HS, 480-Mbps), Full-Speed (FS, 12-Mbps), and Low-Speed (LS, 1.5-Mbps) in Host mode
- Supports High-Speed (HS, 480 Mbps), Full-Speed (FS, 12 Mbps) in Device mode
- Supports the UTMI+ Level 3 interface. The 8-bit bidirectional data buses are used.
- Supports bi-directional endpoint0 for Control transfer
- Supports up to 10 User-Configurable Endpoints for Bulk, Isochronous and Interrupt bi-directional transfers (Endpoint1, Endpoint2, Endpoint3, Endpoint4, Endpoint5)
- Supports up to (8KB+64Bytes) FIFO for 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

2.3.10.2 UART

- Up to 3 UART controllers (UART0, UART1, and UART2)
- Compatible with industry-standard 16450/16550 UARTs
- 64-Byte Transmit and receive data FIFOs
- Supports DMA controller interface
- Supports Software/ Hardware Flow Control
- Supports IrDA 1.0 SIR
- Supports RS-485 mode

2.3.10.3 SPI and SPI_DBI

- Up to 2 SPI controllers (SPI0, SPI1)
- The SPI0 only supports SPI mode. The SPI1 supports SPI mode and display bus interface (DBI) mode

SPI mode:

- Full-duplex synchronous serial interface
- Master/slave configurable
- 8-bit wide by 64-entry FIFO for both transmitting and receiving data
- Polarity and phase of the Chip Select (SPI_SS) and SPI Clock (SPI_SCLK) are configurable
- Supports interrupts and DMA
- Supports mode0, mode1, mode2, and mode3
- Supports 3-wire/4-wire SPI
- Supports programmable serial data frame length: 0 bit to 32 bits
- Supports standard SPI, dual-output/dual-input SPI, dual I/O SPI, quad-output/quad-input SPI
- Supports maximum IO rate of the mass production: 96 MHz
- Supports 5 clock sources, Interrupt or DMA

DBI mode:

- Supports DBI Type C 3 Line/4 Line Interface Mode

- Supports 2 Data Lane Interface Mode
- Supports data source from CPU or DMA
- Supports RGB111/444/565/666/888 video format
- Maximum resolution of RGB666 240 x 320@30Hz with single data lane
- Maximum resolution of RGB888 240 x 320@60Hz or 320 x 480@30Hz with dual data lane
- Supports tearing effect
- Supports software flexible control video frame rate

2.3.10.4 TWI

- Supports 2 TWIs
- Software-programmable for Slave or Master
- Supports Repeated START signal
- Allows 10-bit addressing with TWI bus
- Performs arbitration and clock synchronization
- Owns address and General Call address detection
- Interrupt on address detection
- Supports speeds up to 400 kbits/s in fast mode
- Allows operation from a wide range of input clock frequencies
- TWI Driver Supportss packet transmission and DMA when TWI works in Master mode

2.3.10.5 PWM

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

- Supports any plural channels to form a group, and output the same duty-cycle pulse
- In group mode, the relative phase of the output waveform for each channel is configurable

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

2.3.10.6 GPADC

- 12-bit Resolution and 7-bit effective SAR type A/D converter
- 9-channel multiplexer including 7 channels general purpose ADC (ADC0-ADC6) and 2 channels special ADC (ADC8, ADC12) for R128-S1 and R128-S2
- 10-channel multiplexer including 8 channels general purpose ADC (ADC0-ADC7) and 2 channels special ADC (ADC8, ADC12) for R128-S3
- The ADC8 is used for VBAT voltage detection and the ADC12 is used for temperature sensor
- 64 FIFO depth of data register
- Power Supply Voltage: 2.5V, Analog Input Range: 0 to 2V
- Maximum Sampling frequency: 1 MHz
- Support self-calibration
- Support data compare and interrupt
- Support four operation modes
  - Single conversion mode
  - Single-cycle conversion mode
  - Continuous conversion mode
  - Outbreak conversion mode

2.3.10.7 CIR Receiver

- Supports CIR remote control receiver
- Supports NEC IR protocol
- 64x8 bits RX FIFO for data buffer
- Programmable RX FIFO thresholds
- Supports interrupt
- Sample clock up to 1 MHz

2.3.10.8 CIR Transmitter

- Supports CIR remote control transmitter
- 128 Bytes FIFO for data buffer
- Configurable carrier frequency
- Supports Interrupt and DMA
- Supports handshake mode and waiting mode of DMA

2.3.10.9 LEDC

- Configurable LED input high-/low-level width
- Configurable LED reset time
- LEDC data supports DMA configuration mode and CPU configuration mode
- Maximum 1024 LEDs serial connect
- LED data transmission rate up to 800 kbit/s
- Configurable RGB display mode

2.3.10.10 SCR

- Supports the ISO/IEC 7816-3:1997(E) and EMV2000 (4.0) Specifications
- Performs functions needed for complete smart card sessions, including:
  - Card activation and deactivation
  - Cold/warm reset
  - Answer to Reset (ATR) response reception
  - Data transfers to and from the card
- Supports adjustable clock rate and bit rate
- Configurable automatic byte repetition
- Supports commonly used communication protocols:
  - T=0 for asynchronous half-duplex character transmission
  - T=1 for asynchronous half-duplex block transmission
- 128bits FIFO for data transmit & receive
- Supports FIFOs for receive and transmit buffers (up to 128 bits) with threshold
- Supports configurable timing functions:
  - Smart card activation time

- Smart card reset time
- Guard time
- Timeout timers

- Supports synchronous and any other non-ISO 7816 and non-EMV cards

2.3.11 Wi-Fi MAC

- 802.11d: Regulatory domain operation
- 802.11e: QoS including WMM
- 802.11h: Transmit power control dynamic and frequency selection
- 802.11i: Security including WPA2 compliance
- 802.11w: Supports STA Mode with PMF, SA Query, SAE

2.3.11.1 Wi-Fi Baseband

- Compatible with IEEE 802.11 b/g/n standard on 2.4 GHz
- Up to 20 MHz bandwidth
- 802.11n MCS0-7 with data rate up to 72.2Mbps (BPSK, r=1/2 through 64QAM, r=5/6)
- 6 MB-54MB data rate for 802.11g
- DSSS, CCK modulation with long and short preamble
- Short Guard Interval
- Long Guard Interval
- RX antenna Diversity
- Supports RX STBC

2.3.11.2 Wi-Fi Radio

- Integrated 2.4GHz PA, LNA, and T/R switch
- Internal impedance matching network and harmonic filter allow chip to connect to antenna directly
- High Power Amplifier with 3 V-5.5 V full range directly support XRADIOTECH's MPDTM technology ensure linearity tracking automatically to always keep EVM and mask within specifications
- Special Architecture and Device design to keep the reliability of PA and also deliver high output power (>25 dBm)

2.3.12 Bluetooth Subsystem

- BLE V5.0
  - Bluetooth 5.0 Dual Mode complies with V2.1/4.0/4.2/5.0

- Supports GFSK, π/4 DQPSK, and 8DPSK modulation
- Data rates support: 125Kbps, 500Kbps, 1Mbps, 2Mbps
- supports long range
- TRNG generator
- AES-128 data encryption with ECB and CCM mode
- Supports advertising extension
- Packet assembly and disassembly
- Data Whitening and De-whitening
- Data CRC generation and checking
- Packet filtering based on filter policies (white and resolving lists)
- Private address generation and Accelerate address resolution
- Access address generation and matching
- Frequency hopping and channel mapping
- RSSI Reporting to host

- BR/EDR
  - Adaptive Frequency Hopping
  - SCO and eSCO support
  - 1, 3 and 5 slots all packet types support
  - Transcoders for A-law, μ-law and CVSD voice over air
  - Supports piconet and scatternet
  - Secure simple pairing
  - Supports sniff/ low power mode
- Transmits Power: -20 dBm (0.01 mW) to +10 dBm (10 mW)
- Receiver Sensitivity:
  - 95.0 dBm@BR
  - 98 dBm@BLE 1 Mbit/s
  - 105 dBm@BLE S = 8

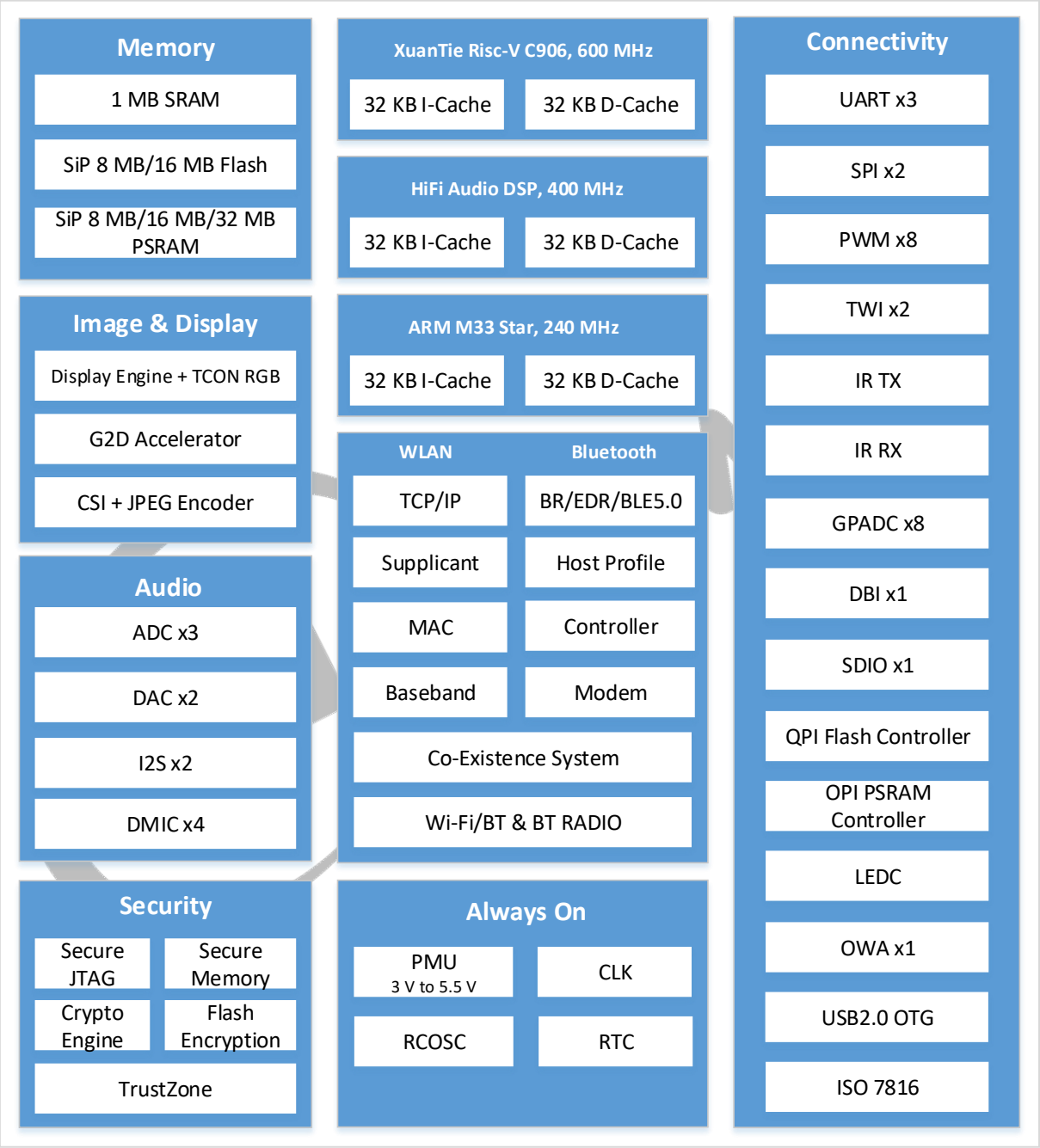
2.3.13 Package

- QFN80 balls, 0.35mm ball pitch, 8mm x 8mm body

2.4 Block Diagram

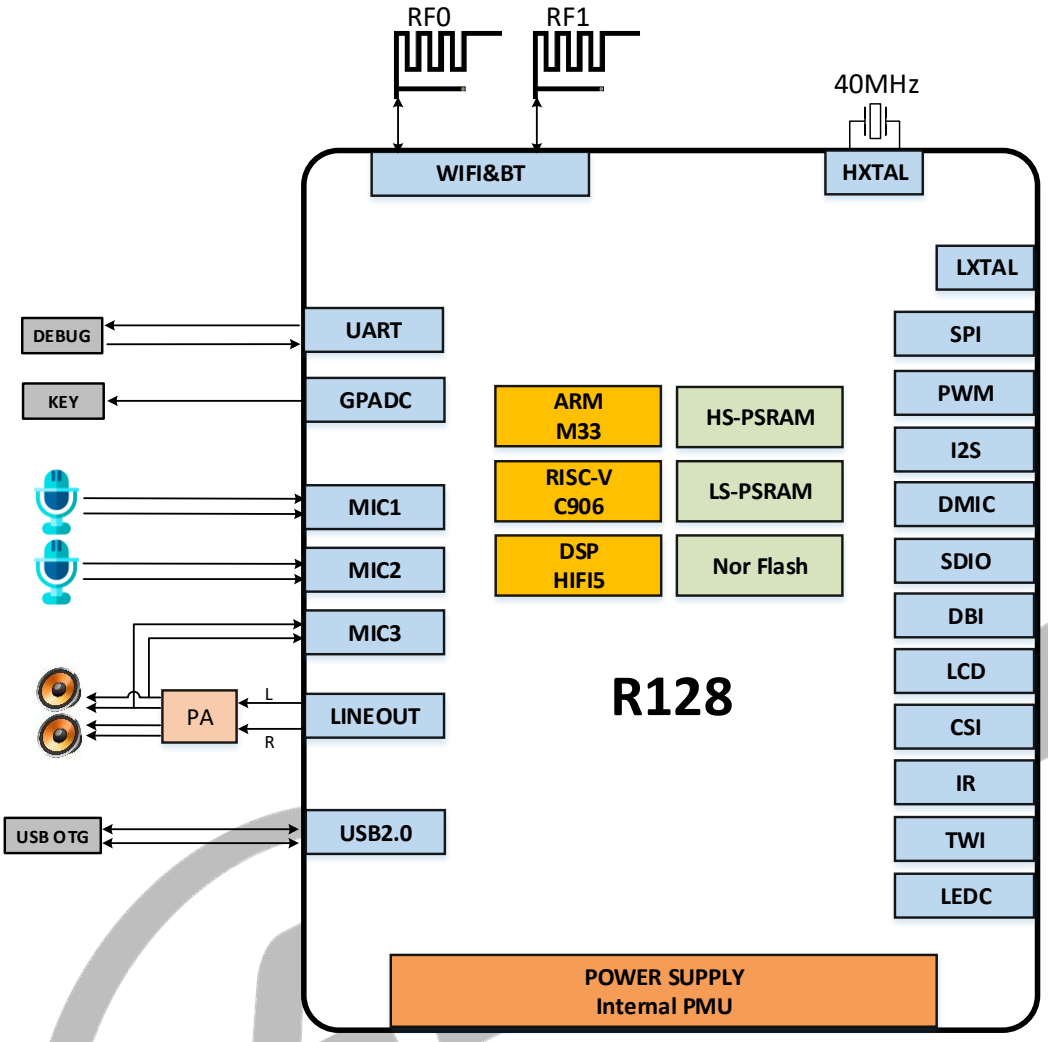
The following figure shows the system block diagram of the R128.

Figure 2-1 R128 System Block Diagram



The following figure shows the intelligent white goods solution of the R128.

Figure 2-2 R128 Intelligent White Goods Solution



NOTE

a. 8 MB PSRAM in R128-S1

b. 16 MB PSRAM in R128-S2

c. 32 MB PSRAM in R128-S3

Contents

Figures

Tables

3 System

3.1 Memory Mapping

Module	Address (It is for Cluster CPU)	Size
BROM & SRAM & PSRAM & FLASH (Code Region)		
BROM_N	0x00000000—0x0000FFFF	64 KB
BROM_S	0x00000000—0x0000FFFF	64 KB
SRAM	0x04000000---0x040FFFFF	1 MB
MAD SRAM	0x04100000---0x0410FFFF	64 KB
OPI L_PSRAM	0x08000000---0x08FFFFFF	16 MB
H_PSRAM	0x0C000000---0x0DFFFFFF	32 MB
SPI FLASH	0x10000000---0x1BFFFFFF	192 MB
AHB_DEC0 Related		
DMAC0	0x40000000---0x40000FFF	4 KB
DMAC1	0x40001000---0x40001FFF	4 KB
SPINLOCK	0x40003000---0x40003FFF	4 KB
CE	0x40004000---0x40004FFF	4 KB
SMC	0x40005000---0x40005FFF	4 KB
SDIO	0x40008000---0x40008FFF	4 KB
SPI0	0x40009000---0x40009FFF	4 KB
SYSCTRL	0x4000A000---0x4000AFFF	4 KB
FLASH CTRL	0x4000B000---0x4000BFFF	4 KB
L_PSRAM CTRL	0x4000D000---0x4000DFFF	4 KB
WIFI_C	0x4000E000---0x4000EFFF	4 KB
SPI1	0x4000F000---0x4000FFFF	4 KB
MSI	0x40100000---0x402FFFFF	2 MB
CSI	0x40300000---0x406FFFFF	4 MB
VIDEO_OUT_SYS	0x40700000---0x40AFFFFF	4 MB (DE)
	0x40B00000---0x40B3FFFF	256 KB (G2D)
	0x40B40000---0x40B40FFF	4 KB (DISPLAY_TOP)
	0x40B41000---0x40B41FFF	4 KB (TCON_LCD0)
USB_DRD	0x40B42000---0x40C41FFF	1 MB
CPU_SYS Related		
CPU_WDG	0x40020400---0x400207FF	1 KB
CPU_MBOX	0x40020800---0x40020FFF	2 KB
DSP_SYS Related		
DSP_WDG	0x40023800---0x40023BFF	1 KB
DSP_MBOX	0x40022000---0x40022FFF	4 KB
RISCV_SYS Related		
RISCV_CFG	0x40028000---0x40028FFF	4 KB
RISCV_WDG	0x40029000---0x40029FFF	4 KB

Module	Address (It is for Cluster CPU)	Size
RISCV_TIMESTAMP	0x4002A000---0x4002AFFF	4 KB
RISCV_MSGBOX	0x4002B000---0x4002BFFF	4 KB
APB_DEC		
H_PSRAM CTRL	0x40038000---0x4003BFFF	16 KB
CCMU	0x4003C000---0x4003CFFF	4 KB
CODEC_DAC	0x4003D000---0x4003DFFF	4 KB
L_PSRAM TZMA	0x4003E000---0x4003E3FF	1 KB
TZMA0	0x4003E400---0x4003E7FF	1 KB
TZMA1	0x4003E800---0x4003EBFF	1 KB
TZMA2	0x4003EC00---0x4003EFFF	1 KB
TZMA3	0x4003F000---0x4003F3FF	1 KB
FLASH TZMA	0x4003F400---0x4003F7FF	1 KB
FLASH ENC	0x4003F800---0x4003FBFF	1 KB
EXPSRAM_TZMA	0x4003FC00---0x4003FFFF	1 KB
Timer (Normal)	0x40043000---0x40043FFF	4 KB
SPC	0x40044000---0x40044FFF	4 KB
PWM	0x40045000---0x400453FF	1 KB
SMCARD	0x40045400---0x4004 57FF	1 KB
I2S	0x40045800---0x40045BFF	1 KB
OWA	0x40045C00---0x40045FFF	1 KB
IRRX	0x40046000---0x400463FF	1 KB
IRTX	0x40046400---0x400467FF	1 KB
UART0	0x40047000---0x400473FF	1 KB
UART1	0x40047400---0x400477FF	1 KB
UART2	0x40047800---0x40047BFF	1 KB
TWD	0x40047C00---0x40047FFF	1 KB
LEDC	0x40048000---0x400483FF	1 KB
AHB_BW_MON0	0x40048400---0x400487FF	1 KB
TRNG	0x40048C00---0x40048FFF	1 KB
TWI0	0x40049000---0x400493FF	1 KB
TWI1	0x40049400---0x400497FF	1 KB
AHB_DEC1 (AON Domain)		
GPADC	0x4004A000---0x4004A3FF	1 KB
GPIO	0x4004A400---0x4004A7FF	1 KB
LPUART0	0x4004A800---0x4004ABFF	1 KB
LPUART1	0x4004AC00---0x4004AFFF	1 KB
CODEC_ADC	0x4004B000---0x4004BFFF	4 KB
DMIC	0x4004C000---0x4004C3FF	1 KB
CCMU_AON	0x4004C400---0x4004C7FF	1 KB
RCOSC_CAL	0x4004C800---0x4004CBFF	1 KB
WAKEUP_TIMER	0x4004CC00---0x4004CFFF	1 KB
MAD	0x4004E000---0x4004E3FF	1 KB
SID	0x4004E400---0x4004EFFF	3 KB

Module	Address (It is for Cluster CPU)	Size
AHB_DEC1 (RTC Domain)		
GPRCM	0x40050000---0x40050FFF	4 KB
RTC_TIMER	0x40051000---0x400513FF	1 KB
PMC	0x40051400---0x400517FF	1 KB
RISCV_PERI (only for RISCV)	0x50000000---0x5100FFFF	64 MB+64 KB
BLE Related		
BLE	0xA0000000---0xA01FFFFF	2 MB
WIFI SRAM Related		
WIFI_SRAM	0x68000000---0x68FFFFFF	16 MB
WIFI Peripheral		
WIFI_PERI	0xB9C00000---0xBAFFFFFF	20 MB

3.2 Power Management Unit (PMU)

3.2.1 Overview

Power Management Unit (PMU) controls system power supply. It is responsible for controlling the power-on/off and parameter configurations of DCDC/LDO/SW, the opening/close of independent power supply circuit, and the clocks like DCXO/PLL. In addition, it supports various situations including power-up, power-down, hibernation, and wakeup, where there are independent trigger sources and the on-off actions executed by PMU. The trigger sources include external PIN_RST or VBAT in power up/down, software in hibernation status, and wakeup in interrupt events.

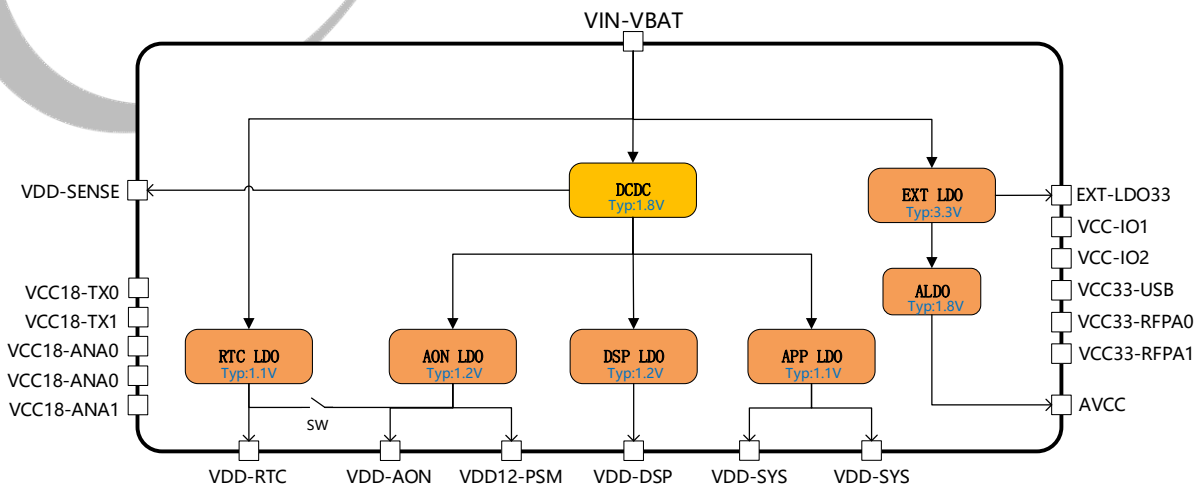
The most actions of PMU are automatically executed by the hardware, and some configurable parameters are configured by the software.

The PMU has the following features:

- Supports 3.0 V-5.5 V external single supply
- Integrates DCDC/LDO and other power modules, and powers all circuits within the IC
- The internal digital circuit is divided into power domains. Each of them has independent power switch, which is determined by system low-power status.
- Supports standby, hibernation and other low-power modes, which can be switched over by PMU.
- Manages the opening and close of analog modules like DCDC/LDO, DCXO/DPLL. The starting duration is configured by the software.

3.2.2 Block Diagram

The following figure shows the block diagram of the PMU circuit structure.



3.2.3 Functional Description

3.2.3.1 Reference Time Configuration

PMU is responsible for enabling/disabling the analog modules like DCXO/DPLL/LDO/SW/BG/DCDC. When these modules are enabled, there is a period of settling time for PMU to release enable and wait effective output through 32K or DCXO. 32K or DCXO is used as counter, whose target value is the settling time configured by the software.

The rules of software configuring settling time: a. The shorter settling time can reduce the waiting time of power-up or wake-up; b. The settling time should be longer than that of analog circuit start-up. In the first power-up, analog modules should be enabled to wait for the default settling time set by the hardware. After the software configuration, the new settling time will take effect when the system is waked up again.

3.2.3.2 System Low-power Control

The following table shows the system low-power statuses and their definitions.

Table 3-1 system low-power status

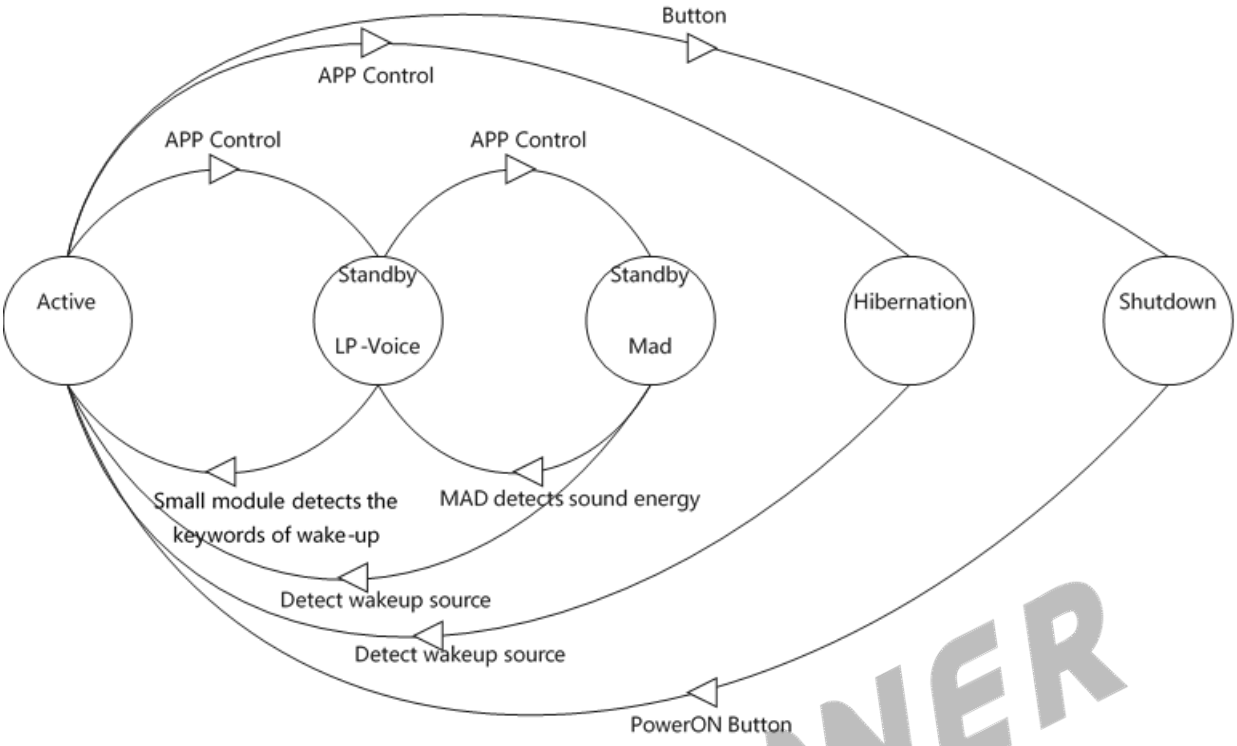
The following figure shows the low-power system status switch.

Power Mode	M33	RISC-V	DSP	EXT LDO	DC-DC	RTC LDO	DCXO /DPLL
ACTIVE[1]	ACTIVE	ACTIVE/OFF	ACTIVE/OFF	ON	ON	ON	ON
STANDBY - LP Voice[2]	SLEEP	OFF	ACTIVE	ON	ON	ON	ON
STANDBY - MAD[3]	RESET/OFF	OFF	OFF	ON	ON/LP[7]	ON	ON/OFF
STANDBY - Network[4]	RESET/OFF	OFF	OFF	ON	ON/LP[7]	ON	ON/OFF
HIBERNATION[5]	OFF	OFF	OFF	ON	OFF	ON	OFF
SHUT DOWN[6]	OFF	OFF	OFF	OFF	OFF	OFF	OFF

NOTE

1. M33 stay active, RISC-V/DSP/Wi-Fi/BT stay active or off.
2. M33/RISC-V keep powered off, DSP stay active. Waiting for waking up Wi-Fi/BT on schedule and M33/RISC-V/DSP/Wi-Fi/BT through voice detection.
3. M33 keeps reset or powered off, RISC-V/DSP keep powered off. Waiting for Wi-Fi/BT to be waked up on schedule and M33/RISC-V/DSP/Wi-Fi/BT to be waked up through acoustic energy detection.
4. M33 keeps reset or powered off, RISC-V/DSP keep powered off. Waiting for Wi-Fi/BT to be waked up on schedule (Wi-Fi/BT may be in DTIM or OFF state).
5. Only RTC on. Waiting for RTC timer or wake-up IO to be waked up.
6. The chip keeps in reset state.
7. “LP” means low power mode.

Figure 3-1 Low-power System Status Switch



3.2.4 Programming Guidelines

3.2.4.1 DSP/RV Low-power Control

DSP/RV have their own independent power domain whose switches are controlled by main CPU. If main program judges that there is no mission, RV/DSP will be switched off to save power; if not, RV/DSP will be switched on.

Switch off DSP: Main program sets SYS_LP_CTRL[12] to 0. Read SYS_LP_STA[13] until the value of DSP_SLEEP is 1, and then DSP is completely switched off. Read SYS_LP_STA[9] until the value of RV_SLEEP is 1, and then RV is completely switched off.

Switch on DSP: Main program sets SYS_LP_CTRL [12] to 1. Read SYS_LP_STA[12] until the value of DSP_ALIVE is 1, and then DSP is completely switched on. Read SYS_LP_STA[8] until the value of RV_ALIVE is 1, and then RV is completely switched off.

3.2.4.2 RTC Wake-Up Timer

RTC Wake-Up Timer is one of the wake-up source to wake up system in hibernation mode.

Step 1 The software configures the of RTC_WUPTIME_CTRL[RTC_WAKE_TIMER_CNT] based on the needed timing length.

Step 2 The software sets the RTC_WUPTIME_CTRL[RTC_WAKE_TIMER_EN] as 1. The wake-up timer starts working.

Step 3 System enters the hibernation mode.

Step 4 Timer counts to the set time and triggers PMU to wake up system.

3.2.4.3 RTC Wake-Up IO

RTC Wake-Up IO is one of the wake-up source to wake up system in hibernation mode.

Step 1 Configure the RTC IO Wake-up Debounce Clock Register to control debounce clock frequency by setting clock division ratio, which is used to debounce and debur before sampling.

Step 2 Configure the RTC IO Wake-up Debounce cycle Register to control the cycles of debounce clock.

Step 3 Enable the responding bits of IO INT Enable Register if necessary. If they are enabled, CPU will receive the IO interrupts when system is booted.

Step 4 Configure the RTC IO Wake-up Enable Register to enable the corresponding wake-up enable signals.

3.2.5 Register List

Module Name	Base Address
PMC	0x40051400

Register Name	Offset Address	Description
DCXO_STABLE_REF_TIME	0x0040	DCXO Stable Reference Time
DPLL_STABLE_REF_TIME	0x0044	DPLL Stable Reference Time
LDO_STABLE_REF_TIME	0x0048	LDO Stable Reference Time
DIGITAL_SWITCH_REF_TIME	0x004C	Digital Switch Reference Time
BANDGAP_STABLE_REF_TIME	0x0050	Band Gap Stable Reference Time
DCDC_STABLE_REF_TIME	0x0054	DCDC Stable Reference Time
SYS_LOW_POWER_CTRL	0x0100	System Low-Power Control
SYS_LOW_POWER_STATUS	0x0104	System Low-Power Status
RTC_WUP_TIMER_CNT	0x0108	RTC wakeup timer counter
RTC_WUP_TIMER_CTRL	0x010C	RTC wakeup timer control
RTC_IO_WAKE_EN	0x0110	RTC IO Wakeup Enable
RTC_IO_WAKE_DEB_CLK	0x0114	RTC IO Wakeup debounce clock
RTC_IO_WAKE_ST	0x0118	RTC IO Wakeup Status
RTC_IO_HOLD_CTRL	0x011C	RTC IO Hold Control
RTC_IO_WUP_GEN	0x0120	RTC IO Wakeup Global Enable
RTC_IO_WUP_DEB_CYCLES0	0x0124	RTC IO Wakeup debouce clock cycles 0
RTC_IO_WUP_DEB_CYCLES1	0x0128	RTC IO Wakeup debouce clock cycles 1
VDD_APP_PWROFF_GATING	0x0180	VDD_APP Power Off Gating Register
POWERCTRL_CFG_REG	0x0184	Power control configuration register

3.2.6 Register Description

3.2.6.1 0x0040 DCXO Stable Reference Time Register (Default Value: 0x0000_0080)

Offset: 0x0040			Register Name: DCXO_STABLE_REF_TIME
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9: 0	R/W	0x80	DCXO Stable Reference Time REF_TIME = T_{LFCLK} * n Typical Value <= 500 us Set Value = 14 us * 128 = 1.8 ms

3.2.6.2 0x0044 DPLL Stable Reference Time Register (Default Value: 0x0000_000C)

Offset: 0x0044			Register Name: DPLL_STABLE_REF_TIME
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7: 0	R/W	0xC	DPLL Stable Reference Time REF_TIME = T_{LFCLK} * n Typical Value <= 50 us Set Value = 14 us * 12 =168 us

3.2.6.3 0x0048 LDO Stable Reference Time Register (Default Value: 0x1400_0308)

Offset: 0x0048			Register Name: LDO_STABLE_REF_TIME
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x1400	LDO_SETTLING_TIME_HFCLK LDO Stable Reference Time in number of HF_CLK REF_TIME = T_{HFCLK} * n Typical Value <= 100 us Set Value = 25 ns * 5120 = 128 us (@40 MHz OSC)
15:12	/	/	/
11:8	R/W	0x3	RFIP_LDO_DIG Stable Reference Time REF_TIME = T_{LFCLK} * n Typical Value <= 10 us Set Value = 14 us * 3 = 42 us
7: 0	R/W	0x8	LDO_SETTLING_TIME_LFCLK LDO Stable Reference Time in number of LFCLK REF_TIME = T_{LFCLK} * n Typical Value <= 100 us Set Value = 16 us * 8 = 128 us (RCOSLP frequency between 1 MHz-2 MHz)

3.2.6.4 0x004C Digital Switch Reference Time Register (Default Value: 0x0002_0020)

Offset: 0x004C			Register Name: DIGITAL_SWITCH_REF_TIME
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	R/W	0x2	Reset Up Reference Time Control isolation/retention/clk_gate/reset settling time Typical Value >= 1 cycles Set Value = 25 ns * 1 = 50 ns (@40 MHz OSC)
15:10	/	/	/
9: 0	R/W	0x20	MTCMOS_SW_Settling_Time One MTCMOS Power Switch Chain Reference Time.There are 8 Chains in each domain, total SW setting time is 8*MTCMOS_SW_Settling_Time REF_TIME = T_{HFCLK} * n Typical Value >= 200 ns Set Value = 25 ns * 32 = 0.8 us (@40 MHz OSC)

3.2.6.5 0x0050 BANDGAP Stable Reference Time Register (Default Value: 0x0000_0017)

Offset: 0x0050			Register Name: BG_STABLE_REF_TIME
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7: 0	R/W	0x17	BG Stable Reference Time REF_TIME = T_{LFCLK} * n Typical Value <= 100 us Set Value = 14 us * 23 = 322 us

3.2.6.6 0x0054 DCDC Stable Reference Time Register (Default Value: 0x0080_000C)

Offset: 0x0054			Register Name: DCDC_STABLE_REF_TIME
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	R/W	0x80	DCDC to PWM Mode Stable Reference Time REF_TIME = T_{LFCLK} * n Typical Value <= 500 us Set Value = 14 us * 128 = 1.8 ms
15:10	/	/	/
9: 0	R/W	0xC	Other Work Mode Switch Stable Reference Time REF_TIME = T_{LFCLK} * n Typical Value <= 50 us Set Value = 14 us * 16 = 168 us

3.2.6.7 0x0100 System Low-Power Control Register (Default Value: 0x0000_0002)

Offset: 0x0100			Register Name: SYS_LP_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
12	R/W	0x0	DSP_WUP_EN DSP system wake up enable 1: Wake-up DSP system 0: Shut-down DSP system
11:9	/	/	/
8	R/W	0x0	RV_WUP_EN RISC-V system wake up enable 1: Wake-up RISC-V system 0: Shut-down RISC-V system
7:1	/	/	/
0	R/W	0x0	STANDBY_HIBERNATE_MODE_SEL This bit decide IC goes into hibernation-mode or standby-mode when APP goes into sleep 1: AON_LDO will be disabled when the APP system goes into sleep(hibernation-mode) 0: AON_LDO is kept on when the APP system goes into sleep(standby-mode)

3.2.6.8 0x0104 System Low-Power Status Register (Default Value: 0x0000_0000)

Offset: 0x0104			Register Name: SYS_LP_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:14	/	/	/
13	R	0x0	DSP_SLEEP 1: DSP system is in SLEEP mode 0: DSP system is not in SLEEP mode
12	R	0x0	DSP_ALIVE 1: DSP system is alive 0: DSP system is not alive
11:10	/	/	/
9	R	0x0	RV_SLEEP 1: RV system is in SLEEP mode 0: RV system is not in SLEEP mode
8	R	0x0	RV_ALIVE 1: RV system is alive 0: RV system is not alive
7	/	/	/

Offset: 0x0104			Register Name: SYS_LP_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
6	R	0x0	APP_SLEEP 1: APP system is in SLEEP mode 0: APP system is not in SLEEP mode
5	R	0x0	APP_DEEPSLEEP 1: APP system is in DEEPSLEEP mode 0: APP system is not in DEEPSLEEP mode
4	R	0x0	APP_ALIVE 1: DSP system is alive 0: DSP system is not alive
3	/	/	/
2	R	0x0	WLAN_SLEEP 1: WLAN system is in SLEEP mode 0: WLAN system is not in SLEEP mode
1	R	0x0	WLAN_Standby 1: WLAN system is in Standby mode 0: WLAN system is not in Standby mode
0	R	0x0	WLAN_ALIVE 1: DSP system is alive 0: DSP system is not alive

3.2.6.9 0x0108 RTC Wakeup Timer Control Register (Default Value: 0x0000_0000)

Offset: 0x0108			Register Name: RTC_WUPTIME_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	RTC_WAKE_TIMER_EN
30: 0	R	0x0	RTC_WAKE_TIMER_CNT **Note**: *This is a free running counter and when the value of this counter equals to that of RTC_WAKE_TIMER_VAL, a wakeup signal will be issued to the Wakeup Interrupt Controller.* *If RTC_WAKE_TIMER_EN is set to '0', this counter will be reset to 0 and stop counting.*

3.2.6.10 0x010C RTC Wakeup Timer Value Register (Default Value: 0x0000_0000)

Offset: 0x010C			Register Name: RTC_WAKE_TIMER_VAL
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x010C			Register Name: RTC_WAKE_TIMER_VAL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	RTC_WAKEUP_TIMER_IRQ_STATUS 0: The RTC wakeup timer irq is not pending 1: The RTC wakeup timer irq is pending **Note**: *This is a write-1-to-clear register. This is an asynchronies operation and this bit will not become '0' immediately after writing '1' to clear the irq state. We should read the register until the value of this bit becomes '0', which means the irq pending state is actually cleared.*
30: 0	R/W	0x0	RTC_WAKE_TIMER_VAL

3.2.6.11 0x0110 RTC IO Wakeup Enable Register (Default Value: 0x0000_0000)

Offset: 0x0110			Register Name: RTC_WAKE_IO_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	/	0x0	RTC_WAKEUP_IOx_MODE 1: Positive Edge 0: Negative Edge bit0- bit14: WUPIO0 – WUPIO9
15:10	/	/	/
9: 0	R/W	0x0	RTC_WAKEUP_IOx_ENB 1: IOx wakeup detection enable 0: IOx wakeup detection disable bit0- bit14: WUPIO0 – WUPIO9

3.2.6.12 0x0114 RTC IO Wakeup Debounce Clock Register (Default Value: 0x0000_0000)

Offset: 0x0114			Register Name: RTC_WAKE_IO_DEB_CLK
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	RTC_WAKEUP_DEB_CLK_SET1 IO wakeup debounce clk1 select, the DEB_CLK1 frequency CLK_SEL1<=14, Freq=LFCLK/(2^{CLK_SEL1}); CLK_SEL1==15, reserved.
27:24	R/W	0x0	RTC_WAKEUP_DEB_CLK_SET0 IO wakeup debounce clk0 select, the DEB_CLK0 frequency CLK_SEL0<=14, Freq=LFCLK/(2^{CLK_SEL0}); CLK_SEL0==15, reserved.
23:16	/	/	/
15:10	/	/	/

Offset: 0x0114			Register Name: RTC_WAKE_IO_DEB_CLK
Bit	**Read/Write**	**Default/Hex**	**Description**
9: 0	R/W	0x0	RTC_WAKEUP_IOx_CLK _SEL 0: IOx select DEB_CLK0 1: IOx select DEB_CLK1 bit0-bit14: WUPIO0 – WUPIO13

3.2.6.13 0x0118 RTC IO Wakeup Status Register (Default Value: 0x0000_0000)

Offset: 0x0118			Register Name: RTC_WAKE_IO_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9: 0	R/W1C	0x0	RTC_WAKEUP_IOx_ST 1: Wakeup event is detected on IOx 0: Nothing bit0-bit14: WUPIO0 – WUPIO13 **Note**: Write *1 to clear the status.*

3.2.6.14 0x011C RTC IO Hold Control Register (Default Value: 0x0000_0000)

Offset: 0x011C			Register Name: RTC_IO_HODL_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9: 0	R/W	0x0	RTC_WAKEUP_IOx_ST 1: IO Configure Hold Enable 0: IO Configure Hold Disable bit0- bit14: WUPIO0 – WUPIO13 **Note**: *When enabling the IO hold function, the configuration (IEN, OEN, PULL, ODAT) of the relative IO will be hold to the current state even if the GPIO controller is powered down or the origin configuration is modified.*

3.2.6.15 0x0120 RTC IO Wakeup Global INT Enable Register (Default Value: 0x0000_0000)

Offset: 0x0120			Register Name: RTC_IO_WUP_GEN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	RTC_IO_WUP_GEN 1: Enable IO Wakeup Interrupt 0: Disable IO Wakeup Interrupt

3.2.6.16 0x0124 RTC IO Wakeup Debounce Cycles Register0 (Default Value: 0x0000_0000)

Offset: 0x0124	Register Name: RTC_WAKEUP_IO_DEB_CYCLES0

Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	RTC_WAKEUP_IO_DEB_CYCLES0 Bit[4N+3:4*N]: IO[N] debounce cycles (N: 0~7) WUPIO0 – WUPIO7 Select clock for debounce (configured by register RTC_WAKE_IO_DEB_CLK), then the IO input debounce clock cycles is DEBC_CYCLE[3: 0]+1.

3.2.6.17 0x0128 RTC IO Wakeup Debounce Cycles Register1 (Default Value: 0x0000_0000)

Address: 0x0128			Register Name: RTC_WAKEUP_IO_DEB_CYCLES1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7: 0	R/W	0x0	RTC_WAKEUP_IO_DEB_CYCLES1 Bit[4(N-8) + 3:4*(N-8)]: IO[N] debounce cycles (N:8~9) WUPIO8 – WUPIO9 Select clock for debounce (configured by register RTC_WAKE_IO_DEB_CLK), then the IO input debounce clock cycles is DEBC_CYCLE[3: 0]+1.

3.2.6.18 0x0180 VDD_APP Power Off Gating Register (Default Value: 0x0000_0004)

Offset: 0x0180			Register Name: VDD_APP_PWROFF_GATING
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x1	DDR_PHY_POCH disable DDR_PHY IO output when VDD/VDDQ(1.2v, supplied by APP_LDO/AON_LDO) turns off. 0: Enable IO output 1: Disable IO output **Note**: *When DDR_PHY_POCH is 1, the IO output will be disabled. The desired IO value is obtained by pulling up/down resistor. Set 0 after AON_LDO and APP_LDO is turned on and stable. Set 1 before turning on AON_LDO.*
1	R/W	0x0	DDR_PHY_IO_LATCH0 Latch The DRAM IO (Except ZQ PAD) Output when VDD_APP power switch is turned off. 0: Not latch 1: Latch **Note**: *This bit should be set to 1 after PSRAM change to auto self-refresh and before VDD_APP power switch turns off in super standby mode.*

Offset: 0x0180			Register Name: VDD_APP_PWROFF_GATING
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	DDR_PHY_IO_LATCH1 Latch the DRAM IO (Only For ZQ PAD) Output when VDD_APP power switch is turned off. 0: Not latch 1: Latch **Note**: *After PSRAM changes to auto self-refresh, this bit should be set to 1. Then, VDD_APP power switch is turned off in super standby mode.*

3.2.6.19 0x0184 Power_ctrl Configure Register (Default Value: 0x0000_05CE)

Offset: 0x0184			Register Name: POWERCTRL_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:15	/	/	/
14	R/W	0x0	DSP_LDO_SHUT_DOWN This bit is used if en external LDO outside IC is used to replace DSP_LDO. 0: Do nothing. 1: Override the control of DSP_LDO by PMU, and disable it.
13	R/W	0x0	BT_RSTN_HIGHLVEL_WAKEUP 0: Wakeup in the posedge of BT_RSTN. 1: Wakeup in the high level of BT_RSTN.
12	R/W	0x0	WLAN_IRQ_HIGHLVEL_WAKEUP 0: WLAN_IRQ can only wakeup powerctrl through WIC. 1: Wakeup in the high level of WLAN_IRQ while BT_RSTN is low.
11	R/W	0x0	IGNORE_WICWAKEUP 0: WIC WAKEUP can take into effect at any time. 1: WIC WAKEUP can't take into effect when BT_RSTN was low.
10	R/W	0x1	IGNORE_WICENACK 0: Use WICENACK in sleep/wakeup process. 1: Do not use WICENACK in sleep/wakeup process.
9	R/W	0x0	WAKEUP_MODE 0: Wakeup in the posedge of interrupt. 1: Wakeup in the high level of interrupt.
8	R/W	0x1	RTC_ALARM_WUP_TO_POWERCTRL 0: RTC alarm0/1 interrupt can not wake up powerctrl directly. 1: RTC alarm0/1 interrupt can wake up powerctrl directly.

Offset: 0x0184			Register Name: POWERCTRL_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
7:6	R/W	0x3	RCOSC_CALIB_START_SRC RCOSC Calibration Start Source. 00: APP sleep 01: WLAN sleep 10: APP and WLAN sleep 11: AON_LDO shut down
5	R/W	0x0	CPU_SLEEP_NO_RETENTION 0: When CPU enters the sleep mode, APP RAM can be retained. 1: When CPU enters the sleep mode, no need to retain APP RAM.
4	R/W	0x0	BT_SLEEP_RETENTION_OVR 0: BT reset retention should not overwrite CPU retention. 1: BT reset retention could overwrite CPU retention.
3	R/W	0x1	BLE_SLEEP_TIMER_IRQ_TO_POWERCTRL 0: BLE Sleep Timer interrupt can not wake up powerctrl directly. 1: BLE Sleep Timer interrupt can wake up powerctrl directly.
2	R/W	0x1	WUPTIMER_IRQ_TO_POWERCTRL 0: Wakeup Timer interrupt can not wake up powerctrl directly. 1: Wakeup Timer interrupt can wake up powerctrl directly.
1	R/W	0x1	WUPIO_IRQ_TO_POWERCTRL 0: Wakeup IO interrupt can not wake up powerctrl directly. 1: Wakeup IO interrupt can wake up powerctrl directly.
0	R/W	0x0	BT_RSTN_WAKEUPPIN 0: BT reset PAD can not be used as wakeup pin. 1: BT reset PAD can be used as wakeup pin. Positive edge means wakeup, and negative edge means going to sleep.

3.3 Generic Power, Reset, and Clock Management

3.3.1 Overview

The Generic Power, Reset, and Clock Management (GPRCM) manages the power, reset and input clocks for this system and is placed in RTC domain.

The GPRCM includes the following features:

- Manages the power of this system
- Manages the reset of each system
- Manages the OSC clock

NOTE

All registers should be placed in RTC power domain and will keep the value unchanged when the system is powered off by disabling the internal DCDC and LDO. The system can automatically start up when detecting external events like GPIO or power events. The value of this register should be read correctly after the system starts up.

3.3.2 Functional Description

3.3.2.1 Power Module Control

The PMU is integrated with DCDC, LDO, BG, and other power modules, which reserve control interfaces like TRIM, voltage-select, work mode to be configured by system. These configurations can be completed through 0x0000~0x004C registers within the GPRCM, and the configured value will remain unchanged even in the hibernation or standby and other low-power statuses.

3.3.2.2 VBAT Monitor

VBAT monitor is used to monitor the external power and voltage of VBAT. When the voltage is lower than the threshold, it will generate interrupts to notice CPU to initiate power failure protection or reset the whole CHIP to ensure the circuit only works in the normal voltage range.

Detailed Configuration:

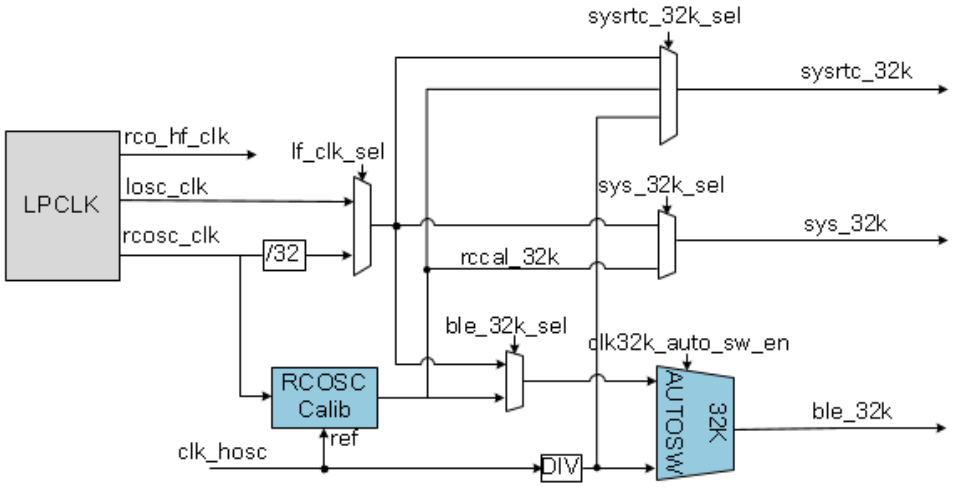
a. Configure the threshold voltage VBAT_MON_VSEL.

b. Configure LOW_VOL_MODE_SEL and determine how the system responds when VBAT is lower than its threshold.

c. Enable VBAT_MON_EN.

3.3.2.3 LFCLK

LFCLK is the low-frequency 32K clock used by system. As shown below, the input source is LPCLK, and the output includes sysrtc_32k, sys_32k, and ble_32k. The MUX selection of input and output is determined by the SYS_LFCLK_CTRL.



3.3.2.4 Reset de-bounce Configuration

The signal of system function reset will be filtered through the de-bounce circuit and then transmitted to the internal circuit. The clock source of De-bounce is 32K, and its division and cycle number can be set. The lower divided clock frequency and the larger cycle numbers contribute to the longer effective resetting time.

3.3.2.5 BLE RCOSC Calibration

BT wireless transmission requires high precision for 32K low frequency clock. However, the internal RCOSC precision cannot meet this requirement. Therefore, the RCOSC calibration is integrated to measure the clock frequency of RCOSC with high-frequency DCXO. Then, the RCOSC clock will be divided into fractions to obtain the 32kHz or 32.576kHz accurate clock.

As the output clock frequency of RCOSC has larger temperature drift, it requires calibration on schedule. The calibration time can be configured by the software. Meanwhile, calibration can still be performed even in the low-power state. When the system enters the hibernation state, system will be waked up and enable DCXO on schedule and then go back to the hibernation state after calibration.

3.3.3 Register List

Module Name	Base Address
GPRCM	0x40050000

Register Name	Offset Address	Description
DIG_BG_CTRL	0x0000	Digital Bandgap Control Register
DCDC_CTRL0	0x0004	DCDC Control Register0

Register Name	Offset Address	Description
DCDC_CTRL1	0x0008	DCDC Control Register1
DCDC_LDO_MODE_SW_SEL	0x000C	DCDC_LDO Work Mode Control Register
RTC_LDO_CTRL	0x0020	RTC LDO Control Register
EXT_LDO_CTRL	0x0024	EXT LDO Control Register
AON_LDO_CTRL	0x0040	AON LDO Control Register
APP_LDO_CTRL	0x0044	APP LDO Control Register
DSP_LDO_CTRL	0x004C	DSP LDO Control Register
VBAT_MON_CTRL	0x0060	VBAT Monitor Control Register
USB_BIAS_CTRL	0x0064	USB Bias Control Register
SYS_LFCLK_CTRL	0x0080	System Low Frequency Clock Control
SYS_LFCLK_STATUS	0x0084	System Low Frequency Clock Status
RST_DBC_CTRL	0x00C0	RESET pin Debounce Control Register
BOOT_MEDIA_RECORD	0x00C8	Record which kind of media for booting
SYS_RCOSC_FREQ_DET	0x0140	System RC-OSC Frequency Detect
BLE_RCOSC_CALIB_REG0	0x0144	RCOSC calibration control register0
BLE_RCOSC_CALIB_REG1	0x0148	RCOSC calibration control register1
BLE_CLK32K_SWITCH_REG0	0x014C	Clock 32k auto switch register0
BLE_CLK32K_SWITCH_REG1	0x0150	Clock 32k auto switch register1
BT_WLAN_CONN_MODE_CTRL	0x0180	BT_WLAN Connect mode control
CFG_IO_STATUS	0x0188	Config IO Status
WLAN_HIF_OV_CTRL	0x018C	WLAN HIF override control register
SRAM_CFG	0x0190	SRAM configure register
FLASH_ENCRYPT_AES_NONCE0	0x0194	Flash encrypt module AES nonce low bits register
FLASH_ENCRYPT_AES_NONCE1	0x0198	Flash encrypt module AES nonce high bits register
CPU_BOOT_FLAG	0x01C0	CPU boot flag
CPU_BOOT_ADDR	0x01C4	CPU boot address
CPU_BOOT_ARG	0x01C8	CPU boot argument
DSP_BOOT_FLAG	0x01CC	DSP boot flag
DSP_BOOT_ADDR	0x01D0	DSP boot address
DSP_BOOT_ARG	0x01D4	DSP boot argument
RV_BOOT_FLAG	0x01D8	RV boot flag
RV_BOOT_ADDR	0x01DC	RV boot address
RV_BOOT_ARG	0x01E0	RV boot argument
SYSTEM_PRIV_REG0	0x0200	SYSTEM Private Register
SYSTEM_PRIV_REG1	0x0204	SYSTEM Private Register
SYSTEM_PRIV_REG2	0x0208	SYSTEM Private Register
SYSTEM_PRIV_REG3	0x020C	SYSTEM Private Register
SYSTEM_PRIV_REG4	0x0210	SYSTEM Private Register
SYSTEM_PRIV_REG5	0x0214	SYSTEM Private Register
SYSTEM_PRIV_REG6	0x0218	SYSTEM Private Register
SYSTEM_PRIV_REG7	0x021C	SYSTEM Private Register

3.3.4 Register Description

3.3.4.1 0x0000 DIG BANDGAP Control Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: DIG_BG_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3: 0	R/W	0x0	BG_OUTPUT_TRIM BG output trimming **Note**: *eFuse override when wake-up*

3.3.4.2 0x0004 DCDC Control Register0 (Default Value: 0x0002_0000)

Offset: 0x0004			Register Name: DCDC_CTRL0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17:12	R/W	0x20	DCDC_VOL_SEL (1.0 V-2.5 V) DCDC(DC-DC) voltage select, 25 mV step, 1.8 V by default; 1.0 V-2.5 V range **Note**: *This bit will only be reset to the default value in power-on-reset operation. When the system is waked up from the deep sleep mode in some external events, this bit will keep the value set by the last write operation.*
11:10	/	/	/
9	R/W	0x0	AONLDO_OFF_DCDC_ON 0: When AON_LDO is turned off, DCDC will be turned off either. 1: When AON_LDO is turned off, DCDC will remain turned on.
8	R/W	0x0	SYS_STANDBY_DCDC_OFF 0: When system is in standby mode, DCDC can remain turned on 1: When system is in standby mode, DCDC should be turned off
7	R/W	0x0	DCDC_PWM_SEL_EN Enable DCDC_PWM_SEL function.
6: 4	R/W	0x0	DCDC_PWM_SEL DCDC_PWM could from the following source: 000: clk_RfipDpllStart 001: clk_RfipDcxoStart 010: WLAN PHY enable 011: APP CPU enable 100: WLAN CPU enable Others: BLE RF enable
3	/	/	/
2	R/W	0x0	OVR_DCDC_DETCT Overwrite DCDC_DETECT by software

Offset: 0x0004			Register Name: DCDC_CTRL0
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x0	OVR_DCDC_DETCT_VALUE Overwrite the value of DCDC_DETECT
0	R	0x0	DCDC_DETECT DCDC detect value from powerctrl

3.3.4.3 0x0008 DCDC Control Register1 (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: DCDC_CTRL1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	R/W	0x0	DCDC_OCP DCDC over current protection threshold tuning Default Value: 400mA
28:24	R/W	0x0	DCDC_BGTR [4:1]: for DCDC BANDGAP trimming [0]: for OCP disable
22:16	R/W	0x0	DCDC_FTR [4:0]: for switching frequency tuning [6:5]: for option to decrease frequency
15:8	R/W	0x0	DCDC_TEST_ITEM_SEL
7:4	R/W	0x0	DCDC_TEST_MODE
3:2	R/W	0x0	DCDC_BYPASS 00: DCDC bypass disable 01: DCDC power pmos 1/6 bypass when PFM mode 1x: DCDC all power pmos bypass
1	/	/	/
0	R	0x0	DCDC_WORK_MODE 0: PWM Mode 1: PFM Mode **Note**: The *DCDC starts up from PWM mode by default.* *This field is read-only and can only be changed by POWERCTRL Module.*

3.3.4.4 0x000C DCDC_LDO Work Mode Override Register (Default Value: 0x0000_0000)

Offset: 0x000C			Register Name: DCDC_MODE_SW_SEL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
rtc7	R/W	0x0	SW_DSP_LDO_LQ_MODE 1: DSP_LDO select LQ mode 0: DSP_LDO select normal mode

Offset: 0x000C			Register Name: DCDC_MODE_SW_SEL
Bit	**Read/Write**	**Default/Hex**	**Description**
6	R/W	0x0	SW__APP_LDO_LQ_MODE 1: APP_LDO select LQ mode 0: APP_LDO select normal mode
5	R/W	0x0	SW_AON_LDO_LQ_MODE 1: AON_LDO select LQ mode 0: AON_LDO select normal mode
4	R/W	0x0	LDO_MODE_SW_OVERIDE 1: AON/APP/DSP/RV_LDO work mode select by this register 0: AON/APP/DSP/RV_LDO work mode select by power control
3:2	/	/	/
1	R/W	0x0	DCDC_PFM_MODE 1: DCDC select PFM mode 0: DCDC select PWM mode
0	R/W	0x0	DCDC_MODE_SW_OVERIDE 1: DCDC PWM/PFM mode select by this register 0: DCDC PWM/PFM mode select by power control

3.3.4.5 0x0020 RTC LDO Control Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: RTC_LDO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6:4	R/W	0x0	SEL_RTCLDO_SLP RTC_LDO output voltage level select when system is in hibernation mode (0.6V-1.3V). The default voltage is 1.075 V. 000: 1.075 V 001: 0.975 V 010: 0.875 V 011: 0.775 V 100: 0.675 V 101: 0.575 V 110: 1.175 V 111: 1.275 V
3	/	/	/

Offset: 0x0020			Register Name: RTC_LDO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
2: 0	R/W	0x0	SEL_RTCLDO_ACT RTC_LDO output voltage level select when IC is in active mode (0.6V-1.3V). The default voltage is 1.075 V. 000: 1.075 V 001: 0.975 V 010: 0.875 V 011: 0.775 V 100: 0.675 V 101: 0.575 V 110: 1.175 V 111: 1.275 V

3.3.4.6 0x0024 EXT LDO Control Register (Default Value: 0x0000_0002)

Offset: 0x0024			Register Name: EXT_LDO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	EXT_LDO_LQB EXT_LDO lq mode control 0: For lq mode 1: For active mode
23	R/W	0x0	EXT_LDO_BYPASS 0: EXT_LDO is not bypassed 1: EXT_LDO is bypassed
22:21	R/W	0x0	EXT_LDO_SW_TRIM Bit[1: 0]: option select of the regulate cycle when EXT_LDO in switch mode
20	R/W	0x0	EXT_LDO_SW_MODE 0: EXT_LDO doesn't enter switch mode 1: EXT_LDO enters switch mode
19:6	/	/	/
5:4	R/W	0x0	EXT_LDO_VOL_SEL ext_ldo output value regulate; default value is 3.3 V 00=3.3 V 01=3.1 V 10=2.8 V 11=2.5 V 2.8 V only for option
3:2	/	/	/

Offset: 0x0024			Register Name: EXT_LDO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
1: 0	R/W	0x2	EXT_LDO_EN EXT_LDO enable 00: Keep off 01: EXT_LDO is controlled by PMU 1x: Always on

3.3.4.7 0x0040 AON LDO Control Register (Default Value: 0x0008_2B51)

Offset: 0x0040			Register Name: AON_LDO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/
19	R/W	0x1	TESTPEAK_LDO enable the auxiliary circuit to avoid output large ripple during select LDO work 1: Enable 0: Disable
18	R	0x0	STARTICC_LDO Decrease over-current protection threshold at startup 1: Enable 0: Disable **Note**: *This field is set by POWERCTRL module.*
17	R/W	0x0	TESTICC_LDO Over current protection enable 1: Enable 0: Disable
16:14	/	/	/
13:9	R/W	0x15	LDO_VOL_SEL_ACT Select the AON_LDO output voltage level when AON domain is in active mode (0.6 V-1.375 V). The default voltage is 1.125 V. Output: 0.6 V + LDO_VOL_SEL*0.025 V
8:4	R/W	0x15	LDO_VOL_SEL_RET AON_LDO output voltage level select when AON domain in sleep mode (0.6 V-1.375 V). The default voltage is 1.125 V. Output: 0.6 V + LDO_VOL_SEL*0.025 V
3:2	/	/	/
1	R	0x0	LDO_LQ_MODE AON_LDO LQ mode enable 0: Active mode 1: LQ mode **Note**: *This field is set by POWERCTRL module*

Offset: 0x0040			Register Name: AON_LDO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R	0x1	LDO_EN LDO enable control 1: Enable 0: Disable **Note**: *This field is set by POWERCTRL module*

3.3.4.8 0x0044 APP LDO Control Register (Default Value: 0x0008_2B51)

Offset: 0x0044			Register Name: APP_LDO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/
19	R/W	0x1	TESTPEAK_LDO Enable the auxiliary circuit to avoid output large ripple during select LDO work 1: Enable 0: Disable
18	R	0x0	STARTICC_LDO Decrease over-current protection threshold at startup 1: Enable 0: Disable **Note**: *This field is set by POWERCTRL module.*
17	R/W	0x0	TESTICC_LDO Over current protection enable 1: Enable 0: Disable
16:14	/	/	/
13:9	R/W	0x15	LDO_VOL_SEL_ACT APP_LDO output voltage level select when APP domain in active mode (0.6 V-1.375 V). The default voltage is 1.125 V. Output: 0.6 V + LDO_VOL_SEL*0.025V
8:4	R/W	0x15	LDO_VOL_SEL_RET APP_LDO output voltage level select when APP domain in sleep mode (0.6 V-1.375 V). The default voltage is 1.125 V. Output: 0.6 V + LDO_VOL_SEL*0.025V
3:2	/	/	/
1	R	0x0	LDO_LQ_MODE APP_LDO LQ mode enable 0: Active mode 1: LQ mode **Note**: *This field is set by POWERCTRL module.*

Offset: 0x0044			Register Name: APP_LDO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R	0x1	LDO_EN LDO enable control 1: Enable 0: Disable **Note**: *This field is set by POWERCTRL module.*

3.3.4.9 0x004C DSP LDO Control Register (Default Value: 0x0008_0150)

Offset: 0x004C			Register Name: DSP_LDO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/
19	R/W	0x1	TESTPEAK_LDO Enable the auxiliary circuit to avoid output large ripple during select LDO work 1: Enable 0: Disable
18	R	0x0	STARTICC_LDO Decrease over current protection threshold at start up 1: Enable 0: Disable **Note**: *This field is set by POWERCTRL module*
17	R/W	0x0	TESTICC_LDO Over current protection enable 1: Enable 0: Disable
16:9	/	/	/
8:4	R/W	0x15	LDO_VOL_SEL DSP_LDO output voltage level select (0.6 V-1.375 V). The default voltage is 1.125 V. Output: 0.6 V + LDO_VOL_SEL*0.025V
3:2	/	/	/
1	R	0x0	LDO_LQ_MODE DSP_LDO LQ mode enable 0: Active mode 1: lq mode **Note**: *This field is set by POWERCTRL module.*
0	R	0x0	LDO_EN LDO enable control 1: Enable 0: Disable **Note**: *This field is set by POWERCTRL module.*

3.3.4.10 0x0060 VBAT Monitor Control Register (Default Value: 0x0000_0000)

Offset: 0x0060			Register Name: VBAT_MON_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8:4	R/W	0x0	VBAT_MON_VSEL Regulate monitor voltage threshold 1.4+vsel*0.1V VBAT monitor output will be 1 when VBAT voltage is lower than threshold.
3	/	/	/
2	R/W	0x0	LOW_VOL_MODE_SEL Control how to do when VBAT voltage lower than threshold 1: Reset whole chip 0: Give IRQ to CPU
1	R/W	0x0	VBAT_MON_SW_MODE VBAT monitor enters switch mode
0	R/W	0x0	VBAT_MON_EN VBAT monitor enable 1: Enable 0: Disable

3.3.4.11 0x0064 USB Bias Control Register (Default Value: 0x0000_0100)

Offset: 0x0064			Register Name: USB_BIAS_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
9:4	R/W	0x10	USB_BIAS_TRIM Bit[4: 0]: for USB inp_usb_25u trim, Bit[5]: no use
3:1	/	/	/
0	R/W	0x0	EN_BIAS_USB USB BIAS enable. Set 0 if USB is disabled to save power at sleep status. 0: Disable USB BIAS 1: Enable USB BIAS

3.3.4.12 0x0080 System LFCLK Control Register (Default Value: 0x4000_0000)

Offset: 0x0080			Register Name: SYS_LFCLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	LOSC_EN 1: Enable the 32768 Oscillator 0: Disable the 32768 Oscillator

Offset: 0x0080			Register Name: SYS_LFCLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
30	R/W	0x1	RCOSC_EN 1: Enable the RC Oscillator 0: Disable the RC Oscillator
29	R/W	0x0	RC_HF_EN 1: Enable the RC_HF Oscillator 0: Disable the RC_HF RC Oscillator
28	R/W	0x0	SYS_32K_SEL The source of 32kHz clock select 1: The clock from RCOSC Calibration output 0: The LFCLK output
27	R/W	0x0	BLE_SEL_RCCAL_32K The source of 32kHz clock select 1: The clock from RCOSC Calibration output 0: The LFCLK output
26:25	R/W	0x0	SYSRTC_32K_SEL 00: Select LFCLK as SYSRTC function clock 01: Select rco_calib_clk as SYSRTC function clock 10: Select hosc_div32k as SYSRTC function clock 11: Reserved
24	R/W	0x0	LFCLK_SRC_SEL Low frequency clock source select 0: Source the LFCLK from the internal RC oscillator. 1: Source the LFCLK from the external 32768 crystal oscillator
23:16	R/W	0x0	RCO_HF_TRIM Trim for RCOSC_HF
15:3	R/W	0x0	PAD_CLOCK_OUT_FACTOR_M factor M, M= Factor + 1 Output clock frequency Fout = Fsrc/M
2:1	R/W	0x0	PAD_CLOCK_OUT_SEL GPIO PAD 32KOSCO clock source select 0: Source the LFCLK from the internal RC oscillator. 1: Source the LFCLK from the 32K AUTO SW oscillator 2: Source the RCO_CALIB 32K 3: Source the HFCLK from the external crystal oscillator
0	R/W	0x0	PAD_CLOCK_OUT_EN GPIO PAD 32KOSCO clock output enable. 0: Disable 1: Enable

3.3.4.13 0x0084 System LFCLK Status (Default Value: 0x0000_0000)

Offset: 0x0084	Register Name: SYS_LFCLK_STATUS

Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R	0x0	LOSC_READY 1: LOSC clock is ready 0: LOSC clock is not ready

3.3.4.14 0x00C0 RESET Debounce Control Register (Default Value: 0x0000_020C)

Offset: 0x00C0			Register Name: Reset_DBC_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11:8	R/W	0x2	CFG_DBCOUNCE_CLK_DLY Set RESETn debounce clock cycle number (N: 0~15) First must select clock for debounce (configurate by register CFG_DBCOUNCE_CLK_FREQ_SEL), then the debounce clock cycles is (CFG_DBCOUNCE_CLK_DLY+1).
7:4	/	/	/
3: 0	R/W	0xC	CFG_DBCOUNCE_CLK_FREQ_SEL RESETn debounce clock frequency select CFG_DBCOUNCE_CLK_DIV<=14, Freq=LFCLK/($2^{CFG\_DBCOUNCE\_CLK\_FREQ\_SEL}$); CFG_DBCOUNCE_CLK_DIV==15, reserved.

3.3.4.15 0x00C8 BOOT MEDIA RECORD Register (Default Value: 0x0000_0000)

Offset: 0x00C8			Register Name: BOOT_MEDIA_RECORD
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	BOOT_MEDIA_RECORD Record which kind of memory to booting the system SW read and write only.

3.3.4.16 0x0140 System RCOSC Frequency Detect Register (Default Value: 0x000C_8001)

Offset: 0x0140			Register Name: SYS_RCOSC_FREQ_DET
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:8	R	0xC80	RCOSC Frequency V=FREQ/10
7:1	/	/	/
0	R/W	0x1	RCOSC Calibration Enable

3.3.4.17 0x0144 BLE RCOSC Calibration Control Register0 (Default Value: 0x0000_0000)

Offset: 0x0144	Register Name:BLE_RCOSC_CALIB_REG0

Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29	R/W	0x0	RCO_CALIB_EN RCOSC calibration function enable 1: Enable 0: Disable
28	R/W	0x0	RCO_CALIB_RST_PUL When writing 1 to this bit, reset RCOSC calibration, then run again. When writing 0, there is no effect. It is always read as 0.
27	/	/	/
26:24	R	0x0	RCO_CALIB_CS RCOSC calibration current state. It is read only. 000: IDLE 001: WAIT_DCXO_READY 011: WAIT_CAL_FINISH 110: CAL_FINISH 010: WAIT_TIMER Others: Reserved
23:21	/	/	/
20	R/W	0x0	RCO_CALIB_SW_REQ_PUL When writing 1 to this bit, begin RCOSC calibration process at once. When writing 0, there is no effect. It is always read as 0.
19:17	/	/	/
16	R/W	0x0	RCO_WUP_TIME_EN RCOSC calibration periodically wake-up function enable 1: Enable 0: Disable
15:13	/	/	/
12: 0	R/W	0x0	RCO_WUP_TIME RCO_SLP calibration periodically wake-up time. Configure by software to wake up system, and begin RCO_SLP calibration process. The time unit of RCO_WUP_TIME is 32 kHz (divided from RCO_SLP).

3.3.4.18 0x0148 BLE RCOSC Calibration Control Register1 (Default Value: 0x0000_0000)

Offset: 0x0148			Register Name:BLE_RCOSC_CALIB_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/
19:16	R/W	0x0	RCO_SCALE_PHASE2_NUM Wake-up times in phase 2

Offset: 0x0148			Register Name:BLE_RCOSC_CALIB_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
15:12	R/W	0x0	RCO_SCALE_PHASE1_NUM Wake-up times in phase1
11	/	/	/
10:8	R/W	0x0	RCO_SCALE_PHASE3_WUP_TIMES Multiply factor of RCO_WUP_TIME in wake-up phase3 0x0: 10 times 0x1: 16 times 0x2: 20 times 0x3: 24 times 0x4: 32 times 0x5: 40 times 0x6: 64 times 0x7: 128 times
7	/	/	/
6:4	R/W	0x0	RCO_SCALE_PHASE2_WUP_TIMES Multiply factor of RCO_WUP_TIME in wake-up phase2 0x0: 2 times 0x1: 4 times 0x2: 6 times 0x3: 8 times 0x4: 10 times 0x5: 12 times 0x6: 14 times 0x7: 16 times
3:2	/	/	/
1	R/W	0x0	RCO_NORMAL_WUP_TIMES_SEL In normal wakeup mode, actual wake-up time is the multiply of RCO_WUP_TIME * factor 1: Select RCO_SCALE_PHASE3_WUP_TIMES as the multiply factor of RCO_WUP_TIME 0: Select RCO_SCALE_PHASE2_WUP_TIMES as the multiply factor of RCO_WUP_TIME
0	R/W	0x0	RCO_WUP_MODE_SEL Scale wakeup and normal wakeup are two wakeup methods 1: Select scale wake up method 0: Select normal wake up method

3.3.4.19 0x014C BLE CLK32K Auto Switch Register0 (Default Value: 0x0271_0000)

Offset: 0x014C			Register Name: BLE_CLK32K_SWITCH_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/

Offset: 0x014C			Register Name: BLE_CLK32K_SWITCH_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
25:16	R/W	0x271	DIV_HALFCYCLE_TARGET Bit[9:0]: The target half cycle number for 32kHz clock divide
15:5	/	/	/
4	R/W	0x0	DIV_CLK_EN The 32kHz divider enable 1: Enable 0: Disable
3:2	R/W	0x0	Reserved
2	R/W	0x0	Reserved
1	R/W	0x0	DIV_CLK_SRC_SEL The source clock is selected for dividing to 32 kHz clock 1: 32 MHz Clock 0: HFCLK
0	R/W	0x0	CLK32K_AUTO_SW_EN 32K clock auto switch enable 1: Enable 0: Disable

3.3.4.20 0x0150 BLE CLK32K Auto Switch Register1 (Default Value: 0x0000_0000)

Offset: 0x0150			Register Name: BLE_CLK32K_SWITCH_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:27	/	/	/
26:16	R	0x0	CLK32K_SW_OFFSET_DOWN The offset when switch from divide 32k clock to 32 kHz input clock
15:11	/	/	/
10: 0	R	0x0	CLK32K_SW_OFFSET_ON The offset when switch from 32 kHz input clock to divide 32k clock

3.3.4.21 0x0180 BT_WIFI Connect Mode Control Register (Default Value: 0x0000_0000)

Offset: 0x0180			Register Name: CONN_BOOT_FLAG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R	0x0	WLAN_RSTN_STATU The WRSTN PIN status in connect mode 0: WLAN reset signal hold 1: WLAN reset signal release

Offset: 0x0180			Register Name: CONN_BOOT_FLAG
Bit	**Read/Write**	**Default/Hex**	**Description**
16	R	0x0	BLE_RSTN_STATU The BRSTN PIN status in connect mode 0: BLE reset signal hold 1: BLE reset signal release
15: 0	/	/	/

3.3.4.22 0x0188 Config IO Status Register (Default Value: 0x0000_0001)

Offset: 0x0188			Register Name: CFG_IO_ST
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R	0x0	SIP FLASH 0: No flash sip 1: With flash sip in package

3.3.4.23 0x018C WLAN HIF override control register (Default Value: 0x0000_0003)

Offset: 0x018C			Register Name: WLAN_HIF_OV_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	Reserved
7	R/W	0x0	WLAN_IRQ_OVER_HIF Use wlan_irq generated in WLAN_MISC logic to replace the corresponding output of WLAN_HIF 0: Not override WLAN_HIF output 1: Override WLAN_HIF output
6	R/W	0x0	WLAN_WUP_OVR_HIF Use WLAN_WUP(bit2) to override the corresponding output of WLAN_HIF 0: Not override WLAN_HIF output 1: Override WLAN_HIF output
5	R/W	0x0	RESET_CPU_OVR_HIF Use RESET_CPU(bit1) to override the corresponding output of WLAN_HIF 0: Not override WLAN_HIF output 1: Override WLAN_HIF output
4	R/W	0x0	DISABLE_CPU_CLK_OVR_HIF Use DISABLE_CPU_CLK(bit0) to override the corresponding output of WLAN_HIF 0: Not override WLAN_HIF output 1: Override WLAN_HIF output
3	/	/	Reserved

Offset: 0x018C			Register Name: WLAN_HIF_OV_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x0	WLAN_WUP Wake up wlan from sleep mode, host firmware need to clear it after WLAN is waked up
1	R/W	0x1	RESET_CPU Reset WLAN CPU 0: Release WLAN CPU reset 1: WLAN CPU is reset
0	R/W	0x1	DISABLE_CPU_CLK Disable WLAN CPU clock 0: WLAN CPU clock enable 1: WLAN CPU clock disable

3.3.4.24 0x0190 SRAM Configuration Register (Default Value: 0x2222_2222)

Offset: 0x0190			Register Name: SRAM_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	R/W	0x22	Reserved
7: 0	R/W	0x22	ARM_SRAM_CFG ram_cfg[7:0] bit [2:0]: Connect to EMA bit [4:3]: Connect to EMAW others: Reserved The recommend value of S40 is that EMA=3'b010, EMAW=2'b10

3.3.4.25 0x0194 Flash Encrypt AES Nonce Register0 (Default Value: 0x0000_0000)

Offset: 0x194			Register Name: FLASH_ENCRYPT_AES_NONCE0
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	FLASH_ENCRYPT_AES_NONCE0 Flash encrypt module AES nonce bit [31: 0]. These bits are only changed once after system is power up.

3.3.4.26 0x0198 Flash Encrypt AES Nonce Register1 (Default Value: 0x0000_0000)

Offset: 0x0198			Register Name: FLASH_ENCRYPT_AES_NONCE1
Bit	**Read/Write**	**Default/Hex**	**Description**
15: 0	R/W	0x0	FLASH_ENCRYPT_AES_NONCE1 Flash encrypt module AES nonce bit [47:32]. These bits are only changed once after system is powered up.

3.3.4.27 0x01C0 CPU Boot Flag Register (Default Value: 0x0000_0000)

Offset: 0x01C0			Register Name: BOOT_FLAG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	BOOT_FLAG_WRITE_LOCK 0x429B: CPU_BOOT_FLAG filed is writable Other: CPU_BOOT_FLAG filed is read only **Note**: The *field of CPU_BOOT_FLAG only can be changed when this bit is set to 0x429B. In one write operation, the GRPCM will check this field whether it equals to 0x429B. If not, the value of CPU_BOOT_FLAG will be ignored.* This field is always read with value 0x0000. For example: After writing 0x429B0001, then read value = 0x00000001 After writing 0x11110000, then read value = 0x00000001 After writing 0x429B0000, then read value = 0x00000000
15:4	/	/	/
3: 0	R/W	0x0	CPU_BOOT_FLAG 0: Boot from cold reset (By default) 1: Boot from deep sleep 2: Boot for system update 3: Boot for recovery Others: Reserved

3.3.4.28 0x01C4 CPU Boot Address Register (Default Value: 0x0000_0000)

Offset: 0x01C4			Register Name: BOOT_ADDR
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	CPU_BOOT_ADDR

3.3.4.29 0x01C8 CPU Boot Argument Register (Default Value: 0x0000_0000)

Offset: 0x01C8			Register Name: BOOT_ARG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	CPU_BOOT_ARG

3.3.4.30 0x01CC DSP Boot Flag Register (Default Value: 0x0000_0000)

Offset: 0x01CC			Register Name: BOOT_FLAG
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x01CC			Register Name: BOOT_FLAG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	BOOT_FLAG_WRITE_LOCK 0x429B: DSP_BOOT_FLAG filed is writable other: DSP_BOOT_FLAG filed is read only **Note**: The *field of DSP_BOOT_FLAG only can be changed when this bit is set to 0x429B. In one write operation, the GRPCM will check this field whether it equals to 0x429B. If not, the value of DSP_BOOT_FLAG will be ignored.* This field is always read with value 0x0000. For example: After writing 0x429B0001, then read value = 0x00000001 After writing 0x11110000, then read value = 0x00000001 After writing 0x429B0000, then read value = 0x00000000
15:4	/	/	/
3: 0	R/W	0x0	DSP_BOOT_FLAG 0: Boot from cold reset (by default) 1: Boot from deep sleep 2: Boot for system update 3: Boot for recovery Others: Reserved

3.3.4.31 0x01D0 DSP Boot Address Register (Default Value: 0x0000_0000)

Offset: 0x01D0			Register Name: BOOT_ADDR
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	DSP_BOOT_ADDR

3.3.4.32 0x01D4 DSP Boot Argument Register (Default Value: 0x0000_0000)

Offset: 0x01D4			Register Name: BOOT_ARG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	DSP_BOOT_ARG

3.3.4.33 0x01D8 RV Boot Flag Register (Default Value: 0x0000_0000)

Offset: 0x01D8			Register Name: BOOT_FLAG
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x01D8			Register Name: BOOT_FLAG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	BOOT_FLAG_WRITE_LOCK 0x429B: RV_BOOT_FLAG filed is writable Other: RV_BOOT_FLAG filed is read only **Note**: The *field of RV_BOOT_FLAG only can be changed when this bit is set to 0x429B. In one write operation, the GRPCM will check this field whether equals to 0x429B. If not, the value of RV_BOOT_FLAG will be ignored.* This field is always read with value 0x0000. For example: After writing 0x429B0001, then read value = 0x00000001 After writing 0x11110000, then read value = 0x00000001 After writing 0x429B0000, then read value = 0x00000000
15:4	/	/	/
3: 0	R/W	0x0	RV_BOOT_FLAG 0: Boot from cold reset(default) 1: Boot from deep sleep 2: Boot for system update 3: Boot for recovery Others: Reserved

3.3.4.34 0x01DC RV Boot Address Register (Default Value: 0x0000_0000)

Offset: 0x01DC			Register Name: BOOT_ADDR
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	RV_BOOT_ADDR

3.3.4.35 0x01E0 RV Boot Argument Register (Default Value: 0x0000_0000)

Offset: 0x01E0			Register Name: BOOT_ARG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	RV_BOOT_ARG

3.3.4.36 0x0200 SYSTEM Private Register0(Default Value: 0x0000_0000)

Offset: 0x0200			Register Name: SYS_PRIV_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	SYS_PRIV_REG0

3.3.4.37 0x0204 SYSTEM Private Register1 (Default Value: 0x0000_0000)

Offset: 0x0204			Register Name: SYS_PRIV_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0204			Register Name: SYS_PRIV_REG1
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	SYS_PRIV_REG1

3.3.4.38 0x0208 SYSTEM Private Register2 (Default Value: 0x0000_0000)

Offset: 0x0208			Register Name: SYS_PRIV_REG2
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	SYS_PRIV_REG2

3.3.4.39 0x020C SYSTEM Private Register3 (Default Value: 0x0000_0000)

Offset: 0x020C			Register Name: SYS_PRIV_REG3
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	SYS_PRIV_REG3

3.3.4.40 0x0210 SYSTEM Private Register4 (Default Value: 0x0000_0000)

Offset: 0x0210			Register Name: SYS_PRIV_REG4
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	SYS_PRIV_REG4

3.3.4.41 0x0214 SYSTEM Private Register5 (Default Value: 0x0000_0000)

Offset: 0x0214			Register Name: SYS_PRIV_REG5
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	SYS_PRIV_REG5

3.3.4.42 0x0218 SYSTEM Private Register6 (Default Value: 0x0000_0000)

Offset: 0x0218			Register Name: SYS_PRIV_REG6
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	SYS_PRIV_REG6

3.3.4.43 0x021C SYSTEM Private Register7 (Default Value: 0x0000_0000)

Offset: 0x021C			Register Name: SYS_PRIV_REG7
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	SYS_PRIV_REG7

3.4 Clock Controller Unit (CCU)

3.4.1 Overview

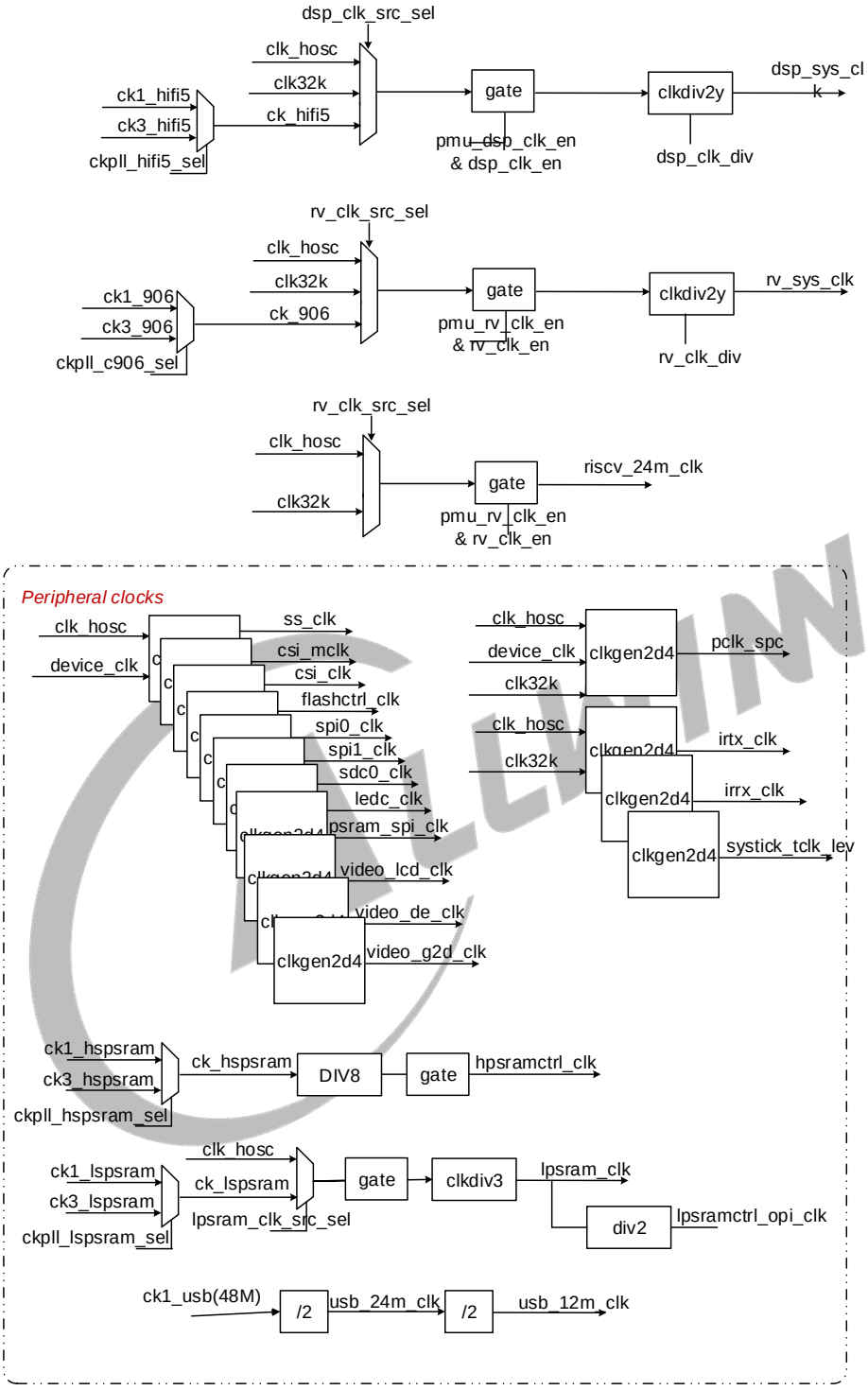
The clock controller unit (CCU) manages the reset and the HFCLK for each module and provides the configurations for each module to adjust its module clock.

CCU includes the following features:

- Supports configuring module clock
- Supports clock output control
- Supports bus clock gating
- Supports bus software reset

3.4.2 Functional Description

3.4.2.1 Clock Structure Diagram on APP Domain



3.4.2.2 MSI Clock Instruction

MSI is located between HSPSRAM (slave) and CPU, DSP, RV, DMA, USB, etc. (master). MSI is deeply coupled with HSPSRAM, and both use the same clock hpsram_clk.

3.4.3 Operation Guides

3.4.3.1 Configuring Bus Clock

Bus clock supports dynamic switch, but the switching process needs to observe the following rules:

- When bus is switched from the low frequency clock source to the high frequency clock source, configure the corresponding frequency division factor, and then switch the clock source. The frequencies of AHB, APB and MBUS are 192MHz, 96MHz and 200MHz respectively.
- When bus is switched from the high frequency clock source to the low frequency clock source, switch the clock source, and then configure the corresponding frequency division factor.

3.4.3.2 Configuring Module Clock

- For the gating and reset registers of modules, release reset and then open CLK_GATE to ensure that modules release reset synchronously to avoid exceptions.
- For the module clocks (except DDR clocks), configure the clock source and frequency division factor, and then open CLK_GATE (namely, set it as 1). For the configuring sequence of clock source and frequency division factor, refer to section 3.5.3.1.

3.4.4 Register List

Module Name	Base Address
CCMU	0x4003C000

Register Name	Offset Address	Description
BUS_CLK_GATING_CTRL0	0x0004	Bus Clock Gating Control Register0
BUS_CLK_GATING_CTRL1	0x0008	Bus Clock Gating Control Register1
DEV_RST_CTRL0	0x000C	Software Device Reset Control Register0
DEV_RST_CTRL1	0x0010	Software Device Reset Control Register1
CPU_DSP_RV_CLK_GATING_CTRL	0x0014	CPU_DSP_RV system Clock Gating Control
CPU_DSP_RV_RST_CTRL	0x0018	CPU_DSP_RV system Reset Control
MBUS_CLK_GATING_CTRL	0x001C	MBUS Clock Gating Control Register
SPI0_CLK_CTRL	0x0020	SPI0 Clock Control Register
SPI1_CLK_CTRL	0x0024	SPI1 Clock Control Register
SDC_CLK_CTRL	0x0028	SDC Clock Control Register
SS_CLK_CTRL	0x002C	SS Clock Control Register
CSI_DCLK_CTRL	0x0030	CSI Device Clock Control Register
LEDC_CLK_CTRL	0x0034	LEDC Clock Control Register
IRRX_CLK_CTRL	0x0038	IRRX Clock Control Register
IRTX_CLK_CTRL	0x003C	IRTX Clock Control Register
SYSTICK_REFCLK_CTRL	0x0040	System Tick Reference Clock Register
SYSTICK_CALIB_CTRL	0x0044	System Tick Clock Calibration Register
CSI_OUT_MCLK_CTRL	0x0050	CSI Output MCLK Control Register

Register Name	Offset Address	Description
FLASHC_MCLK_CTRL	0x0054	FLASH Controller Module Clock Control Register
SQPI_PSRAMC_CLK_CTRL	0x0058	SQPI LS-PSRAM Module Clock Control Register
APB_SPC_CLK_CTRL	0x005C	Special APB Clock Control Register (UART APB bus clock)
USB_CLK_CTRL	0x0060	USB clock control register
RISCV_CLK_CTRL	0x0064	RISCV clock control register
DSP_CLK_CTRL	0x0068	DSP clock control register
HSPSRAM_CLK_CTRL	0x006C	High-Speed PSRAM clock control register
LSPSRAM_CLK_CTRL	0x0070	Low-Speed PSRAM clock control register
G2D_CLK_CTRL	0x0074	G2D Clock Control Register
DE_CLK_CTRL	0x0078	DE Clock Control Register
LCD_CLK_CTRL	0x007C	LCD Clock Control Register
RESET_SOURCE_RECORD	0x0100	WDG Reset Source Record Register

3.4.5 Register Description

3.4.5.1 0x0004 BUS Clock Gating Control Register0 (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: BUS_CLK_GATING_CTRL0
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	USB_EHCI_CLK_GATING 0: The clock is gated off. 1: The clock is running.
30	R/W	0x0	USB_OHCI_CLK_GATING 0: The clock is gated off. 1: The clock is running.
29	/	/	/
28	R/W	0x0	LEDC_CLK_GATING 0: The clock is gated off. 1: The clock is running.
27	R/W	0x0	USB_OTG_CLK_GATING 0: The clock is gated off. 1: The clock is running.
26	R/W	0x0	SMCARD_CLK_GATING 0: The clock is gated off. 1: The clock is running.
25:22	/	/	/
21	R/W	0x0	HSPSRAM_CTRL_CLK_GATING 0: The clock is gated off. 1: The clock is running.

Offset: 0x0004			Register Name: BUS_CLK_GATING_CTRL0
Bit	**Read/Write**	**Default/Hex**	**Description**
20:17	/	/	/
16	R/W	0x0	IRRX_CLK_GATING 0: The clock is gated off. 1: The clock is running.
15	R/W	0x0	IRTX_CLK_GATING 0: The clock is gated off. 1: The clock is running.
14	R/W	0x0	PWM_CLK_GATING 0: The clock is gated off. 1: The clock is running.
13:12	/	/	/
11	R/W	0x0	TWI1_CLK_GATING 0: The clock is gated off. 1: The clock is running.
10	R/W	0x0	TWI0_CLK_GATING 0: The clock is gated off. 1: The clock is running.
9	/	/	/
8	R/W	0x0	UART2_CLK_GATING 0: The clock is gated off. 1: The clock is running.
7	R/W	0x0	UART1_CLK_GATING 0: The clock is gated off. 1: The clock is running.
6	R/W	0x0	UART0_CLK_GATING 0: The clock is gated off. 1: The clock is running.
5	/	/	/
4	R/W	0x0	SDC0_CLK_GATING 0: The clock is gated off. 1: The clock is running.
3:2	/	/	/
1	R/W	0x0	SPI1_CLK_GATING 0: The clock is gated off. 1: The clock is running.
0	R/W	0x0	SPI0_CLK_GATING 0: The clock is gated off. 1: The clock is running.

3.4.5.2 0x0008 BUS Clock Gating Control Register1 (Default Value: 0x0000_0000)

Offset: 0x0008	Register Name:BUS_CLK_GATING_CTRL1

Bit	Read/Write	Default/Hex	Description
31:29	/	/	/
28	R/W	0x0	AHB_MONITOR_CLK_GATING 0: The clock is gated off. 1: The clock is running.
27	R/W	0x0	G2D_CLK_GATING 0: The clock is gated off. 1: The clock is running.
26	R/W	0x0	DE_CLK_GATING 0: The clock is gated off. 1: The clock is running.
25	R/W	0x0	DISPLAY_CLK_GATING 0: The clock is gated off. 1: The clock is running.
24	R/W	0x0	LCD_CLK_GATING 0: The clock is gated off. 1: The clock is running.
23:13	/	/	/
12	R/W	0x0	SS_CLK_GATING 0: The clock is gated off. 1: The clock is running.
11	/	/	/
10	R/W	0x0	SPINLOCK_CLK_GATING 0: The clock is gated off. 1: The clock is running.
9:8	/	/	/
7	R/W	0x0	DMA1_CLK_GATING 0: The clock is gated off. 1: The clock is running.
6	R/W	0x0	DMA0_CLK_GATING 0: The clock is gated off. 1: The clock is running.
5:3	/	/	/
2	R/W	0x0	OWA_CLK_GATING 0: The clock is gated off. 1: The clock is running.
1	R/W	0x0	I2S_CLK_GATING 0: The clock is gated off. 1: The clock is running.
0	R/W	0x0	Reserved

3.4.5.3 0x000C Module Reset Control Register0 (Default Value: 0x0000_0000)

Offset: 0x000C	Register Name: MOD_RST_CTRL0

Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	USB_EHCI_RST 0: The module is in reset state. 1: The module is released to work.
30	R/W	0x0	USB_OHCI_RST 0: The module is in reset state. 1: The module is released to work.
29	R/W	0x0	CSI_ JPE_RST 0: The module is in reset state. 1: The module is released to work.
28	R/W	0x0	LEDC_RST 0: The module is in reset state. 1: The module is released to work.
27	R/W	0x0	USB_OTG_RST 0: The module is in reset state 1: The module is released to work
26	R/W	0x0	SMCARD_RST 0: The module is in reset state. 1: The module is released to work.
25	R/W	0x0	USB_PHY_RST 0: The module is in reset state. 1: The module is released to work.
24	/	/	/
23	R/W	0x0	FLASH_ENC_RST 0: The module is in reset state. 1: The module is released to work.
22	R/W	0x0	FLASH_CTRL_RST 0: The module is in reset state 1: The module is released to work
21	R/W	0x0	HSPSRAM_CTRL_RST 0: The module is in reset state. 1: The module is released to work.
20	R/W	0x0	LSPSRAM_CTRL_RST 0: The module is in reset state. 1: The module is released to work.
19:17	/	/	/
16	R/W	0x0	IRRX_RST 0: The module is in reset state. 1: The module is released to work.
15	R/W	0x0	IRTX_RST 0: The module is in reset state. 1: The module is released to work.
14	R/W	0x0	PWM_RST 0: The module is in reset state. 1: The module is released to work.

Offset: 0x000C			Register Name: MOD_RST_CTRL0
Bit	**Read/Write**	**Default/Hex**	**Description**
13:12	/	/	/
11	R/W	0x0	TWI1_RST 0: The module is in reset state. 1: The module is released to work.
10	R/W	0x0	TWI0_RST 0: The module is in reset state. 1: The module is released to work.
9	/	/	/
8	R/W	0x0	UART2_RST 0: The module is in reset state. 1: The module is released to work.
7	R/W	0x0	UART1_ RST 0: The module is in reset state 1: The module is released to work
6	R/W	0x0	UART0_RST 0: The module is in reset state. 1: The module is released to work.
5	/	/	/
4	R/W	0x0	SDC0_RST 0: The module is in reset state. 1: The module is released to work.
3:2	/	/	/
1	R/W	0x0	SPI1_RST 0: The module is in reset state. 1: The module is released to work.
0	R/W	0x0	SPI0_RST 0: The module is in reset state. 1: The module is released to work.

3.4.5.4 0x0010 Module Reset Control Register1 (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: MOD_RST_CTRL1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27	R/W	0x0	G2D_RST 0: The module is in reset state. 1: The module is released to work.
26	R/W	0x0	DE_RST 0: The module is in reset state. 1: The module is released to work.

Offset: 0x0010			Register Name: MOD_RST_CTRL1
Bit	**Read/Write**	**Default/Hex**	**Description**
25	R/W	0x0	DISPLAY_RST 0: The module is in reset state. 1: The module is released to work.
24	R/W	0x0	LCD_RST 0: The module is in reset state. 1: The module is released to work.
23:22	/	/	/
21	R/W	0x0	BT_CORE_RST 0: The module is in reset state. 1: The module is released to work.
20:15	/	/	/
14	R/W	0x0	TRNG_RST 0: The module is in reset state. 1: The module is released to work. The default value is 1, used for TRNG retention registers.
13	/	/	/
12	R/W	0x0	SS_RST 0: The module is in reset state. 1: The module is released to work.
11	/	/	/
10	R/W	0x0	SPINLOCK_RST 0: The module is in reset state. 1: The module is released to work.
9:8	/	/	/
7	R/W	0x0	DMA1_RST 0: The module is in reset state. 1: The module is released to work.
6	R/W	0x0	DMA0_RST 0: The module is in reset state. 1: The module is released to work.
5:3	/	/	/
2	R/W	0x0	OWA_RST 0: The module is in reset state. 1: The module is released to work.
1	R/W	0x0	I2S_RST 0: The module is in reset state. 1: The module is released to work.
0	/	/	/

3.4.5.5 0x0014 CPU_DSP_RV Systems Clock Gating Control Register (Default Value: 0x0000_0000)

Offset: 0x0014	Register Name: CPU_SYS_CLK_GATING_CTRL

Bit	Read/Write	Default/Hex	Description
31:20	/	/	/
19	R/W	0x0	RISCV_CFG_CLK_GATING 0: The clock is gated off. 1: The clock is running.
18	R/W	0x0	RISCV_MSGBOX_CLK_GATING 0: The clock is gated off. 1: The clock is running.
17:12	/	/	/
11	R/W	0x0	DSP_CFG_CLK_GATING 0: The clock is gated off. 1: The clock is running.
10	R/W	0x0	DSP_MSGBOX_CLK_GATING 0: The clock is gated off. 1: The clock is running.
9:2	/	/	/
1	R/W	0x0	CPU_MSGBOX_CLK_GATING 0: The clock is gated off. 1: The clock is running.
0	/	/	/

3.4.5.6 0x0018 CPU_DSP_RV Systems Reset Control Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: CPU_SYS_RST_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21	R/W	0x0	RISCV_SYS_APB_SOFT_RST Reset for RISCV Debug bus 0: The module is in reset state. 1: The module is released to work.
20	/	/	/
19	R/W	0x0	RISCV_CFG_RST 0: The module is in reset state. 1: The module is released to work.
18	R/W	0x0	RISCV_MSGBOX_RST 0: The module is in reset state. 1: The module is released to work.
17	/	/	/
16	R/W	0x0	RISCV_CORE_RST 0: The module is in reset state. 1: The module is released to work.
15	/	/	/

Offset: 0x0018			Register Name: CPU_SYS_RST_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
14	R/W	0x0	DSP_DEBUG_RST 0: The module is in reset state. 1: The module is released to work.
13:12	/	/	/
11	R/W	0x0	DSP_CFG_RST 0: The module is in reset state. 1: The module is released to work.
10	R/W	0x0	DSP_MSGBOX_RST 0: The module is in reset state. 1: The module is released to work.
9	/	/	/
8	R/W	0x0	DSP_CORE_RST 0: The module is in reset state. 1: The module is released to work.
7:2	/	/	/
1	R/W	0x0	CPU_MSGBOX_RST 0: The module is in reset state. 1: The module is released to work.
0	/	/	/

3.4.5.7 0x001C MBUS Clock Gating Control Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: MBUS_CLK_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9	R/W	0x0	DE_MBUS_CLK_EN 0: The clock is gated off. 1: The clock is released to work.
8	R/W	0x0	G2D_MBUS_CLK_EN 0: The clock is gated off. 1: The clock is released to work.
7	/	/	/
6	R/W	0x0	DMA1_MBUS_CLK_EN 0: The clock is gated off. 1: The clock is released to work.
5	R/W	0x0	DMA0_MBUS_CLK_EN 0: The clock is gated off. 1: The clock is released to work.
4	/	/	/
3	R/W	0x0	CE_MBUS_CLK_EN 0: The clock is gated off. 1: The clock is released to work.

Offset: 0x001C			Register Name: MBUS_CLK_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
2: 0	/	/	/

3.4.5.8 0x0020 SPI0 Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: SPI0_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: DEVICE_CLK 1x: Reserved
23:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.4.5.9 0x0024 SPI1 Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: SPI1_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: DEVICE_CLK 1x: Reserved

Offset: 0x0024			Register Name: SPI1_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
23:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N. 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M. M = value + 1 (1~16)

3.4.5.10 0x0028 SDC Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: SDC0_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: DEVICE_CLK 1x: Reserved
23:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.4.5.11 0x002C SS Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: SS_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x002C			Register Name: SS_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off 1: The module clock is released to work $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: DEVICE_CLK 1x: Reserved
23:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N Clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.4.5.12 0x0030 CSI_JPEG Device CLK Control Register (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: CSI_JPE_DEV_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CSI_JPE_DEV_CLK_ENABLE 0: The module clock is gated off 1: The module clock is released to work $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: DEVICE_CLK 1x: Reserved
23:18	/	/	/

Offset: 0x0030			Register Name: CSI_JPE_DEV_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N. 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.4.5.13 0x0034 LEDC Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: LEDC_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: DEVICE_CLK 1x: Reserved
23:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N Clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.4.5.14 0x0038 IRRX Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: IRRX_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0038			Register Name: IRRX_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: CLK32K 1x: Reserved
23:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.4.5.15 0x003C IRTX Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x003C			Register Name: IRTX_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: CLK32K 1x: Reserved
23:18	/	/	/

Offset: 0x003C			Register Name: IRTX_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.4.5.16 0x0040 System Tick Reference Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x0040			Register Name: SYSTICK_REFCLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: CLK32K 1x: Reserved
23:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.4.5.17 0x0044 System Tick Clock Calibration Register (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: SYSTICK_CALIB_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/

Offset: 0x0044			Register Name: SYSTICK_CALIB_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
25	R/W	0x0	ST_NOREF 1: There is no reference clock 0: Use the reference clock
24	R/W	0x0	ST_SKEW **Note**: Set *this bit to 0 if the system timer clock, the external reference clock, or FCLK indicated by ST_NOREF, can guarantee an exact multiple of 10ms. Otherwise, set this bit to 1.*
23: 0	R/W	0x0	ST_10MS_COUNTER **Note**: *This field provides an integer value to compute a 10ms (100Hz) delay from either the reference clock, or FCLK if the reference clock is not implemented.* *For example: No reference clock, FCLK=50MHz, CNT = 0x7A11F (49999). REFCLK = 1MHz, CNT = 0x270F(9999).*

3.4.5.18 0x0050 CSI Output MCLK Control Register (Default Value: 0x0000_0000)

Offset: 0x0050			Register Name: CSI_MCLK _CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CSI_MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: DEVICE_CLK 1x: Reserved
23:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N. 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M. M = value + 1 (1~16)

3.4.5.19 0x0054 Flash Controller SPI Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x0054			Register Name: FLASH _SPI_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: DEVICE_CLK 1x: Reserved
23:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.4.5.20 0x0058 SQPI_PSRAM SPI Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x0058			Register Name: SQPI_PSRAM_SPI_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: DEVICE_CLK 1x: Reserved
23:18	/	/	/

Offset: 0x0058			Register Name: SQPI_PSRAM_SPI_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.4.5.21 0x005C APB_SPC Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x005C			Register Name: PCLK_SPC_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: DEVICE_CLK (output from DPLL1/DPLL3, 384 Mhz at max) 10: CLK32K 11: Reserved
23:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.4.5.22 0x0060 USB Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x0060			Register Name: USB_CLK_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	Reserved

3.4.5.23 0x0064 RISCV Clock Control Register (Default Value: 0x0000_0100)

Offset: 0x0064			Register Name: RV_CLKCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CLK_ENABLE 0: The clock is gated off. 1: The clock is released to work. $CLK_{rv\_sys\_clk} = CLK_{src} / M$ $CLK_{riscv\_24m\_clk} = CLK_{src}$
30:10	/	/	/
9:8	R/W	0x1	RISCV_AXI_DIV_CFG. Factor N. (N = FACTOR_N +1) FACTOR_N is from 1 to 3. **Note**: *This clock is generated by the N division of RISCV_CORE clock and used for the AXI bus of RISC-V subsystem. DIV supports dynamic frequency adjustment and configuration.*
7:6	/	/	/
5:4	R/W	0x0	RISCV_CLK_SRC_SEL RISCV clock source select 00: CLK_HOSC 01: CLK32K 1X: CK_906 **Note**: *The clock source of rv_sys_clk is configured by the RISCV_CLK_SRC_SEL. When the clock source of rv_sys_clk is configured as CK_906, the clock of riscv_24m_clk will source from CLK_HOSC. When the clock source of rv_sys_clk is configured as CLK_HOSC or CLK32K, riscv_24m_clk and rv_sys_clk will share the same clock source.*
3:2	/	/	/
1: 0	R/W	0x0	RISCV_CLK_DIV Clock divide ratio M 00: /1 01: /2 10: /4 11: /8

3.4.5.24 0x0068 DSP Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x0068			Register Name: DSP_CLKCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CLK_ENABLE 0: The clock is gated off. 1: The clock is released to work.
30:6	/	/	/

Offset: 0x0068			Register Name: DSP_CLKCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
5:4	R/W	0x0	DSP_CLK_SRC_SEL DSP clock source select 00: CLK_HOSC 01: CLK32K 1X: CK_HIFI5
3:2	/	/	/
1: 0	R/W	0x0	DSP_CLK_DIV 00: /1 01: /2 10: /4 11: /8

3.4.5.25 0x006C HSPSRAM Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x006C			Register Name: HSPSRAM_CLKCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CLK_ENABLE 0: The clock is gated off. 1: The clock is released to work.
30:5	/	/	/
4	R/W	0x0	Reserved
3:2	/	/	/
1: 0	R/W	0x0	Reserved

3.4.5.26 0x0070 LSPSRAM Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x0070			Register Name: LSPSRAM_CLKCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CLK_ENABLE 0: The clock is gated off. 1: The clock is released to work.
30:5	/	/	/
4	R/W	0x0	LPSRAM_CLK_SRC_SEL LPSRAM clock source select 0: CLK_HOSC 1: CK_LSPSRAM
3:2	/	/	/
1: 0	R/W	0x0	LSPSRAM_CLK_DIV 00: /1 01: /2 10: /4 11: /8

3.4.5.27 0x0074 G2D Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x0074			Register Name: G2D_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: DEVICE_CLK 1x: Reserved
23:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.4.5.28 0x0078 DE Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x0078			Register Name: DE_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: DEVICE_CLK 1x: Reserved
23:18	/	/	/

Offset: 0x0078			Register Name: DE_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.4.5.29 0x007C LCD Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x007C			Register Name: LCD_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: DEVICE_CLK 1x: Reserved
23:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.4.5.30 0x0100 WDG Reset Record Register (Default Value: 0x0000_0000)

Offset: 0x0100			Register Name: WDG_RST_RECORD
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/

Offset: 0x0100			Register Name: WDG_RST_RECORD
Bit	**Read/Write**	**Default/Hex**	**Description**
3	R/W	0x0	TWD_RST Record TWD reset. Write 1 to clear this bit. 0: No watch dog reset happens. 1: Watch dog reset happens.
2	R/W	0x0	M33_WDG_RST Record M33 WDG reset. Write 1 to clear this bit. 0: No watch dog reset happens. 1: Watch dog reset happens.
1	R/W	0x0	RV_WDG_RST Record RV WDG reset. Write 1 to clear this bit. 0: No watch dog reset happens. 1: Watch dog reset happens.
0	R/W	0x0	DSP_WDG_RST Record DSP WDG reset. Write 1 to clear this bit. 0: No watch dog reset happens. 1: Watch dog reset happens.

3.5 CCU_AON

3.5.1 Overview

The CCU_AON module manages the power, reset and input clocks for this system. The CCU_AON is placed in always on power domain.

CCU_AON includes the following features:

- Supports managing the OSC clock
- Supports bus source and divisions
- Supports configuring module clock
- Supports clock output control
- Supports bus clock gating
- Supports bus software reset

3.5.2 Functional Description

3.5.2.1 DCXO

DCXO module generates the reference clock frequency through external crystal oscillator and internal capacitor resonance. clk_RfipDcxoTrim [7: 0] is used to adjust the capacitor in series with the crystal oscillator and select the suitable capacitor size according to the crystal oscillator manual; dcxo_ictrl<3: 0> is used to adjust the DCXO current.

It is suggested to configure the gear of clk_dcxo_ctrl [4: 0] as 5'b00000 to reduce the crystal oscillator startup time when starting. After startup, adjust its gear to 5'b10000 to reduce power consumption.

As shown below, the control capacitance of clk_RfipDcxoTrim [7: 0] ranges from 0.2p to 25.4p.

clk_RfipDcxoTrim [7: 0]	0	1	2	3	4	5	6	7
capacitance	0.2p	0.4p	0.4p*2	0.4p*4	3.2p	3.2p*2	3.2p*4	-

clk_dcxo_ctrl [4: 0] can influence the negative resistane of DCXO circuit by adjusting current. It needs to increase current to ensure the normal operation of circuit in the case of higher crystal oscillator or oscillation frequency.

3.5.2.2 PLL

PLL Features

The following tables show the PLL features and the PLL usage.

Table 3-2 PLL Features

PLL	Fvco	PLL Output Frequency (Actual)	PLL Output Frequency (Default)	Spread Spectrum	LFM	Gradient Control	RMS Request	Lock Time
DPLL1	1.344G ~2.496 G	1.92G	1.92G	No	Yes	No	＜40 ps	20 us
DPLL2	1.344G ~2.496 G	1.92G	1.92G	No	Yes	No	＜40 ps	20 us
DPLL3	800M~ 2.1G	960M~1.92G	1.6G	No	Yes	No	＜40 ps	20 us
AUDPLL	63.22 M~127 .8M	4*24.576M 4*22.5792M	4*24.576 M 4*22.5792 M	No	Yes	No	＜200 ps	120 us
USBPLL	336M~ 624M	480M	480M	No	No	No	＜30 ps	50 us

Table 3-3 Usage Notes of PLL Frequencies

PLL Type	PLL Calculation Formula	PLL Output Frequency Range	Post Frequency-Divider of Module	Typical Working-Frequency of Module
DPLL1	Xtal_fre*N/M	1.92 GHz	RF:32/20/12/10/6/4/2 SoC:40/8/7/6/5/4/3/2.5/1 Audio:39*2/85*2	1.92B GHz
DPLL2	Xtal_fre*N/M	1.92 GHz	RF:16/6/5/3/2(960M)	1.92 GHz
DPLL3	Xtal_fre*N/M	960 MHz~1.92 GHz	SoC:8/7/6/5/4/3/2.5/1	1.6 GHz /1.5 GHz /1.4 GHz
AUDPLL	Xtal_fre/M*N	4*24.576 MHz 4*22.5792 MHz	-	4*24.576 MHz 4*22.5792 MHz
USBPLL	Xtal_fre*N(24M/24.576M/32M/40M) Xtal_fre/M*N (26M)	480 MHz	-	480 MHz

DPLL1/2/3

DPLL1/DPLL2

The output frequency of DPLL1/DPLL2 is set by the equation:

$$Fre_{DPLL1/DPLL2} = \frac{Fre_{DCXO} \bullet N}{M}$$

M is the reference clock prescale factor configured by clk_RefClkCfg [3: 0]. Its default value is 4'b0001 and not recommended to be modified, as the larger prescale factor will aggravate the noise. N is the loop frequency multiplication factor whose integer part is configured by clk_DpllNdiv [7: 0]. As the fractional part (clk_DpllFracCtrl) of N is enabled, it can be configured by clk_DpllFrac [15: 0].

Set an example of 26M crystal oscillator. If the PLL clock output frequency is 1920M, the prescale factor will be 1920/26=73.8461.... Then clk_DpllNdiv [7: 0] configures its integer part as 73 (binary code is 8'b0100_1001), and clk_DpllFrac [15: 0] configures its fractional part by being multiplied by 2^16, namely, 0.8461...*2^16=55453.538...≈55454 (binary code is 16'b1101_1000_1001_1110). During the fractional division, DITHER can be enabled by clk_DPLL_DITHER_DISABLE [1: 0] to decrease spur.

Set dpll_[0] as 0 to adjust the LPF order in the loop to the second order (set 1 to adjust to the third order), thereby optimizing performance.

DPLL1/DPLL2 are fixed frequency output (1920 MHz), whose configuration varies based on Crystal Frequencies.

Table 3-4 Corresponding Configurations of Different Crystal Frequencies

Crystal Frequencies	PLL Output Frequencies	clk_DpllNdiv<7: 0>	clk_DpllFrac<15: 0>
24M	1920M	8'b0101_0000	clk_DpllFracCtrl disable
24.576M	1920M	8'b0100_1110	16'b0010_0000_0000_0000
26M	1920M	8'b0100_1001	16'b1101_1000_1001_1110
32M	1920M	8'b0011_1100	clk_DpllFracCtrl disable
40M	1920M	8'b0011_0000	clk_DpllFracCtrl disable

DPLL3

The output frequency of DPLL3 is set by the equation:

$$Fre_{DPLL3} = \frac{Fre_{DCXO} \bullet N}{M}$$

M is the reference clock prescale factor that is configured by clk_RefClkCfg [3: 0]. Its default value is 4'b0001 and not recommended to be modified, as the larger prescale factor will aggravate the noise. N is the loop frequency multiplication factor whose integer part is configured by clk_DpllNdiv [7: 0]. As the fractional part (clk_DpllFracCtrl) of N is enabled, it can be configured by clk_DpllFrac [15: 0].

Set an example of 26M crystal oscillator. If the PLL clock output frequency is 1600M, the prescale factor will be 1600/26=61.5384.... Then clk_DpllNdiv [7: 0] configures its integer part as 61 (binary code is 8'b0011_1101), and clk_DpllFrac [15: 0] configures its fractional part by being multiplied by 2^16, namely, 0.5384.....*2^16=35288.615≈35289 (binary code is 16'b1000_1001_1101_1001). During the fractional division, DITHER can be enabled by clk_DPLL_DITHER_DISABLE [1: 0] to decrease spur.

Set dpll_[0] as 0 to adjust the LPF order in the loop to the second order (set 1 to adjust to the third order), thereby optimize performance.

The DPLL3 output frequency can be adjusted from 960M to 1920M, and the typical frequencies include 1600M, 1500M, and 1400M. To work properly, DPLL3 needs to be powered by CLK_LDO1 and CLK_LDO2, and enables CLK_LDO1_EN and CLK_LDO2_EN.

Table 3-5 Corresponding Configurations of Different Crystal Frequencies and PLL Output Frequencies

Crystal Frequencies	PLL Output Frequencies	clk_DpllNdiv [7: 0]	clk_DpllFrac [15: 0]
24M	1600M	8'b0100_0010	16'b1010_1010_1010_1011
	1500M	8'b0011_1110	16'b1000_0000_0000_0000
	1400M	8'b0011_1010	16'b0101_0101_0101_0101
24.576M	1600M	8'b0100_0001	16'b0001_1010_1010_1011
	1500M	8'b0011_1101	16'b0000_1001_0000_0000
	1400M	8'b0011_1000	16'b1111_0111_0101_0101
26M	1600M	8'b0011_1101	16'b1000_1001_1101_1001
	1500M	8'b0011_1001	16'b1011_0001_0011_1011
	1400M	8'b0011_0101	16'b1101_1000_1001_1110
32M	1600M	8'b0011_0010	clk_DpllFracCtrl Disable
	1500M	8'b0010_1110	16'b1110_0000_0000_0000
	1400M	8'b0010_1011	16'b1100_0000_0000_0000
40M	1600M	8'b0010_1000	clk_DpllFracCtrl Disable
	1500M	8'b0010_0101	16'b1000_0000_0000_0000
	1400M	8'b0010_0011	clk_DpllFracCtrl Disable

AudioPLL

AUDPLL outputs two frequency points (90.3168M and 98.304M). Its output frequency is set by the equation:

$$Fre_{AUDPLL} = \frac{Fre_{DCXO} \bullet N \bullet 2}{M}$$

M is the reference clock prescale factor that is configured by clk_aud_pll_prediv [4: 0]. N is the loop frequency multiplication factor that is configured by clk_aud_pll_n [6: 0].

Set an example of 26M crystal oscillator. If the clock output is 4*24.576M, configure M as 32 (binary code is 5'b1_1111) and N as 121 (7'b111_1000). Then the output frequency is 26M/32*121=98.3125M, which differs from 4*24.576M by -19ppm.

Adjust the current of charge pump in the loop and the bandwidth of loop by adjusting clk_aud_cp_s [4: 0]. The larger current can improve phase noise but decrease phase margin. The recommended value of clk_aud_pll_vco_s [4: 0] is 5'b10000.

To work properly, AUDIOPLL needs to be powered by CLK_LDO2 and controlled by CLK_LDO1_EN. Thus, CLK_LDO1_EN and CLK_LDO2_EN should be enabled.

Table 3-6 Corresponding Configurations of Different Crystal Frequencies and PLL Output Frequencies

Crystal Frequencies	PLL Output Frequencies	clk_aud_pll_prediv<4: 0>	clk_aud_pll_n<6: 0>	±400ppm (prediv,n,ppm)	±1000ppm (except ±400ppm) (prediv,n,ppm)

Crystal Frequencies	PLL Output Frequencies	clk_aud_pll_prediv<4: 0>	clk_aud_pll_n<6: 0>	±400ppm (prediv,n,ppm)	±1000ppm (except ±400ppm) (prediv,n,ppm)
24M	22.5792M * 8	5'b1_0100	7'b100_1110	(21,79,-345ppm)	
		5'b1_0000	7'b011_1111		(17,64,401ppm)
	24.576M * 8	5'b1_0100	7'b101_0101	(21,86,-187ppm)	
		5'b0_1001	7'b010_1000		(10,41,977ppm)
24.576M	22.5792M * 8	5'b1_1011	7'b110_0110		(3,11,-2268ppm)
	24.576M * 8	5'b0_0000	7'b000_0011	(1,4,0ppm)	
26M	22.5792M * 8	5'b1_0010	7'b100_0001	(19,66,-12ppm)	
		5'b1_0000	7'b011_1010		(17,59,-903ppm)
	24.576M * 8	5'b0_1000	7'b010_0001		(9,34,-832ppm)
32M	22.5792M * 8	5'b1_0000	7'b010_1111		(17,48,401ppm)
	24.576M * 8	5'b0_1101	7'b010_1010	(14,43,-187ppm)	
40M	22.5792M * 8	5'b1_1010	7'b011_1100		(27,61,594ppm)
	24.576M * 8	5'b1_0111	7'b011_1010	(24,59,299ppm)	

NOTE

a. The configuration values with grey background are recommended; and the configuration value with red font goes beyond ±1000ppm.

b. The fractional division is not supported.

USBPLL

USBPLL generates 10-phase 480M clock fixed frequency output, and xtal_mode<2: 0> should be configured based on the crystal frequency. The following table shows the specific configuration.

Table 3-7 Corresponding Configurations of Different Crystal Frequencies

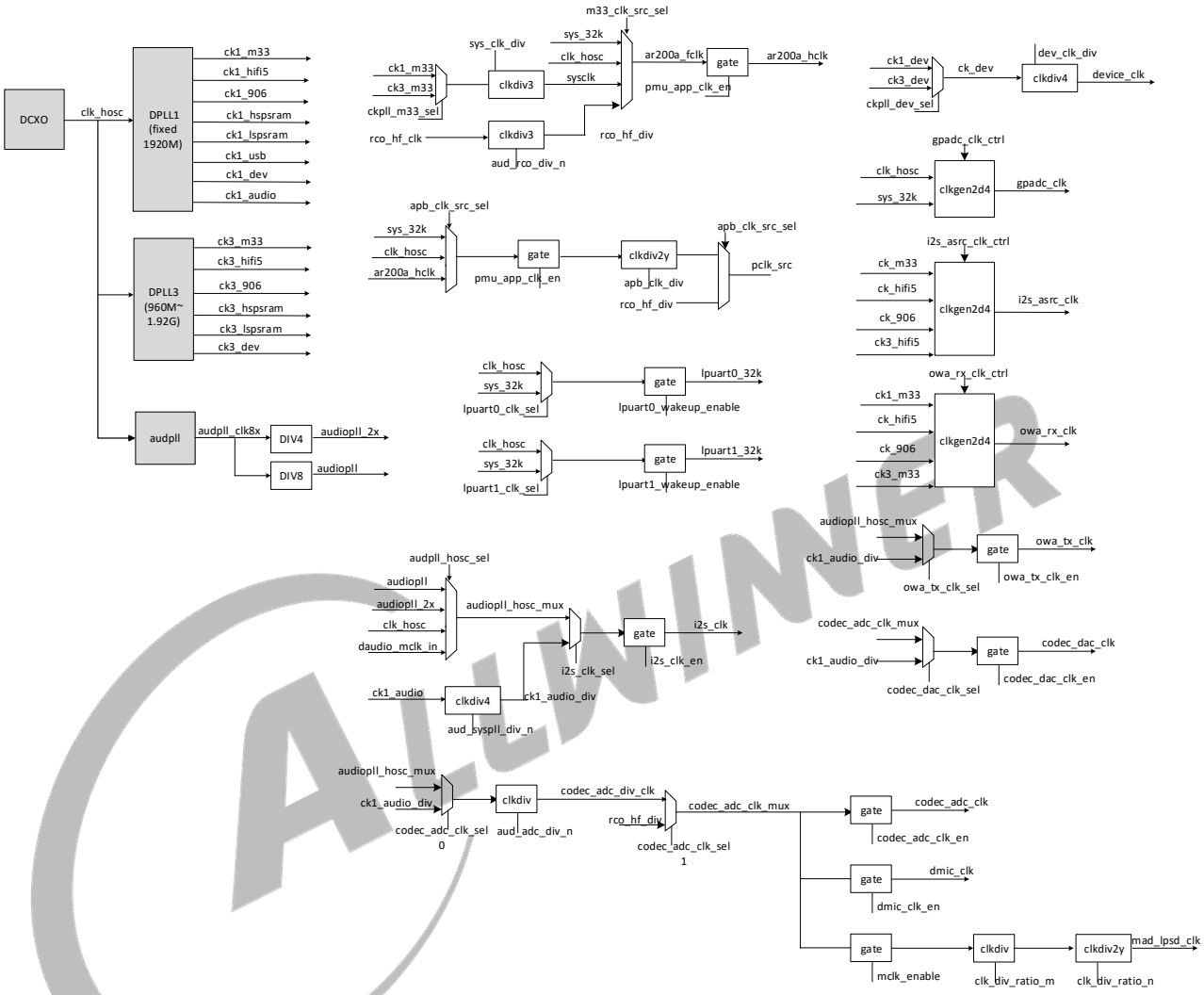
Crystal Frequency	PLL Output Frequency	xtal_mode<2: 0>	cp_cont<1: 0>
24M	480M	3'b000	2'b10
26M	480M	3'b001	2'b10
32M	480M	3'b010	2'b10
40M	480M	3'b011	2'b10
24.576M(DPLL1_ref)	480M	3'b100	2'b10
26M(DPLL1_ref)	480M	3'b101	2'b10
40M	480M	3'b110	2'b10
40M	480M	3'b111	2'b10

The crystal frequencies of 24M, 26M, 32M and 40M are all obtained by the loop integer multiplication to obtain the final output frequency. Since the 24.567M crystal frequency cannot be multiplied by an integer, the USBPLL reference clock input is the 24M clock output by DPLL1. As the performance of output jitter is poor, USBPLL may need to slow down and the value of cp_cont needs to be increased.

3.5.2.3 System/Audio Clock Generation

The following figure shows the diagram of the system/audio clock generation.

Figure 3-2 System/Audio Clock Generation



3.5.3 Programming Guidelines

3.5.3.1 Configuring PLL

Enable DPLL1

Step 1 Configure the values of DPLL_NDIV and FACTOR_M of the DPLL1_CTRL register.

Step 2 Set the value of DPLL1_CTRL[bit31] as 1.

Step 3 Wait the value of DPLL1_OUT_CTRL[bit30] to be 1.

Step 4 Delay 20 us.

Step 5 Configure the DIV and EN of output clock of the DPLL1_OUT_CTRL register. Then, the output clocks of DPLL1 can be used.

Disable DPLL1

Set the value of DPLL1_CTRL[bit31] as 0.

Enable DPLL3

Step 1 Configure the values of DPLL_NDIV and FACTOR_M of the DPLL3_CTRL register.

Step 2 Set the values of CLK_LDO1_EN and CLK_LDO2_EN of the CLK_LDO_CTRL register as 1.

Step 3 Set the value of DPLL3_CTRL[bit 31] as 1.

Step 4 Wait the value of DPLL3_OUT_CTRL[bit30] to be 1.

Step 5 Delay 20 us.

Step 6 Configure the DIV and EN of output clock of the DPLL3_OUT_CTRL register. Then, the output clocks of DPLL3 can be used.

Disable DPLL3

Step 1 Set the value of DPLL3_CTRL [bit31] is 0.

Step 2 Set the CLK_LDO1_EN and CLK_LDO2_EN of the CLK_LDO_CTRL register as 0.

Open AUDIOPLL

Step 1 Configure the values of PLL_FACTOR_N and PLL_PRETDIV_M of the AUDIO_PLL_CTRL register.

Step 2 Set the values of CLK_LDO1_EN and CLK_LDO2_EN of the CLK_LDO_CTRL register as 1.

Step 3 Set the value of the AUDIO_PLL_CTRL[bit31] as 1.

Step 4 Set the value of the AUDIO_PLL_CTRL[bit29] as 1.

Step 5 Wait the value of the AUDIO_PLL_CTRL[bit28] to be 1.

Step 6 Delay 150 us, then the AUDIOPLL can be used.

Close AUDIOPLL

Step 1 Configure the value of AUDIO_PLL_CTRL[bit31] as 0.

Step 2 Configure the values of CLK_LDO1_EN and CLK_LDO2_EN of the CLK_LDO_CTRL register as 0.

Step 3 Configure the value of the AUDIO_PLL_CTRL[bit29] as 0.

NOTE

When DPLL3 and AUDIOPLL needs to use the CLK_LDO1_EN and CLK_LDO2_EN, the LDOs cannot be disabled if either of PLLs is using them.

3.5.3.2 Configuring Bus Clock

Bus clock supports dynamic switch, but the switching process needs to observe the following rules:

- When bus is switched from the low frequency clock source to the high frequency clock source, configure the corresponding frequency division factor, and then switch the clock source.
- When bus is switched from the high frequency clock source to the low frequency clock source, switch the clock source, and then configure the corresponding frequency division factor. The frequencies of AHB, APB and MBUS are 192MHz, 96MHz and 200MHz respectively.

3.5.3.3 Configuring Module Clock

- For the gating and reset registers of modules, release reset and then open CLK_GATE to ensure that modules release reset synchronously to avoid exceptions.
- For the module clocks (except DDR clocks), configure the clock source and frequency division factor, and then open CLK_GATE. For the configuring sequence of clock source and frequency division factor, refer to section 3.6.2.2.

3.5.4 Register List

Module Name	Base Address
CCMU_AON	0x4004C400

Register Name	Offset Address	Description
HOSC_FREQ_DET	0x0080	HOSC Frequency Detect Register0
HOSC_TYPE	0x0084	HOSC Type Register
DCXO_CTRL	0x0088	DCXO control Register
DPLL1_CTRL	0x008C	DPLL1 Control Register
DPLL2_CTRL	0x0090	DPLL2 Control Register
DPLL3_CTRL	0x0094	DPLL3 Control Register
AUDIO_PLL_CTRL	0x0098	Audio PLL Control Register
AUDIO_PLL_BIAS	0x009C	Audio PLL Bias Register
AUDIO_PATTERN	0x00A0	Audio Pattern Control Register
DPLL1_OUT_CONFIG	0x00A4	DPLL1 Output Clock frequency configure Register
DPLL3_OUT_CONFIG	0x00A8	DPLL3 Output Clock frequency configure Register
CLK_LDO_CTRL	0x00AC	AUDPLL and DPLL3 LDO control Register
WLAN_BT_RFIP_CTRL	0x00C4	WLAN BT RFIP Control Register
MODULE_RST_CTRL	0x00C8	Module Reset control Register
MODULE_CLK_EN_CTRL	0x00CC	Module clock control register
LPUART0_WAKEUP_CTRL	0x00D0	LPUART0 wakeup control register

Register Name	Offset Address	Description
LPUART1_WAKEUP_CTRL	0x00D4	LPUART1 wakeup control register
GPADC_CLK_CTRL	0x00D8	GPADC clock control register
AUDIO_CLK_CTRL	0x00DC	audio clock control register
SYS_CLK_CTRL	0x00E0	system clock control register
MAD_LPSD_CLK_CTRL	0x00E4	MAD_lpsd_clk control register
OWA_RX_CLK_CTRL	0x00E8	OWA_RX_clock control register
I2S_ASRC_CLK_CTRL	0x00EC	I2S_ASRC_clk control register

3.5.5 Register Description

3.5.5.1 0x0080 HOSC Frequency Detect Register0 (Default Value: 0x0000_0000)

Offset: 0x0080			Register Name: HOSC_FREQ_DET
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:4	R	0x0	HOSC_FREQ_DET HOSC counter value for 60000 RC_HF cycles
3:2	/	/	/
1	R	0x0	HOSC_FREQ_READY It indicates whether HOSC_FREQ_DET is ready, which will be cleared during detection and be set after detection finishes.
0	R/W	0x0	HOSC Frequency Detect Enable Use RC_HF (8.192 MHz) to detect the frequency of HOSC. Detecting time fix to 60000 RC_HF cycles, about 10 ms. 0: Disable detect 1: Enable detect When Detect Enable is set, RC_HF counter and HOSC counter begin to count at the same time. After 60000 RC_HF cycles, the two counters will stop. The counter value of HOSC counter will be save to HOSC_FREQ_DET

3.5.5.2 0x0084 HOSC Type Register (Default Value: 0x0000_0000)

Offset: 0x0084			Register Name: HOSC_TYPE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/

Offset: 0x0084			Register Name: HOSC_TYPE
Bit	**Read/Write**	**Default/Hex**	**Description**
2: 0	R/W	0x0	HOSC_TYPE 0: System uses 26 MHz HOSC 1: System uses 40 MHz HOSC 2: System uses 24 MHz HOSC 3: System uses 32 MHz HOSC 4: System uses 24.576 MHz HOSC Others: Reserved **Note**: *This bit is set by application software and must be consistent with the real situation. A wrong configuration will cause the RCOSC calibration failure.*

3.5.5.3 0x0088 DCXO Control Register (Default Value: 0x9409_0B10)

Offset: 0x0088			Register Name: DCXO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x1	DCXO_EN DCXO Enable 0: Disable 1: Enable
30	/	/	/
29:28	R/W	0x1	CLK_MODE_SEL Clock Mode Select (When AUTO_DET_EN is disabled) 00: Internal Clock 01: Crystal 10: Analog 11: Digital
27:20	R/W	0x40	FRE_TRI Frequency Trimming
19	R/W	0x1	ALC_EN ALC Enable 0: Disable 1: Enable
18:17	/	/	/
16:12	R/W	0x10	ICTRL BIAS Current Control
11	R/W	0x1	RF_DIG_REF_CLK_EN 0: Disable 1: Enable
10	R/W	0x0	Reserved
9	R/W	0x1	BUFFER_DPLL_EN 0: Disable 1: Enable

Offset: 0x0088			Register Name: DCXO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
8:6	R/W	0x4	Reserved
5:4	R	0x1	DCXO_DET 00: Internal clock 01: Crystal 10: Analog 11: Digital
3:1	/	/	/
0	R/W	0x0	AUTO_DET_EN Auto Detect Enable 0: Disable 1: Enable

3.5.5.4 0x008C DPLL1 Control Register (Default Value: 0x6000_0181)

Offset: 0x008C			Register Name: DPLL1_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE Enable or disable the system PLL 0: Disable the system PLL 1: Enable the system PLL **Note:** *PLL frequency F_{output} = (F_{hosc}* N.f) / M. The output clock must be in range of 960 MHz. The default value is set for 960 MHz with 40 MHz OSC.*
30:29	R/W	0x3	DPLL_DITHER_DISABLE
28:13	R/W	0x0	DPLL_FRAC **Note** : V_{FRAC} = $f * 2^{16}$
12	R/W	0x0	DPLL_FRAC_CTRL
11:4	R/W	0x18	DPLL_NDIV PLL N
3: 0	R/W	0x1	FACTOR_M PLL factor M (1 ~ 8). M = Factor M **Note**: *When the factor is set to 0, M will be set to 1*

NOTE

a. The default value of this register will be 0x6000_0281 when the frequency of dcxo is 24 MHz.

b. The default value of this register will be 0x1D89_D241 when the frequency of dcxo is 26 MHz.

c. The default value of this register will be 0x6000_01E1 when the frequency of dcxo is 32 MHz.

d. The default value of this register will be 0x6000_0181 when the frequency of dcxo is 40 MHz.

3.5.5.5 0x0090 DPLL2 Control Register (Default Value: 0x6000_0181)

Offset: 0x0090			Register Name: DPLL2_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE Enable or disable the system PLL 0: Disable the system PLL 1: Enable the system PLL **Note:** *PLL frequency F_{output} = (F_{hosc}* N.f) / M. The output clock must be in range of 960 MHz. The default value is set for 960 MHz with 40 MHz OSC.*
30:29	R/W	0x3	DPLL_DITHER_DISABLE
28:13	R/W	0x0	DPLL_FRAC **Note**: $V_{FRAC} = f * 2^{16}$
12	R/W	0x0	DPLL_FRAC_CTRL
11:4	R/W	0x18	DPLL_NDIV PLL N
3: 0	R/W	0x1	FACTOR_M PLL factor M (1 ~ 8). M = Factor M **Note**: *When the factor is set to 0, M will be set to 1*

NOTE

a. The default value of this register will be 0x6000_0281 when the frequency of dcxo is 24 MHz.

b. The default value of this register will be 0x1D89_D241 when the frequency of dcxo is 26 MHz.

c. The default value of this register will be 0x6000_01E1 when the frequency of dcxo is 32 MHz.

d. The default value of this register will be 0x6000_0181 when the frequency of dcxo is 40 MHz.

3.5.5.6 0x0094 DPLL3 Control Register (Default Value: 0x6000_0181)

Offset: 0x0094			Register Name: DPLL3_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0094			Register Name: DPLL3_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE Enable or disable the system PLL. 0: Disable the system PLL 1: Enable the system PLL **Note:** *PLL frequency F_{output} = (F_{hosc}* N.f) / M. The output clock must be in range of 960 MHz. The default value is set for 960 MHz with 40 MHz OSC.*
30:29	R/W	0x3	DPLL_DITHER_DISABLE
28:13	R/W	0x0	DPLL_FRAC **Note**: $V_{FRAC} = f * 2^{16}$
12	R/W	0x0	DPLL_FRAC_CTRL
11:4	R/W	0x18	DPLL_NDIV PLL N
3: 0	R/W	0x1	FACTOR_M PLL factor M (1 ~ 8). M = Factor M **Note**: *When the factor is set to 0, M will be set to 1*

NOTE

a. The default value of this register will be 0x6000_0281 when the frequency of dcxo is 24 MHz.

b. The default value of this register will be 0x1D89_D241 when the frequency of dcxo is 26 MHz.

c. The default value of this register will be 0x6000_01E1 when the frequency of dcxo is 32 MHz.

d. The default value of this register will be 0x6000_0181 when the frequency of dcxo is 40 MHz.

3.5.5.7 0x0098 Audio PLL Control Register (Default Value: 0x0203_3A17)

Offset: 0x0098			Register Name: AUDIO_PLL_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PLL_ENABLE 1: Enable the PLL 0: Disable the PLL **Note:** *This PLL is used for Audio application.* *AUDPLL_CLK8X = (CLK_HOSC * N * 2) / M* *3 ≤ N / M ≤ 21* *CLK_HOSC * N must be in the range of 74 MHz ~ 504 MHz* *AUDPLL_CLK8X frequency is 8*24.583 MHz*

Offset: 0x0098			Register Name: AUDIO_PLL_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
30	/	/	/
29	R/W	0x0	LOCK_EN 0: Disable 1: Enable
28	R	0x0	LOCK 0: Unlocked 1: Locked (it means the pll is stable)
27:26	/	/	/
25	R/W	0x1	REF_CLK_EN Audio pll reference clock enable from DCXO
24	R/W	0x0	PLL_SDM_EN 0: Disable 1: Enable In this case, the PLL_FACTOR_N only low 4 bits are valid (1~16) **Note**: *Fractional Division is not supported.*
23:20	/	/	/
19:16	R/W	0x3	PLL_POSTDIV_P Post-div factor (P = Factor + 1) The range from 1 to 16
15	/	/	/
14:8	R/W	0x3A	PLL_FACTOR_N N = Factor + 1 The range from 1 to 128
7:5	/	/	/
4: 0	R/W	0x17	PLL_PRETDIV_M Pre-div factor (M = Factor + 1) The range from 1 to 32

3.5.5.8 0x009C Audio PLL Bias Register (Default Value: 0x1010_0000)

Offset: 0x009C			Register Name: AUDIO_PLL_BIAS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28:24	R/W	0x10	PLL_VCO_BIAS_CTRL PLL VCO Bias Control [4: 0]
23:21	/	/	/
20:16	R/W	0x10	PLL_CUR_BIAS_CTRL PLL Current Bias Control [4: 0]
15: 0	/	/	/

3.5.5.9 0x00A0 Audio PLL Pattern Control Register (Default Value: 0x0000_0000)

Offset: 0x00A0			Register Name: AUDIO_PLL_PATTERN_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	DIG_DELT_PAT_EN Sigma-Delta Pattern Enable
30:29	R/W	0x0	SPR_FREQ_MODE Spread Frequency Mode 00: DC = 0 01: DC = 1 1X: Triangular
28:20	R/W	0x0	WAVE_STEP Wave Step
19	/	/	/
18:17	R/W	0x0	FREQ 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16: 0	R/W	0x0	WAVE_BOT Wave Bottom

NOTE

a. The default value of this register will be 0xC001_1FAA when the frequency of dcxo is 26 MHz.

b. The default value of this register will be 0xC000_EA4A when the frequency of dcxo is 40 MHz.

3.5.5.10 0x00A4 DPLL1 Output Configure Register (Default Value: 0x8080_0008)

Offset: 0x00A4			Register Name: DPLL1_OUT_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x1	CK1_USB_EN 1: Enable USB 48 MHz clock output 0: Disable USB 48 MHz clock output
30	R	0x0	LOCK 0: Unlocked 1: Locked (it means the pll is stable)
29:28	/	/	/

Offset: 0x00A4			Register Name: DPLL1_OUT_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
27	R/W	0x0	CK1_AUD_EN 1: Enable clock output 0: Disable clock output
26:25	/	/	/
24	R/W	0x0	CK1_AUD_DIV DPLL1 output factor M 0: M=85 1: M=39 Clock frequency $F_{ck1\_audio} = F_{dpll1}/ M$. F_{dpll1} is 1920 MHz.
23	R/W	0x1	CK1_DEV_EN 1: Enable clock output 0: Disable clock output
22	/	/	/
21:20	R/W	0x0	CK1_DEV_DIV DPLL1 output factor M 00: M=7 01: M=6 10: M=5 11: M=4 Clock frequency $F_{ck1\_dev} = F_{dpll1}/ M$. F_{dpll1} is 1920 MHz.
19	R/W	0x0	CK1_LSPSRAM_EN 1: Enable clock output 0: Disable clock output
18:16	R/W	0x0	CK1_LSPSRAM_DIV DPLL1 output factor M 000: M=8 001: M=7 010: M=6 011: M=5 1XX: M=4 Clock frequency $F_{ck1\_lspsram} = F_{dpll1}/ M$. F_{dpll1} is 1920 MHz.
15	R/W	0x0	CK1_HSPSRAM_EN 1: Enable clock output 0: Disable clock output
14	/	/	/
13:12	R/W	0x0	CK1_HSPSRAM_DIV DPLL1 output factor M 00: M=3 01: M=2.5 10: M=2 11: M=1 Clock frequency $F_{ck1\_hspsram} = F_{dpll1}/ M$. F_{dpll1} is 1920 MHz.

Offset: 0x00A4			Register Name: DPLL1_OUT_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
11	R/W	0x0	CK1_HIFI5_EN 1: Enable clock output 0: Disable clock output
10:8	R/W	0x0	CK1_HIFI5_DIV DPLL1 output factor M 000: M=7 001: M=6 010: M=5 011: M=4 1XX: M=3 Clock frequency $F_{ck1\_hifi5} = F_{dpll1}/M$. F_{dpll1} is 1920 MHz.
7	R/W	0x0	CK1_C906_EN 1: Enable clock output 0: Disable clock output
6:4	R/W	0x0	CK1_C906_DIV DPLL1 output factor M 000: M=7 001: M=4 010: M=3 011: M=2.5 1XX: M=2 Clock frequency $F_{ck1\_906} = F_{dpll1}/ M$. F_{dpll1} is 1920 MHz.
3	R/W	0x1	CK1_M33_EN 1: Enable clock output 0: Disable clock output
2: 0	R/W	0x0	CK1_M33_DIV DPLL1 output factor M 000: M=8 001: M=7 010: M=6 011: M=5 1XX: M=4 Clock frequency $F_{ck1\_m33} = F_{dpll1}/ M$. F_{dpll1} is 1920 MHz.

3.5.5.11 0x00A8 DPLL3 Output Configure Register (Default Value: 0x0011_1001)

Offset: 0x00A8			Register Name: DPLL3_OUT_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R	0x0	LOCK 0: Unlocked 1: Locked (means the pll is stable)

Offset: 0x00A8			Register Name: DPLL3_OUT_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
29:24	/	/	/
23	R/W	0x0	CK3_DEV_EN 1: Enable clock output 0: Disable clock output
22	/	/	/
21:20	R/W	0x1	CK3_DEV_DIV DPLL3 output factor M 00: M=7 01: M=6 10: M=5 11: M=4 Clock frequency $F_{ck3\_dev} = F_{dpll3}/M$. F_{dpll3} is 960 MHz ~ 1920 MHz.
19	R/W	0x0	CK3_LSPSRAM_EN 1: Enable clock output 0: Disable clock output
18:16	R/W	0x1	CK3_LSPSRAM_DIV DPLL3 output factor M 000: M=8 001: M=7 010: M=6 011: M=5 1XX: M=4 Clock frequency $F_{ck3\_lspsram} = F_{dpll3}/M$. F_{dpll3} is 960 MHz ~ 1920 MHz.
15	R/W	0x0	CK3_HSPSRAM_EN 1: Enable clock output 0: Disable clock output
14	/	/	/
13:12	R/W	0x1	CK3_HSPSRAM_DIV DPLL3 output factor M 00: M=3 01: M=2.5 10: M=2 11: M=1 Clock frequency $F_{ck3\_hspsram} = F_{dpll3}/M$. F_{dpll3} is 960 MHz ~ 1920 MHz.
11	R/W	0x0	CK3_HIFI5_EN 1: Enable clock output 0: Disable clock output

Offset: 0x00A8			Register Name: DPLL3_OUT_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
10:8	R/W	0x0	CK3_HIFI5_DIV DPLL3 output factor M 000: M=7 001: M=6 010: M=5 011: M=4 1XX: M=3 Clock frequency F_{ck3_hifi5} = F_{dpll3}/M. F_{dpll3} is 960 MHz ~ 1920 MHz.
7	R/W	0x0	CK3_C906_EN 1: Enable clock output 0: Disable clock output
6:4	R/W	0x0	CK3_C906_DIV DPLL3 output factor M 000: M=7 001: M=4 010: M=3 011: M=2.5 1XX: M=2 Clock frequency F_{ck3_906} = F_{dpll3}/M. F_{dpll3} is 960 MHz ~ 1920 MHz.
3	R/W	0x0	CK3_M33_EN 1: Enable clock output 0: Disable clock output
2: 0	R/W	0x1	CK3_M33_DIV DPLL3 output factor M 000: M=8 001: M=7 010: M=6 011: M=5 1XX: M=4 Clock frequency F_{ck3_m33} = F_{dpll3}/M. F_{dpll3} is 960 MHz ~ 1920 MHz.

3.5.5.12 0x00AC AUDPLL and DPLL3 LDO Control Register (Default Value: 0x0800_0800)

Offset: 0x0AC			Register Name: CLK_LDO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:24	R/W	0x8	CLK_LDO2_TRIM CLK_LDO output voltage trim
23:18	/	/	/

Offset: 0x0AC			Register Name: CLK_LDO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
17	R/W	0x0	CLK_LDO2_BYPASS bypass CLK_LDO
16	R/W	0x0	CLK_LDO2_EN Clock ldo&vref enable, used for DPLL3 and AUD_PLL 1: Enable CLK_LDO 0: Disable CLK_LDO
15:12	/	/	/
11:8	R/W	0x8	CLK_LDO1_TRIM CLK_LDO output voltage trim
7:2	/	/	/
1	R/W	0x0	CLK_LDO1_BYPASS Bypass CLK_LDO
0	R/W	0x0	CLK_LDO1_EN Clock ldo&vref enable, used for DPLL3 and AUD_PLL 1: Enable CLK_LDO 0: Disable CLK_LDO

3.5.5.13 0x00C4 WLAN BT RFIP Control Register (Default Value: 0x0000_0001)

Offset: 0x00C4			Register Name: WLAN_BT_RFIP_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11:4	R/W	0x0	WLAN_BT_DEBUG_IO_SEL Wlan_debug_io [7: 0] and bt_debug_io [7: 0] share PA7~PA14, each bit controls which debug IO to be selected. 0: Select bt_debug_io 1: Select wlan_debug_io
3	/	/	/
2	R/W	0x0	WLAN_CLK_SOURCE_SEL Both RFIP0_DPLL/RFIP1_DPLL outputs 160 MHz clock for WLAN, and one of them needs to be selected for use. 0: WLAN clock source chooses RFIP0_DPLL 1: WLAN clock source chooses RFIP1_DPLL
1	R/W	0x0	BT_CLK_SOURCE_SEL Both RFIP0_DPLL/RFIP1_DPLL outputs 192 MHz clock for BT, and one of them needs to be selected for use. 0: BT clock source chooses RFIP0_DPLL 1: BT clock source chooses RFIP1_DPLL

Offset: 0x00C4			Register Name: WLAN_BT_RFIP_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x1	RFIP2_ENABLE There are two RFIPs in chip. There are two options: 1) WIFI/BT both use RFIP1, in this case RFIP2 can be disabled; 2) WIFI/BT each use one RFIP, in this case RFIP2 needs to be enabled. 0: Disable RFIP2 1: Enable RFIP2

3.5.5.14 0x00C8 Module Reset Control Register (Default Value: 0x0000_0100)

Offset: 0x00C8			Register Name: MOD_RST_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	BLE_RTC_RSTN BLE RTC Reset 1: Release the global reset of the BLE RTC 0: Hold the global reset of the BLE RTC
15	R/W	0x0	MADCFG_RSTN 0: The module is in reset state. 1: The module is released to work.
14:13	/	/	/
12	R/W	0x0	WLAN_RSTN WLAN Reset 1: Release the global reset of the WLAN 0: Hold the global reset of the WLAN
11	/	/	/
10	R/W	0x0	CODEC_DAC_RSTN 0: The module is in reset state. 1: The module is released to work.
9	R/W	0x0	RFAS_RSTN 1: Release the global reset of RFAS 0: Hold the global reset of RFAS
8	R/W	0x1	RCCAL_RSTN 0: The module is in reset state. 1: The module is released to work.
7	R/W	0x0	LPSD_RSTN 0: The module is in reset state. 1: The module is released to work.
6	/	/	/
5	R/W	0x0	CODEC_ADC_RSTN 0: The module is in reset state. 1: The module is released to work.

Offset: 0x00C8			Register Name: MOD_RST_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
4	R/W	0x0	MAD_RSTN 0: The module is in reset state. 1: The module is released to work.
3	R/W	0x0	DMIC_RSTN 0: The module is in reset state. 1: The module is released to work.
2	R/W	0x0	GPADC_RSTN 0: The module is in reset state. 1: The module is released to work.
1	R/W	0x0	LPUART1_RSTN 0: The module is in reset state. 1: The module is released to work.
0	R/W	0x0	LPUART0_RSTN 0: The module is in reset state. 1: The module is released to work.

3.5.5.15 0x00CC Module Clock Enable Register (Default Value: 0x0000_0000)

Offset: 0x00CC			Register Name: MOD_CLK_EN_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R/W	0x0	BLE_32M_CLK_EN BLE 32M clock enable 0: The clock is gated off. 1: The clock is running.
16	R/W	0x0	BLE_48M_CLK_EN BLE 48M clock enable 0: The clock is gated off. 1: The clock is running.
15	R/W	0x0	MAD_AHB_CLK_GATING 0: The clock is gated off. 1: The clock is running.
14:12	/	/	/
11	R/W	0x0	GPIO_BUS_CLK_GATING 0: The clock is gated off. 1: The clock is running.
10	R/W	0x0	CODEC_DAC_BUS_CLK_GATING 0: The clock is gated off. 1: The clock is running.
9	/	/	/

Offset: 0x00CC			Register Name: MOD_CLK_EN_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
8	R/W	0x0	RCCAL_CLK_GATING 0: The clock is gated off. 1: The clock is running.
7:6	/	/	/
5	R/W	0x0	CODEC_ADC_BUS_CLK_GATING 0: The clock is gated off. 1: The clock is running.
4	R/W	0x0	MAD_APB_CLK_GATING 0: The clock is gated off. 1: The clock is running.
3	R/W	0x0	DMIC_BUS_CLK_GATING 0: The clock is gated off. 1: The clock is running.
2	R/W	0x0	GPADC_CLK_GATING 0: The clock is gated off. 1: The clock is running.
1	R/W	0x0	LPUART1_WKUP_CLK_GATING 0: The clock is gated off. 1: The clock is running.
0	R/W	0x0	LPUART0_WKUP_CLK_GATING 0: The clock is gated off. 1: The clock is running.

3.5.5.16 0x00D0 LPUART0 Control Register (Default Value: 0x0000_0000)

Offset: 0x00D0			Register Name: LPUART0_WAKEUP_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	LPUART0_WAKEUP_ENABLE 0: The module clock is gated off. 1: The module clock is released to work.
30:18	/	/	/
17:16	R/W	0x0	LPUART0_WAKEUP_IN_SEL 0: LPUART0 wakeup input use UART0 serial in 1: LPUART0 wakeup input use UART1 serial in 2: LPUART0 wakeup input use UART2 serial in 3: Reserved
15:1	/	/	/
0	R/W	0x0	LPUART0_CLK_SEL 0: LPUART0 wakeup clock use SYS_32K 1: LPUART0 wakeup clock use CLK_HOSC (used for test)

3.5.5.17 0x00D4 LPUART1 Control register (Default Value: 0x0001_0000)

Offset: 0x00D4			Register Name: LPUART1_WAKEUP_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	LPUART1_WAKEUP_ENABLE 0: The module clock is gated off. 1: The module clock is released to work.
30:18	/	/	/
17:16	R/W	0x1	LPUART1_WAKEUP_IN_SEL 0: LPUART1 wakeup input uses UART0 serial in. 1: LPUART1 wakeup input uses UART1 serial in. 2: LPUART1 wakeup input uses UART2 serial in. 3: Reserved
15:1	/	/	/
0	R/W	0x0	LPUART1_CLK_SEL 0: LPUART1 wakeup clock uses SYS_32K. 1: LPUART1 wakeup clock uses CLK_HOSC (used for test).

3.5.5.18 0x00D8 GPADC Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x00D8			Register Name: GPADC_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CLK_HOSC 01: SYS_32K 1x: Reserved
23:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N. 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M. M = value + 1 (1~16)

3.5.5.19 0x00DC Audio Clock Control Register (Default Value: 0x0000_0000)

Offset: 0x00DC			Register Name: AUDIO_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28	R/W	0x0	DMIC_CLK_EN DMIC function clock enable 0: Disable 1: Enable
27	R/W	0x0	OWA_TX_CLK_EN OWA clock enable 0: Disable 1: Enable
26	R/W	0x0	I2S_CLK_EN I2S clock enable 0: Disable 1: Enable
25	R/W	0x0	CODEC_DAC_CLK_EN CODEC DAC clock enable 0: Disable 1: Enable
24	R/W	0x0	CODEC_ADC_CLK_EN CODEC_ADC function clock enable 0: Disable 1: Enable
23:22	/	/	/
21	R/W	0x0	OWA_TX_CLK_SEL 0: Select audpll_hosc_mux (from AUDPLL or DCXO) 1: Select ck1_audio_div(from DPLL1)
20	R/W	0x0	I2S_CLK_SEL 0: Select audpll_hosc_mux (from AUDPLL or DCXO) 1: Select ck1_audio_div(from DPLL1)
19	R/W	0x0	CODEC_DAC_CLK_SEL 0: Select audpll_hosc_mux (from AUDPLL or DCXO) 1: Select ck1_audio_div(from DPLL1)
18	R/W	0x0	CODEC_ADC_CLK_SEL1 Select clock for DMIC, CODEC_ADC, and MAD 0: Select F(codec_adc_div_clk) as the clock source of ADC 1: Select rco_hf_div(from RCO_HF) as the clock source of ADC
17	R/W	0x0	CODEC_ADC_CLK_SEL0 Select clock for DMIC, CODEC_ADC, and MAD 0: Select audpll_hosc_mux (from AUDPLL or DCXO) as the clock source of ADC 1: Select ck1_audio_div(from DPLL1) as the clock source of ADC

Offset: 0x00DC			Register Name: AUDIO_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
16:15	R/W	0x0	AUDPLL_HOSC_SEL 00: Select audiopll as the clock source of audio modules 01: Select audiopll_2x as the clock source of audio modules 10: Select CLK_HOSC(DCXO) as the clock source of audio modules 11: Select i2s_mclk_in as the clock source of audio modules
14:12	/	/	/
11:8	R/W	0x0	AUD_ADC_DIV_N Factor = 0~15 N = Factor + 1 **Note:** *$F_{codec\_adc\_div\_clk}$ = Freq(codec_adc_clk_sel0)/N. This clock is used as the source clock for ADC/MAD/DMIC.*
7:4	R/W	0x0	CK1_AUDIO_DIV factor = (0 ~ 15). N = Factor + 1 **Note**: *$F_{ck1\_audio\_div}$ = 24.615Mhz/N or 11.294Mhz/N. This clock is used as the source clock for I2S/OWA/DAC/ADC/MAD/DMIC.*
3	/	/	/
2: 0	R/W	0x0	AUD_RCO_DIV_N Factor = 0~7 N = Factor + 1 **Note:** *$F_{AUD\_RCO\_DIV\_CLK}$ = Freq(RCOSC)/N. This clock is used as the source clock for ADC/MAD/DMIC.*

3.5.5.20 0x00E0 System Clock Control Register (Default Value: 0x0000_0111)

Offset: 0x00E0			Register Name: SYS_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21	R/W	0x0	CKPLL_HSPSRAM_SEL select HSPRAM clock source from DPLL1 and DPLL3 0: CK1_HSPSRAM 1: CK3_HSPSRAM
20	R/W	0x0	CKPLL_LSPSRAM_SEL Select LSPRAM clock source from DPLL1 and DPLL3 0: CK1_LSPSRAM 1: CK3_LSPSRAM
19	R/W	0x0	CKPLL_M33_SEL Select M33 CPU clock source from DPLL1 and DPLL3 0: CK1_M33 1: CK3_M33

Offset: 0x00E0			Register Name: SYS_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
18	R/W	0x0	CKPLL_HIFI5_SEL select HIFI5 CPU clock source from DPLL1 and DPLL3 0: CK1_HIFI5 1: CK3_HIFI5
17	R/W	0x0	CKPLL_C906_SEL select C906 DSP clock source from DPLL1 and DPLL3 0: CK1_906 1: CK3_906
16	R/W	0x0	CKPLL_DEV_SEL select device clock source from DPLL1 and DPLL3 0: CK1_DEV 1: CK3_DEV
15:14	/	/	/
13:12	R/W	0x0	M33_CLK_SRC_SEL M33 clock source select 00: CLK_HOSC (default) 01: SYS_32K 10: SYSCLK 11: rco_hf_div (from RCO_HF)
11:8	R/W	0x1	SYS_CLK_DIV Factor = 0~15 N = Factor + 1 **Note:** *F_{sysclk} = Freq(ck_m33)/N. This clock is used as the source clock for AHB bus and M33.*
7:6	R/W	0x0	APB_CLK_SRC_SEL APB bus clock source select 00: CLK_HOSC 01: SYS_32K 10: AHBCLK 11: rco_hf_div (from RCO_HF) **Note**: *rco_hf_div is not divided by APB_CLK_DIV. The first three clock sources that are selected and divided will be selected together with rco_hf_div.*
5:4	R/W	0x1	APB_CLK_DIV 00: /1 01: /2 10: /4 11: /8
3: 0	R/W	0x1	DEV_CLK_DIV Factor = 0~15 N = Factor + 1 **Note**: *F_{device_clk} = Freq(ck_dev)/N. This clock is used as the source clock for some devices.*

3.5.5.21 0x00E4 MAD_lpsd_clk control register (Default Value: 0x0000_0000)

Offset: 0x00E4			Register Name: MAD_LPSD_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src} / N / M$
30:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3: 0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.5.5.22 0x00E8 OWA_RX_clock control register (Default Value: 0x0000_0000)

Offset: 0x00E8			Register Name: OWA_RX_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CK1_M33 (from DPLL) 01: CK_HIFI5 (from DPLL) 10: CK_906 (from DPLL) 11: CK3_M33 (from DPLL)
23:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/

Offset: 0x00E8			Register Name: OWA_RX_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
3:0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.5.5.23 0x00EC I2S_ASRC_clk control register (Default Value: 0x0000_0000)

Offset: 0x00EC			Register Name: I2S_ASRC_CLK_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	MCLK_ENABLE 0: The module clock is gated off. 1: The module clock is released to work. $CLK_m = CLK_{src}/N/M$
30:26	/	/	/
25:24	R/W	0x0	MCLK_SRC_SEL Clock source selection 00: CK_M33 (from DPLL) 01: CK_HIFI5 (from DPLL) 10: CK_906 (from DPLL) 11: CK3_HIFI5
23:18	/	/	/
17:16	R/W	0x0	CLK_DIV_RATIO_N clock pre-divide ratio N 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3:0	R/W	0x0	CLK_DIV_RATIO_M Clock divide ratio M M = value + 1 (1~16)

3.6 Boot ROM

3.6.1 Overview

The system has several ways to boot. It has an integrated on-chip Boot ROM (BROM) that is considered the primary program-loader. When system is powered on, the reset signal will be released. It is detected that the secure_enable_bit status of eFuse is configured as safe Boot ROM (SBROM) and not-safe Boot ROM (NBROM). If the secure_enable_bit is 0, NBROM will be configured as address 0x0, and CPU will be released to execute. If the secure_enable_bit is 1, SBROM will be configured as address 0x0, and CPU will be released to execute.

NBROM is used for loading BOOT0, validating, jumping to execute and flashing. If boot fails, NBROM will enter the upgrading mode.

SBROM is used for loading BOOT0, signature verify and jumping to execute. If boot fails, SBROM will set the statues of corresponding systems as unsafe, and NBROM will enter the upgrading mode.

The BROM includes the following features:

- On-chip memory
- Supports system boot from the following devices:
  - SD/eMMC
  - SPI Nor Flash
  - SPI Nand Flash
- Supports secure boot and normal boot
- Secure Brom supports load only certified firmware
- Secure Brom ensures that the secure boot is a trusted environment
- Supports USB eFEX protocol and UART mboot protocol for firmware upgrade

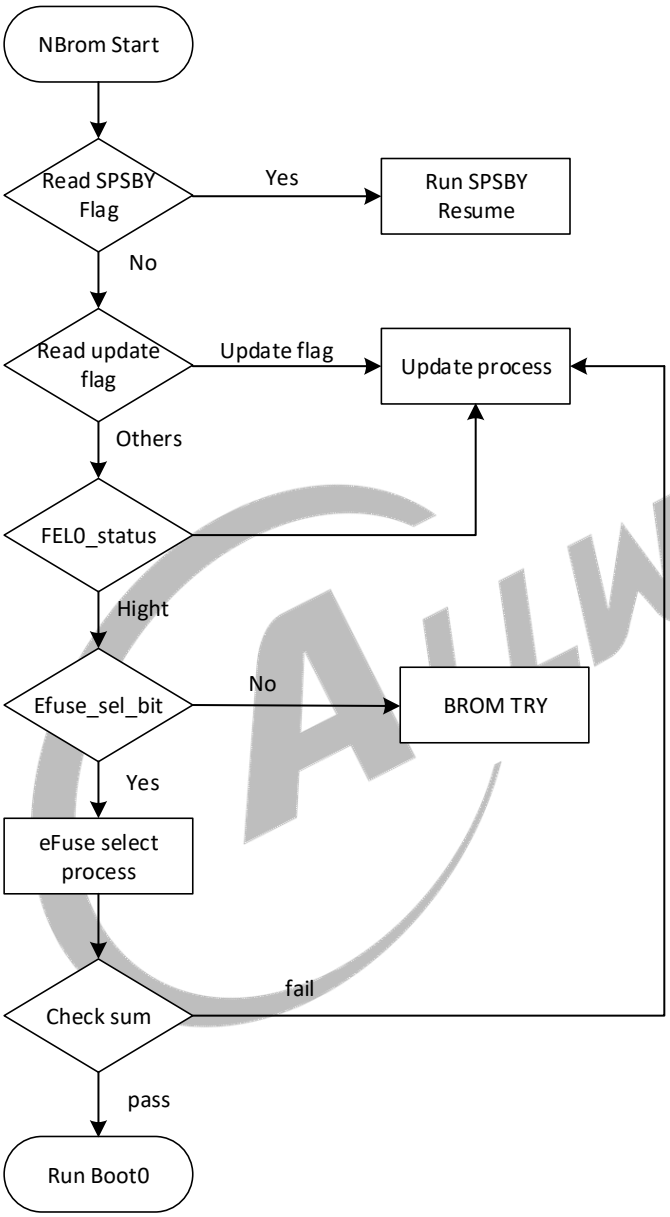
3.6.2 Functional Description

3.6.2.1 Startup Process

NBROM startup process

The following figure shows the startup process of NBROM.

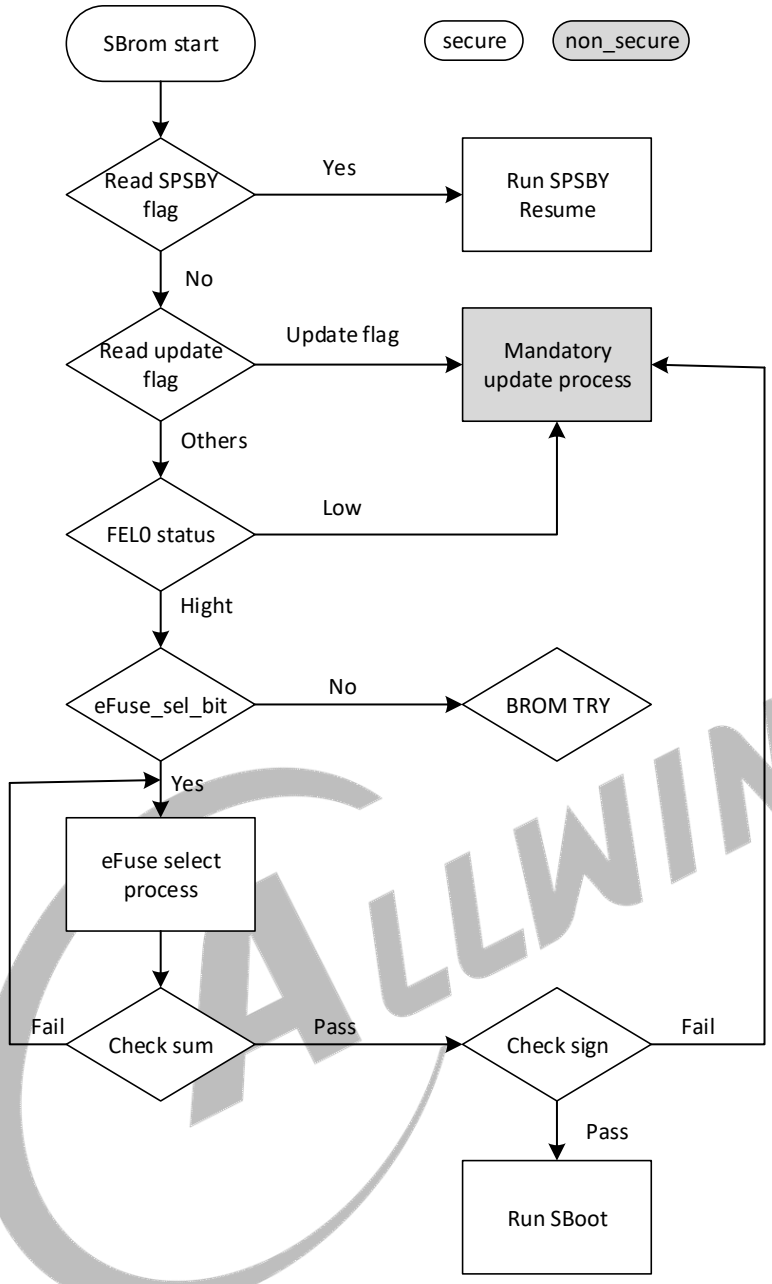
Figure 3-3 NBROM startup process



SBROM startup process

SBROM supports the signature algorithm of RSA2048-SHA256 and ECC256-SHA256. The following figure shows the startup process of SBROM.

Figure 3-4 SBROM startup process



3.6.2.2 Boot Medium Select

The BROM system supports the following boot media:

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

There are two ways to select the boot media: The TRY and the eFuse sel. On startup, the BROM will read the state of BOOT_SEL to decide the type of boot media.

The following table shows the BOOT_SEL setting.

Table 3-8 BOOT_MODE Setting

BOOT SEL	Boot_Select Type
0	the TRY
1	the eFuse sel

NOTE

BOOT SEL is the bit0 in system field.

The TRY

If the state of the BOOT SEL is 0, the boot media priority is SDC0->SPINOR -> SPI NAND->EMMC.

NOTE

a. The TRY merely polls the boot media of PA interface.

b. When the BROM starts by loading the SPINOR, it will query whether the package is SiP Flash. If yes, it will try PB8 first; in the case of failure, it will try other interfaces.

The eFuse sel

If the state of the BOOT_MODE is 1, the boot media is decided by the state of Select_Media_Field_Sel. It is divided into 2 groups. The following table shows the groups of Select_Media_Field_Sel.

Table 3-9 Groups of eFuse Boot Select

BOOT SEL[9:1]	Description
Select_Media_Field_Sel[0]	0: Select_Media_Field0 valid 1: Select_Media_Field1 valid
Select_Media_Field0[4:1]	Select_Media_Field0
Select_Media_Field1[8:5]	Select_Media_Field1

To meet the requirements of PIN selection in different schemes, each group of boot media priority is as follows:

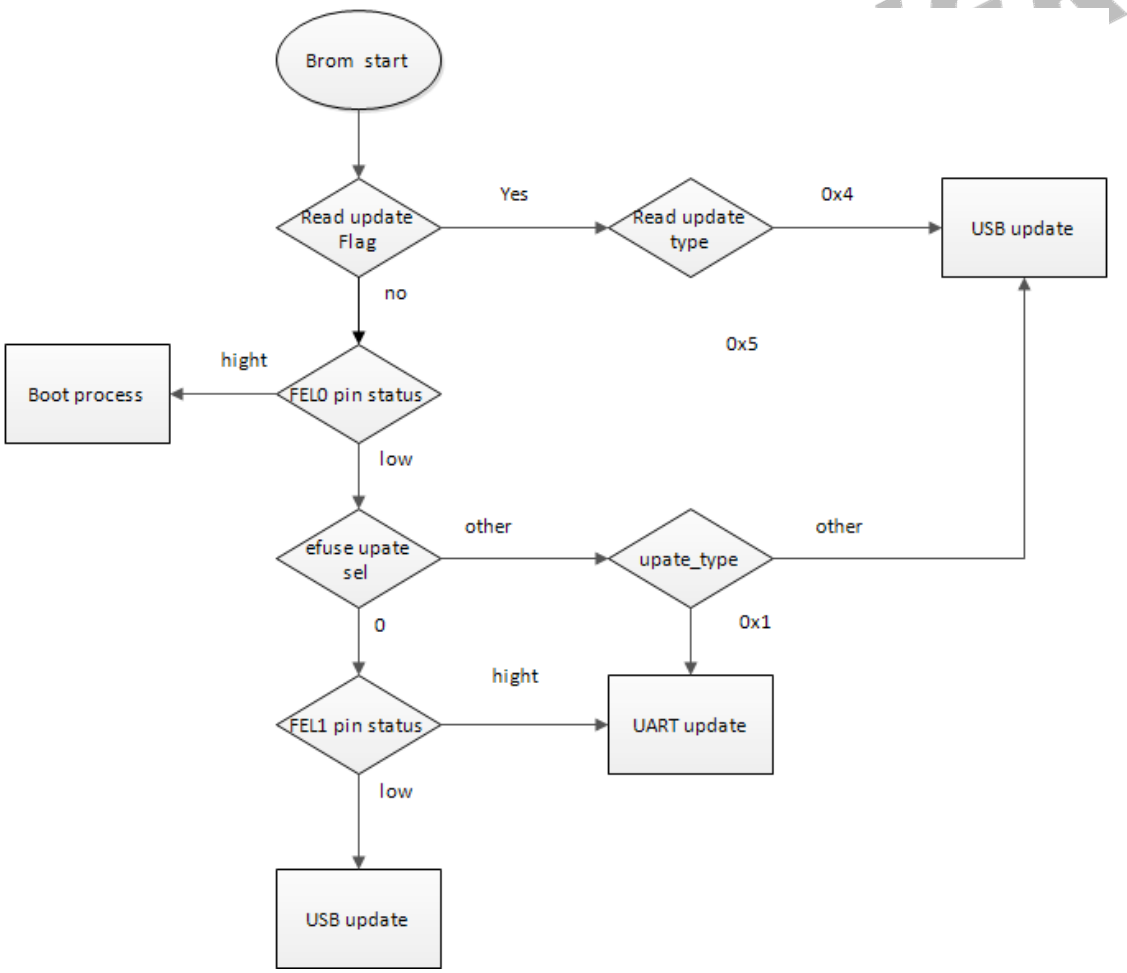
eFuse_Boot_Select	Boot media
0000	Nor_PB8->Nor_PB4->Nor_PA24->Nand_PA24->Nand_PB4->Nand_PB8->EMMC_PA24->EMMC_PB4->SD_PA24->SD_PA2
0001	Nor_PB4->Nor_PB8->Nor_PA24->Nand_PA24->Nand_PB4->Nand_PB8->EMMC_PA24->EMMC_PB4->SD_PA24->SD_PA2
0010	Nor_PA24->Nor_PB8->Nor_PB4->Nand_PA24->Nand_PB4->Nand_PB8->EMMC_PA24->EMMC_PB4->SD_PA24->SD_PA2
0011	Nand_PA24->Nand_PB4->Nand_PB8->Nor_PB8->Nor_PB4->Nor_PA24->EMMC_PA24->EMMC_PB4->SD_PA24->SD_PA2

eFuse_Boot_Select	Boot media
0100	Nand_PB4->Nand_PA24->Nand_PB8->Nor_PB8->Nor_PB4->Nor_PA24->EMMC_PA24->EMMC_PB4->SD_PA24->SD_PA2
0101	Nand_PB8->Nand_PA24->Nand_PB4->Nor_PB8->Nor_PB4->Nor_PA24->EMMC_PA24->EMMC_PB4->SD_PA24->SD_PA2
0110	EMMC_PA24->EMMC_PA8->Nor_PB8->Nor_PB4->Nor_PA24->Nand_PA24->Nand_PB4->Nand_PB8->SD_PA24->SD_PA2
0111	EMMC_PA8->EMMC_PA24->Nor_PB8->Nor_PB4->Nor_PA24->Nand_PA24->Nand_PB4->Nand_PB8->SD_PA24->SD_PA2

NOTE

Boot SEL[9:1] is [bit9:bit1] in the system filed.

3.6.2.3 Mass Production Upgrading Process



NOTE

a. The register of update flag and supper standby flag is the 0x1c0 of GPRCM module.

b. The status of FEL0 Pin is PA1.

c. The status of FEL1Pin is PA2

d. efuse update select is the bit [22:21] in system field.

e. The following table shows the meaning of different values of Boot flag.

Boot Flag	Meaning
0x0	reset
0x1	Supper standby
0x2	Upate process
0x3	HIBERNATE process
0x4	USB update
0x5	UART update

USB eFuse Communication Protocol

MAIN-CMD	SUB-CMD	CMD value	Description	Remarks
0x00	0x01	0x001		
0x01	0x01	0x101	Fel_down	Memory
	0x02	0x102	run	
	0x03	0x103	Fel_up	
	0x04	0x104	Set jtag	Enable/Disable JTAG
	0x05	0x105	reboot	
0x02	0x00	0x200	Obtain flash information	
	0x01	0x201	Chip erase	Flash erase
	0x02	0x202	Read n sectors	Flash reading operation
	0x03	0x203	Read n sectors	Flash reading operation

UART mBoot Communication Protocol

Host	MAIN-CMD	SUB-CMD	CMD value	Description	Remarks
PC	0x00	0x0	0x00	Read 1 byte	
		0x1	0x01	Write 1 byte	
		0x2	0x02	Read 2 bytes	
		0x3	0x03	Write 2 bytes	
		0x4	0x04	Read 4 bytes	
		0x5	0x05	Write 4 bytes	
		0x6	0x06	Read 8 bytes	
		0x7	0x07	Write 8 bytes	
	0x01	0x0	0x08	Read n bytes	

Host	MAIN-CMD	SUB-CMD	CMD value	Description	Remarks
		0x1	0x09	Write n bytes	
	0x02	0x0	0x10		Change the UART transmission condition
		0x1	0x11		Enable/Disable JTAG
		0x2	0x12		reboot
		0x3	0x13	Set PC pointer	
		0x4	0x14		Enable/disable MCU transmission and validation
		0x5	0x15		Obtain baud rate list
		0x6	0x16		Modify the buffer
	0x0	0x0	0x18	Obtain flash information	
	0x1	0x1	0x19	Chip erase	
		0x2	0x1A	Read n sectors	
		0x3	0x1B	Write n sectors	
		0x4	0x1C	Obtain flash information	
		0x5	0x1D	Chip erase	Flash erase
		0x6	0x1E	Read n sectors	Flash reading operation
		0x7	0x1F	Write n sectors	Flash reading operation
MCU	0x00	0			Send message to PC

3.7 RISC-V Subsystem

3.7.1 Overview

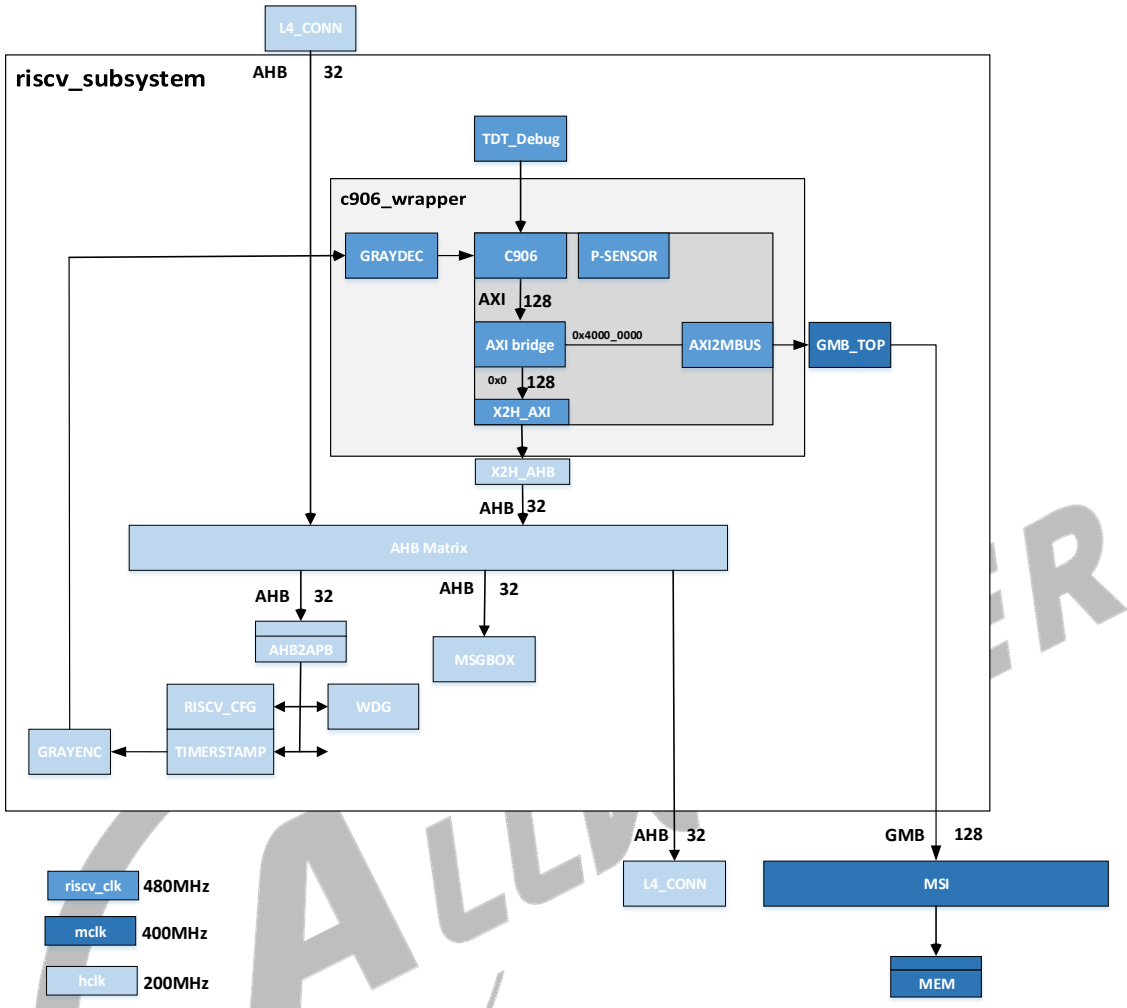
The RISC-V system includes RISCV IP core and related peripheral devices (RISCV_CFG, RISCV_TIMESTAMP, Watchdog, PSENSOR, BROM, and so on), which are interconnected by BUS Matrix.

The RISC-V system has the following features:

- RISC 64 GCV instruction architecture
- The 8-level single engine executes the pipeline in sequence
- Instruction and data cache of the first level Harvard, 32 KB I-cache + 32 KB D-cache, 64 B cacheline
- Instruction high-cache can configure parity, and data high-cache can configure ECC or parity
- Two-level TLB memory management unit to realize the virtual-real address translation and memory management
- Supports hardware automatic detection and instruction prefetch
- Supports AXI4.0 128-bit master interface
- Supports core-internal interrupt (CLINT) and interrupt controller (PLIC)
- Floating-point process unit
- Vector execution unit

3.7.2 Block Diagram

The following figure shows the block diagram of RISC-V system.



3.7.3 Register List

Module Name	Base Address
RISCV_CFG	0x40028000

Register Name	Offset	Description
RISCV_CFG		
RISCV_STA_ADD0_REG	0x0004	RISCV Start Address0 Register
RISCV_STA_ADD1_REG	0x0008	RISCV Start Address1 Register
RF1P_CFG_REG	0x0010	RF1P Configuration Register
ROM_CFG_REG	0x001C	ROM Configuration Register
WAKEUP_EN_REG	0x0020	Wakeup Enable Register
WAKEUP_MASK0_REG	0x0024	Wakeup Mask0 Register
WAKEUP_MASK1_REG	0x0028	Wakeup Mask1 Register
WAKEUP_MASK2_REG	0x002C	Wakeup Mask2 Register

Register Name	Offset	Description
WAKEUP_MASK3_REG	0x0030	Wakeup Mask3 Register
WAKEUP_MASK4_REG	0x0034	Wakeup Mask4 Register
TS_TMODE_SEL_REG	0x0040	Timestamp Test Mode Select Register
SRAM_ADDR_TWIST_REG	0x0044	SRAM Address Twist Register
WORK_MODE_REG	0x0048	Work Mode Register
IRQ_MODE0_REG	0x0060	IRQ Mode0 Register
IRQ_MODE1_REG	0x0064	IRQ Mode1 Register
IRQ_MODE2_REG	0x0068	IRQ Mode2 Register
IRQ_MODE3_REG	0x006C	IRQ Mode3 Register
IRQ_MODE4_REG	0x0070	IRQ Mode4 Register
RISCV_AXI_PMU_CTRL	0x0104	RISCV AXI PMU Control Register
RISCV_AXI_PMU_PRD	0x0108	RISCV AXI PMU Period Register
RISCV_AXI_PMU_LAT_RD	0x010C	RISCV AXI PMU Read Latency Register
RISCV_AXI_PMU_LAT_WR	0x0110	RISCV AXI PMU Write Latency Register
RISCV_AXI_PMU_REQ_RD	0x0114	RISCV AXI PMU Read Request Register
RISCV_AXI_PMU_REQ_WR	0x0118	RISCV AXI PMU Write Request Register
RISCV_AXI_PMU_BW_RD	0x011C	RISCV AXI PMU Read Bandwidth Register
RISCV_AXI_PMU_BW_WR	0x0120	RISCV AXI PMU Write Bandwidth Register

3.7.4 Register Description

3.7.4.1 0x0004 RISCV Start Address0 Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: RISCV_STA_ADD0_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	STA_ADD_L Start Address Low 32bit The bit0 is fixed as 0 and cannot be written.

3.7.4.2 0x0008 RISCV Start Address1 Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: RISCV_STA_ADD1_REG
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7: 0	R/W	0x0	STA_ADD_H. Start Address High 8bit.

NOTE

The 0x0004 register and 0x0008 register are grouped into a 40-bit address which is the running PC address after RISC releases reset. Before releasing reset, this register should be configured. This register doesn't support dynamic configuration.

3.7.4.3 0x0010 RF1P Configuration Register (Default Value: 0x0001_0002)

Offset: 0x0010			Register Name: RF1P_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
17:16	R/W	0x1	FAST_RF1P_CFG FAST_RF1P Configuration FAST_RF1P delay setting parameter
15:8	/	/	/
7: 0	R/W	0x2	RF1P_CFG RF1P Configuration RF1P setting parameter

3.7.4.4 0x001C ROM Configuration Register (Default Value: 0x0000_0002)

Offset: 0x001C			Register Name: ROM_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7: 0	R/W	0x2	ROM_CFG ROM Configuration. ROM setting parameter

3.7.4.5 0x0020 Wakeup Enable Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: WAKEUP_EN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	WP_EN Wakeup Enable CPUX will wake up the enabled bit when riscv enters the low-power mode.

3.7.4.6 0x0024 Wakeup Mask0 Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: WAKEUP_MASK0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	WP_MASK0 Wakeup Mask0

NOTE

The 0x0024 to 0x0034 registers correspond to the wakeup enable bits of 160 interrupts, which are disabled by default. When some interrupt needs to be waken-up, the corresponding bit needs to be set to 1.

3.7.4.7 0x0028 Wakeup Mask1 Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: WAKEUP_MASK1_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	WP_MASK1 Wakeup Mask1

3.7.4.8 0x002C Wakeup Mask2 Register (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: WAKEUP_MASK2_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	WP_MASK2 Wakeup Mask2

3.7.4.9 0x0030 Wakeup Mask3 Register (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: WAKEUP_MASK3_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	WP_MASK3 Wakeup Mask3

3.7.4.10 0x0034 Wakeup Mask4 Register (Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: WAKEUP_MASK4_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	WP_MASK4 Wakeup Mask4

3.7.4.11 0x0040 Timestamp Test Mode Select Register (Default Value: 0x0000_0000)

Offset: 0x0040			Register Name: TS_TMODE_SEL_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	TS_TEST_MODE_EN Timestamp Test Mode Enable 0: Normal Mode 1: Test Mode **Note**: *In the Test Mode, this Counter Low/Hi registers will count simultaneously.*

3.7.4.12 0x0044 SRAM Address Twist Register (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: SRAM_ADDR_TWIST_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	W	UDF	SRAM_TS_KF SRAM Twist Keyfield The bit 0 can be written only if this key field is written by 0x16AA.
15:1	/	/	/
0	R/W	0x0	SRAM_ADDR_TS_FG SRAM Address Twist Flag When this bit is set to 1, the RISCV_BROM area will become invisible, and the start address of SRAM A1 will be mapped to 0x20000 and 0x0 at the same time. In other words, the data read from 0x0 to 0Xffff is the same as that read from 0x20000 to 0x2FFFF.

3.7.4.13 0x0048 Work Mode Register (Default Value: 0x0000_0003)

Offset: 0x0048			Register Name: WORK_MODE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R	0x0	DM_STA Debug Mode Status 0: Normal Mode 1: Debug Mode
1: 0	R	0x3	LP_STA Low Power Status 00: Low Power Mode 01: Reserved 10: Reserved 11: Normal Mode is reserved

3.7.4.14 0x0060 IRQ Mode0 Register (Default Value: 0x0000_0000)

Offset: 0x0060			Register Name: Irq_MODE0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	IRQ_MD0 IRQ Mode 0 0: It is triggered in the high level. 1: It is triggered in the rising edge.

3.7.4.15 0x0064 IRQ Mode1 Register (Default Value: 0x0000_0000)

Offset: 0x0064			Register Name: Irq_MODE1_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	IRQ_MD1 IRQ Mode1 0: It is triggered in the high level. 1: It is triggered in the rising edge.

3.7.4.16 0x0068 IRQ Mode2 Register (Default Value: 0x0000_0000)

Offset: 0x0068			Register Name: Irq_MODE2_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	IRQ_MD2 IRQ Mode2 0: It is triggered in the high level. 1: It is triggered in the rising edge.

3.7.4.17 0x006C IRQ Mode3 Register (Default Value: 0x0000_0000)

Offset: 0x006C			Register Name: Irq_MODE3_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	IRQ_MD3 IRQ Mode3 0: It is triggered in the high level. 1: It is triggered in the rising edge.

3.7.4.18 0x0070 IRQ Mode4 Register (Default Value: 0x0000_0000)

Offset: 0x0070			Register Name: Irq_MODE4_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	IRQ_MD4 IRQ Mode4 0: It is triggered in the high level. 1: It is triggered in the rising edge.

3.7.4.19 0x0104 RISCV AXI PMU Control Register (Default Value: 0x0000_0000)

Offset: 0x0104			Register Name: RISCV_AXI_PMU_CTRL
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/

Offset: 0x0104			Register Name: RISCV_AXI_PMU_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
1	WC	0x0	PMU_CLR PMU Clear 0: No operation 1: PMU is cleared.
0	R/W	0x0	PMU_EN PMU Enable 0: PMU is disabled. 1: PMU is enabled.

3.7.4.20 0x0108 RISCV AXI PMU Period Register (Default Value: 0x0000_0000)

Offset: 0x0108			Register Name: RISCV_AXI_PMU_PRD
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	PRD Monitor Period Suggest that the field is in units of 1 us (1 ms).

3.7.4.21 0x010C RISCV AXI PMU Read Latency Register (Default Value: 0x0000_0000)

Offset: 0x010C			Register Name: RISCV_AXI_PMU_LAT_RD
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	LAT_RD Monitor the total latency of read-channel within the period

3.7.4.22 0x0110 RISCV AXI PMU Write Latency Register (Default Value: 0x0000_0000)

Offset: 0x0110			Register Name: RISCV_AXI_PMU_LAT_WR
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	LAT_WR Monitor the total latency of write-channel within the period

3.7.4.23 0x0114 RISCV AXI PMU Read Request Register (Default Value: 0x0000_0000)

Offset: 0x0114			Register Name: RISCV_AXI_PMU_REQ_RD
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	REQ_RD Monitor the total command numbers of read-channel within the period

3.7.4.24 0x0118 RISCV AXI PMU Write Request Register (Default Value: 0x0000_0000)

Offset: 0x0118	Register Name: RISCV_AXI_PMU_REQ_WR

Bit	Read/Write	Default/Hex	Description
31: 0	R	0x0	REQ_WR Monitor the total command numbers of write-channel within the period

3.7.4.25 0x011C RISCV AXI PMU Read Bandwidth Register (Default Value: 0x0000_0000)

Offset: 0x011C			Register Name: RISCV_AXI_PMU_BW_RD
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	BW_RD Monitor the total data (KB) of read-channel within the period

3.7.4.26 0x0120 RISCV AXI PMU Write Bandwidth Register (Default Value: 0x0000_0000)

Offset: 0x0120			Register Name: RISCV_AXI_PMU_BW_WR
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	BW_WR Monitor the total data (KB) of write-channel within the period

3.8 Platform-Level Interrupt Controller (PLIC)

3.8.1 Overview

The PLIC is only used for sampling, priority arbitration and distribution for the external interrupt sources of C906.

- The interrupt can be configured as machine mode and super user mode
- Up to 256 interrupt source sampling, supporting level interrupt and pulse interrupt
- 32 levels of interrupt priority
- Independently maintains the interrupt enable for each interrupt mode (machine/super user)
- Independently maintains the interrupt threshold for each interrupt mode (machine/super user)
- Configurable access permission for PLIC registers

3.8.2 Functional Description

The following table describes the details of interrupt sources.

Table 3-10 Interrupt Sources

Interrupt Number	Interrupt Source	Interrupt Vector	Description
0~15	Reserved	0x00~0x3C	Not Used
16	RTC_WUP_TIMER	0x40	
17	CPU_WDG	0x44	
18	CPU_MBOX_R	0x48	
19	CPU2RV_MBOX_W	0x4C	
20	CPU2DSP_MBOX_W	0x50	
21		0x54	
22		0x58	
23		0x5C	
24		0x60	
25		0x64	
26	ALARM0	0x68	
27	ALARM1	0x6C	
28	DMA0_0_S	0x70	
29	DMA0_0_NS	0x74	
30	DMA0_1_S	0x78	
31	DMA0_1_NS	0x7C	
32	DMA1_S	0x80	
33	DMA1_NS	0x84	
34		0x88	
35		0x8C	

Interrupt Number	Interrupt Source	Interrupt Vector	Description
36	TIMER0	0x90	
37	TIMER1	0x94	
38	TIMER2	0x98	
39	TIMER3	0x9C	
40	TIMER4	0xA0	
41		0xA4	
42		0xA8	
43		0xAC	
44	WKUP_TIMER0	0xB0	
45	WKUP_TIMER1	0xB4	
46	WKUP_TIMER2	0xB8	
47		0xBC	
48	TWD	0xC0	
49	UART0	0xC4	
50	UART1	0xC8	
51	UART2	0xCC	
52		0xD0	
53		0xD4	
54		0xD8	
55	LPUART0	0xDC	
56	LPUART1	0xE0	
57	TWI0	0xE4	
58	TWI1	0xE8	
59		0xEC	
60		0xF0	
61	SPI0	0xF4	
62	SPI1	0xF8	
63		0xFC	
64	PWM	0x100	
65	IRRX	0x104	
66	IRTX	0x108	
67	LEDC	0x10C	
68		0x110	
69	LPSRAM	0x114	
70	SPI_FLASH	0x118	
71	ADC	0x11C	
72	DAC_RX	0x120	
73	DAC_TX	0x124	
74	MAD_WAKE	0x128	
75	MAD_DATA_REQ	0x12C	
76	DMIC	0x130	
77	I2S0	0x134	
78	OWA	0x138	

Interrupt Number	Interrupt Source	Interrupt Vector	Description
79	USB0_EHCI	0x13C	
80	USB0_OHCI	0x140	
81	USB0_OTG	0x144	
82		0x148	
83		0x14C	
84		0x150	
85	SCR	0x154	
86	SPINLOCK	0x158	
87	SD0	0x15C	
88	SRAM0_TZMA	0x160	
89	SRAM1_TZMA	0x164	
90	SRAM2_TZMA	0x168	
91	SRAM3_TZMA	0x16C	
92	LPSRAM_TZMA	0x170	
93	FLASH_TZMA	0x174	
94	EXPSRAM_TZMA	0x178	
95	CE_S	0x17C	
96	CE_NS	0x180	
97	SMC	0x184	
98	MSI	0x188	
99		0x18C	
100		0x190	
101		0x194	
102		0x198	
103	G2D	0x19C	
104	DE	0x1A0	
105	LCD0	0x1A4	
106		0x1A8	
107		0x1AC	
108		0x1B0	
109	CSI	0x1B4	
110	VBAT_MON	0x1B8	
111		0x1BC	
112		0x1C0	
113	DSP_DEE	0x1C4	
114	DSP_PE	0x1C8	
115	DSP_WDG	0x1CC	
116	DSP2CPU_MBOX_W	0x1D0	
117	DSP2RV_MBOX_W	0x1D4	
118	DSP_TZMA	0x1D8	
119		0x1DC	
120		0x1E0	
121	RV_MBOX_R	0x1E4	

Interrupt Number	Interrupt Source	Interrupt Vector	Description
122	RV2DSP_MBOX_W	0x1E8	
123	RV2CPU_MBOX_W	0x1EC	
124	RV_WDG	0x1F0	
125		0x1F4	
126	BT1	0x1F8	
127	BT2	0x1FC	
128	BT3	0x200	
129	BTCOEX	0x204	
130	BLE_LL	0x208	
131	GPIOA	0x20C	
132	GPIOB	0x210	
133	GPIOC	0x214	
134		0x218	
135	WLAN	0x21C	
136	RCCAL	0x220	
137	GPADC	0x224	
138		0x228	
139		0x22C	

3.8.3 Register List

Module Name	Base Address
RISCV_PERI	0x50000000

Register Name	Offset	Description
PLIC_PRIO_REGn	0x0000+n*0x0004(0<n<256)	PLIC Priority Register n
PLIC_IP_REGn	0x1000+n*0x0004(0≤n<9)	PLIC Interrupt Pending Register n
PLIC_MIE_REGn	0x2000+n*0x0004(0≤n<9)	PLIC Machine Mode Interrupt Enable Register n
PLIC_SIE_REGn	0x2080+n*0x0004(0≤n<9)	PLIC Superuser Mode Interrupt Enable Register n
PLIC_CTRL_REG	0x1FFFFC	PLIC Control Register
PLIC_MTH_REG	0x200000	PLIC Machine Threshold Register
PLIC_MCLAIM_REG	0x200004	PLIC Machine Claim Register
PLIC_STH_REG	0x201000	PLIC Superuser Threshold Register
PLIC_SCLAIM_REG	0x201004	PLIC Superuser Claim Register

3.8.4 Register Description

3.8.4.1 0x0000+n*0x0004 (0<n<256) PLIC Priority Register n (Default Value: 0x0000_0000)

Offset: 0x0000+n*0x0004 (0<n<256)			Register Name: PLIC_PRIO_REGn
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/

Offset: 0x0000+n*0x0004 (0<n<256)			Register Name: PLIC_PRIO_REGn
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	PLIC_PRIO PLIC Priority Supports for 32 different levels of priority Where, setting a priority as 0 indicates that the interrupt is invalid. Machine mode interrupts have unconditionally higher priority than super-user mode interrupts. When the interrupt target mode is the same, priority 1 is the lowest priority, priority 31 is the highest priority. When the same interrupts of multiple priorities are waiting arbitration, the interrupt source IDs will be compared, and the smaller one has the higher priority.

3.8.4.2 0x1000+n*0x0004 (0≤n<9) PLIC Interrupt Pending Register n (Default Value: 0x0000_0000)

Offset: 0x1000+n*0x0004 (0≤n<9)			Register Name: PLIC_IP_REGn
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	PLIC_IP PLIC Interrupt Pending

3.8.4.3 0x2000+n*0x0004 (0≤n<9) PLIC Machine Mode Interrupt Enable Register n (Default Value: 0x0000_0000)

Offset: 0x2000+n*0x0004 (0≤n<9)			Register Name: PLIC_MIE_REGn
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	PLIC_MIE PLIC Machine Mode Interrupt Enable

3.8.4.4 0x2080+n*0x0004 (0≤n<9) PLIC Superuser Mode Interrupt Enable Register n (Default Value: 0x0000_0000)

Offset: 0x2080+n*0x0004 (0≤n<9)			Register Name: PLIC_SIE_REGn
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	PLIC_SIE PLIC Superuser Mode Interrupt Enable

3.8.4.5 0x1FFFFC PLIC Control Register (Default Value: 0x0000_0000)

Offset: 0x1FFFFC			Register Name: PLIC_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/

Offset: 0x1FFFFC			Register Name: PLIC_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	PLIC_CTRL PLIC Control 0: Only the machine mode can access to all registers in PLIC. The super-user mode cannot access PLIC_CTRL, PLIC_PRIO, PLIC_IP, and PLIC_IE registers except the interrupt threshold register and the interrupt response/completion register. The normal-user mode cannot access any registers in PLIC. 1: The machine mode can access to all registers in PLIC. The super-user mode can access all registers except PLL_CTRL in PLIC. The normal-user mode cannot access any registers in PLIC.

3.8.4.6 0x200000 PLIC Machine Threshold Register (Default Value: 0x0000_0000)

Offset: 0x200000			Register Name: PLIC_MTH_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4: 0	R/W	0x0	PLIC_MTH. PLIC Machine Threshold Indicate the interrupt threshold of the current interrupt mode. If the threshold is configured to 0, it indicates that all interrupts are permitted.

3.8.4.7 0x200004 PLIC Machine Claim Register (Default Value: 0x0000_0000)

Offset: 0x200004			Register Name: PLIC_MCLAIM_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9: 0	R/W	0x0	PLIC_MCLAIM PLIC Machine Claim Read register: Return the current stored ID value of the register. The read operation indicates that the interrupt of the corresponding ID starts to perform. The PLIC starts to process the interrupt response. Write register: Indicate that the interrupt of the corresponding ID is complete. The writing operation cannot update the response/completion register. The PLIC starts to process the interrupt completion.

3.8.4.8 0x201004 PLIC Superuser Threshold Register (Default Value: 0x0000_0000)

Offset: 0x201000			Register Name: PLIC_STH_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4: 0	R/W	0x0	PLIC_STH PLIC Superuser Threshold Indicate the interrupt threshold of the current interrupt mode. If the threshold is configured to 0, it indicates that all interrupts are permitted.

3.8.4.9 0x201004 PLIC Superuser Claim Register (Default Value: 0x0000_0000)

Offset: 0x201004			Register Name: PLIC_SCLAIM_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9:0	R/W	0x0	PLIC_SCLAIM. PLIC Superuser Claim Read register: Return the current stored ID value of the register. The read operation indicates that the interrupt of the corresponding ID starts to perform. The PLIC starts to process the interrupt response. Write register: Indicate that the interrupt of the corresponding ID is complete. The writing operation cannot update the response/completion register. The PLIC starts to process the interrupt completion.

3.9 Nested Vectored Interrupt Controller (NVIC)

The following table describes the details of M33 interrupt sources.

Interrupt Number	Interrupt Source	Interrupt Vector	Description
0	RTC_WUP_TIMER	0x7C	
1	CPU_WDG	0x80	
2	CPU_MBOX_R	0x84	
3		0x88	
4		0x8C	
5		0x90	
6		0x94	
7		0x98	
8		0x9C	
9	BLE_SLPTMR_WUP	0xA0	ble wake up M33 irq(only for M33)
10	ALARM0	0xA4	
11	ALARM1	0xA8	
12	DMA0_0_S	0xAC	
13	DMA0_0_NS	0xB0	
14	DMA0_1_S	0xB4	
15	DMA0_1_NS	0xB8	
16	DMA1_S	0xBC	
17	DMA1_NS	0xC0	
18		0xC4	
19		0xC8	
20	TIMER0	0xCC	
21	TIMER1	0xD0	
22	TIMER2	0xD4	
23	TIMER3	0xD8	
24	TIMER4	0xDC	secure timer
25		0xE0	
26		0xE4	
27		0xE8	
28	WKUP_TIMER0	0xEC	
29	WKUP_TIMER1	0xF0	
30	WKUP_TIMER2	0xF4	
31		0xF8	
32	TWD	0xFC	
33	UART0	0x100	
34	UART1	0x104	
35	UART2	0x108	
36		0x10C	
37		0x110	

Interrupt Number	Interrupt Source	Interrupt Vector	Description
38		0x114	
39	LPUART0	0x118	
40	LPUART1	0x11C	
41	TWI0	0x120	
42	TWI1	0x124	
43		0x128	
44		0x12C	
45	SPI0	0x130	
46	SPI1	0x134	
47		0x138	
48	PWM	0x13C	
49	IRRX	0x140	
50	IRTX	0x144	
51	LEDC	0x148	
52		0x14C	
53	LPSRAM	0x150	
54	SPI_FLASH	0x154	
55	ADC	0x158	
56	DAC_RX	0x15C	
57	DAC_TX	0x160	
58	MAD_WAKE	0x164	
59	MAD_DATA_REQ	0x168	
60	DMIC	0x16C	
61	I2S0	0x170	
62	SPDIF	0x174	
63	USB0_EHCI	0x178	
64	USB0_OHCI	0x17C	
65	USB0_OTG	0x180	
66		0x184	
67		0x188	
68		0x18C	
69	SCR	0x190	
70	SPINLOCK	0x194	
71	SD0	0x198	
72	SRAM0_TZMA	0x19C	
73	SRAM1_TZMA	0x1A0	
74	SRAM2_TZMA	0x1A4	
75	SRAM3_TZMA	0x1A8	
76	LPSRAM_TZMA	0x1AC	
77	FLASH_TZMA	0x1B0	
78	EXPSRAM_TZMA	0x1B4	
79	CE_S	0x1B8	
80	CE_NS	0x1BC	

Interrupt Number	Interrupt Source	Interrupt Vector	Description
81	SMC	0x1C0	
82	MSI	0x1C4	
83		0x1C8	
84		0x1CC	
85		0x1D0	
86	PSENSOR	0x1D4	
87	G2D	0x1D8	
88	DE	0x1DC	
89	LCD0	0x1E0	
90		0x1E4	
91		0x1E8	
92		0x1EC	
93	CSI	0x1F0	
94	VBAT_MON	0x1F4	
95		0x1F8	
96		0x1FC	
97	DSP_DEE	0x200	
98	DSP_PE	0x204	
99	DSP_WDG	0x208	
100	DSP2CPU_MBOX_W	0x20C	
101		0x210	
102	DSP_TZMA	0x214	
103		0x218	
104		0x21C	
105		0x220	
106		0x224	
107	RV2CPU_MBOX_W	0x228	
108	RV_WDG	0x22C	
109		0x230	
110	BT1	0x234	
111	BT2	0x238	
112	BT3	0x23C	
113	BTCOEX	0x240	
114	BLE_LL	0x244	
115	GPIOA	0x248	
116	GPIOB	0x24C	
117	GPIOC	0x250	
118		0x254	
119	WLAN	0x258	
120	RCCAL	0x25C	
121	GPADC	0x260	
122		0x264	
123		0x268	

Interrupt Number	Interrupt Source	Interrupt Vector	Description
124		0x26C	
125		0x270	
126		0x274	
127		0x278	

3.10 DMA Controller

3.10.1 Overview

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

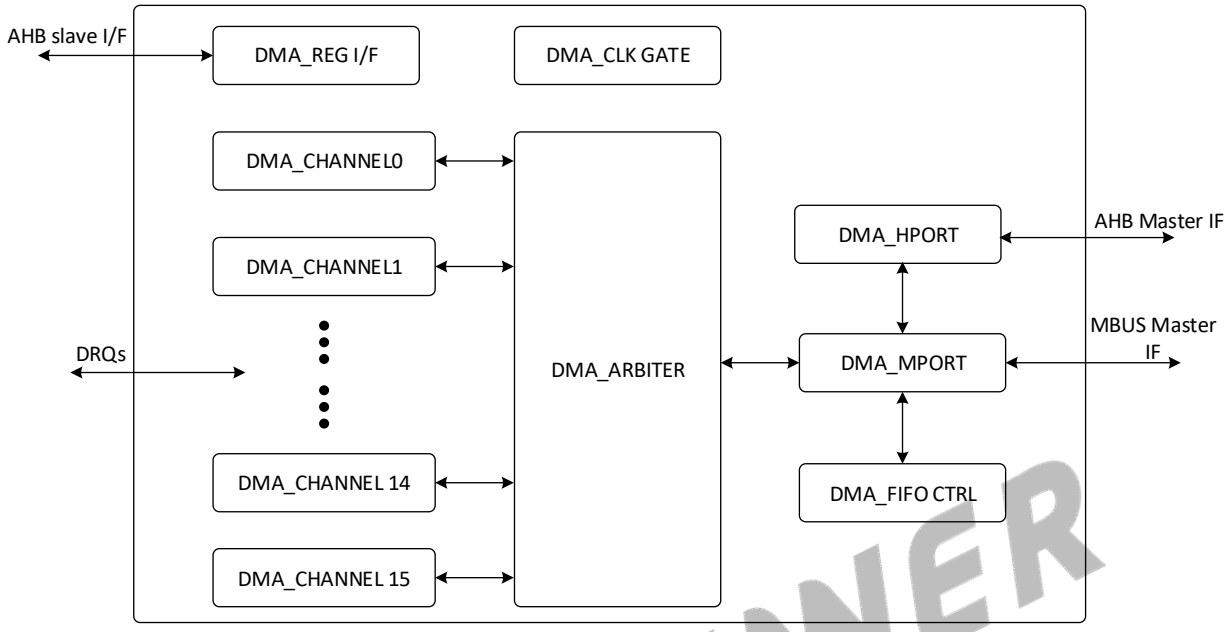
The DMAC has the following features:

- Up to 2 DMACs
- Up to 16 DMA channels
- Provides 32 peripheral DMA requests for data reading and 32 peripheral DMA requests for data writing
- Supports transferring data with a linked list
- Supports programmable 8-bit, 16-bit, 32-bit, and 64-bit data width
- Supports programmable DMA burst length
- DRQ response includes the waiting mode and handshake mode
- DMA channel supports pause function
- Memory devices support non-aligned transform

3.10.2 Block Diagram

The following figure shows a block diagram of DMAC.

Figure 3-5 DMAC Block Diagram



DMAC contains the following sub-blocks:

Table 3-11 DMAC Sub-blocks

Sub-block	Description
DMA_ARBITER	Arbitrates the DMA read/write requests from all channels, and converts the requests to the read/write requests of ports.
DMA_CHANNELs	DMA transfer engine. Each channel is independent. When the DMA requests from multiple peripherals are valid simultaneously, the channel with the highest priority starts data transmission first. The system uses the polling mechanism to decide the priorities of DMA channels.
DMA_MPORT	Receives the read/write requests from DMA_ARBITER, and converts the requests to the corresponding MBUS access requests. It is mainly used for accessing the DRAM.
DMA_HPORT	The port for accessing the AHB Master. It is mainly used for accessing the SRAM and IO devices.
DMA_FIFO CTRL	Internal FIFO cell control module.
DMA_REG Interface	DMA_REG is the common register module that is mainly used to resolve AHB demands.
DMA_CLKGATE	The control module for hardware auto clock gating.

The DMAC integrates 16 independent DMA channels and each channel has an independent FIFO controller. When the DMA channel starts, the DMAC gets a DMA descriptor from the DMA_DESC_ADDR_REG and uses it as the configuration information for the data transmission of the current DMA package. Then the DMAC can transfer data between the specified devices. After transferring a DMA package, the DMAC judges if the

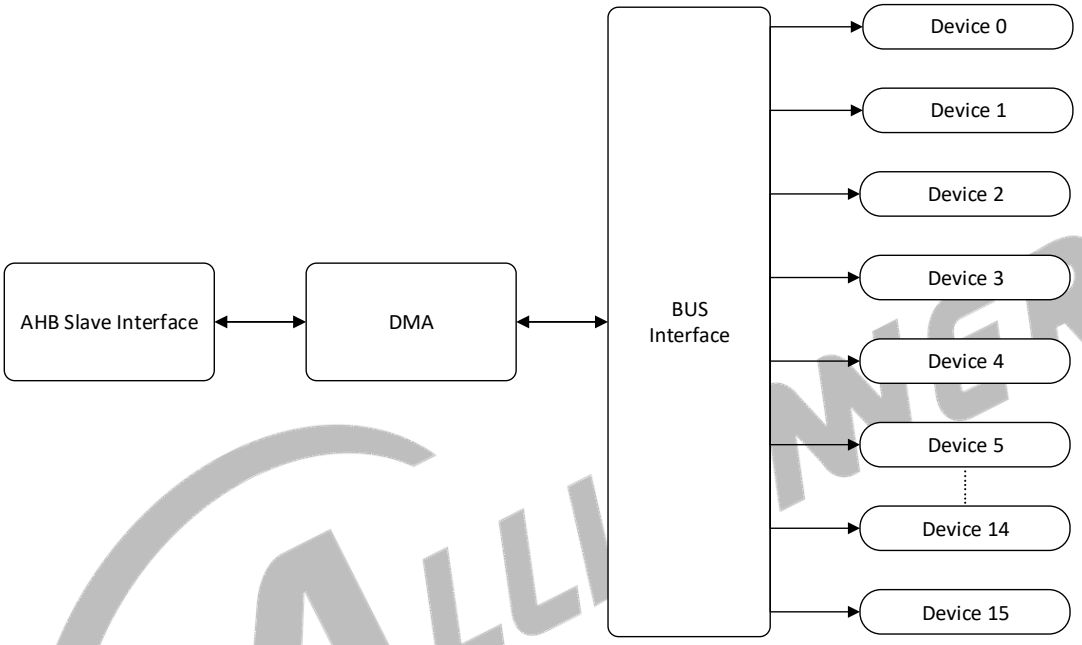
current channel transfer is finished via the linked address in the descriptor. If the linked address shows all the packages are transferred, the DMAC will end the chain transmission and close the channel.

3.10.3 Functional Description

3.10.3.1 Typical Application

The following figure shows a typical application of the DMAC.

Figure 3-6 DMAC Typical Application Diagram



3.10.3.2 DRQ Port of Peripherals

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

Table 3-12 DMA DRQ Type

Source DRQ Type		Destination DRQ Type	
port0	SRAM	port0	SRAM
port1	L-PSRAM	port1	L-PSRAM
port2		port2	
port3	I2S0_RX	port3	I2S0_TX
port4		port4	
port5		port5	
port6	MAD	port6	MAD
port7	CODEC_ADC	port7	
port8	DMIC	port8	
port9	OWA	port9	OWA
port10	L-PSRAM CTRL	port10	L-PSRAM CTRL
port11	FLASH CTRL	port11	FLASH CTRL
port12	GPADC	Port12	

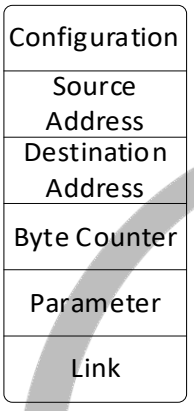
Source DRQ Type		Destination DRQ Type	
port13	CODEC_DAC_RX	port13	CODEC_DAC_TX
port14	UART0-RX	port14	UART0-TX
port15	UART1-RX	port15	UART1-TX
port16	UART2-RX	port16	UART2-TX
port17		port17	
port18		port18	
port19		port19	
port20		Port20	
port21		port21	
port22	SPI0-RX	port22	SPI0-TX
port23	SPI1-RX	port23	SPI1-TX
port24		port24	
port25		port25	
port26		port26	
port27		port27	
port28		port28	
port29		port29	
Port30	OTG0_EP1	Port30	OTG0_EP1
Port31	OTG0_EP2	Port31	OTG0_EP2
Port32	OTG0_EP3	Port32	OTG0_EP3
Port33	OTG0_EP4	Port33	OTG0_EP4
Port34	OTG0_EP5	Port34	OTG0_EP5
Port35		Port35	
Port36		Port36	
Port37		Port37	
Port38		Port38	
Port39		Port39	
Port40		Port40	
Port41		Port41	IRTX
Port42		Port42	LEDC
Port43	TWI0	Port43	TWI0
Port44	TWI1	Port44	TWI1
Port45		Port45	
Port46		Port46	
Port47		Port47	
Port48		Port48	
Port49		Port49	
Port50		Port50	
Port51		Port51	
Port52		Port52	
Port53		Port53	
Port54			
Port55			

Source DRQ Type		Destination DRQ Type	
Port56			
Port57			
Port58			
Port59			
Port60			
Port61			
Port62			
Port63			

3.10.3.3 DMA Descriptor

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

Figure 3-7 DMA Descriptor



- Configuration: Configure the following information by DMA_CFG_REG.
  - DRQ type: DRQ type of the source and destination devices.
  - Address counting mode: For both the source and destination devices, there are two address counting modes: The IO mode and linear mode. The IO mode is for IO devices whose address is fixed during the data transmission and the linear mode is for the memory whose address is increasing during the data transmission.
  - Transferred block length: The amount of data that non-memory peripherals can transfer in a valid DRQ. The block length supports 1 bit, 4 bits, 8 bits, and 16 bits.
  - Transferred data width: The data width of operating the non-memory peripherals. The data width supports 8 bits, 16 bits, 32 bits, and 64 bits.

NOTE

The configuration supports BMODE mode. The BMODE is used in the following scenario: The source is an IO device, and the destination is a memory device. Setting the BMODE mode can limit the amount of block data

transmission in DMA block transmission to the amount of data transmission when the DRQ threshold of the source IO device is 1.

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

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

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

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

Table 3-13 Source/Destination Address Distribution

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

From the above table, you can get:

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

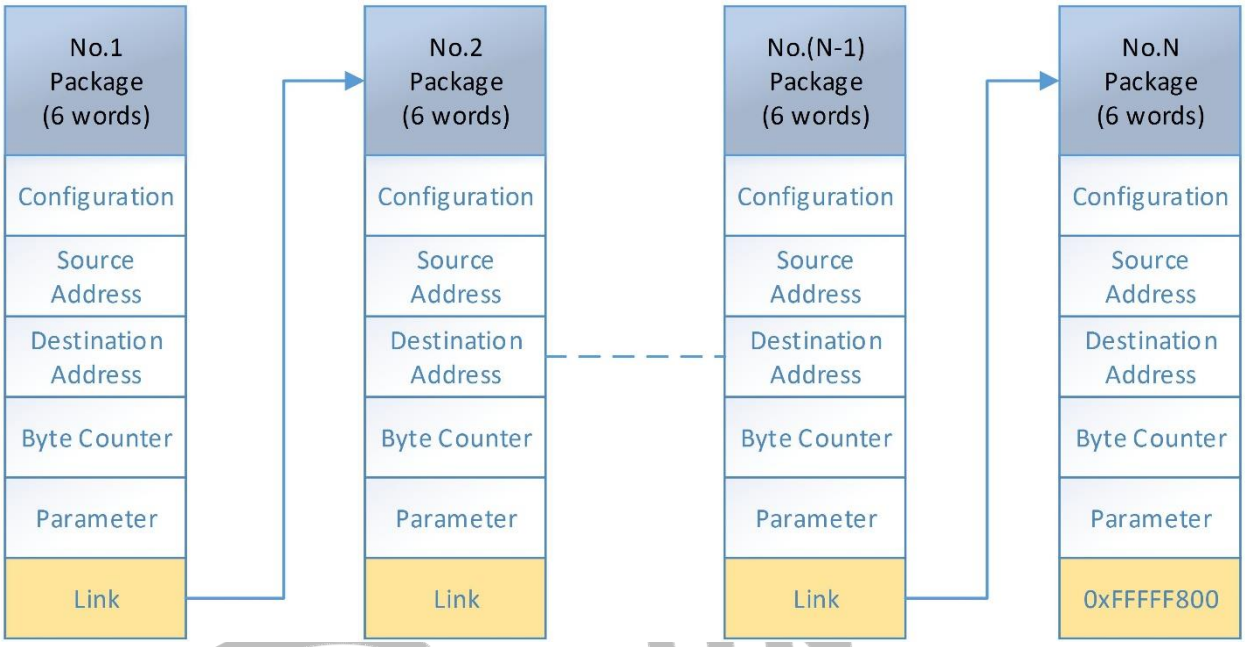
Real DMA destination address (in byte mode) = {Parameter [19:18], Destination Address [31: 0]};

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

- Byte counter: Configure the data amount of a package. The maximum value is (2^25-1) bytes. If the data amount of the package reaches the maximum value, even if DRQ is valid, the DMA will stop the current transfer.
- Parameter: Configure the interval between the data block. The parameter is valid for non-memory peripherals. When DMA detects that the DRQ is high, the DMA transfers the data block and ignores the status changes of the DRQ until the data transmission finishes. After that, the DMA waits for certain clock cycles (WAIT_CYC) and executes the next DRQ detection.

- Link: If the value of the link is 0xFFFFF800, the current package is at the end of the linked list. The DMAC will stop the data transmission after transferring the package; otherwise, the value of the link is considered as the descriptor address of the next package.

Figure 3-1 DMA Chain Transfer



3.10.3.4 Interrupts

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

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

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

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

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

NOTE

The DMA0 uses two groups of 8 channels. The channel [7: 0] corresponds to one group of interrupt, the channel [15:8] corresponds to another group of interrupt. Both of the two channels include safe and unsafe interrupts.

The DMA1 uses one group of 16 channels which correspond to one group of interrupts. The channel includes safe and unsafe interrupts.

3.10.3.5 Clock Gating

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

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

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

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

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

3.10.3.6 Transferring Mode

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

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

The principle of waiting mode

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

The principle of handshake mode

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

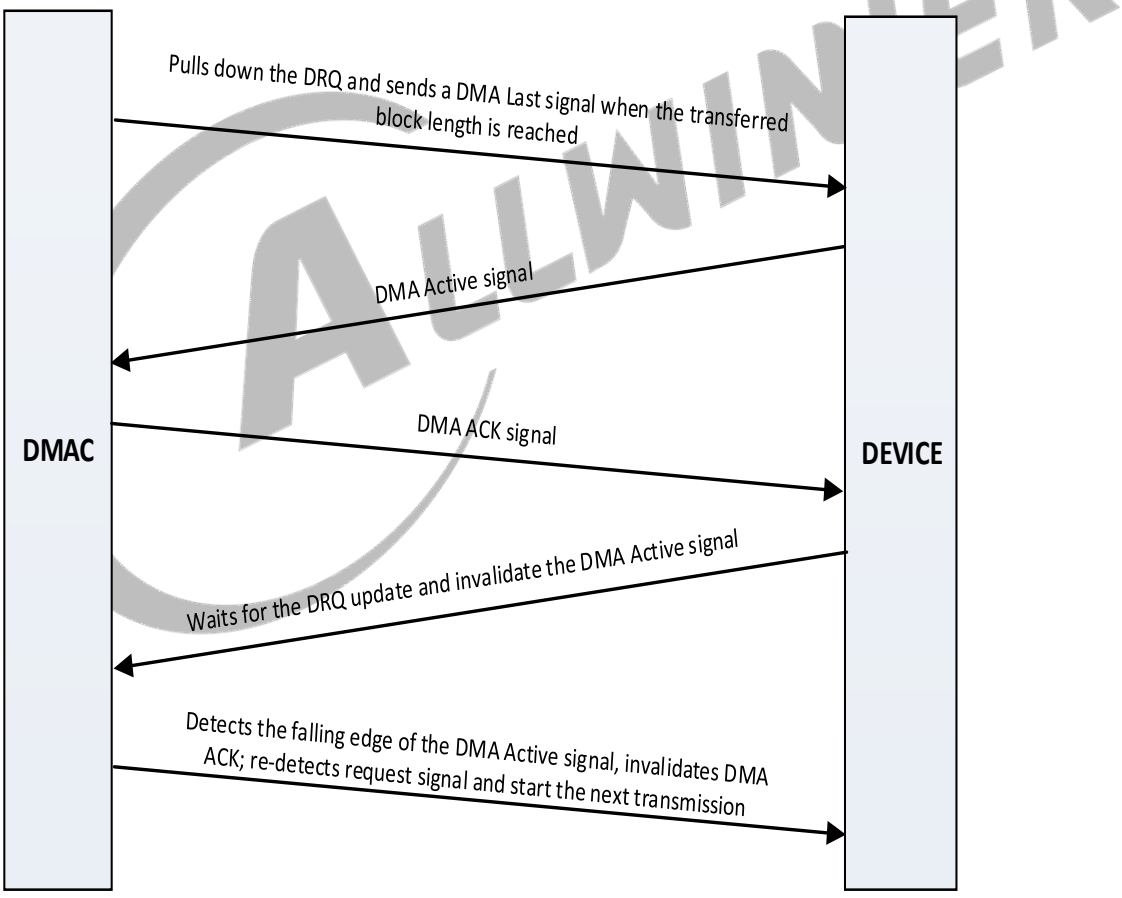
NOTE

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

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

The following figure shows the workflow of the handshake mode.

Figure 3-8 Workflow of the DMAC Handshake Mode



3.10.3.7 Address Auto-Alignment

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

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

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

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

3.10.3.8 DMAC Clock Control

The DMAC clock is synchronous with the AHB0 clock. Make sure that the DMAC gating bit of AHB0 clock is enabled before accessing the DMAC register.

The reset input signal of the DMAC is asynchronous with AHB0 and is low valid by default. Make sure that the reset signal of the DMAC is de-asserted before accessing the DMA register.

To avoid the indefinite state within registers, de-assert the reset signal first, and then open the gating bit of AHB0.

The DMAC supports Clock Auto Gating function to reduce power consumption, the system will automatically disable the DMAC clock in the DMAC idle state. Clock Auto Gating is enabled by default.

3.10.4 Programming Guidelines

3.10.4.1 Transfer Process Using DMAC

The DMAC transfer process is as follows.

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

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

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

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

Step 5 The DMA obtains the descriptor information.

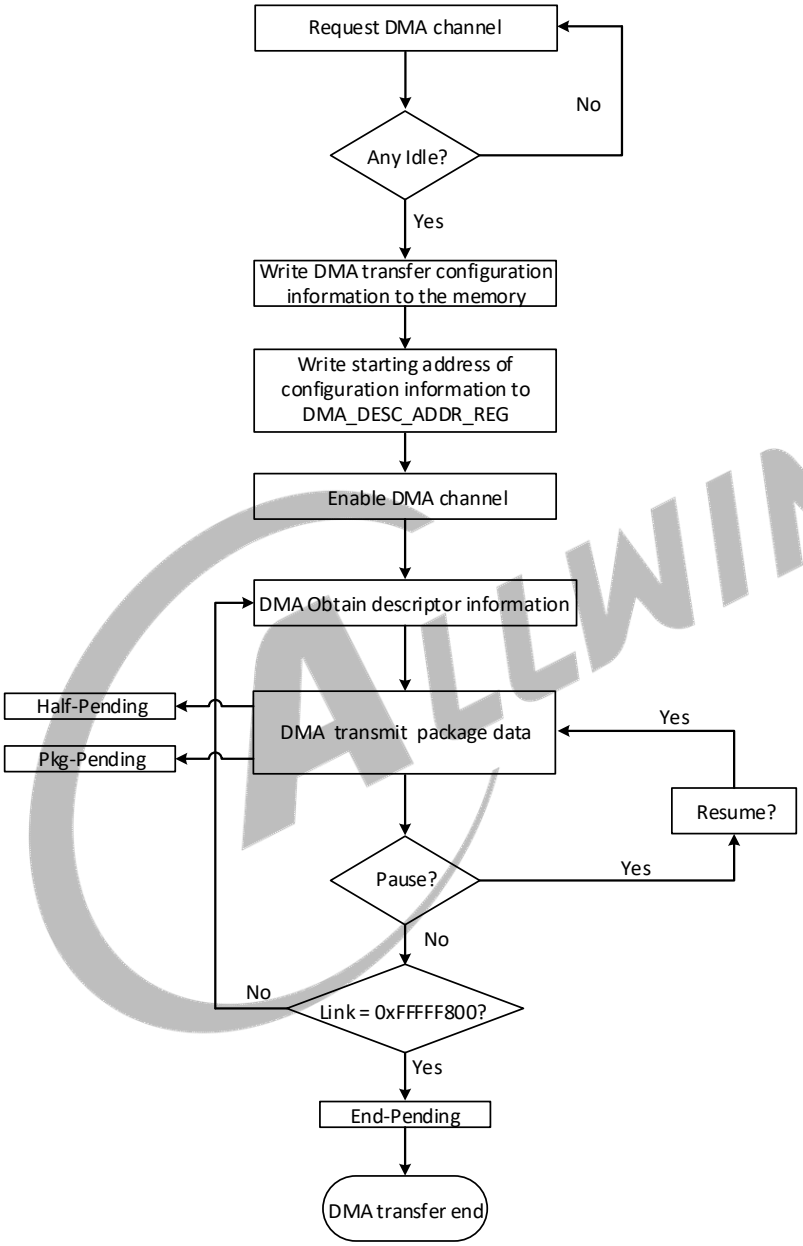
Step 6 Start to transmit a package. When half of the package is completed, the DMA sends a Half Package Transfer Interrupt; when a total package is completed, the DMA sends a Package End Transfer Interrupt. These interrupt statuses can be read by DMA_IRQ_PEND_REG0.

Step 7 Set DMA_PAU_REGN to pause or resume the data transmission.

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

Step 9 Disable the DMA channel.

Figure 3-9 DMAC Transfer Process



3.10.4.2 Processing DMAC Interrupt

Follow the steps below to process the DMAC interrupt:

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

Step 2 After entering the interrupt process, write to clear the interrupt pending and execute the process of waiting for the interrupt.

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

3.10.4.3 Configuring DMAC

To configure the DMAC, follow the guidelines below:

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

DMAC application example:

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

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

- The 0xFFFFF800 value of DMA_FDESC_ADDR_REGN indicates that the DMA channel has got back the descriptor of the last package. The DMA channel will automatically stop the data transmission after transferring the current package.
- To add a package during the data transmission, check if the DMA channel has got back the descriptor of the last package. If yes, you cannot add any package in the current queue. Request another DMA channel with a new DRQ to transfer the package. Otherwise, you can add the package by modifying the

DMA_FDESC_ADDR_REGN of the last package from 0xFFFFF800 to the start address of the to-be-added package.

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

  Another problem is, the system needs some time to process the modification, during which the DMA channel may get back the descriptor of the last package. You can read DMA_CUR_SRC_REGN and DMA_CUR_DEST_REGN and check if the increasing memory address accords with the information of the added package. If yes, the package is added successfully; otherwise, the modification failed.

- To ensure a higher rate of success, it is suggested that you add the package before the half package interrupt of the penultimate package.

3.10.5 Register List

Module Name	Base Address
DMAC0	0x40000000
DMAC1	0x40001000

Register Name	Offset	Description
DMAC0		
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+0x0000	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)
DMAC1		

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+0x0000	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 Register	0x0100+N*0x0040+0x0028	DMA_MODE_REG(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)

3.10.6 DMAC0 Register Description

3.10.6.1 0x0000 DMA IRQ Enable Register0 (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 DMA7 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
29	R/W	0x0	DMA7_PKG_IRQ_EN DMA7 Package End Transfer Interrupt Enable 0: Disable 1: Enable
28	R/W	0x0	DMA7_HLAF_IRQ_EN DMA7 Half Package Transfer Interrupt Enable 0: Disable 1: Enable

Offset: 0x0000			Register Name: DMA_IRQ_EN_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
27	/	/	/
26	R/W	0x0	DMA6_QUEUE_IRQ_EN DMA6 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
25	R/W	0x0	DMA6_PKG_IRQ_EN DMA6 Package End Transfer Interrupt Enable 0: Disable 1: Enable
24	R/W	0x0	DMA6_HLAF_IRQ_EN DMA6 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
23	/	/	/
22	R/W	0x0	DMA5_QUEUE_IRQ_EN DMA5 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
21	R/W	0x0	DMA5_PKG_IRQ_EN DMA5 Package End Transfer Interrupt Enable 0: Disable 1: Enable
20	R/W	0x0	DMA5_HLAF_IRQ_EN DMA5 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
19	/	/	/
18	R/W	0x0	DMA4_QUEUE_IRQ_EN DMA4 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
17	R/W	0x0	DMA4_PKG_IRQ_EN DMA4 Package End Transfer Interrupt Enable 0: Disable 1: Enable
16	R/W	0x0	DMA4_HLAF_IRQ_EN DMA4 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
15	/	/	/

Offset: 0x0000			Register Name: DMA_IRQ_EN_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
14	R/W	0x0	DMA3_QUEUE_IRQ_EN DMA3 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
13	R/W	0x0	DMA3_PKG_IRQ_EN DMA3 Package End Transfer Interrupt Enable 0: Disable 1: Enable
12	R/W	0x0	DMA3_HLAF_IRQ_EN DMA3 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
11	/	/	/
10	R/W	0x0	DMA2_QUEUE_IRQ_EN DMA2 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 DMA2 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
7	/	/	/
6	R/W	0x0	DMA1_QUEUE_IRQ_EN DMA1 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
5	R/W	0x0	DMA1_PKG_IRQ_EN DMA1 Package End Transfer Interrupt Enable 0: Disable 1: Enable
4	R/W	0x0	DMA1_HLAF_IRQ_EN DMA1 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
3	/	/	/

Offset: 0x0000			Register Name: DMA_IRQ_EN_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x0	DMA0_QUEUE_IRQ_EN DMA0 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
1	R/W	0x0	DMA0_PKG_IRQ_EN DMA0 Package End Transfer Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	DMA0_HLAF_IRQ_EN DMA0 Half Package Transfer Interrupt Enable 0: Disable 1: Enable

3.10.6.2 0x0004 DMA IRQ Enable Register1 (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 DMA15 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
29	R/W	0x0	DMA15_PKG_IRQ_EN DMA15 Package End Transfer Interrupt Enable 0: Disable 1: Enable
28	R/W	0x0	DMA15_HLAF_IRQ_EN DMA15 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
27	/	/	/
26	R/W	0x0	DMA14_QUEUE_IRQ_EN DMA14 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
25	R/W	0x0	DMA14_PKG_IRQ_EN DMA14 Package End Transfer Interrupt Enable 0: Disable 1: Enable

Offset: 0x0004			Register Name: DMA_IRQ_EN_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
24	R/W	0x0	DMA14_HLAF_IRQ_EN DMA14 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
23	/	/	/
22	R/W	0x0	DMA13_QUEUE_IRQ_EN DMA13 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
21	R/W	0x0	DMA13_PKG_IRQ_EN DMA13 Package End Transfer Interrupt Enable 0: Disable 1: Enable
20	R/W	0x0	DMA13_HLAF_IRQ_EN DMA13 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
19	/	/	/
18	R/W	0x0	DMA12_QUEUE_IRQ_EN DMA12 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
17	R/W	0x0	DMA12_PKG_IRQ_EN DMA12 Package End Transfer Interrupt Enable 0: Disable 1: Enable
16	R/W	0x0	DMA12_HLAF_IRQ_EN DMA12 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
15	/	/	/
14	R/W	0x0	DMA11_QUEUE_IRQ_EN DMA11 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
13	R/W	0x0	DMA11_PKG_IRQ_EN DMA11 Package End Transfer Interrupt Enable 0: Disable 1: Enable

Offset: 0x0004			Register Name: DMA_IRQ_EN_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
12	R/W	0x0	DMA11_HLAF_IRQ_EN DMA11 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
11	/	/	/
10	R/W	0x0	DMA10_QUEUE_IRQ_EN DMA10 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
9	R/W	0x0	DMA10_PKG_IRQ_EN DMA10 Package End Transfer Interrupt Enable 0: Disable 1: Enable
8	R/W	0x0	DMA2_HLAF_IRQ_EN DMA2 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
7	/	/	/
6	R/W	0x0	DMA9_QUEUE_IRQ_EN DMA9 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
5	R/W	0x0	DMA9_PKG_IRQ_EN DMA9 Package End Transfer Interrupt Enable 0: Disable 1: Enable
4	R/W	0x0	DMA9_HLAF_IRQ_EN DMA9 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
3	/	/	/
2	R/W	0x0	DMA8_QUEUE_IRQ_EN DMA8 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
1	R/W	0x0	DMA8_PKG_IRQ_EN DMA8 Package End Transfer Interrupt Enable 0: Disable 1: Enable

Offset: 0x0004			Register Name: DMA_IRQ_EN_REG1
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	DMA8_HLAF_IRQ_EN DMA8 Half Package Transfer Interrupt Enable 0: Disable 1: Enable

3.10.6.3 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 DMA7 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 DMA7 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 DMA7 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 DMA6 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 DMA6 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 DMA6 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
23	/	/	/

Offset: 0x0010			Register Name: DMA_IRQ_PEND_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
22	R/W1C	0x0	DMA5_QUEUE_IRQ_PEND DMA5 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 DMA5 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 DMA5 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 DMA4 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 DMA4 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 DMA4 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 DMA3 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 DMA3 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending

Offset: 0x0010			Register Name: DMA_IRQ_PEND_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
12	R/W1C	0x0	DMA3_HLAF_IRQ_PEND DMA3 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 DMA2 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 DMA2 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 DMA2 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 DMA1 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 DMA1 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 DMA1 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
3	/	/	/

Offset: 0x0010			Register Name: DMA_IRQ_PEND_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W1C	0x0	DMA0_QUEUE_IRQ_PEND DMA0 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 DMA0 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 DMA0 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending

3.10.6.4 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 DMA15 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 DMA15 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 DMA15 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 DMA14 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending

Offset: 0x0014			Register Name: DMA_IRQ_PEND_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
25	R/W1C	0x0	DMA14_PKG_IRQ_ PEND DMA14 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 DMA14 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 DMA13 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 DMA13 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 DMA13 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 DMA12 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 DMA12 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 DMA12 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending

Offset: 0x0014			Register Name: DMA_IRQ_PEND_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
15	/	/	/
14	R/W1C	0x0	DMA11_QUEUE_IRQ_PEND DMA11 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 DMA11 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 DMA11 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 DMA10 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 DMA10 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 DMA10 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 DMA9 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 DMA9 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending

Offset: 0x0014			Register Name: DMA_IRQ_PEND_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
4	R/W1C	0x0	DMA9_HLAF_IRQ_PEND DMA9 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 DMA8 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 DMA8 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 DMA8 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending

3.10.6.5 0x0020 DMA Security Register (Default Value: 0x0000_FFFF)

Offset: 0x0020			Register Name: DMA_SEC_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15	R/W	0x1	DMA15_SEC DMA channel 15 security 0: Secure 1: Non-secure
14	R/W	0x1	DMA14_SEC DMA channel 14 security 0: Secure 1: Non-secure
13	R/W	0x1	DMA13_SEC DMA channel 13 security 0: Secure 1: Non-secure

Offset: 0x0020			Register Name: DMA_SEC_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
12	R/W	0x1	DMA12_SEC DMA channel 12 security 0: Secure 1: Non-secure
11	R/W	0x1	DMA11_SEC DMA channel 11 security 0: Secure 1: Non-secure
10	R/W	0x1	DMA10_SEC DMA channel 10 security 0: Secure 1: Non-secure
9	R/W	0x1	DMA9_SEC DMA channel 9 security 0: Secure 1: Non-secure
8	R/W	0x1	DMA8_SEC DMA channel 8 security 0: Secure 1: Non-secure
7	R/W	0x1	DMA7_SEC DMA channel 7 security 0: Secure 1: Non-secure
6	R/W	0x1	DMA6_SEC DMA channel 6 security 0: Secure 1: Non-secure
5	R/W	0x1	DMA5_SEC DMA channel 5 security 0: Secure 1: Non-secure
4	R/W	0x1	DMA4_SEC DMA channel 4 security 0: Secure 1: Non-secure
3	R/W	0x1	DMA3_SEC DMA channel 3 security 0: Secure 1: Non-secure

Offset: 0x0020			Register Name: DMA_SEC_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x1	DMA2_SEC DMA channel 2 security 0: Secure 1: Non-secure
1	R/W	0x1	DMA1_SEC DMA channel 1 security 0: Secure 1: Non-secure
0	R/W	0x1	DMA0_SEC DMA channel 0 security 0: Secure 1: Non-secure

3.10.6.6 0x0028 DMA Auto Gating Register (Default Value: 0x00000000)

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, bit-2 should be set up.

3.10.6.7 0x0030 DMA Status Register (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: DMA_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0030			Register Name: DMA_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x0	MBUS FIFO Status 0: Empty 1: Not Empty
30:16	/	/	/
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

Offset: 0x0030			Register Name: DMA_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
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

3.10.6.8 0x0100+N*0x0040+0x0000 (N=0~15) DMA Channel Enable Register (Default Value: 0x0000_0000)

Offset: 0x0100+N*0x0040+0x0000 (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

3.10.6.9 0x0100+N*0x0040+0x0004 (N=0~15) 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

3.10.6.10 0x0100+N*0x0040+0x0008 (N=0~15) 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};

3.10.6.11 0x0100+N*0x0040+0x000C (N=0~15) 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	/	/	/
30	R	0x0	BMODE_SEL 0: Normal Mode 1: BMODE
29:27	/	/	/
26:25	R	0x0	DMA_DEST_DATA_WIDTH DMA Destination Data Width 00: 8 bit 01: 16 bit 10: 32 bit 11: 64 bit

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

3.10.6.12 0x0100+N*0x0040+0x0010 (N=0~15) 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**

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.

3.10.6.13 0x0100+N*0x0040+0x0014 (N=0~15) 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.

3.10.6.14 0x0100+N*0x0040+0x0018 (N=0~15) 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.

3.10.6.15 0x100+N*0x40+0x001C (N=0~15) DMA Channel Parameter Register (Default Value: 0x0000_0000)

Offset: 0x100+N*0x40+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

3.10.6.16 0x100+N*0x40+0x0028 (N=0~15) DMA Mode Register (Default Value: 0x0000_0000)

Offset: 0x100+N*0x40+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

Offset: 0x100+N*0x40+0x0028 (N=0~15)			Register Name: DMA_MODE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x0	DMA_SRC_MODE 0: Wait mode 1: Handshake mode
1: 0	/	/	/

3.10.6.17 0x0100+N*0x0040+0x002C (N=0~15) 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.

3.10.6.18 0x0100+N*0x0040+0x0030 (N=0~15) 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.

3.10.7 DMAC1 Register Description

3.10.7.1 0x0000 DMA IRQ Enable Register0 (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

Offset: 0x0000			Register Name: DMA_IRQ_EN_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
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

Offset: 0x0000			Register Name: DMA_IRQ_EN_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
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

Offset: 0x0000			Register Name: DMA_IRQ_EN_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
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

3.10.7.2 0x0004 DMA IRQ Enable Register1 (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: DMA_IRQ_EN_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31-7	/	/	/
6	R/W	0x0	DMA9_QUEUE_IRQ_EN DMA9 Queue End Transfer Interrupt Enable 0: Disable 1: Enable
5	R/W	0x0	DMA9_PKG_IRQ_EN DMA9 Package End Transfer Interrupt Enable 0: Disable 1: Enable
4	R/W	0x0	DMA9_HLAF_IRQ_EN DMA9 Half Package Transfer Interrupt Enable 0: Disable 1: Enable
3	/	/	/
2	R/W	0x0	DMA8_QUEUE_IRQ_EN DMA8 Queue End Transfer Interrupt Enable 0: Disable 1: Enable

Offset: 0x0004			Register Name: DMA_IRQ_EN_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x0	DMA8_PKG_IRQ_EN DMA8 Package End Transfer Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	DMA8_HLAF_IRQ_EN DMA8 Half Package Transfer Interrupt Enable 0: Disable 1: Enable

3.10.7.3 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

Offset: 0x0010			Register Name: DMA_IRQ_PEND_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
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.

Offset: 0x0010			Register Name: DMA_IRQ_PEND_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
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

Offset: 0x0010			Register Name: DMA_IRQ_PEND_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
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

3.10.7.4 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	/	/	/
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	/	/	/

Offset: 0x0014			Register Name: DMA_IRQ_PEND_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W1C	0x0	DMA8_QUEUE_IRQ_PEND DMA8 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 DMA8 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending

3.10.7.5 0x0020 DMA Security Register (Default Value: 0x0000_FFFF)

Offset: 0x0020			Register Name: DMA_SEC_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15	R/W	0x1	DMA15_SEC DMA channel 15 security 0: Secure 1: Non-secure
14	R/W	0x1	DMA14_SEC DMA channel 14 security 0: Secure 1: Non-secure
13	R/W	0x1	DMA13_SEC DMA channel 13 security 0: Secure 1: Non-secure
12	R/W	0x1	DMA12_SEC DMA channel 12 security 0: Secure 1: Non-secure
11	R/W	0x1	DMA11_SEC DMA channel 11 security 0: Secure 1: Non-secure

Offset: 0x0020			Register Name: DMA_SEC_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
10	R/W	0x1	DMA10_SEC DMA channel 10 security 0: Secure 1: Non-secure
9	R/W	0x1	DMA9_SEC DMA channel 9 security 0: Secure 1: Non-secure
8	R/W	0x1	DMA8_SEC DMA channel 8 security 0: Secure 1: Non-secure
7	R/W	0x1	DMA7_SEC DMA channel 7 security 0: Secure 1: Non-secure
6	R/W	0x1	DMA6_SEC DMA channel 6 security 0: Secure 1: Non-secure
5	R/W	0x1	DMA5_SEC DMA channel 5 security 0: Secure 1: Non-secure
4	R/W	0x1	DMA4_SEC DMA channel 4 security 0: Secure 1: Non-secure
3	R/W	0x1	DMA3_SEC DMA channel 3 security 0: Secure 1: Non-secure
2	R/W	0x1	DMA2_SEC DMA channel 2 security 0: Secure 1: Non-secure
1	R/W	0x1	DMA1_SEC DMA channel 1 security 0: Secure 1: Non-secure

Offset: 0x0020			Register Name: DMA_SEC_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x1	DMA0_SEC DMA channel 0 security 0: Secure 1: Non-secure

3.10.7.6 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, bit-2 should be set up.

3.10.7.7 0x0030 DMA Status Register (Default Value: 0x0000_0000)

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

Offset: 0x0030			Register Name: DMA_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
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

Offset: 0x0030			Register Name: DMA_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
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

3.10.7.8 0x0100+N*0x0040+0x0000 (N=0~15) DMA Channel Enable Register (Default Value: 0x0000_0000)

Offset: 0x0100+N*0x0040+0x0000 (N=0~9)			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

3.10.7.9 0x0100+N*0x0040+0x0004 (N=0~15) DMA Channel Pause Register (Default Value: 0x0000_0000)

Offset: 0x0100+N*0x0040+0x0004 (N=0~9)			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

3.10.7.10 0x0100+N*0x0040+0x0008 (N=0~15) DMA Channel Descriptor Address Register (Default Value: 0x0000_0000)

Offset: 0x0100+N*0x0040+0x0008 (N=0~9)			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};

3.10.7.11 0x0100+N*0x0040+0x000C (N=0~15) DMA Channel Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0100+N*0x0040+0x000C (N=0~9)			Register Name: DMA_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R	0x0	BMODE_SEL 0: Normal Mode 1: BMODE
29:27	/	/	/
26:25	R	0x0	DMA_DEST_DATA_WIDTH DMA Destination Data Width 00: 8-bit 01: 16-bit 10: 32-bit 11: 64-bit
24	R	0x0	DMA_ADDR_MODE DMA Destination Address Mode 0: Linear Mode 1: IO Mode
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

Offset: 0x0100+N*0x0040+0x000C (N=0~9)			Register Name: DMA_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
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

3.10.7.12 0x0100+N*0x0040+0x0010 (N=0~15) DMA Channel Current Source Address Register (Default Value: 0x0000_0000)

Offset: 0x0100+N*0x0040+0x0010 (N=0~9)			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

3.10.7.13 0x0100+N*0x0040+0x0014 (N=0~15) DMA Channel Current Destination Address Register (Default Value: 0x0000_0000)

Offset: 0x0100+N*0x0040+0x0014 (N=0~9)			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

3.10.7.14 0x0100+N*0x0040+0x0018 (N=0~15) DMA Channel Byte Counter Left Register (Default Value: 0x0000_0000)

Offset: 0x0100+N*0x0040+0x0018 (N=0~9)			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

3.10.7.15 0x100+N*0x0040+0x001C (N=0~15) DMA Channel Parameter Register (Default Value: 0x0000_0000)

Offset: 0x100+N*0x0040+0x001C (N=0~9)			Register Name: DMA_PARA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7: 0	R	0x0	WAIT_CYC Wait Clock Cycles

3.10.7.16 0x100+N*0x0040+0x0028 (N=0~15) DMA Mode Register (Default Value: 0x0000_0000)

Offset: 0x100+N*0x0040+0x0028 (N=0~9)			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	/	/	/

3.10.7.17 0x0100+N*0x0040+0x002C (N=0~15) DMA Former Descriptor Address Register (Default Value: 0x0000_0000)

Offset: 0x0100+N*0x0040+0x002C (N=0~9)			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.

3.10.7.18 0x0100+N*0x0040+0x0030 (N=0~15) DMA Package Number Register (Default Value: 0x0000_0000)

Offset: 0x0100+N*0x0040+0x0030 (N=0~9)			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.

3.11 Timer

3.11.1 Overview

The timer module implements the timing and counting functions. It includes timers 0-7. The main features of this timer are as follows:

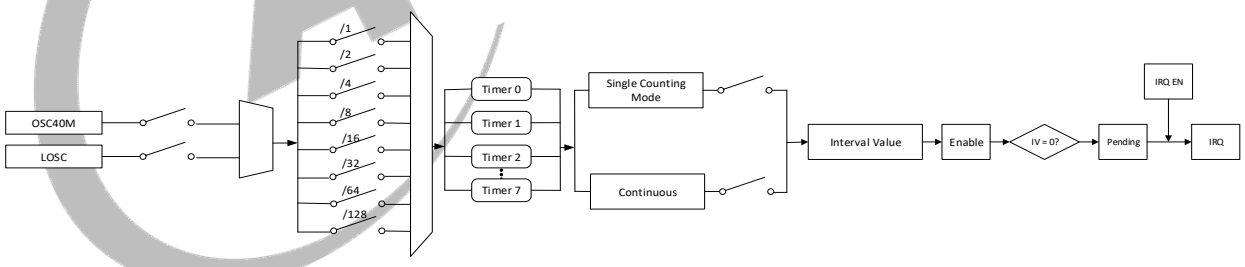
- 8 timers: 5 for SW domain, and 3 for AON domain (used for awaking system)
- Configurable counting clock: LOSC and OSC40M. Whether LOSC is internal low-frequency clock or external low-frequency clock (with greater accuracy) depends on LOSC_SRC_SEL.
- Supports 8 prescale factors
- Programmable 32-bit down timer
- Supports two timing modes: periodic mode and single counting mode
- Generates an interrupt when the count is decreased to 0

3.11.2 Block Diagram

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

The following figure shows the block diagram for the timer.

Figure 3-10 Block Diagram for the Timer



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

3.11.3 Functional Description

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

- Single Counting Mode

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

- Continuous Mode

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

Formula for Calculating the Timer Time

The following formula describes the relationship among timer parameters.

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

Where,

The parameter n is either 0 or 1;

Ttimer is the remaining time of the timer;

TMRn_INTV_VALUE_REG is the interval value of the timer;

TMRn_CUR_VALUE_REG is the current value of the timer;

TMRn_CLK_SRC is the frequency of the timer clock source;

TMRn_CLK_PRES is the prescale ratio of the timer clock.

3.11.4 Programming Guidelines

3.11.4.1 Initializing the Timer

Follow the steps below to initialize the timer:

Step 1 Configure the timer parameters clock source, prescale factor, and timing mode by writing TMRn_CTRL_REG (n=0~4). There is no sequence requirement when configuring the parameters.

Step 2 Write the interval value.

- Write TMRn_INTV_VALUE_REG (n=0~4) to configure the interval value for the timer.
- Write bit[1] of TMRn_CTRL_REG (n=0~4) to load the interval value to the timer. The value of the bit will be cleared automatically after loading the interval value.

Step 3 Write bit[0] of TMRn_CTRL_REG (n=0~4) to start the timer. To get the current value of the timer, read TMRn_CUR_VALUE_REG (n=0~4).

3.11.4.2 Processing the Interrupt

Follow the steps below to process the interrupt:

Step 1 Enable interrupts for the timer: Write the enable bit of the corresponding interrupt in TMR_IRQ_EN_REG for the timer. The timer will generate an interrupt every time when the count value reaches 0.

Step 2 After entering the interrupt process, write the pending bit of the corresponding interrupt in TMR_IRQ_STA_REG to clear the interrupt pending, and execute the process of waiting for the interrupt.

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

3.11.4.3 Configuring Timer

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

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

3.11.5 Register List

Module Name	Base Address
Timer	0x40043000
WAKEUP_TIMER	0x4004CC00

Register Name	Offset	Description
Timer		
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
TMR2_CTRL_REG	0x0030	Timer 2 Control Register
TMR2_INTV_VALUE_REG	0x0034	Timer 2 Interval Value Register
TMR2_CUR_VALUE_REG	0x0038	Timer 2 Current Value Register
TMR3_CTRL_REG	0x0040	Timer 3 Control Register
TMR3_INTV_VALUE_REG	0x0044	Timer 3 Interval Value Register
TMR3_CUR_VALUE_REG	0x0048	Timer 3 Current Value Register
TMR4_CTRL_REG	0x0050	Timer 4 Control Register
TMR4_INTV_VALUE_REG	0x0054	Timer 4 Interval Value Register
TMR4_CUR_VALUE_REG	0x0058	Timer 4 Current Value Register
WAKEUP_TIMER		

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
TMR2_CTRL_REG	0x0030	Timer 2 Control Register
TMR2_INTV_VALUE_REG	0x0034	Timer 2 Interval Value Register
TMR2_CUR_VALUE_REG	0x0038	Timer 2 Current Value Register

3.11.6 Register Description

3.11.6.1 0x0000 Timer IRQ Enable Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: TMR_IRQ_EN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4	R/W	0x0	TMR4_IRQ_EN Timer 4 Interrupt Enable 0: Disable 1: Enable
3	R/W	0x0	TMR3_IRQ_EN Timer 3 Interrupt Enable 0: Disable 1: Enable
2	R/W	0x0	TMR2_IRQ_EN Timer 2 Interrupt Enable 0: Disable 1: Enable
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

3.11.6.2 0x0004 Timer IRQ Status Register (Default Value: 0x0000_0000)

Offset: 0x0004	Register Name: TMR_IRQ_STA_REG

Bit	Read/Write	Default/Hex	Description
31:5	/	/	/
4	R/W1C	0x0	TMR4_IRQ_PEND Timer 4 IRQ Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending. Timer 0 interval value is reached.
3	R/W1C	0x0	TMR3_IRQ_PEND Timer 3 IRQ Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending. Timer 0 interval value is reached.
2	R/W1C	0x0	TMR2_IRQ_PEND Timer 2 IRQ Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending. Timer 0 interval value is reached.
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.

3.11.6.3 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 is reached, the timer will not be disable automatically. 1: Single mode. When interval value is reached, the timer will be disabled 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

Offset: 0x0010			Register Name: TMR0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
3:2	R/W	0x1	TMR0_CLK_SRC 00: LFCLK 01: HFCLK 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 cannot be written again before it's cleared automatically.
0	R/W	0x0	TMR0_EN Timer 0 Enable 0: Stop/Pause 1: Start If the timer is started, it will reload the interval value to internal register, and the current counter will count from interval value to 0. If the current counter does not reach the zero, and the timer enable bit is set to "0", the current value counter will pause. Wait for 2 cycles at least, and then the start bit can be set to 1. In the timer pause state, the interval value register can be modified. If the timer is started again, and the Software hope that the current value register can down-count from the new interval value. The reload bit and the enable bit should be set to 1 at the same time.

3.11.6.4 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.*

3.11.6.5 0x0018 Timer 0 Current Value Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: TMR0_CUR_VALUE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**

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 **Note**: *Timer 0 current value is a 32-bit down-counter (from interval value to 0).*

3.11.6.6 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 is reached, the timer will be not disabled automatically. 1: Single mode. When interval value is reached, the timer will be disabled 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: LFCLK 01: HFCLK 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 cannot be written again before it’s cleared automatically.

Offset: 0x0020			Register Name: TMR1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	TMR1_EN Timer 1 Enable 0: Stop/Pause 1: Start If the timer is started, it will reload the interval value to internal register, and the current counter will count from interval value to 0. If the current counter does not reach the zero, and the timer enable bit is set to "0", the current value counter will pause. Wait for 2 cycles at least, and the start bit can be set to 1. In the timer pause state, the interval value register can be modified. If the timer is started again, and the Software hope that the current value register can down-count from the new interval value. The reload bit and the enable bit should be set to 1 at the same time.

3.11.6.7 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 setting should consider the system clock and the timer clock source.*

3.11.6.8 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 **Note**: *Timer 1 current value is a 32-bit down-counter (from interval value to 0).*

3.11.6.9 0x0030 Timer 2 Control Register (Default Value: 0x0000_0004)

Offset: 0x0030			Register Name: TMR2_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/

Offset: 0x0030			Register Name: TMR2_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
7	R/W	0x0	TMR2_MODE Timer 2 mode 0: Continuous mode. When interval value is reached, the timer will not be disabled automatically. 1: Single mode. When interval value is reached, the timer will be disabled automatically.
6:4	R/W	0x0	TMR2_CLK_PRES Select the pre-scale of Timer 2 clock source 000: /1 001: /2 010: /4 011: /8 100: /16 101: /32 110: /64 111: /128
3:2	R/W	0x1	TMR2_CLK_SRC 00: LFCLK 01: HFCLK 10: / 11: /
1	R/W	0x0	TMR2_RELOAD Timer 2 Reload 0: No effect 1: Reload Timer 2 Interval value After the bit is set, it cannot be written again before it's cleared automatically.
0	R/W	0x0	TMR2_EN Timer 2 Enable 0: Stop/Pause 1: Start If the timer is started, it will reload the interval value to internal register, and the current counter will count from interval value to 0. If the current counter does not reach the zero, the timer enable bit is set to "0", the current value counter will pause. Wait for 2 cycles at least, and the start bit can be set to 1. In timer pause state, the interval value register can be modified. If the timer is started again, and the Software hope that the current value register can down-count from the new interval value. The reload bit and the enable bit should be set to 1 at the same time.

3.11.6.10 0x0034 Timer 2 Interval Value Register (Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: TMR2_INTV_VALUE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	TMR2_INTV_VALUE Timer 2 Interval Value **Note**: The *value setting should consider the system clock and the timer clock source.*

3.11.6.11 0x0038 Timer 2 Current Value Register (Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: TMR2_CUR_VALUE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	TMR2_CUR_VALUE Timer 2 Current Value **Note**: *Timer 2 current value is a 32-bit down-counter (from interval value to 0).*

3.11.6.12 0x0040 Timer 3 Control Register (Default Value: 0x0000_0004)

Offset: 0x0040			Register Name: TMR3_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	TMR3_MODE Timer 3 mode 0: Continuous mode. When interval value is reached, the timer will not be disabled automatically. 1: Single mode. When interval value is reached, the timer will be disabled automatically.
6:4	R/W	0x0	TMR3_CLK_PRES Select the pre-scale of Timer 3 clock source 000: /1 001: /2 010: /4 011: /8 100: /16 101: /32 110: /64 111: /128
3:2	R/W	0x1	TMR3_CLK_SRC 00: LFCLK 01: HFCLK 10: / 11: /

Offset: 0x0040			Register Name: TMR3_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x0	TMR3_RELOAD Timer 3 Reload 0: No effect 1: Reload Timer 3 Interval value After the bit is set, it cannot be written again before it's cleared automatically.
0	R/W	0x0	TMR3_EN Timer 3 Enable 0: Stop/Pause 1: Start If the timer is started, it will reload the interval value to internal register, and the current counter will count from interval value to 0. If the current counter does not reach the zero, the timer enable bit is set to "0", the current value counter will pause. Wait for 2 cycles at least, and the start bit can be set to 1. In timer pause state, the interval value register can be modified. If the timer is started again, and the Software hope that the current value register can down-count from the new interval value. The reload bit and the enable bit should be set to 1 at the same time.

3.11.6.13 0x0044 Timer 3 Interval Value Register (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: TMR3_INTV_VALUE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	TMR3_INTV_VALUE Timer 3 Interval Value **Note**: The *value setting should consider the system clock and the timer clock source.*

3.11.6.14 0x0048 Timer 3 Current Value Register (Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: TMR3_CUR_VALUE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	TMR3_CUR_VALUE Timer 3 Current Value **Note**: *Timer 3 current value is a 32-bit down-counter (from interval value to 0).*

3.11.6.15 0x0050 Timer 4 Control Register (Default Value: 0x0000_0004)

Offset: 0x0050	Register Name: TMR4_CTRL_REG

Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	TMR4_MODE Timer 4 mode 0: Continuous mode. When interval value is reached, the timer will not be disabled automatically. 1: Single mode. When interval value is reached, the timer will not be disabled automatically.
6:4	R/W	0x0	TMR4_CLK_PRES Select the pre-scale of Timer 4 clock source 000: /1 001: /2 010: /4 011: /8 100: /16 101: /32 110: /64 111: /128
3:2	R/W	0x1	TMR4_CLK_SRC 00: LFCLK 01: HFCLK 10: / 11: /
1	R/W	0x0	TMR4_RELOAD Timer 4 Reload 0: No effect 1: Reload Timer 4 Interval value After the bit is set, it cannot be written again before it's cleared automatically.
0	R/W	0x0	TMR4_EN Timer 4 Enable 0: Stop/Pause 1: Start If the timer is started, it will reload the interval value to internal register, and the current counter will count from interval value to 0. If the current counter does not reach the zero, the timer enable bit is set to "0", the current value counter will pause. Wait for 2 cycles at least, and the start bit can be set to 1. In timer pause state, the interval value register can be modified. If the timer is started again, and the Software hope that the current value register can down-count from the new interval value. The reload bit and the enable bit should be set to 1 at the same time.

3.11.6.16 0x0054 Timer 4 Interval Value Register (Default Value: 0x0000_0000)

Offset: 0x0054			Register Name: TMR4_INTV_VALUE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	TMR4_INTV_VALUE Timer 4 Interval Value **Note**: The *value setting should consider the system clock and the timer clock source.*

3.11.6.17 0x0058 Timer 4 Current Value Register (Default Value: 0x0000_0000)

Offset: 0x0058			Register Name: TMR4_CUR_VALUE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	TMR4_CUR_VALUE Timer 4 Current Value **Note**: *Timer 4 current value is a 32-bit down-counter (from interval value to 0).*

3.12 Watchdog

3.12.1 Overview

Watchdog is used to monitor whether the system is abnormal. If yes, a system reset signal will be triggered.

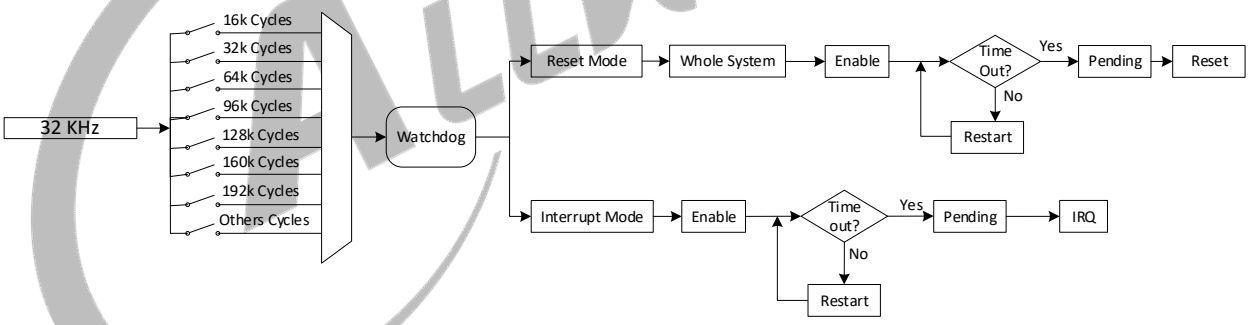
The watchdog has the following features:

- Up to 4 watchdogs, and one of them is secure world watchdog
- HOSC_32K clock sources and 32K system
- Supports 12 configurable initial count value
- Supports generating timeout interrupt
- Supports outputting reset signal
- Supports restarting timer

3.12.2 Block Diagram

The following figure shows the block diagram for the watchdog.

Figure 3-11 Block Diagram for the Watchdog



3.12.3 Operating Modes

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

In the interrupt mode, when the counter value reaches 0 and WDOG_IRQ_EN_REG is enabled, the watchdog generates an interrupt.

In the reset mode, when the counter value reaches 0, the watchdog generates a reset signal to reset the entire system.

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

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

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

3.12.3.1 Initializing the Watchdog

Follow the steps below to initialize the watchdog:

Step 1 Write the bit[1: 0] of WDOG_CFG_REG to configure the watchdog operating mode so that the watchdog can generate interrupts or output reset signals.

Step 2 Write the bit[7:4] of WDOG_MODE_REG to configure the initial count value.

Step 3 Write the bit[0] of WDOG_MODE_REG to enable the watchdog.

3.12.3.2 Processing the Interrupt

In the interrupt mode, the watchdog is used as a counter. It will generate an interrupt every time when the count value reaches 0.

Follow the steps below to process the interrupt:

Step 1 Write the enable bit of WDOG_IRQ_EN_REG to enable the interrupt.

Step 2 After entering the interrupt process, write the pending bit of WDOG_IRQ_STA_REG to clear the interrupt pending and execute the process of waiting for the interrupt.

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

3.12.4 Programming Guidelines

3.12.4.1 Watchdog Reset

In the following instance, make configurations for RISCV Watchdog: configure clock source as 24M/750, configure Interval Value as 1s and configure RISCV 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 RISCV Watchdog
```

3.12.4.2 Watchdog Restart

In the following instance, make configurations for RISCV Watchdog: configure clock source as 24M/750, configure Interval Value as 1s and configure RISCV Watchdog Configuration as To whole system. In the following instance, if the time of other codes is larger than 1s, RISCV Watchdog will reset the whole system. If the sentence of restart RISCV Watchdog is implemented inside 1s, RISCV 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 RISCV Watchdog
writel(readl(WDOG_CTRL)|(0xA57<<1)|(1<<0)，WDOG_CTRL);      //Writel 0xA57 at Key Fieland Restart RISCV Watchdog
```

3.12.5 Register List

Module Name	Base Address
RISCV_WDG	0x40029000
CPU_WDG	0x40020400
DSP_WDG	0x40023800

Register Name	Offset	Description
R/WDOG_IRQ_EN_REG	0x0000	Watchdog IRQ Enable Register
R/WDOG_IRQ_STA_REG	0x0004	Watchdog IRQ Status Register
R/WDOG_CTRL_REG	0x0010	Watchdog Control Register
R/WDOG_CFG_REG	0x0014	Watchdog Configuration Register
R/WDOG_MODE_REG	0x0018	Watchdog Mode Register
R/WDOG_OUTPUT_CFG_REG	0x001C	Watchdog Output Configuration Register

3.12.6 Register Description

3.12.6.1 0x0000 Watchdog IRQ Enable Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: R/WDOG_IRQ_EN_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	R/WDOG_IRQ_EN Watchdog Interrupt Enable. 0: Watchdog interrupt mask. 1: CPUSYS_Watchdog interrupt enable.

3.12.6.2 0x0004 Watchdog IRQ Status Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: WDOG_IRQ_STA_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	R/WDOG_IRQ _PEND Watchdog IRQ Pending. Set 1 to the bit will clear it. 0: No effect. 1: Pending. CPUSYS_Watchdog interval value is reached.

3.12.6.3 0x0010 Watchdog Control Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: R/WDOG_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
12:1	W	0x0	R/WDOG_KEY_FIELD Watchdog Key Field 0xA57 should be written in this field. Writing any other value will abort the write operation of bit[0].
0	R/W	0x0	R/WDOG_RESTART Watchdog Restart 0: No effect 1: Restart the CPUSYS_Watchdog

3.12.6.4 0x0014 Watchdog Configuration Register (Default Value: 0x0000_0001)

Offset: 0x0014			Register Name: R/WDOG_CFG_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 bit 0 can be written with the new value.
15:7	/	/	/
8	R/W	0x0	R/WDOG_CLK_SRC Select Watchdog clock sources 0: HOSC_32K 1: LFCLK
7:2	/	/	/
1: 0	R/W	0x1	R/WDOG_CONFIG Watchdog generates a reset signal 00: / 01: to whole system 10: only interrupt 11: /

3.12.6.5 0x0018 Watchdog Mode Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: R/WDOG_MODE_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 bit 0 can be written with the new value.
15:8	/	/	/

Offset: 0x0018			Register Name: R/WDOG_MODE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
7:4	R/W	0x0	R/WDOG_INTV_VALUE Watchdog Interval Value. When the clock source is HOSC_32K, the time of 16000 cycles is equal to 0.5 sec. When the clock source is 32K, the time of 16000 cycles is based on the number of 32K. 0000: 16000 cycles 0001: 32000 cycles 0010: 64000 cycles 0011: 96000 cycles 0100: 128000 cycles 0101: 160000 cycles 0110: 192000 cycles 0111: 256000 cycles 1000: 320000 cycles 1001: 384000 cycles 1010: 448000 cycles 1011: 512000 cycles 1100: / 1101: / 1110: / 1111: /
3:1	/	/	/
0	R/W	0x0	R/WDOG_EN Watchdog Enable 0: No effect 1: Enable the CPUSYS_Watchdog

3.12.6.6 0x001C Watchdog Output Config Register (Default Value: 0x0000_001F)

Offset: 0x001C			Register Name: R/WDOG_OUTPUT_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11: 0	R/W	0x1F	Watchdog output config Watchdog rst valid time config T=1/32ms*(N+1) The default value is 1 ms.

3.13 RTC

3.13.1 Overview

The real time clock (RTC) is for calendar usage. It is built around a 32-bit counter and used to count elapsed time in YY-MM-DD and HH-MM-SS. The RTC can continue to work while the system is in hibernation and standby. It has a built-in leap year generator.

The alarm generates an alarm signal at a specified time in the hibernation mode or standby mode or active mode. In active mode, both the alarm interrupt and the power management wakeup are activated. In hibernation mode or standby mode, the power management wakeup signal is activated. In this section, there are two kinds of alarm. Alarm 0 is a general alarm; its counter is based on second. Alarm 1 is a weekly alarm; its counter is based on the real time.

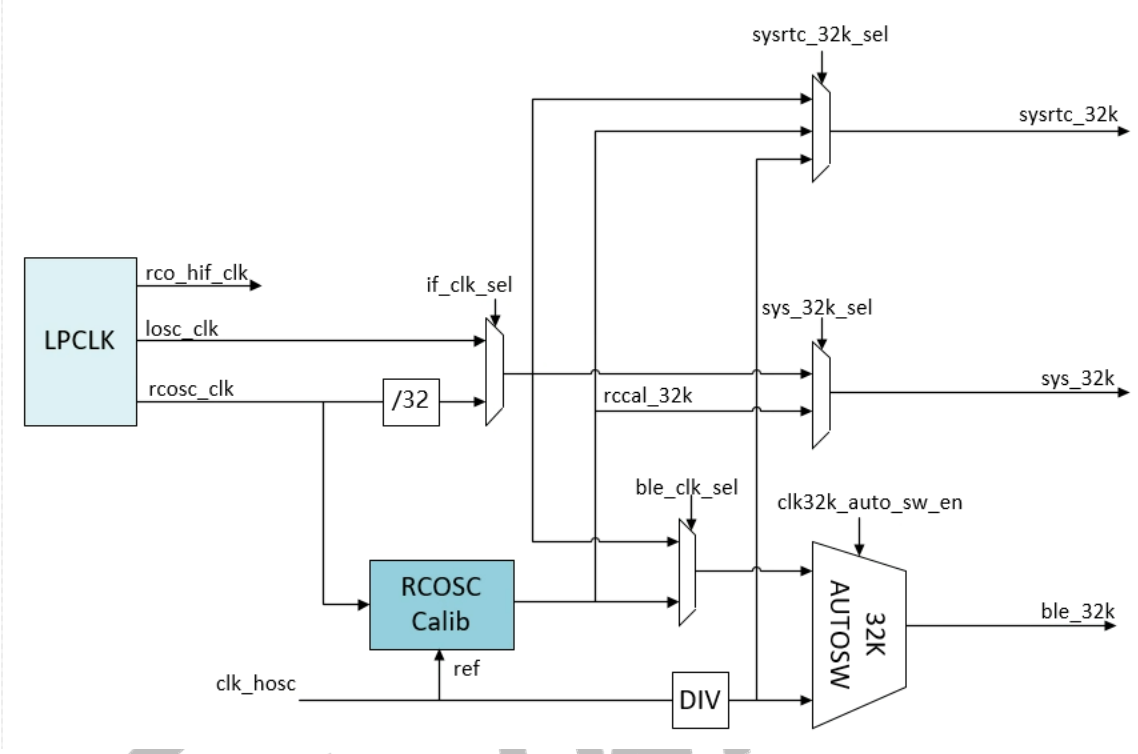
The RTC has the following features:

- Implements time counter and timing wakeup
- Provides counters for counting year, month, day, hour, minute, and second
- 4-channel clock sources: HOSC_32K, RCCAL_32K, RCOSC, LOSC_CLK
- Configures initial value by software anytime
- Supports timing alarm, and generates interrupt and wake up the PMU system

3.13.2 Block Diagram

The following figure shows the block diagram of the RTC.

Figure 3-12 RTC Block Diagram



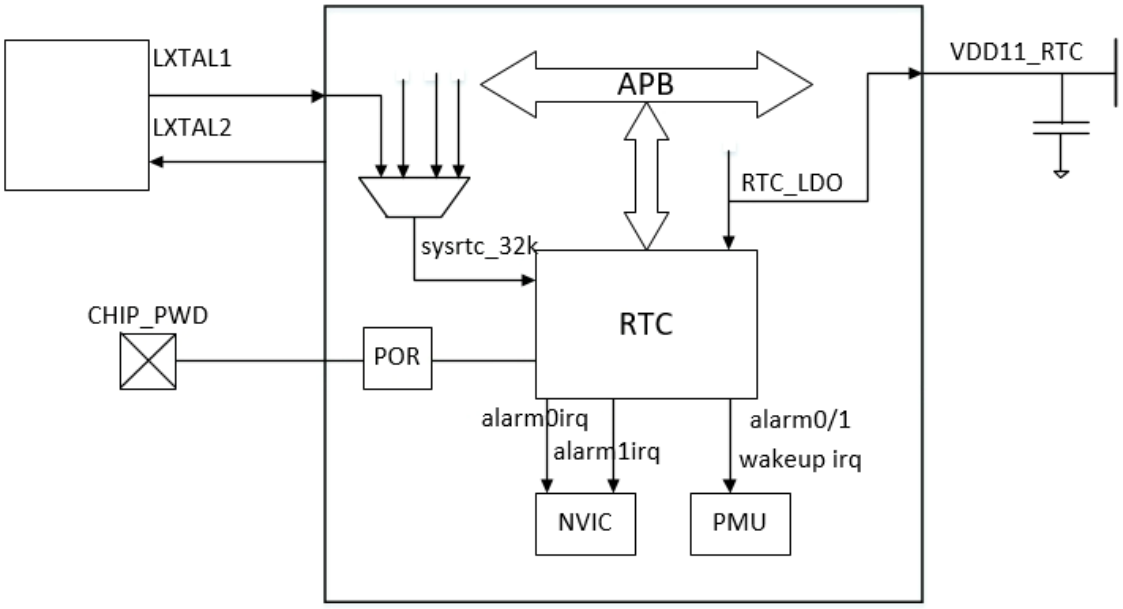
3.13.3 Functional Description

3.13.3.1 External Signals

Signal	Description	Type
LXTAL_IN	32.768 kHz oscillator input	I
LXTAL_OUT	32.768 kHz oscillator output	O
VDD_RTC	RTC domain power supply	P

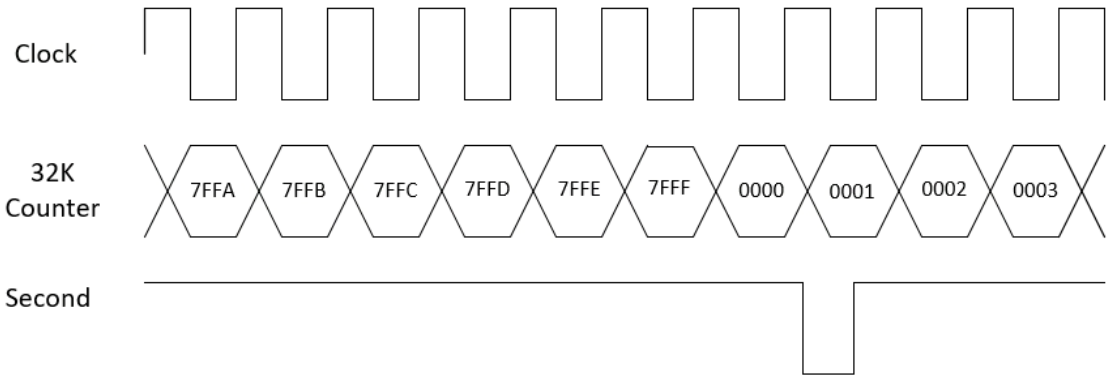
3.13.3.2 Typical Application

The following figure shows the typical application of RTC.



3.13.3.3 Real Time Clock

Figure 3-13 RTC Counter



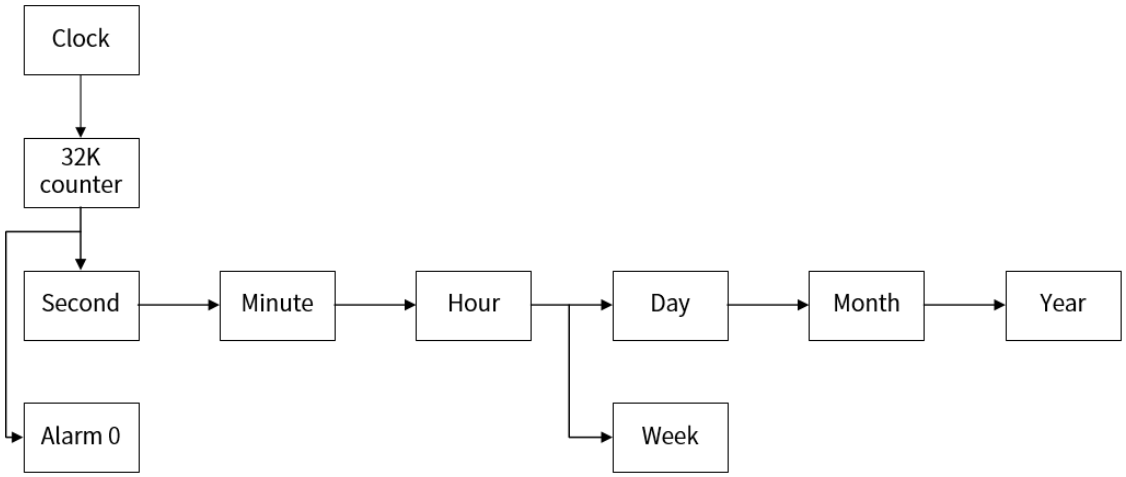
LOSC_CLK is selected as clock source in the above figure. The 32K counter adds 1 on each rising edge of the clock. When the clock number reaches 0x8000 (32768), the 32K counter will start to count again from 0, and the second counter will add 1.

If rcosc_32div is selected as clock source, the 32K counter will add 1 on each rising edge of the clock. RTC validates the rcosc_32div through high-frequency oscillator and auto calculates the counter number within 1 second in real time. When the number of sampled clock is same to that of counter, 32K counter will start to count again from 0, and the second counter will add 1.

If hosc_32k or rccal_32k is selected as clock source, the 32K counter will add 1 on each rising edge of the clock. When the clock number reaches 0x7D00 (32000), the 32K counter will start to count again from 0, and the second counter adds 1.

The step structure of 32K counter is as follows.

Figure 3-14 RTC 32K Counter Step Structure



The varying range of each counter is as follows.

Table 3-14 RTC Counter Changing Range

Counter	Range
Second	00-59 (It auto changes to 59 if not in this range)
Minute	00-59 (It auto changes to 59 if not in this range)
Hour	00-23 (It auto changes to 23 if not in this range)
Week	00-6 (It auto changes to 6 if not in this range)
Day	01-31 (It auto changes to 31 if not in this range)
Month	01-12 (It auto changes to 12 if not in this range)
Year	00-127 (A base year can be set by software; a leap year only can be set by software.)

3.13.3.4 Alarm 0

The 0x24 register is a 32bit number which adds 1 each second. When Alarm0irq is enabled, read the value of the 0x24 register, add n seconds, and write to the 0x20 register. N seconds later, the value of 0x24 register is the same to that of 0x20 register. Then, the Alarm0irq will be triggered.

3.13.3.5 Alarm 1

The weekly alarm function of Alarm 1 is to set the week number of triggering alarm. When the real time of system is the same to the set time, the Alarm1irq will be triggered.

3.13.4 Programming Guidelines

3.13.4.1 RTC Calendar

Step 1 Write RTC_HHMMSS and RTC_YYMMDD based on the current time.

Step 2 As the time elapses, RTC will start to count again from 0. The software can read the current time anytime.

Step 3 The leap year only can be set by the software.

3.13.4.2 Alarm 0

Step 1 The 32bit adding counter adds 1 each second. The value of counter is stored in ALM0_CURVAL (0x24).

Step 2 Enable the alarm 0 interrupt and write ALM0_EN.

Step 3 The software reads the ALM0_CURVAL. Add n seconds to this value and then write them to the ALM0_CNT register. N seconds later, the value of the ALM0_CURVAL is the same to that of the ALM0_CNT. Then, Alarm0irq will be triggered.

Step 4 After entering the interrupt process, write ALM0_IRQST to clear the interrupt pending, and execute the interrupt process.

Step 5 Resume the interrupt and continue to execute the interrupted process.

3.13.4.3 Alarm 1

Step 1 Set the initial value of Alarm 1, and write the hour/minute/second of Alarm 1 to Alarm 1_WK_HHMMSS.

Step 2 Write ALM1_EN based on the week number of triggering alarm, and enable alarm 0.

Step 3 When the value of RTC_HHMMSS [20: 0] is the same to Alarm 1_WK_HHMMSS, and the week number set in ALM1_EN is the same to RTC_HHMMSS [31:29], alarm 1 will generates pending. The ALM1_IRQST will be auto set to 1.

Step 4 When the ALM1_IRQEN is set to 1, enter the interrupt process. Write ALM1_IRQST to clear the interrupt pending, and execute the interrupt process.

Step 5 Resume the interrupt and continue to execute the interrupted process.

3.13.4.4 Alarm 0/1 Wakeup

Step 1 Enable ALM_WAKEUP_EN.

Step 2 Configure Alarm 0/Alarm 1.

Step 3 System enters the hibernation mode or standby mode.

Step 4 rtc_alarm_wakeup triggers PMU wakeup system.

3.13.5 Register List

Module Name	Base Address
RTC_TIMER	0x40051000

Register Name	Offset Address	Description
RTC_CTRL	0x0000	RTC Control Register
RTC_YYMMDD	0x0010	RTC Year Month Day Register
RTC_HHMMSS	0x0014	RTC Hour Minute Second Register
ALM0_CNT	0x0020	Alarm 0 Counter Register
ALM0_CURVAL	0x0024	Alarm 0 Current Value Register
ALM0_EN	0x0028	Alarm 0 Enable Register
ALM0_IRQEN	0x002C	Alarm 0 IRQ Enable Register
ALM0_IRQST	0x0030	Alarm 0 IRQ Status Register
ALM1_WK_HHMMSS	0x0040	Alarm 1 Week Hour-Minute-Second Register
ALM1_EN	0x0044	Alarm 1 Enable Register
ALM1_IRQEN	0x0048	Alarm 1 IRQ Enable Register
ALM1_IRQST	0x004C	Alarm 1 IRQ Status Register
ALM_WAKEUP_EN	0x0050	Alarm wakeup Enable Register
FRUN_CNT_L	0x0060	Free Running Counter bit[31: 0]
FRUN_CNT_H	0x0064	Free Running Counter bit[47:32]

3.13.6 Register Description

3.13.6.1 0x0000 RTC Control Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: RTC_ CTRL
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	RTC_TEST_MODE_CTRL RTC TEST Mode Control bit
30	R/W	0x0	RTC_DEBUG RTC Simulation Control bit 0: No effect 1: Simulation mode
29:2	/	/	/
1	R	0x0	RTC_HHMMSS_ACCE RTC HH-MM-SS access After writing the RTC HH-MM-SS register, this bit is set. It will be cleared until the real writing operation is finished. After the RTC_HHMMSS_ACCE is cleared, the writing operation of HH-MM-SS register should be delayed up to sysrtc_32k cycle once.

Offset: 0x0000			Register Name: RTC_ CTRL
Bit	Read/Write	Default/Hex	Description
0	R	0x0	RTC_YYMMDD_ACCE. RTC YY-MM-DD access. After writing the RTC YY-MM-DD register, this bit is set. It will be cleared until the real writing operation is finished. After the RTC_YYMMDD_ACCE is cleared, the writing operation of YY-MM-DD register should be delayed up to sysrtc_32k cycle once.

3.13.6.2 0x0010 RTC YYMMDD Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: RTC_YYMMDD
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	LEAP 0: Not 1: Leap year This bit cannot be set by hardware. It should be set or cleared by software.
23:16	R/W	x	YEAR Range from 0~256
15:12	/	/	/
11:8	R/W	x	MONTH Month Range from 1~12
7:5	/	/	/
4: 0	R/W	x	DAY Range from 1~31. **Note:** *If the written value is not from 1 to 31, it will turn into 31 automatically. The fileds of month and year are similar.*

3.13.6.3 0x0014 RTC HHMMSS Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: RTC_HHMMSS
Bit	Read/Write	Default/Hex	Description

Offset: 0x0014			Register Name: RTC_HHMMSS
Bit	Read/Write	Default/Hex	Description
31:29	R/W	0x0	WEEK_NO Week number 000: Monday 001: Tuesday 010: Wednesday 011: Thursday 100: Friday 101: Saturday 110: Sunday 111: /
28:21	/	/	/
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

3.13.6.4 0x0020 Alarm 0 Counter Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: ALM0_CNT
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	ALM0_CNT Alarm 0 Counter is based on second.

3.13.6.5 0x0024 Alarm 0 Current Value Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: ALM0_CURVAL
Bit	Read/Write	Default/Hex	Description
31: 0	R	x	ALM0_CURVAL

3.13.6.6 0x0028 Alarm 0 Enable Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: ALM0_EN
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/

Offset: 0x0028			Register Name: ALM0_EN
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	ALM0_EN If this bit is set to 1, the alarm 0 counter register's valid bits will down count to 0, and the alarm pending bit will be set to 1. 0: Disable 1: Enable

3.13.6.7 0x002C Alarm 0 IRQ Enable Register (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: ALM0_IRQEN
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	ALM0_IRQEN 0: Disable 1: Enable

3.13.6.8 0x0030 Alarm 0 IRQ Status Register (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: ALM0_IRQST
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	ALM0_IRQST 0: No effect 1: pending

3.13.6.9 0x0040 Alarm 1 Week HH-MM-SS Register (Default Value: 0x0000_0000)

Offset: 0x0040			Register Name: Alarm 1_WK_HHMMSS
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

3.13.6.10 0x0044 Alarm 1 Enable Register (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: ALM1_EN
Bit	Read/Write	Default/Hex	Description

Offset: 0x0044			Register Name: ALM1_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6	R/W	0x0	SUN_ALM1_EN
5	R/W	0x0	SAT_ALM1_EN Saturday Alarm 1 Enable 0: Disable 1: Enable If this bit is set to “1”, only when the Alarm 1 Week HH-MM-SS register valid bits is equal to RTC HH-MM-SS register, and the register RTC HH-MM-SS bit [31:29] is 5, the alarm IRQ pending bit will be set to “1”.
4	R/W	0x0	FRI_ALM1_EN Friday Alarm 1 Enable 0: Disable 1: Enable If this bit is set to “1”, only when the Alarm 1 Week HH-MM-SS register valid bits is equal to RTC HH-MM-SS register, and the register RTC HH-MM-SS bit [31:29] is 4, the alarm IRQ pending bit will be set to “1”.
3	R/W	0x0	THU_ALM1_EN Thursday Alarm 1 Enable 0: Disable 1: Enable If this bit is set to “1”, only when the Alarm 1 Week HH-MM-SS register valid bits is equal to RTC HH-MM-SS register and the register RTC HH-MM-SS bit [31:29] is 3, the alarm IRQ pending bit will be set to “1”.
2	R/W	0x0	WED_ALM1_EN Wednesday Alarm 1 Enable 0: Disable 1: Enable If this bit is set to “1”, only when the Alarm 1 Week HH-MM-SS register valid bits is equal to RTC HH-MM-SS register, and the register RTC HH-MM-SS bit [31:29] is 2, the alarm IRQ pending bit will be set to “1”.

Offset: 0x0044			Register Name: ALM1_EN
Bit	Read/Write	Default/Hex	Description
1	R/W	0x0	TUE_ALM1_EN Tuesday Alarm 1 Enable 0: Disable 1: Enable If this bit is set to "1", only when the Alarm 1 Week HH-MM-SS register valid bits is equal to RTC HH-MM-SS register and the register RTC HH-MM-SS bit [31:29] is 1, the alarm IRQ pending bit will be set to "1".
0	R/W	0x0	MON_ALM1_EN Monday Alarm 1 Enable 0: Disable 1: Enable If this bit is set to "1", only when the Alarm 1 Week HH-MM-SS register valid bits is equal to RTC HH-MM-SS register, and the register RTC HH-MM-SS bit [31:29] is 0, the alarm IRQ pending bit will be set to "1".

3.13.6.11 0x0048 Alarm 1 IRQ Enable Register (Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: ALM1_IRQEN
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	ALM1_IRQEN 0: Disable 1: Enable

3.13.6.12 0x004C Alarm 1 IRQ Status Register (Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: ALM1_IRQST
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	ALM1_WEEK_IRQST Alarm 1 week (0-7) IRQ Pending 0: No effect 1: pending

3.13.6.13 0x0050 Alarm Wakeup Enable Register (Default Value: 0x0000_0000)

Offset: 0x0050			Register Name: ALM_WAKEUP_EN
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/

Offset: 0x0050			Register Name: ALM_WAKEUP_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x0	ALM1_WAKEUP 0: Disable alarm1 wakeup output 1: Enable alarm1 wakeup output
0	R/W	0x0	ALM0_WAKEUP 0: Disable alarm0 wakeup output 1: Enable alarm0 wakeup output

3.13.6.14 0x0060 Free Running Counter Low Register (Default Value: 0x0000_0000)

Offset: 0x0060			Register Name: FREERUN_CNT_L
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	Free Running Counter bit [31: 0] **Note:** *Free running counter is a 48-bit counter which is driven by LFCLK and starts to count as soon as the system reset is released and the LFCLK is ready. Writing will reset the counter value.*

3.13.6.15 0x0064 Free Running Counter High Register (Default Value: 0x0000_0000)

Offset: 0x0064			Register Name: FREERUN_CNT_H
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	Free Running Counter bit [47:32] **Note:** *Free running counter is a 48-bit counter which is driven by LFCLK and starts to count as soon as the system reset is released and the LFCLK is ready. Writing will reset the counter value.*

3.14 Message Box

3.14.1 Overview

The Message Box (MSGBOX) provides interrupt communication mechanism for on-chip processor.

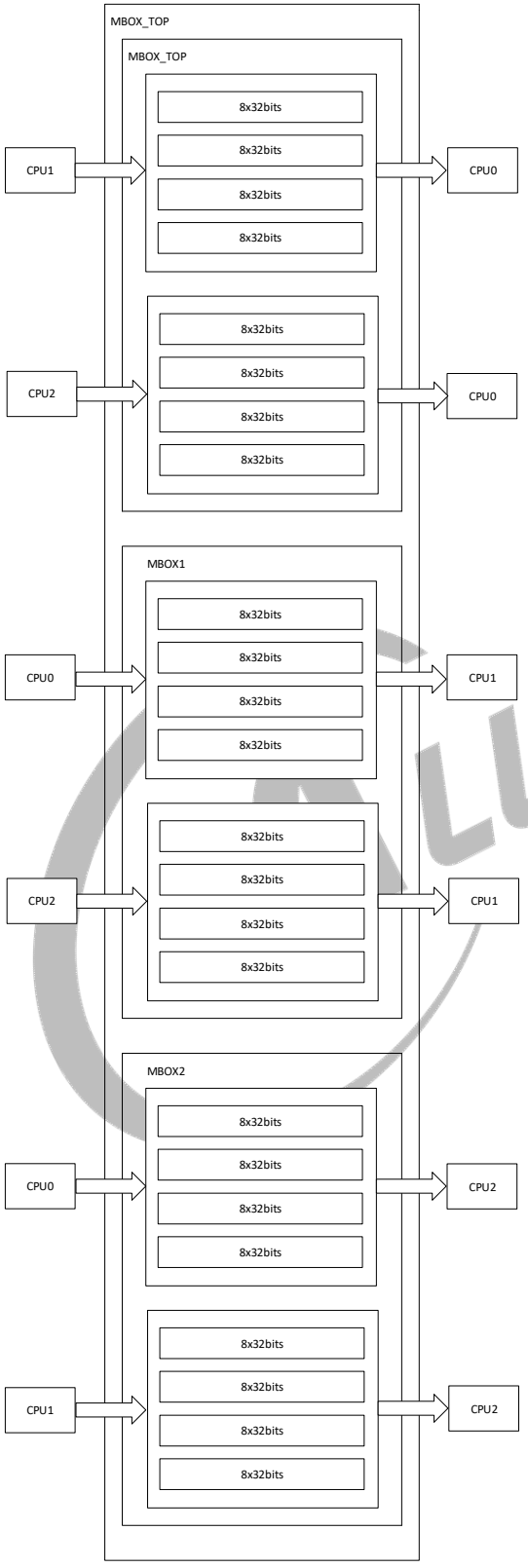
The MSGBOX has the following features:

- Supports 3 CPUs to transmit information through channels. Each CPU has a MSGBOX.
  - CPU0: ARM CPU
  - CPU1: RISC-V
  - CPU2: DSP
- There are four channels every two CPU, and the FIFO depth of one channel is 8 x 32 bits.
- Supports interrupts

3.14.2 Block Diagram

The following figure shows the block diagram of the message box.

Figure 3-15 Block Diagram of Message Box



3.14.3 Functional Description

3.14.3.1 Clock and Reset

The MSGBOX is mounted on AHB. Before accessing the MSGBOX registers, you need to de-assert the MSGBOX reset signal on AHB bus and then open the MSGBOX gating signal on AHB bus.

Typical Application

Several masters can build communication by configuring the MSGBOX. The communication parties have 4 channels. In a channel, the user1 is fixed as the transmitter and the user0 is fixed as the receiver. During the communication process, the current status can be judged through the interrupt or FIFO status.

3.14.3.2 Transmitter/Receiver Mode

At the same channel, user1 is fixed as transmitter, user0 is fixed as receiver.

3.14.3.3 Interrupt

Each channel can independently configure the interrupt enable bit, a read interrupt will be generated when the channel is non-empty, a write interrupt will be generated when the channel is non-full. For each CPU, all channels generate a read interrupt together, that is, if only a channel is non-empty, the read interrupt will be generated. This channel can be obtained by querying the interrupt status register.

3.14.3.4 FIFO Status

When channel FIFO is non-full, the FIFO_FULL_FLAG is 0. Now the FIFO can be written.

When channel FIFO is full, the FIFO_FULL_FLAG is 1. At this moment, if FIFO is written again, the first data of FIFO will be covered.

See MSGBOX_MSG_STATUS_REG for FIFO status.

3.14.4 Programming Guidelines

3.14.4.1 Checking the Transfer Status via the Interrupt

Follow the steps below to check the transfer status. Take an example of CPUX Message Box.

Step 1 Enable the interrupt for the channel: Configure the interrupt enable bits of transmitter/receiver through MSGBOX_WR_IRQ_EN_REG/MSGBOX_RD_IRQ_EN_REG. (user0: RX interrupt enable; user1: TX interrupt enable)

Step 2 Check the IRQ status of the corresponding queue through MSGBOX_WR_IRQ_STATUS_REG/MSGBOX_RD_IRQ_STATUS_REG.

-If the FIFO is not full, the channel generates a transmission interrupt to remind the transmitter to transmit data. Write data to the FIFO in the interrupt handler, then clear the pending bit of the transmitter in MSGBOX_WR_IRQ_STATUS_REG and the enable bit of the transmitter in MSGBOX_WR_IRQ_EN_REG.

-If the FIFO has new data, the channel generates a reception interrupt to remind the receiver to receive data. Read data from the FIFO in interrupt handler, then clear the pending bit of the receiver in MSGBOX_RD_IRQ_STATUS_REG and the enable bit of the receiver in MSGBOX_RD_IRQ_EN_REG.

3.14.4.2 Checking the Transfer Status via the FIFO

Follow the steps below to check the FIFO status of the corresponding queue:

- If the FIFO is not full, the transmitter will fill the FIFO to 8*32 bits.
- If the FIFO is full, the receiver will read the FIFO data and MSGBOX_FIFO_STATUS_REG. The receiver can acquire the current FIFO data amount and the prior-to-reading FIFO data amount to avoid data loss.

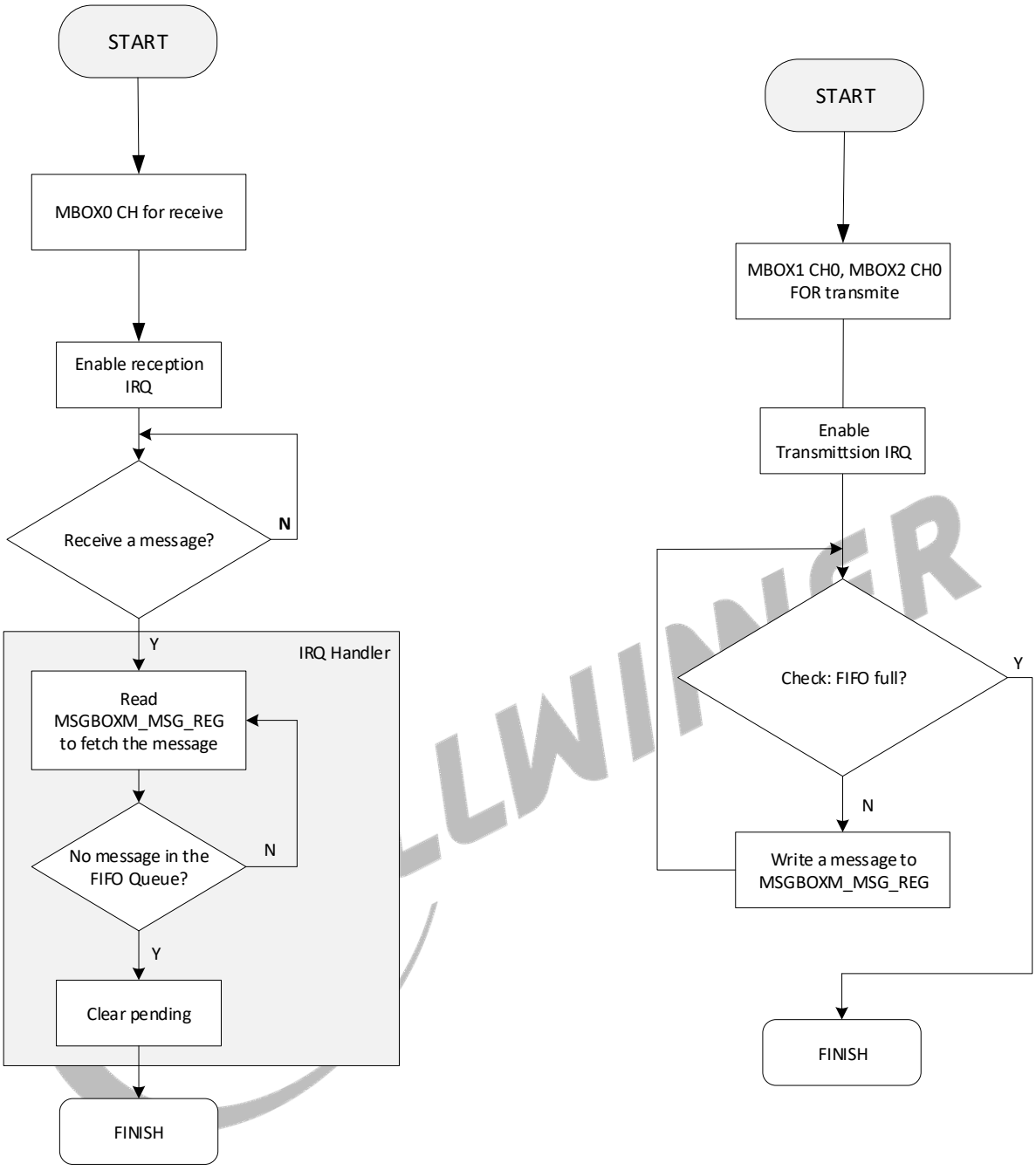
3.14.4.3 Transmitting/Receiving Message

The following figure shows the communication process between MSGBOX0, MSGBOX1 and MSGBOX2.

MSGBOX0: receiving message

MSGBOX1 & MSGBOX2: transmitting message

Figure 3-16 The Communication Process between MSGBOX0, MSGBOX1, and MSGBOX2



3.14.5 Register List

Module Name	Base Address
CPU_MBOX	0x40020800
RISC-V_MSGBOX	0x4002B000
DSP_MSGBOX	0x40023000

Parameter	Description
N	0-1

Parameter	Description
P	0-3

MSGBOX	CPU	The Value of N
MSGBOX (ARM CPUX)	DSP->CPUX	N=0
MSGBOX (ARM CPUX)	RISCV->CPUX	N=1
MSGBOX (DSP)	CPUX->DSP	N=0
MSGBOX (DSP)	RISC-V -> DSP	N=1
MSGBOX (RISC-V)	CPUX->RISCV	N=0
MSGBOX (RISC-V)	DSP -> RISC-V	N=1

Register Name	Offset	Description
MBOX (CPUX)		
MSGBOX_RD_IRQ_EN_REG	0x0020+M*0x400+N*0x100	IRQ enable for user0 read
MSGBOX_RD_IRQ_STATUS_REG	0x0024+M*0x400+N*0x100	IRQ status for user0 read
MSGBOX_WR_IRQ_EN_REG	0x0030+M*0x400+N*0x100	IRQ enable for user1 write
MSGBOX_WR_IRQ_STATUS_REG	0x0034+M*0x400+N*0x100	IRQ status for user1 write
MSGBOX_DEBUG_REG	0x0040+M*0x400+N*0x100	MSGBOX debug register
MSGBOX_FIFO_STATUS_REG	0x0050+N*0x0004+M*0x400+N*0x100+P*0x0004	Message status for message queue P(P=0~1)
MSGBOX_MSG_STATUS_REG	0x0060+M*0x400+N*0x100+P*0x0004	FIFO status for message queue P(P = 0~1)
MSGBOX_MSG_REG	0x0070+M*0x400+N*0x100+P*0x0004	Message Register for message queue P(P=0~1)
MBOX (RISC CPU)		
MSGBOX_RD_IRQ_EN_REG	0x0020+N*0x100 (N=0~1)	MSGBOX Read IRQ Enable Register
MSGBOX_RD_IRQ_STATUS_REG	0x0024+N*0x100 (N=0~1)	MSGBOX Read IRQ Status Register
MSGBOX_WR_IRQ_EN_REG	0x0030+N*0x100 (N=0~1)	MSGBOX Write IRQ Enable Register
MSGBOX_WR_IRQ_STATUS_REG	0x0034+N*0x100 (N=0~1)	MSGBOX Write IRQ Status Register
MSGBOX_DEBUG_REG	0x0040+N*0x100 (N=0~1)	MSGBOX debug register
MSGBOX_FIFO_STATUS_REG	0x0050+N*0x100+P*0x0004 (N=0~1)(P=0~3)	MSGBOX FIFO Status Register
MSGBOX_MSG_STATUS_REG	0x0060+N*0x100+P*0x0004 (N=0~1)(P=0~3)	MSGBOX Message Status Register
MSGBOX_MSG_REG	0x0070+N*0x100+P*0x0004 (N=0~1)(P=0~3)	MSGBOX Message Queue Register
MSGBOX_WR_INT_THRESHOLD	0x0080+N*0x100+P*0x0004 (N=0~1)(P=0~3)	MSGBOX Write IRQ Threshold Register
MBOX (DSP)		
MSGBOX_RD_IRQ_EN_REG	0x0020+M*0x400+N*0x100	IRQ enable for user0 read
MSGBOX_RD_IRQ_STATUS_REG	0x0024+M*0x400+N*0x100	IRQ status for user0 read
MSGBOX_WR_IRQ_EN_REG	0x0030+M*0x400+N*0x100	IRQ enable for user1 write

Register Name	Offset	Description
MSGBOX_WR_IRQ_STATUS_REG	0x0034+M*0x400+N*0x100	IRQ status for user1 write
MSGBOX_DEBUG_REG	0x0040+M*0x400+N*0x100	MSGBOX debug register
MSGBOX_FIFO_STATUS_REG	0x0050+N*0x0004+M*0x400+N*0x100+P*0x0004	Message Status for message queue P(P=0~1)
MSGBOX_MSG_STATUS_REG	0x0060+M*0x400+N*0x100+P*0x0004	FIFO status for message queue P(P = 0~1)
MSGBOX_MSG_REG	0x0070+M*0x400+N*0x100+P*0x0004	Message Register for message queue P(P=0~1)

3.14.6 Register Description

3.14.6.1 0x0020+N*0x100 MSGBox Read IRQ Enable Register (Default Value: 0x0000_0000)

Offset: 0x0020+N*0x100			Register Name: MSGBOX_RD_IRQ_EN_REG
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6	R/W	0x0	RECEPTION_MQ3_IRQ_EN 0: Disable 1: Enable When this bit is set to 1, it will notify user 0 by interrupt when Message Queue 3 has received a new message.
5	/	/	/
4	R/W	0x0	RECEPTION_MQ2_IRQ_EN 0: Disable 1: Enable When this bit is set to 1, it will notify user 0 by interrupt when Message Queue 2 has received a new message.
3	/	/	/
2	R/W	0x0	RECEPTION_MQ1_IRQ_EN 0: Disable 1: Enable When this bit is set to 1, it will notify user 0 by interrupt when Message Queue 1 has received a new message.
1	/	/	/
0	R/W	0x0	RECEPTION_MQ0_IRQ_EN 0: Disable 1: Enable When this bit is set to 1, it will notify user 0 by interrupt when Message Queue 0 has received a new message.

3.14.6.2 0x0024+N*0x100 MSGBox Read IRQ Status Register (Default Value: 0x0000_0000)

Offset: 0x0024+N*0x100			Register Name: MSGBOX_RD_IRQ_STATUS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
6	R/W	0x0	RECEPTION_MQ3_IRQ_PEND 0: No effect 1: Pending. This bit will be pending for user 0 when Message Queue 3 has received a new message. Setting 1 to this bit will clear it.
5	/	/	/
4	R/W	0x0	RECEPTION_MQ2_IRQ_PEND 0: No effect 1: Pending. This bit will be pending for user 0 when Message Queue 2 has received a new message. Setting 1 to this bit will clear it.
3	/	/	/
2	R/W	0x0	RECEPTION_MQ1_IRQ_PEND 0: No effect 1: Pending. This bit will be pending for user 0 when Message Queue 1 has received a new message. Setting 1 to this bit will clear it.
1	/	/	/
0	R/W	0x0	RECEPTION_MQ0_IRQ_PEND 0: No effect 1: Pending. This bit will be pending for user 0 when Message Queue 0 has received a new message. Setting 1 to this bit will clear it.

3.14.6.3 0x0030+N*0x100 MSGBox Write IRQ Enable Register (Default Value: 0x0000_0000)

Offset: 0x0030+N*0x100			Register Name: MSGBOX_WR_IRQ_EN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	TRANSMIT_MQ3_IRQ_EN 0: Disable 1: Enable When this bit is set to 1, it will notify user 1 by interrupt when Message Queue 3 empty level reaches the configured threshold.
6	/	/	/

Offset: 0x0030+N*0x100			Register Name: MSGBOX_WR_IRQ_EN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
5	R/W	0x0	TRANSMIT_MQ2_IRQ_EN 0: Disable 1: Enable When this bit is set to 1, it will notify user 1 by interrupt when Message Queue 2 empty level reach the configured threshold.
4	/	/	/
3	R/W	0x0	TRANSMIT_MQ1_IRQ_EN 0: Disable 1: Enable When this bit is set to 1, it will notify user 1 by interrupt when Message Queue 1 empty level reach the configured threshold.
2	/	/	/
1	R/W	0x0	TRANSMIT_MQ0_IRQ_EN 0: Disable 1: Enable When this bit is set to 1, it will notify user 1 by interrupt when Message Queue 0 empty level reach the configured threshold.
0	/	/	/

3.14.6.4 0x0034+N*0x100 MSGBox Write IRQ Status Register (Default Value: 0x0000_0000)

Offset: 0x0034+N*0x100			Register Name: MSGBOX_WR_IRQ_STATUS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	TRANSMIT_MQ3_IRQ_PEND 0: No effect 1: Pending. This bit will be pending for user 1 when Message Queue 3 empty level reaches the configured threshold. Setting 1 to this bit will clear it.
6	/	/	/
5	R/W	0x0	TRANSMIT_MQ2_IRQ_PEND 0: No effect 1: Pending. This bit will be pending for user 1 when Message Queue 2 empty level reaches the configured threshold. Setting 1 to this bit will clear it.
4	/	/	/
3	R/W	0x0	TRANSMIT_MQ1_IRQ_PEND 0: No effect 1: Pending. This bit will be pending for user 1 when Message Queue 1 empty level reaches the configured threshold. Setting 1 to this bit will clear it.
2	/	/	/

Offset: 0x0034+N*0x100			Register Name: MSGBOX_WR_IRQ_STATUS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x0	TRANSMIT_MQ0_IRQ_PEND 0: No effect 1: Pending. This bit will be pending for user 1 when Message Queue 0 empty level reaches the configured threshold. Setting 1 to this bit will clear it.
0	/	/	/

3.14.6.5 0x0040+N*0x100 MSGBox Debug Register (Default Value: 0x0000_0000)

Offset: 0x0040+N*0x100			Register Name: MSGBOX_DEBUG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:8	R/W	0x0	FIFO_CTRL MQ [7: 0] Control. In the debug mode, the corresponding FIFO channel will be disabled, and only one register space is valid for a message exchange. 0: Normal Mode 1: Disable the corresponding FIFO (Clear FIFO)
7:1	/	/	/

3.14.6.6 0x0050+N*0x100+P*0x0004 MSGBox FIFO Status Register (Default Value: 0x0000_0000)

Offset: 0x0050+N*0x100+P*0x0004			Register Name: MSGBOX_FIFO_STATUS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 1	/	/	/
0	R	0x0	FIFO_NOT_AVA_FLAG FIFO is not available flag 0: The Message FIFO queue empty level reaches the configured threshold. 1: The Message FIFO queue empty level doesn't reach the configured threshold. This FIFO status register has the status related to the message queue.

3.14.6.7 0x0060+N*0x100+P*0x0004 MSGBox Message Status Register (Default Value: 0x0000_0000)

Offset: 0x0060+N*0x100+P*0x0004			Register Name: MSGBOX_MSG_STATUS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/

Offset: 0x0060+N*0x100+P*0x0004			Register Name: MSGBOX_MSG_STATUS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
3: 0	R	0x0	MSG_NUM Number of unread messages in the message queue. Here, there are up to eight messages in the message FIFO queue. 0000: There is no message in the message FIFO queue. 0001: There is 1 message in the message FIFO queue. 0010: There are 2 messages in the message FIFO queue. 0011: There are 3 messages in the message FIFO queue. 0100: There are 4 messages in the message FIFO queue. 0101: There are 5 messages in the message FIFO queue. 0110: There are 6 messages in the message FIFO queue. 0111: There are 7 messages in the message FIFO queue. 1000: There are 8 messages in the message FIFO queue. 1001~1111: /

3.14.6.8 0x0070+N*0x100+P*0x0004 MSGBox Message Queue Register (Default Value: 0x0000_0000)

Offset: 0x0070+N*0x100+P*0x0004			Register Name: MSGBOX_MSG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	The message register stores the to-be-read message of the message FIFO queue. Reads remove the message from the FIFO queue.

3.14.6.9 0x0080+N*0x100+P*0x0004 MSGBox Write IRQ Threshold Register (Default Value: 0000_0000)

Offset: 0x0080+N*0x100+P*0x0004			Register Name: MSGBOX_WR_INT_THRESHOLD_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
1: 0	R/W	0x0	Configure the FIFO empty level to trigger the write interrupt for user1 00: 1 01: 2 10: 4 11: 8

3.15 Spinlock

3.15.1 Overview

The spinlock provides hardware synchronization mechanism in multi-core systems. With the lock operation, the spinlock prevents multiple processors from handling the sharing data simultaneously and thus ensures the coherence of data.

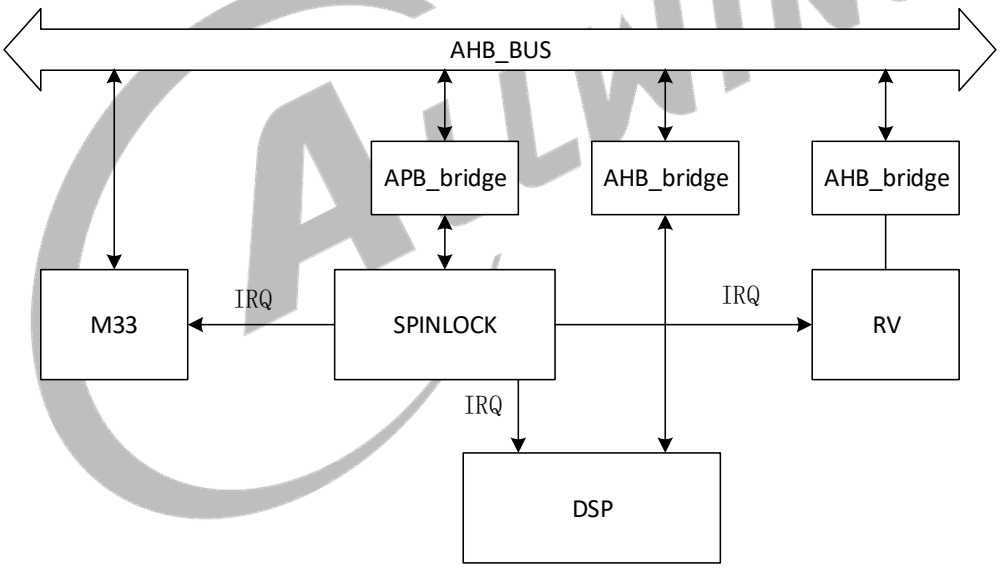
The spinlock has the following features:

- Supports 32 lock units
- Two kinds of lock status: locked and unlocked
- Lock time of the processor is predictable (less than 200 cycles)

3.15.2 Block Diagram

The following figure shows the block diagram of the spinlock.

Figure 3-17 Spinlock Block Diagram



3.15.3 Functional Description

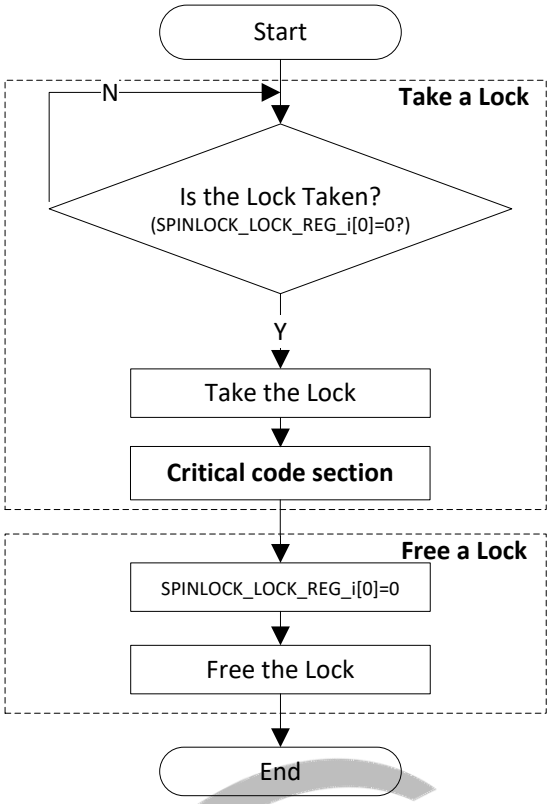
3.15.3.1 Clock and Reset

The spinlock is mounted on AHB. Before accessing the spinlock registers, you need to de-assert the reset signal on AHB bus and then open the corresponding gating signal on AHB bus.

3.15.3.2 Typical Application

The following figure shows a typical application of the spinlock. A processor locks spinlock before executing specific codes, and then unlocks the codes. After the lock is freed, other processors can read or write the data.

Figure 3-18 Spinlock Typical Application Diagram



3.15.3.3 Spinlock State Machine

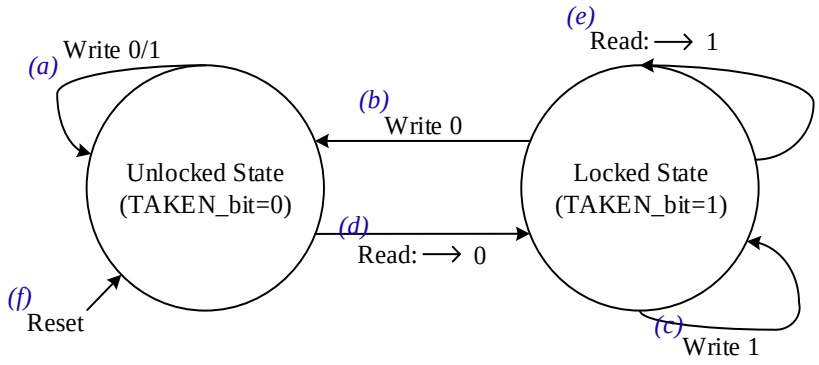
When a processor uses spinlock, it needs to acquire the spinlock status through SPINLOCK_STATUS_REG.

Reading operation: when the return value is 0, it indicates that the spinlock enters the locked status; reading this status bit again can return 1, it indicates that the spinlock is the locked status.

Writing operation: when the spinlock is in the locked status, writing 0 can convert the spinlock to the unlocked status, the writing operation for other status is invalid.

The following figure shows the spinlock state machine.

Figure 3-19 Spinlock State Machine



a) When the spinlock is in the unlocked state, writing 0/1 has no effect.

b) When the spinlock is in the locked state, writing 0 can convert the corresponding spinlock to the unlocked state.

c) When the spinlock is in the locked state, writing 1 has no effect.

d) When the spinlock is in the unlocked state, reading the bit can return 0 (it indicates spinlock enters into the locked state).

e) When the spinlock is in the locked state, reading the bit can return 1 (it indicates spinlock is in the locked state).

f) After reset, the spinlock is in the unlock state by default.

3.15.4 Programming Guidelines

3.15.4.1 Switching the Status

Follow the steps below to switch the lock status of a spinlock.

Step 1 When the read value from SPINLOCKN_LOCK_REG (N=0–31) is 0, the spinlock comes into the locked status.

Step 2 Execute the application codes, and the status of SPINLOCK_STATUS_REG is 1.

Step 3 Write 0 to SPINLOCKN_LOCK_REG (N=0–31), the spinlock converts into the unlocked status, and the corresponding spinlock is released.

3.15.4.2 Processing the Interrupt

The spinlock generates an interrupt when a lock is freed (the lock status converts from the locked status to the unlocked status).

Follow the steps below to process the interrupt:

Step 1 Configure the interrupt enable bit of the corresponding spinlock in SPINLOCK_IRQ_EN_REG to enable the interrupt

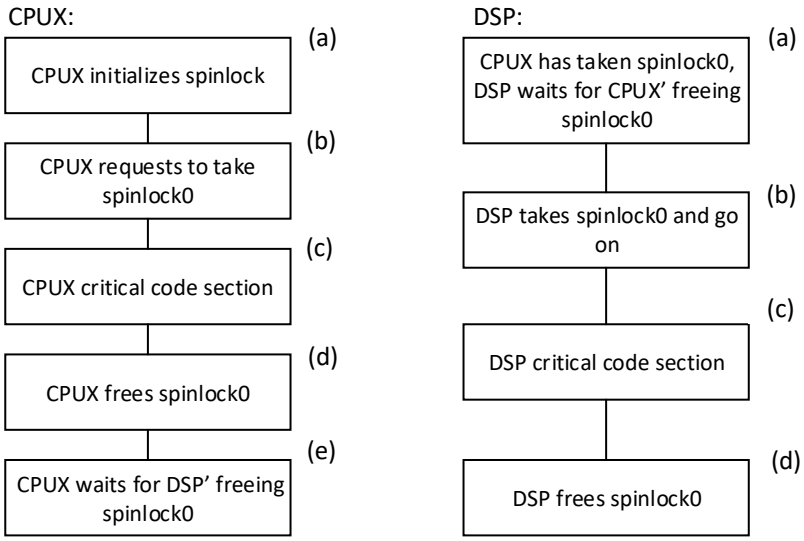
Step 2 The spinlock generates an interrupt when its status converts from the locked status to the unlocked status, and the corresponding bit of the SPINLOCK_IRQ_STA_REG turns to 1.

Step 3 Execute the interrupt handle function and clear the pending bit.

3.15.4.3 Taking/Freeing Spinlock

Take the synchronization between CPUX and DSP with Spinlock0 as an example, the CPUX and DSP perform the following steps.

Figure 3-20 CPUX and DSP Taking/Freeing Spinlock0 Process



CPUX:

a. The CPUX initializes Spinlock.

b. Check lock register0 (SPINLOCK_STATUS_REG0) status. If it is taken, check until CPUX frees spinlock0 and then request to take spinlock0. Otherwise, retry until the lock register0 is taken.

c. Execute CPUX critical code.

d. After executing CPUX critical code, the CPUX frees spinlock0.

e. The CPUX waits for DSP to free spinlock0.

DSP:

a. If the CPUX has taken spinlock0, the DSP waits for CPUX to free spinlock0.

b. The DSP requests to take spinlock0. If it fails, retry until the lock register0 is taken.

c. Execute DSP critical code.

d. After executing DSP critical code, the DSP frees spinlock0.

The following codes are for reference.

----------**CPUX**----------

Step 1 CPUX initializes Spinlock

```
put_wvalue(SPINLOCK_BGR_REG,0x00010000);
put_wvalue(SPINLOCK_BGR_REG,0x00010001);
```

Step 2 CPUX requests to take spinlock0

```
rdata=readl(SPINLOCK_STATUS_REG0);        //Check lock register0 status
```

if(rdata != 0) writel(0, SPINLOCK_LOCK_REG0); //If it is taken, check till CPUX frees spinlock0

rdata=readl(SPINLOCK_LOCK_REG0); //Request to take spinlock0

if(rdata != 0) rdata=readl(SPINLOCK_LOCK_REG0); //If it fails, retry till lock register0 is taken

---------- **CPUX critical code section** ----------

Step 3 CPUX frees spinlock0

writel(0, SPINLOCK_LOCK_REG0); //CPUX frees spinlock0

Step 4 CPUX waits for DSP' freeing spinlock0

writel(readl(SPINLOCK_STATUS_REG0) = 1); //CPUX waits for DSP' freeing spinlock0

----------**DSP**----------

Step 1 CPUX has taken spinlock0, DSP waits for CPUX' freeing spinlock0

while(readl(SPINLOCK_STATUS_REG0) = 1); //DSP waits for CPUX' freeing spinlock0

Step 2 DSP takes spinlock0 and go on

rdata=readl(SPINLOCK_LOCK_REG0); //Request to take spinlock0

if(rdata != 0) rdata=readl(SPINLOCK_LOCK_REG0); //If it fails, retry till lock register0 is taken

---------- **DSP critical code section** ----------

Step 3 DSP frees spinlock0

writel(0, SPINLOCK_LOCK_REG0);//DSP frees spinlock0

3.15.5 Register List

Module Name	Base Address
Spinlock	0x40003000

Register Name	Offset	Description
SPINLOCK_SYSTATUS_REG	0x0000	Spinlock System Status Register
SPINLOCK_STATUS_REG	0x0010	Spinlock Status Register
SPINLOCK_IRQ_EN_REG	0x0020	Spinlock Interrupt Enable Register
SPINLOCK_IRQ_STA_REG	0x0040	Spinlock Interrupt Status Register
SPINLOCK_LOCKID0_REG	0x0080	Spinlock Lockid0 Register
SPINLOCK_LOCKID1_REG	0x0084	Spinlock Lockid1 Register
SPINLOCK_LOCKID2_REG	0x0088	Spinlock Lockid2 Register
SPINLOCK_LOCKID3_REG	0x008C	Spinlock Lockid3 Register
SPINLOCK_LOCKID4_REG	0x0090	Spinlock Lockid4 Register
SPINLOCKN_LOCK_REG	0x0100+N*0x0004	Spinlock Register N (N=0~31)

3.15.6 Register Description

3.15.6.1 0x0000 Spinlock System Status Register (Default Value: 0x1000_0000)

Offset: 0x0000			Register Name: SPINLOCK_SYSTATUS_REG
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:28	R	0x1	LOCKS_NUM Number of lock registers implemented 00: This instance has 256 lock registers. 01: This instance has 32 lock registers. 10: This instance has 64 lock registers. 11: This instance has 128 lock registers.
27:9	/	/	/
8	R	0x0	IU0 In-Use flag 0, covering lock registers 0-31. 0: All lock registers 0-31 are in the Not Taken state. 1: At least one of the lock registers 0-31 is in the Taken state.
7: 0	/	/	/

3.15.6.2 0x0010 Spinlock Status Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: SPINLOCK_STATUS_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x0	LOCK_REG_STATUS Spinlock [i] status (i=0~31) 0: The Spinlock is free. 1: The Spinlock is taken.

3.15.6.3 0x0020 Spinlock Interrupt Enable Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: SPINLOCK_IRQ_EN_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	LOCK_IRQ_EN Spinlock [i] interrupt enable 0: Disable 1: Enable

3.15.6.4 0x0040 Spinlock Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0040			Register Name: SPINLOCK_IRQ_STA_REG
Bit	Read/Write	Default/Hex	Description

Offset: 0x0040			Register Name: SPINLOCK_IRQ_STA_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R/W1C	0x0	LOCK_IRQ _STATUS Spinlock [i] interrupt status. 0: No effect 1: pending Writing 1 will clear this bit.

3.15.6.5 0x0080 Spinlock Lockid0 Register (Default Value: 0x7777_7777)

Offset: 0x0080			Register Name: SPINLOCK_LOCKID0_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x77777777	LOCKID0

3.15.6.6 0x0084 Spinlock Lockid1 Register (Default Value: 0x7777_7777)

Offset: 0x0084			Register Name: SPINLOCK_LOCKID1_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x77777777	LOCKID1

3.15.6.7 0x0088 Spinlock Lockid2 Register (Default Value: 0x7777_7777)

Offset: 0x0088			Register Name: SPINLOCK_LOCKID2_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x77777777	LOCKID2

3.15.6.8 0x008C Spinlock Lockid3 Register (Default Value: 0x7777_7777)

Offset: 0x008C			Register Name: SPINLOCK_LOCKID3_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x77777777	LOCKID3

3.15.6.9 0x0090 Spinlock Lockid4 Register (Default Value: 0x7777_7777)

Offset: 0x0090			Register Name: SPINLOCK_LOCKID4_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x77777777	LOCKID4

3.15.6.10 0x0100+N*0x0004 (N=0~31) Spinlock Register N (N=0 to 31) (Default Value: 0x0000_0000)

Offset: 0x0100+N*0x0004 (N=0~31)			Register Name: SPINLOCKN_LOCK_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/

Offset: 0x0100+N*0x0004 (N=0~31)			Register Name: SPINLOCKN_LOCK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	TAKEN Lock State Read 0x0: The lock was previously Not Taken (free). The requester is granted the lock. Write 0x0: Set the lock to Not Taken (free). Read 0x1: The lock was previously Taken. The requester is not granted the lock and must retry. Write 0x1: No update to the lock value.

Contents

Figures

4 Video & Graphics

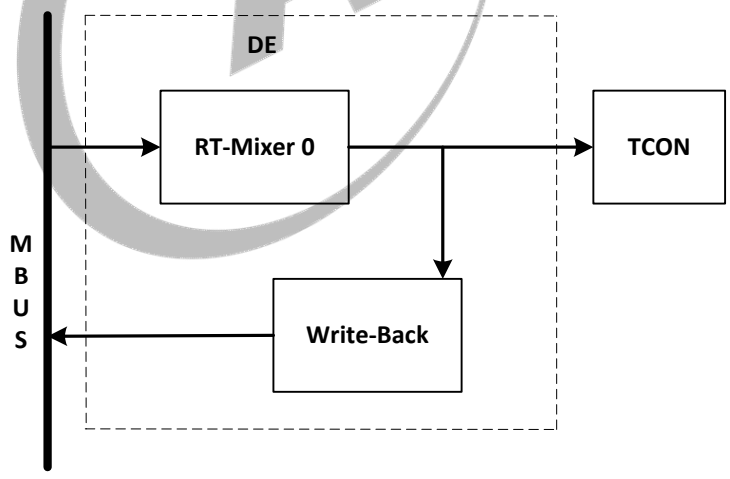
4.1 Display Engine

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 the following figure.

The DE has the following features:

- Supports output size up to 1024x1024.
- Supports two alpha blending UI channels for main display.
- Supports four overlay layers in each channel, and channel 0 has an independent scaler.
- Supports potter-duff compatible blending operation.
- Supports input format ARGB8888/XRGB8888/RGB888/ARGB4444/ARGB1555/RGB565.
- Supports SmartColor2.0 for excellent display experience.
  - Fully programmable color matrix.
  - Dynamic gamma.
- Supports write back only for verification.

Figure 4-1 DE Block Diagram



4.2 Graphic 2D

The Graphic 2D (G2D) engine is hardware accelerator for 2D graphic.

The G2D has the following features:

- Supports layer size up to 1024x1024 pixels
- Supports pre-multiply alpha image data
- Supports color key
- Supports two pipes Porter-Duff alpha blending
- Supports multiple video formats 4:2: 0, 4:2:2, 4:1:1 and multiple pixel formats (8/16/24/32 bits' graphics layer)
- Supports any format convert function above
- Supports 1/16× to 32× resize for video channel
- Supports 32-phase 8-tap horizontal anti-alias filter and 32-phase 4-tap vertical anti-alias filter.
- Supports window clip
  - 32bpp format clip width should be more than 2
  - 24bpp format clip width should be more than 3
  - 16bpp format clip width should be more than 4
- Supports FillRectangle, BitBlit, StretchBlit and MaskBlit
- Supports horizontal and vertical flip, clockwise 0/90/180/270 degree rotate for normal buffer

Contents

Figures

Tables

5 Video Input Interfaces

5.1 CSI_JPEG

5.1.1 Overview

The CSI_JPEG provides image or video capture, and encodes them into JPEG files in real time or non-real time.

The CMOS Sensor Interface Controller(CSIC) includes the following features:

- Supports YUV422 format input and YUV420 format NV12 output
- Supports cropwin
- Supports receiving JPEG streams directly output by sensor
- Supports receiving the images with unconventional resolutions (X and Y can be an integer multiple of 16, such as: 192*192, 304*224)
- Supports receiving the images with conventional resolutions (such as 128*128, 256*256, 320*240, 640*480, 1280*720, 1920*1088)

The JPEG encoder includes the following features:

- Supports 640*480@60fps in the online mode, and 640*480@30fps in the offline mode
- Supports 1280*720@40fps in the online mode, and 1280*720@20fps in the offline mode
- Supports up to 1920*1088 online/offline encoding
- Supports encoding after scaling images down 1/2 and cropwin
- Supports block output in the online mode to reduce SRAM usage and improve bandwidth utilization ratio
- Supports online/offline encoding (such as 192*192 and 304*224) the images with non-conventional resolution
- Supports the online/offline encoding the images with conventional resolution images

5.1.2 Functional Description

5.1.2.1 Block Diagram

CSI supports data input in YUV422, RAW, and JPEG formats, and the interface supports independent SYNC timing and embedded SYNC (CCIR656) timing. The following figures show the CSI hardware interface and CSI separate sync interface timing.

Figure 5-1 CSI Hardware Interface

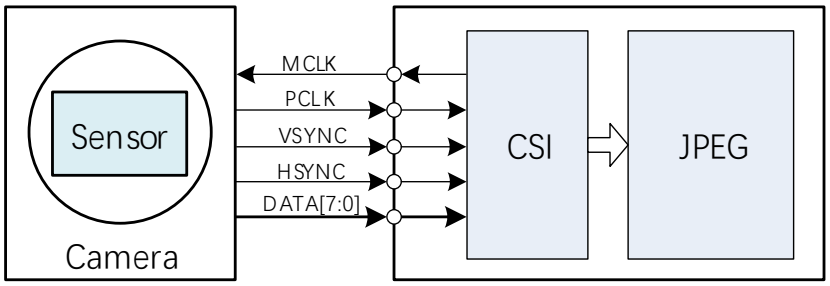
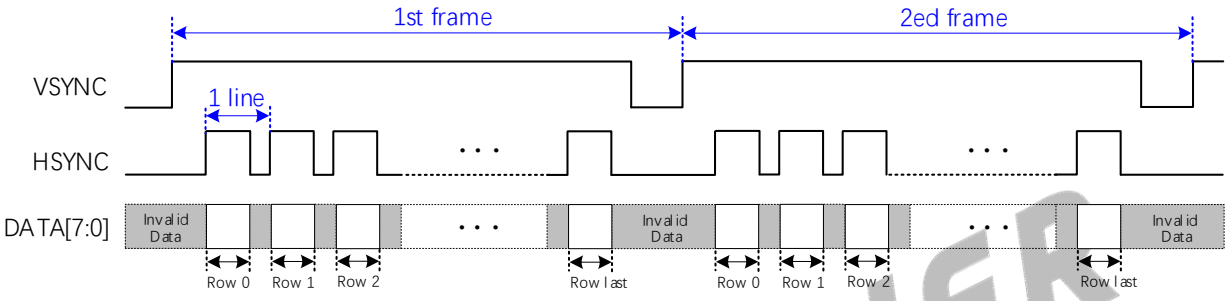


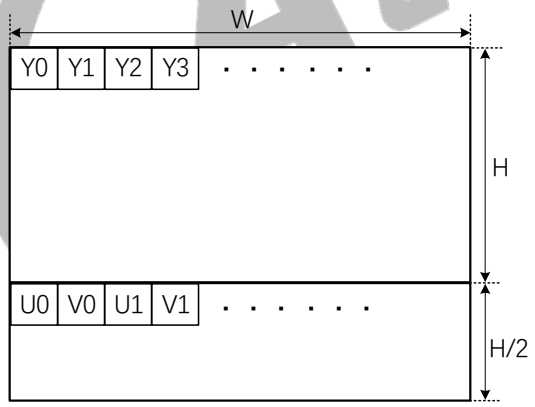
Figure 5-2 CSI Separate Sync Interface Timing



The active levels of VSYNC and HSYNC can be positive or negative. The camera should be consistent with the internal configuration of CSI during configuration.

After the common format YUV422 is input, the CSI will process it into the format YUV420 and output it to memory in the format of NV12. The following figure shows the memory layout of NV12 format.

Figure 5-3 Memory Layout of NV12 Format



CSI supports two image capture modes: still capture and video capture.

Still Capture: When the SCAP_ON is enabled, one frame of image will be captured. The still capture mode can be used for single shot.

Vdieo Capture: When the VCAP_ON is enabled, multiple images will be continuously captured until VCAP_ON is disabled. The video capture mode can be used for video shot.

CSI will generate Frame End each time when a frame is received. Frame End will take effect after the number of lines received per frame reaches VER_LEN(reg0x14[29:16]). If the points X per line or the lines Y per frame is inconsistent with the value INPUT_X_SIZE or INPUT_Y_SIZE, a size change interrupt will be generated. This

may be due to clock or data errors caused by hardware interference, and it may be risky to continue to use the current frame data.

5.1.2.2 External Signals

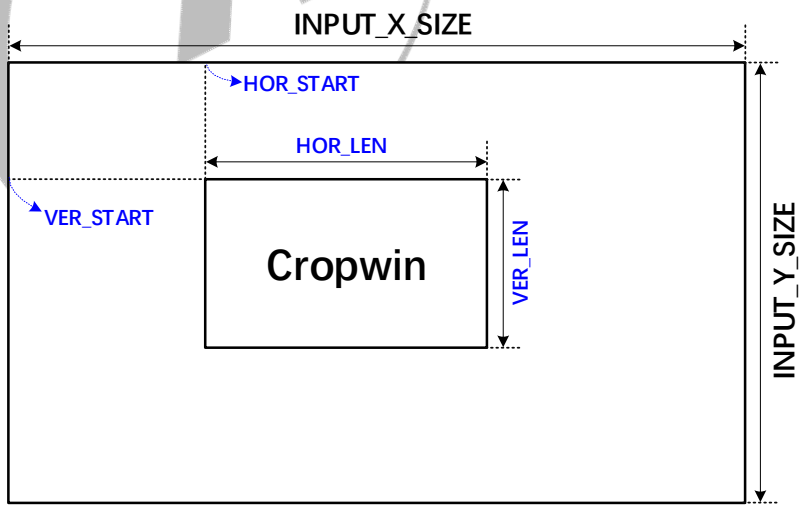
Signal	Description	Type
NCSI_HSYNC	CSI Horizontal sync, indicate one new scan line	I
NCSI_VSYNC	CSI Vertical sync, indicates one new frame	I
NCSI_PCLK	CSI pixel data synchronization clock input	I
NCSI_MCLK	CSI main clock output for camera working clock (not necessary)	O
NCSI_D0	CSI data[0] input	I
NCSI_D1	CSI data[1] input	I
NCSI_D2	CSI data[2] input	I
NCSI_D3	CSI data[3] input	I
NCSI_D4	CSI data[4] input	I
NCSI_D5	CSI data[5] input	I
NCSI_D6	CSI data[6] input	I
NCSI_D7	CSI data[7] input	I

5.1.2.3 CSI Function

Cropwin

If the length HOR_LEN and VER_LEN received by CSI is same to the actual length output by camera, it is full-size image receiving. If the length HOR_LEN or VER_LEN is smaller than the actual length output by camera, cropwin is available. The following figure shows the cropwin function diagram.

Figure 5-4 Cropwin Function Diagram



Cropwin function can be realized by configuring the values of HOR_LEN, VER_LEN, HOR_STERT, and VER_STERT. HOR_LEN and VER_LEN are the length and width of screenshot respectively. HOR_STERT and VER_STERT are the coordinate positions of starting cropwin. The cropwin dimensions of HOR_LEN and VER_LEN should be the integral multiples of 16.

5.1.2.4 JPEG Encoding

There are two working modes for JPEG encoding: online mode and offline mode.

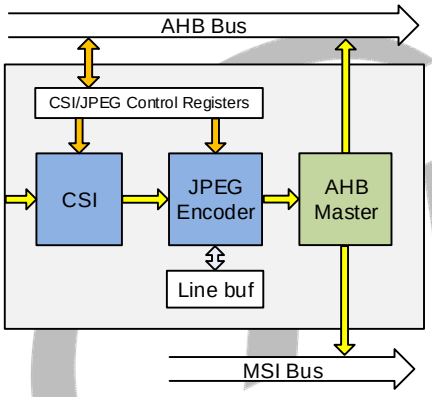
In the online mode, JPEG encoding will start each time when CSI receives 16 lines of data. As the receiving process ends, JPEG encoding will complete. CSI will not store the raw image data but output the data encoded by JPEG. The ways of encoded data output include online overall frame mode and and online block mode.

In the offline mode, the data received by CSI will be stored to memory first. When an overall frame of data is well stored, JPEG encoding will start by software.

Online Overall Frame Mode

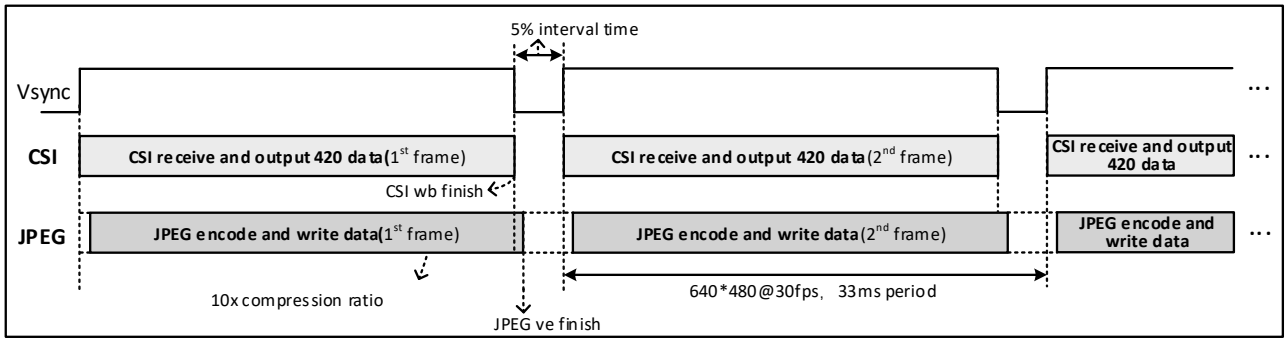
In this mode, CSI converts the format of the data output by external sensor from YUV422 to YUV420. Each time when 16 lines of data is received, hardware will auto initiate JPEG encoding. Then the output bit stream will be written to the designated memory through the AHB2. Take an example of the resolution 640x480, there are 30 encodings. Every time when one encoding is completed, JPEG encoder will generate the ve finish interrupt. Thus, the lines of image resolution should be the integral multiples of 16, and the minimum resolution is 32*32. The following figure shows the signal flow in the online mode.

Figure 5-5 Signal Flow in the Online Mode



CSI converts the format of received data from YUV422 to YUV420, and then transmits them to ENCPP. ENCPP caches YUV420 data in the Line Buf. When 16 lines of data is cached, JPEG encoding will be initiated. The encoded data will be sent to ahb1to2 through mbus2ahb. ahb1to2 will be ultimately written to memory through the AHB or MSI. The following figure shows the timing diagram in online overall frame mode.

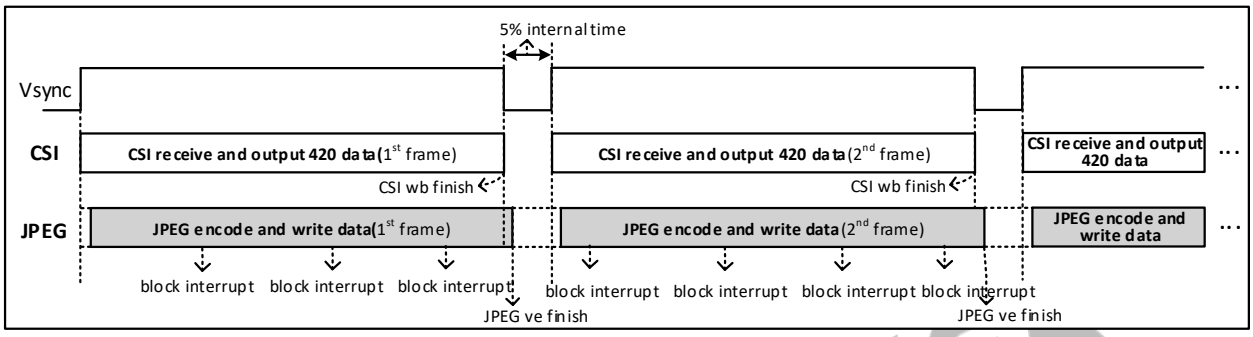
Figure 5-6 Timing Diagram in Online Overall Frame Mode



Online Block Mode

In this mode, when the volume of data output by JPEG encoding reaches the set target (such as 2 KB/4 KB), CSI will generate interrupt and reuse the encoding output space during a frame of image encoding. For example, if in 8 KB encoding output space, there is a frame of 20KB image needs to be encoded., JPEG will start storing data from the start address of the encoding output space when the space is full. Therefore, the previous data must be read, otherwise they will be overwritten. The following figure shows the timing diagram of online block mode.

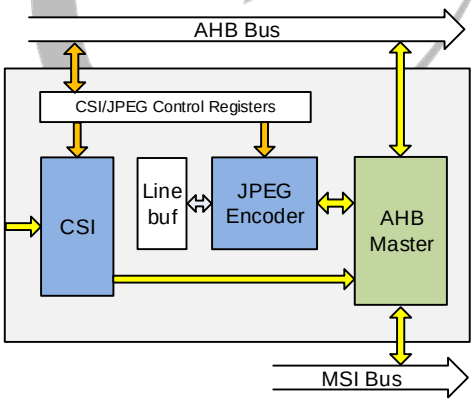
Figure 5-7 Timing Diagram in Online Block Mode



Offline Mode

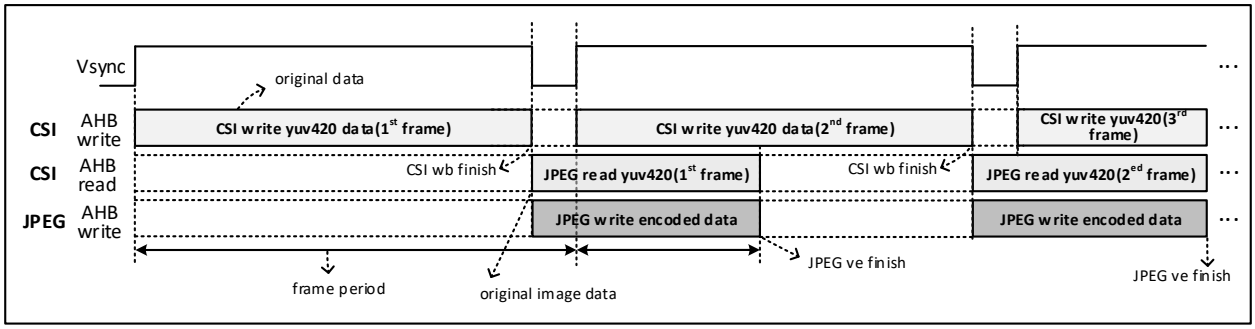
In this mode, CSI converts the format of the data output by external sensor from YUV422 to YUV420, and then writes the data to the designated memory. When a whole frame of data is received and written to memory, CSI will generate the write back finish interrupt. Then, software will initiate JPEG encoder in ISR for encoding, and the output bit stream will be written to memory through the AHB2. Thus, the image receiving and encoding aren't conducted simultaneously in the offline mode. The following figure shows the signal flow in the offline mode.

Figure 5-8 Signal Flow in the Offline Mode



CSI stores the original YUV420 image data in the format of NV12 to memory. When a frame of image is well stored, wb finish interrupt will be generated. Then, software will initiate JPEG encoding, and ENCPP will read the original YUV420 image data for encoding. The encoded data will be sent to ahb1to2 through mbus2ahb. ahb1to2 will be ultimately written to memory through AHB bus or MSI bus. The following figure shows the timing diagram in the offline mode.

Figure 5-9 Timing Diagram in the Offline Mode



5.1.3 Register List

Module Name	Base Address
CSI	0x40300000
VE_TOP	0x40300400
ENCPP	0x40300800
JPEG	0x40300C00

Register Name	Offset Address	Description
CMOS Sensor Interface		
CSI_EN_REG	0x0000	CSI Enable Register
CSI_CFG_REG	0x0004	CSI Configuration Register
CSI_CAP_REG	0x0008	CSI Capture Register
CSI_SIGNAL_STA	0x000C	CSI Signal Status Register
CSI_OUT_HOR_SIZE	0x0010	CSI Output Horizontal Size Register
CSI_OUT_VER_SIZE	0x0014	CSI Output Vertical Size Register
CSI_INPUT_SIZE	0x0018	CSI Input Size Register
CSI_INT_EN_REG	0x001C	CSI Interrupt Enable Register
CSI_INT_STA_REG	0x0020	CSI Interrupt Status Register
VE_TOP		
VE_MODE_REG	0x0000	VE Mode Control Register
VE_RESET_REG	0x0004	VE Reset Control Register
VE_CNT_REG	0x0008	VE Counter Control Register
VE_OVERTIME_REG	0x000C	VE Overtime Register
VE_INT_STA_REG	0x001C	VE Interrupt Status Register
CSI_OUTPUT_ADDR_Y	0x0020	CSI Output Y Address Register
CSI_OUTPUT_ADDR_UV	0x0024	CSI Output UV Address Register
CSI_OUTPUT_STRIDE	0x0028	CSI Output Stride Control Register
CSI_OUTPUT_BUS1_EN	0x002C	CSI Output BUS1 ENABLE
CSI_OUTPUT_BUS1_ST_ADDR	0x0030	CSI Output BUS1 START ADDRESS
CSI_OUTPUT_BUS1_END_ADDR	0x0034	CSI Output BUS1 END ADDRESS
ENCPP		

Register Name	Offset Address	Description
JPEG_INPUT_SIZE	0x0000	JPEG Input Size Register
JPEG_INPUT_STRIDE0	0x0004	JPEG Input Stride0 Control Register
JPEG_INPUT_STRIDE1	0x0014	JPEG Input Stride1 Control Register
JPEG_INPUT_ADDR_Y	0x0078	JPEG Input Y Address Register
JPEG_INPUT_ADDR_UV	0x007C	JPEG Input UV Address Register
JPEG		
JPEG_PARA_REG	0x0004	JPEG Para Config Register
JPEG_BITRATE_CTRL	0x0008	JPEG Bitrate Control Register
VE_INT_EN_REG	0x0014	VE Interrupt Enable Register
VE_START_TRIG	0x0018	VE Start Trigger Register
VE_STA_REG	0x001C	VE Status Register
VE_PUTBITS_DATA	0x0020	VE Putbits Data Register
MBLEVEL_OVERTIME	0x0024	Macroblock Level Overtime Register
OUTSTM_START_ADDR	0x0080	Output Stream Start address Register
OUTSTM_END_ADDR	0x0084	Output Stream End address Register
OUTSTM_OFFSET	0x0088	Output Stream Offset Register
OUTSTM_VSIZE	0x008C	Output Stream Valid Size Register
OUTSTM_LEN	0x0090	Output Stream Length Register
VE_QMINPUT_INDEX	0x00E0	VE Quantiser Matrix Input Index Register
VE_QMINPUT_DATA	0x00E4	VE Quantiser Matrix Input Data Register

5.1.4 CSI Register Description

5.1.4.1 0x0000 CSI Enable Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: CSI_EN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	R/W	0x0	TIMEOUT_THD CSI Timeout threshold
11:5	/	/	/
4	R/W	0x0	TIMEOUT_EN CSI Timeout enable 0: Disable 1: Enable
3	/	/	/
2	R/W	0x0	PCLK_EN 0: Gate pclk input 1: Enable pclk input
1	R/W	0x0	CSIC_EN 0: Reset and disable the CSIC module 1: Enable the CSIC module

Offset: 0x0000			Register Name: CSI_EN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	PRS_EN 0: Reset and disable the parser module 1: Enable the parser module

5.1.4.2 0x0004 CSI Configuration Register (Default Value: 0x0000_0A8A)

Offset: 0x0004			Register Name: CSI_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
12	R/W	0x0	YUV420 Mask Valid when Output_Mode set 'YUV422 to YUV420' 0: mask UV in odd lines (1,3,5……) 1:mask UV in even lines (2,4,6……)
11	R/W	0x1	Output_Mode 0: original output 1: YUV422 to YUV420
10	R/W	0x0	YUV420_LINE_ORDER 0: YUV420 input in Y-YC-Y-YC Line Order 1: YUV420 input in YC-Y-YC-Y Line Order
9:8	R/W	0x2	INPUT_SEQ Input data sequence, only valid for YUV422 and YUV420 input format. 00: YUYV 01: YVYU 10: UYVY 11: VYUY
7:6	R/W	0x2	INPUT_FMT Input data format 0x: RAW stream 10: YUV422 11: YUV420
5:4	/	/	/
3	R/W	0x1	VREF_POL Vref polarity 0: negative 1: positive This register is not apply to CCIR656 interface.
2	R/W	0x0	HERF_POL Href polarity 0: negative 1: positive This register is not apply to CCIR656 interface.

Offset: 0x0004			Register Name: CSI_CFG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x1	CLK_POL Data clock type 0: Active in rising edge 1: Active in falling edge
0	R/W	0x0	CSI_IF 0: separate syncs(DC) 1: embedded syncs(CCIR656)

5.1.4.3 0x0008 CSI Capture Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: CSI_CAP_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6	R/W	0x0	CSI_HMASK Frame rate half down.
5:2	R/W	0x0	CSI_CAP_MASK Vsync number masked before capture.
1	R/W	0x0	CSI_VCAP_ON Video capture control: Capture the video image data stream. 0: Disable video capture If video capture is in progress, the CSI stops capturing image data at the end of the current frame, and all of the current frame data is wrote to output FIFO. 1: Enable video capture The CSI starts capturing image data at the start of the next frame.
0	RC/W	0x0	CSI_SCAP_ON Still capture control: Capture a single still image frame. 0: Disable still capture. 1: Enable still capture The CSI module starts capturing image data at the start of the next frame. The CSI module captures only one frame of image data. This bit is self-clearing and always reads as a 0.

5.1.4.4 0x000C CSI Signal Status Register (Default Value: 0x0000_0000)

Offset: 0x000C			Register Name: CSI_SIGNAL_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:23	/	/	/
22:20	R	0x0	PCLK_CNT_STA Indicates the pclk status

Offset: 0x000C			Register Name: CSI_SIGNAL_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
19	R	0x0	VSYNC_STA Indicates the Vsync status 0: Low 1:high
18	R	0x0	HSYNC_STA Indicates the Hsync status 0: Low 1:high
17:8	/	/	/
7: 0	R	0x0	DATA_STA Indicates the Dn status(n=0~7), MSB for D7, LSB for D0 0: Low 1:high

5.1.4.5 0x0010 CSI Output Horizontal Size Register (Default Value: 0x0500_0000)

Offset: 0x0010			Register Name: CSI_OUT_HOR_SIZE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:16	R/W	0x500	HOR_LEN Horizontal pixel unit length. Valid pixel of a line.
15:14	/	/	/
13: 0	R/W	0x0	HOR_START Horizontal pixel unit start. Pixel is valid from this pixel.

5.1.4.6 0x0014 CSI Output Vertical Size Register (Default Value: 0x02D0_0000)

Offset: 0x0014			Register Name: CSI_OUT_VER_SIZE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:16	R/W	0x2d0	VER_LEN Valid line number of a frame.
15:14	/	/	/
13: 0	R/W	0x0	VER_START Vertical line start. data is valid from this line.

5.1.4.7 0x0018 CSI Input Size Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: CSI_INPUT_SIZE
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30:16	R	0x0	INPUT_Y_SIZE

Offset: 0x0018			Register Name: CSI_INPUT_SIZE
Bit	**Read/Write**	**Default/Hex**	**Description**
15	/	/	/
14: 0	R	0x0	INPUT_X_SIZE

5.1.4.8 0x001C CSI Interrupt Enable Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: CSI_INT_EN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x0	INPUT_TIMEOUT_INT_EN Input timeout interrupt enable 0: Disable 1: Enable
1	R/W	0x0	INPUT_FRM_END_INT_EN Input frame end interrupt enable 0: Disable 1: Enable
0	R/W	0x0	INPUT_SIZE_CHG_INT_EN Input frame size (X or Y) change interrupt enable 0: Disable 1: Enable

5.1.4.9 0x0020 CSI Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: CSI_INT_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
4	R/W1C	0x0	INPUT_Y_SIZE_CHG_PD CSI input Y size change pending When the csi input Y size changed, this flag set to “1”. Write “1” to clear.
3	R/W1C	0x0	INPUT_X_SIZE_CHG_PD CSI input X size change pending When the csi input X size changed, this flag set to “1”. Write “1” to clear.
2	R/W1C	0x0	INPUT_TIMEOUT_PD CSI timeout pending When no valid data input timeout, this flag set to “1”. Write “1” to clear.
1	R/W1C	0x0	INPUT_FRM_END_PD CSI input frame end pending When the captured frame end, this flag set to “1”. Write “1” to clear.

Offset: 0x0020			Register Name: CSI_INT_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W1C	0x0	INPUT_SIZE_CHG_PD CSI input X or Y size change pending When the parser input size changed, this flag set to “1”. Write “1” to clear.

5.1.5 VE_TOP Register Description

5.1.5.1 0x0000 VE Mode Control Register (Default Value: 0x0000 0000)

Offset: 0x0000			Register Name: VE_MODE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	W	0x0	MEM_PARTITION_TAKE Output stream memory partition take away Software reading value always zero. Software write “1” to this bit after taking away one memory partition.
16	R/W	0x0	MEM_PARTITION_MODE Output stream memory partition mode enable 0: Disable memory partition mode 1: Enable memory partition mode
15:14	R/W	0x0	MEM_PARTITION_NUM Output stream memory partition number 00: 2 memory blocks 01: 4 memory blocks 10: 8 memory blocks 11: Reserved
13	R/W	0x0	AHB_WRITE_BL_8W 0: Encoder use burst 4, access 4 words each time; 1: Encoder use burst 8, access 8 words each time; This bit influence bandwidth only. It does not have an effect on the hardware result.
12	R/W	0x0	AHB_READ_BL_8W 0: Encoder use burst 4, access 4 words each time; 1: Encoder use burst 8, access 8 words each time; This bit influence bandwidth only. It does not have an effect on the hardware result.

Offset: 0x0000			Register Name: VE_MODE_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
11	R/W	0x0	JPE_HEIGHT_HALF Used in online mode only. 0: JPEG encoder input picture height is the same with CSI input picture height; 1: JPEG encoder input picture height equals to half of CSI input picture height, CSI input picture height should align to 32.
10	R/W	0x0	JPE_WIDTH_HALF Used in online mode only. 0: JPEG encoder input picture width is the same with CSI input picture height; 1: JPEG encoder input picture width equals to half of CSI input picture height, CSI input picture width should align to 32.
9	R/W	0x0	ONLINE_MODE_EN 0: Off line mode; JPEG encoder need a whole frame as input; 1: On line mode; No need to set JPEG encoder input data buffer.
8	R/W	0x0	JPEG_INPUT_FMT 0: CSI output data format is yuv420 data (NV12 format). 1: CSI output data format is JPEG data. Used in offline mode. Some sensors have the ability to output JPEG picture data. In this case, driver should set this flag as 1 to inform CSI to process JPEG data.
7	R/W	0x0	ENC_CLK_EN Enable encoder top level clock. 0: Disable 1: Enable
6	R/W	0x0	JPE_CLK_EN Enable jpeg encoder level clock. 0: Disable 1: Enable
5	R/W	0x0	CLOCK_GATING_DISABLE This is a black door bit. 0: Use JPE_ENABLE and ENC_ENABLE to control clock. 1: Enable top and jpeg encoder clock at the same time. The same function as ENC_CLK_EN equals one and JPE_CLK_EN equals one. ENC_CLK_EN and JPE_CLK_EN are invalid in this case.
4: 0	/	/	/

5.1.5.2 0x0004 VE Reset Control Register (Default Value: 0x0000_0000)

Offset: 0x0004	Register Name: VE_RESET_REG

Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3	R/W	0x0	CFG_AHBO_RSTN 0: Reset MBUS2AHB and AHBBRIDGE2 1: Release
2	R/W	0x0	CFG_CORE_RSTN 0: Reset VE2_CORE except jpe_io_regs 1: Release
1	R/W	0x0	CFG_PARSER_RSTN 0: Reset PARSER except parser_ioregs 1: Release
0	R/W	0x1	CFG_MBUS2AHB_RSTN 0: Reset MBUS2AHB 1: Release

5.1.5.3 0x0008 VE Counter Control Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: VE_CNT_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	VE_CNT_EN 0: Counter stop 1: Counter enable VE_CNT will be clear to 0 when write “1” to this bit.
30: 0	R/W	0x0	VE_CNT JPEG encoder counter.

5.1.5.4 0x000C VE Overtime Register (Default Value: 0x0000_0000)

Offset: 0x000C			Register Name: VE_OVERTIME_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30:8	R/W	0x0	VE_OVERTIME When VE_CNT > VE_OVERTIME, VE_OVER_FLAG will be set, and VE_CNT will stop until VE_CNT be initial. VE_OVER_FLAG will be send to all modules for interrupt output.
7: 0	/	/	Hardware wired to 8’h00

5.1.5.5 0x001C VE Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: VE_INT_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/

Offset: 0x001C			Register Name: VE_INT_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
25	R	0x0	AHB_SYNC_IDLE Read only. 0: Some data still in ahb bus; 1: All data has been written to memory;
24	R/W1C	0x0	CSI_WB_FINISH_INT_PD CSI write back finish interrupt pending. Software write "1" to clear this pending. Only used in csi offline mode.
23	R/W1C	0x0	FIFO_OVERFLOW_INT_PD CSI&JPEG fifo overflow interrupt pending. Software write "1" to clear this pending.
22	R	0x0	CSI_ERROR CSI path error flag. Read only.
21	R	0x0	CSI_TIMEOUT CSI timeout flag. Read only.
20:11	/	/	/
10	R/W1C	0x0	MEM_PARTITION_OVERFLOW All memory blocks are full. To avoid overflow, software should take away a memory block data when MEM_PARTITION_INTERRUPT equals "1", and write "1" to OUTMEM_PARTITION_TAKE (VE_MODE_REG[17]).
9	R/W1C	0x0	MEM_PARTITION_INTERRUPT The bit is set "1" by hardware when one memory block data is available for software.
8	R/W	0x0	VE_TIMEOUT_INT_EN VE timeout interrupt enable. 0: Disable 1: Enable
7	R/W1C	0x0	MB_OVERTIME_INT_PD Marcoblock overtime interrupt pending. Software write "1" to clear this pending.
6	R/W1C	0x0	BS_STALL_INT_PD Bitstream stall interrupt pending. Software write "1" to clear this pending.
5:4	/	/	/
3	R/W1C	0x0	ENC_FINISH_INT_PD JPEG Encode finish interrupt pending. Software write "1" to clear this pending.
2	R	0x0	CSI_FRAME_END CSI frame end flag. Read only.
1	R	0x0	CSI_SIZE_CHG CSI size change flag. Read only.

Offset: 0x001C			Register Name: VE_INT_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W1C	0x0	VE_TIMEOUT_INT_PD VE timeout interrupt pending. Software write “1” to clear this pending.

5.1.5.6 0x0020 CSI Output Y Address Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: CSI_OUTPUT_ADDR_Y
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	R/W	0x0	CSI_OUTPUT_ADDR_Y[31:4] CSI output Luma(Y) data address. NV12 format. For offline mode only. When SENSOR_OUTPUT_JPEG equals one, JPEG data is stored in CSI_OUTPUT_ADDR_Y, and CSI_OUTPUT_ADDR_UV would be ignored.
3: 0	R	0x0	CSI_OUTPUT_ADDR_Y[3: 0] wired to 4h'0.

5.1.5.7 0x0024 CSI Output UV Address Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: CSI_OUTPUT_ADDR_UV
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	R/W	0x0	CSI_OUTPUT_ADDR_UV [31:3] CSI output Chroma(UV) data address. NV12 format. For offline mode only. When SENSOR_OUTPUT_JPEG equals one, JPEG data is stored in CSI_OUTPUT_ADDR_Y, and CSI_OUTPUT_ADDR_UV would be ignored.
2: 0	R	0x0	CSI_OUTPUT_ADDR_UV[2: 0] wired to 3h'0.

5.1.5.8 0x0028 CSI Output Stride Control Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: CSI_OUTPUT_STRIDE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:27	/	/	/
26:16	R/W	0x0	CSI_OUTPUT_Y_STRIDE_DIV16 CSI output Luma stride in 16x16 block. For offline mode only.
15:12	/	/	/
11: 0	R/W	0x0	CSI_OUTPUT_UV_STRIDE_DIV8 CSI output Chroma stride in 8x8 block. For offline mode only.

5.1.5.9 0x002C CSI Output BUS1 ENABLE (Default Value: 0x0000_0000)

Offset: 0x002C	Register Name: CSI_OUTPUT_BUS1_EN

Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	BUS1_ENABLE Enable BUS1(MSI IN aw1883).If BUS1 is enable and address is matched, BUS request will send to BUS1 instead of original BUS0.

5.1.5.10 0x0030 CSI Output BUS1 START ADDRESS (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: CSI_OUTPUT_BUS1_ST_ADDR
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	BUS1_START_ADDRESS

5.1.5.11 0x0034 CSI Output BUS1 END ADDRESS (Default Value: 0x0000 0000)

Offset: 0x0034			Register Name: CSI_OUTPUT_BUS1_END_ADDR
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	BUS1_START_ADDRESS

5.1.6 ENCPP Register Description

5.1.6.1 0x0000 JPEG Input Size Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: JPEG_INPUT_SIZE
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26:16	R/W	0x0	PIC_WIDTH_IN_8x8 Block The width of each picture in units of 8X8 macroblock.
15:11	/	/	/
10: 0	R/W	0x0	PIC_HEIGHT_IN_8x8 Block The height of each picture in units of 8X8 macroblock.

5.1.6.2 0x0004 JPEG Input Stride Control Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: JPEG_INPUT_STRIDE
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26:16	R/W	0x0	JPEG_INPUT_STRIDE_DIV16 JPEG input picture stride in macroblocks. This parameter should be set to even to get better memory efficiency. If input mode has two or three planar, this parameter only used for Y planar.
15: 0	/	/	/

5.1.6.3 0x0014 JPEG Input Stride1 Control Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: JPEG_INPUT_STRIDE1
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11: 0	R/W	0x0	JPEG_INPUT_C_STRIDE_DIV8 JPEG input picture stride of Chroma planar. If input mode only has one planar, this parameter is not used.

5.1.6.4 0x0078 JPEG Input Y Address Register (Default Value: 0x0000_0000)

Offset: 0x0078			Register Name: JPEG_INPUT_ADDR_Y
Bit	Read/Write	Default/Hex	Description
31:4	R/W	0x0	JPEG_INPUT_ADDR_Y[31:4] JPEG input Luma(Y) data address. NV12 format. For offline mode only.
3: 0	R	0x0	JPEG_INPUT_ADDR_Y[3: 0] wired to 4h'0.

5.1.6.5 0x007C JPEG Input UV Address Register (Default Value: 0x0000_0000)

Offset: 0x007C			Register Name: JPEG_INPUT_ADDR_UV
Bit	Read/Write	Default/Hex	Description
31:3	R/W	0x0	JPEG_INPUT_ADDR_UV[31:3] JPEG input Chroma(UV) data address. NV12 format. For offline mode only.
2: 0	R	0x0	JPEG_INPUT_ADDR_UV[2: 0] wired to 3h'0.

5.1.7 JPEG Register Description

5.1.7.1 0x0004 JPEG Para Control Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: JPEG_PARA_REG
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30	R/W	0x0	STUFF_ZERO_EN Stuffing Zero Insert Disable 0: Disable insert stuffing zero after 0xff 1: Insert stuffing zero after 0xff
29:27	/	/	/
26:16	R/W	0x0	DC_CHRO JPEG Encoder Chroma Default DC
15:11	/	/	/
10: 0	R/W	0x0	DC_LUMA JPEG Encoder Luma Default DC

5.1.7.2 0x0008 JPEG Bitrate Control Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: JPEG_BITRATE_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	run_length_opt_disable 0: Run_length_th enable 1: Run_length_th disable Suggested val is 1.
30	R/W	0x0	coef_down_sample_disable 0: Classfy_th enable 1: Classfy_th disable Suggested val is 1.
29:12	/	/	/
11:8	R/W	0x0	run_length_th Value: 0~15. The larger run_length_th val, the larger bit stream size. Suggested val is 8.
7: 0	R/W	0x0	classfy_th The larger classfy_th val, the smaller bit stream size. Suggested val is 48.

5.1.7.3 0x0014 VE Interrupt Enable Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: VE_INT_EN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x0	MB_OVERTIME_INT_EN Marcolever overtime interrupt enable 0: Disable 1: Enable
1	R/W	0x0	BS_STALL_INT_EN Bitstream stall interrupt enable 0: Disable 1: Enable
0	R/W	0x0	VE_FINISH_INT_EN VE encode finish interrupt enable 0: Disable 1: Enable

5.1.7.4 0x0018 VE Start Trigger Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: VE_START_TRIG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:14	/	/	/

Offset: 0x0018			Register Name: VE_START_TRIG
Bit	Read/Write	Default/Hex	Description
13:8	R/W	0x0	PUTBITS_LEN The bits number to put.
7:4	/	/	/
3: 0	R/W	0x0	1000: VE encode start Others: Reserved

5.1.7.5 0x001C VE Status Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: VE_STA_REG
Bit	Read/Write	Default/Hex	Description
31	R	0x0	PIC_ENC_BUSY Read-only (for debug)
30:24	/	/	/
23	/	/	READ_MB_BUSY Read-only (for debug)
22:14	/	/	/
13	R	0x0	IPTIT_BUSY Read-only (for debug)
12	R	0x0	VLC_BUSY Read-only (for debug)
11:10	/	/	/
9	R	0x0	PUTBITS_STATUS 0: Put bits busy 1: Put bits free
8	/	/	/
7	R	0x0	ENC_MEM_BUSY Read-only (for debug)
6	R	0x0	ENC_ISP Read-only (for debug)
5: 0	/	/	/

5.1.7.6 0x0020 VE Putbits Data Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: VE_PUTBITS_DATA
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	PUTBITS_DATA The bits data to put.

5.1.7.7 0x0024 Macroblock Level Overtime Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: MBLEVEL_OVERTIME
Bit	Read/Write	Default/Hex	Description

Offset: 0x0024			Register Name: MBLEVEL_OVERTIME
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15	R/W	0x0	MBOVERTIME_EN 0: Disable mblevel overtime counter 1: Enable mblevel overtime counter
14:12	R/W	0x0	MBOVERTIME_THRESHOLD [14:12] Macroblock level encode overflow threshold, if mb level encode time overflow, it will set MBOVERTIME_INTERRUPT to "1", but it continue encode next macroblock if it can go on.
11: 0	R	0x0	MBOVERTIME_THRESHOLD [11: 0] Hardware wired to 12'h0.

5.1.7.8 0x0080 Output Stream Start address Register (Default Value: 0x0000_0000)

Offset: 0x0080			Register Name: OUTSTM_START_ADDR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	R/W	0x0	STREAM_START_ADDR [31:10] The start address of JPEG encoded output stream. It must configure OUTSTRM_OFFSET register before set this register.
9: 0	R	0x0	STREAM_START_ADDR [9: 0] Hardware wired to 10'h0.

5.1.7.9 0x0084 Output Stream End Address Register (Default Value: 0xFFFF_FFFF)

Offset: 0x0084			Register Name: OUTSTM_END_ADDR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	R/W	0x0	STREAM_END_ADDR [31:10] The end address of JPEG encoded output stream.
9: 0	R	0x3FF	STREAM_END_ADDR [9: 0] Hardware wired to 10'h3FF.

5.1.7.10 0x0088 Output Stream Offset Register (Default Value: 0x0000_0000)

Offset: 0x0088			Register Name: OUTSTM_OFFSET
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27: 0	R0/W	0x0	STREAM_OFFSET JPEG Encoded bitstream bit offset Software configuration register, but it's read value hardware_offset register. It must be 256 bit align.

5.1.7.11 0x008C Output Stream Valid Size Register (Default Value: 0x0FFF_0000)

Offset: 0x008C			Register Name: OUTSTM_VSIZE
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0x0	STREAM_VALID_SIZE JPEG Encoded bitstream valid output memory space size.
15: 0	/	/	/

5.1.7.12 0x0090 Output Stream Length Register (Default Value: 0x0000_0000)

Offset: 0x0090			Register Name: OUTSTM_LEN
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27: 0	R	0x0	OUTSTM_LEN JPEG Encoded stream bit offset. Attention of that it is always great than STREAMOFFSET even it is overflow. Output bitstream bit size is OUTSTM_LEN minus STREAM_OFFSET.

5.1.7.13 0x00E0 VE Quantiser Matrix Input Index Register (Default Value: 0x0000_0000)

Offset: 0x00E0			Register Name: VE_QMINPUT_INDEX
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6: 0	R/W	0x0	QM_INDEX [6: 0] The Index of Quantiser Matrix.

5.1.7.14 0x00E4 VE Quantiser Matrix Input Data Register (Default Value: 0x0000_0000)

Offset: 0x00E4			Register Name: VE_QMINPUT_DATA
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	QM_DATA [31: 0] The Quantiser Matrix Data.

Contents

Figures

Tables

6 Video Output Interfaces

6.1 Timing Controller_LCD (TCON_LCD)

6.1.1 Overview

The Timing Controller_LCD (TCON_LCD) is a module that processes video signals received from system through a complicated arithmetic and then generates control signals and transmits them to the LCD panel driver IC.

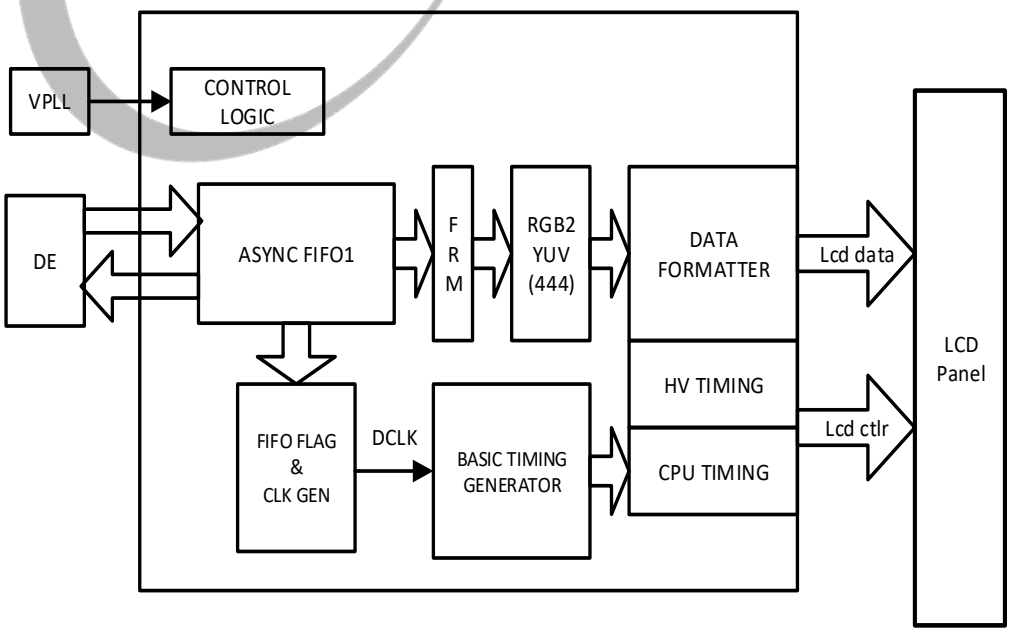
The TCON_LCD includes the following features:

- Supports RGB interface with DE/SYNC mode, up to 1024x768@60fps
- Supports serial RGB/dummy RGB interface, up to 800x480@60fps
- Supports i8080 interface, up to 800x480@60fps
- Supports BT656 interface for NTSC and PAL
- Supports RGB666 and RGB565 with dither function

6.1.2 Block Diagram

The following figure shows the block diagram of the TCON LCD.

Figure 6-1 TCON_LCD Block Diagram



6.1.3 Functional Description

6.1.3.1 External Signals

The LCD external signals are used to connect to panel interface. The panel interface has various types.

Table 6-1 LCD External Signals

Signal	Description	Type
LCD_CLK	LCD clock, pixel data are sync by this clock	O
LCD_VSYNC	LCD Vertical sync, indicates one new frame	O
LCD_HSYNC	LCD Horizontal sync, indicate one new scan line	O
LCD_DE	LCD data output enable	O
LCD_D23	LCD data[23] output or input	O/I
LCD_D22	LCD data[22] output or input	O/I
LCD_D21	LCD data[21] output or input	O/I
LCD_D20	LCD data[20] output or input	O/I
LCD_D19	LCD data[19] output or input	O/I
LCD_D18	LCD data[18] output or input	O/I
LCD_D15	LCD data[15] output or input	O/I
LCD_D14	LCD data[14] output or input	O/I
LCD_D13	LCD data[13] output or input	O/I
LCD_D12	LCD data[12] output or input	O/I
LCD_D11	LCD data[11] output or input	O/I
LCD_D10	LCD data[10] output or input	O/I
LCD_D7	LCD data[7] output or input	O/I
LCD_D6	LCD data[6] output or input	O/I
LCD_D5	LCD data[5] output or input	O/I
LCD_D4	LCD data[4] output or input	O/I
LCD_D3	LCD data[3] output or input	O/I
LCD_D2	LCD data[2] output or input	O/I

6.1.3.2 Clock Sources

The following table describes the clock sources of TCON_LCD. The following table describes the clock sources of TCON_LCD.

Table 6-2 TCON_LCD Clock Sources

Clock Sources	Description
hclk	AHB interface working clock (200 MHz)
lcd_clk	Working clock of TCON_L (390 MHz at max). PCLK is obtained after frequency division.
de_clk	Working clock of DE (200 MHz), which is input to TCON_LCD module together with DE data.
testclk	Test clock

6.1.3.3 Control Signal and Data Port Mapping

		SYNC RGB					DC	CPU Cmd	CPU 18-bit	CPU 16bit								CPU 8bit							CPU 9bit	
External I/O	Internal pin	Para RGB	Serial RGB			CCIR 656	YUV422		256K	256K							65K	256K				65K			256K	
			1st	2nd	3rd		Connection			1s	2n	3r	1s	2n	1s	2n		1s	2n	3r	1s	2n	1s	2n		
LCD_VSYNC	IO0	VSYNC					VSYNC	CS																		
LCD_HSYNC	IO1	HSYNC						RD																		
LCD_CLK	IO2	DCLK					DCLK	WR																		
LCD_DE	IO3	DE					HSYNC	RS																		
LCD_D23	D23	R7						D23	R5	R	B	G	R		R	B	R4									
LCD_D22	D22	R6						D22	R4	R	B	G	R		R	B	R3									
LCD_D21	D21	R5						D21	R3	R	B	G	R		R	B	R2									
LCD_D20	D20	R4						D20	R2	R	B	G	R		R	B	R1									
LCD_D19	D19	R3						D19	R1	R	B	G	R		R	B	R0									
LCD_D18	D18	R2						D18	R0	R	B	G	R		R	B	G5									
LCD_D17	D17	R1						D17																		
LCD_D16	D16	R0						D16																		
LCD0_D15	D15	G7						D15	G5								G4									
LCD_D14	D14	G6						D14	G4								G3									
LCD_D13	D13	G5						D13	G3																	
LCD_D12	D12	G4	D7	D72	D7	D7	D7	D12	G2	G	R	B	G	B	G		G2	R	G	B	R	G	R	G		
LCD_D11	D11	G3	D6	D62	D6	D6	D6	D11	G1	G	R	B	G	B	G		G1	R	G	B	R	G	R	G		
LCD_D10	D10	G2	D5	D52	D5	D5	D5	D10	G0	G	R	B	G	B	G		G0	R	G	B	R	G	R	G		
LCD_D9	D9	G1						D9																		
LCD_D8	D8	G0						D8																		
LCD_D7	D7	B7	D4	D42	D4	D4	D4	D7	B5	G	R	B	G	B	G		B4	R	G	B	R	B	R	B		
LCD_D6	D6	B6	D3	D32	D3	D3	D3	D6	B4	G	R	B	G	B	G		B3	R	G	B	R	B	R	B		
LCD_D5	D5	B5	D2	D22	D2	D2	D2	D5	B3	G	R	B	G	B	G		B2	R	G	B	G	B	R	B		
LCD_D4	D4	B4	D1	D12	D1	D1	D1	D4	B2								B1				G	B	G	B		
LCD_D3	D3	B3	D0	D02	D0	D0	D0	D3	B1								B0				G	B	G	B		
LCD_D2	D2	B2						D2	B0														G	B		
LCD_D1	D1	B1						D1																		
LCD_D0	D0	B0						D0																		

6.1.3.4 HV Interface (Sync+DE mode)

HV interface is also known as Sync + DE mode, which is widely used in TFT LCD module for PMP/MP4 applications. Its signals are defined as:

Table 6-3 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
DE	LCD data enable	O
D[23: 18]&[15:10]&[7:2]	18-Bit RGB output from input FIFO for panel	O

The timing diagram of HV interface is as follows.

Figure 6-2 HV Interface Vertical Timing

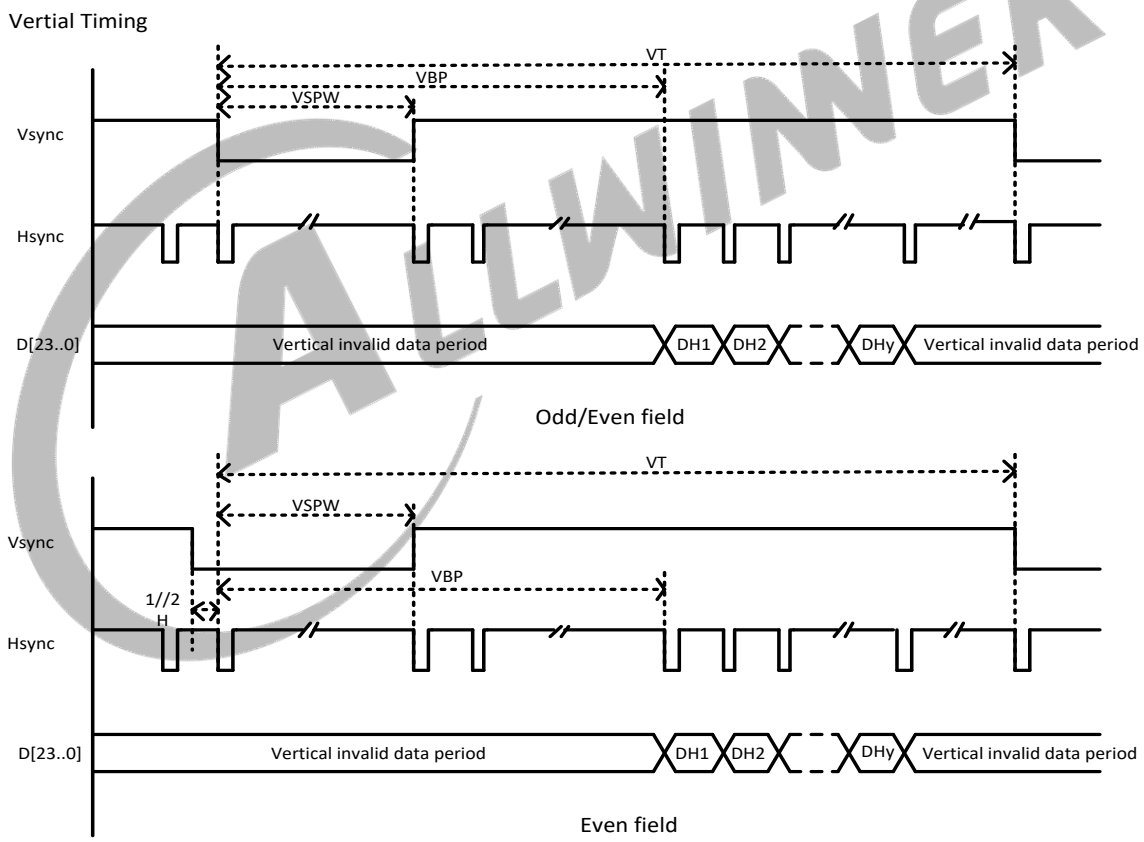
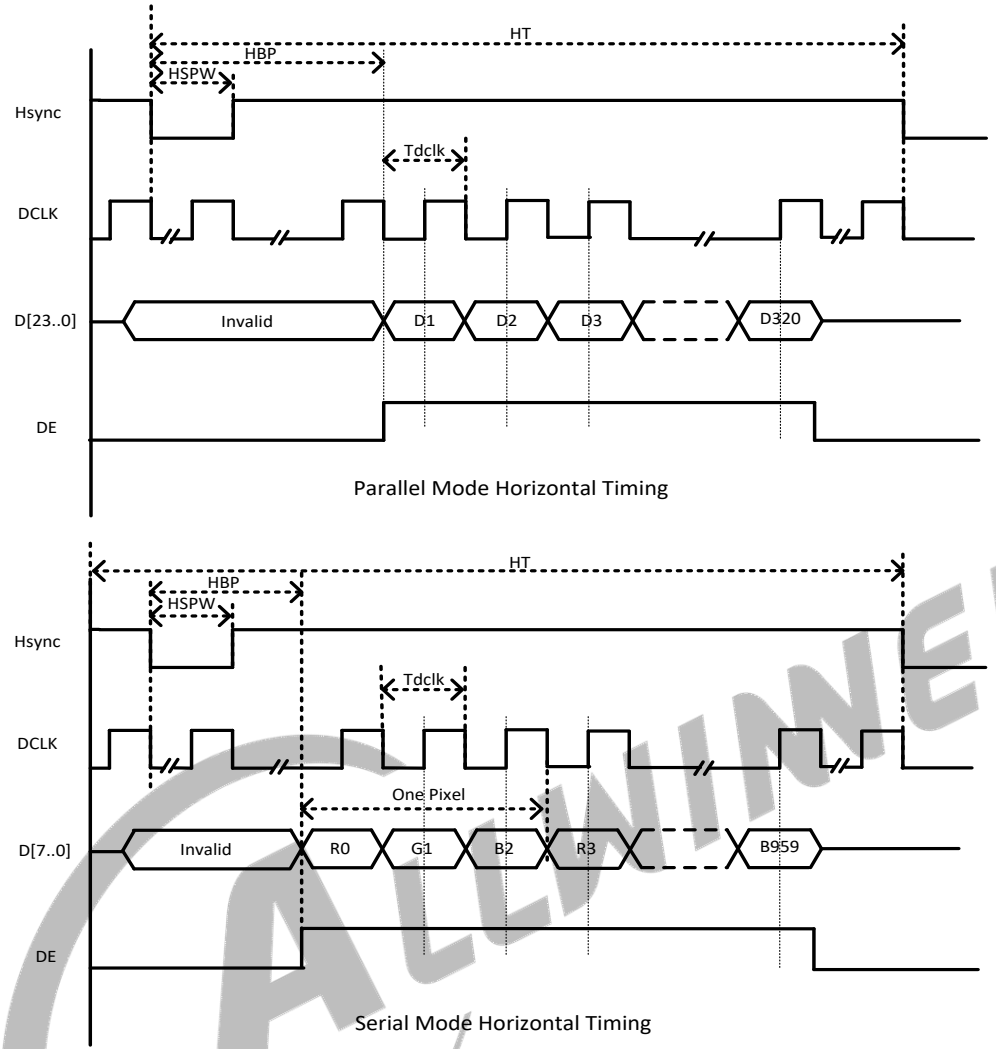


Figure 6-3 HV Interface Horizontal Timing



6.1.3.5 BT656 Interface

In HV serial YUV output mode, its timing is BT656 compatible. SAV adds right before active area every line; EAV adds right after active area every line.

Table 6-4 BT656 Panel Signals

Signal	Description	Type
DCLK	Clock	O
DATA[7: 0]	Data	O

Its logic is:

F = “0” for Field 1 F = “1” for Field 2

V = “1” during vertical blanking

H = “0” at SAV H = “1” at EAV

P3-P0 = protection bits

P3 = V ⊕H
P2 = F ⊕H
P1 = F ⊕V
P0 = F ⊕V ⊕H
Where ⊕represents the exclusive-OR function

The 4 byte SAV/EAV sequences are as follows.

Table 6-5 EAV and SAV Sequence

	8-bit Data								10-bit Data	
	D9 (MSB)	D8	D7	D6	D5	D4	D3	D2	D1	D0
Preamble	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
Status word	1	F	V	H	P3	P2	P1	P0	0	0

6.1.3.6 i8080 Interface

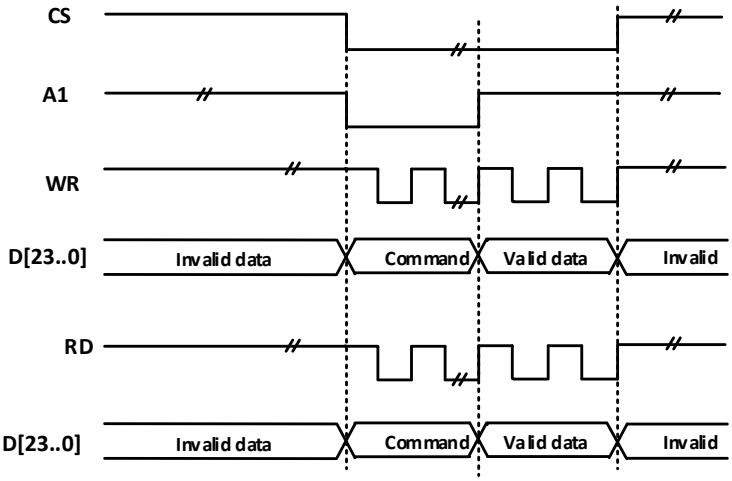
The i8080 I/F LCD panel is most common interface for small size, low resolution LCD panels. The CPU control signals are active low.

Table 6-6 CPU Panel Signals

Signal Name	Description	Type
CS	Chip select, active low	O
WR	Write strobe, active low	O
RD	Read strobe, active low	O
A1	Address bit, controlled by "LCD_CPUI/F" BIT26/25	O
D[23..0]	Digital RGB output signal	I/O

The following figure relationship between basic timing and CPU timing. WR is 180o delay of DCLK; CS is active when pixel data is valid; RD is always set to 1; A1 is set by “LCD_CPUI/F”.

Figure 6-4 i8080 Interface Timing



When CPU I/F is in IDLE state, it can generate WR/RD timing by setting "Lcd_CPUI/F". The CS strobe is one DCLK width, and the WR/RD strobe is half DCLK width.

6.1.3.7 Capture Engine Unit (CEU) Module

This module enhances color data from DE.

R' = Rr*R + Rg*G + Rb*B + Rc

G' = Gr*R + Gg*G + Gb*B + Gc

B' = Br*R + Bg*G + Bb*B + Bc

Rr, Rg, Rb, Gr, Gg, Gb, Br, Bg, Bb s13(-16, 16)

Rc, Gc, Bc s19 (-16384, 16384)

R, G, B u8 [0-255]

R' has the range of [Rmin, Rmax]

G' has the range of [Rmin, Rmax]

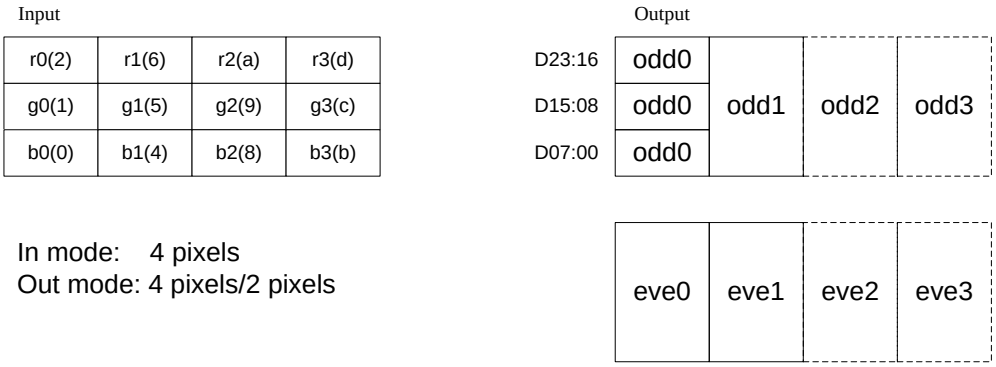
B' has the range of [Rmin, Rmax]

6.1.3.8 Colormap (CMAP) Module

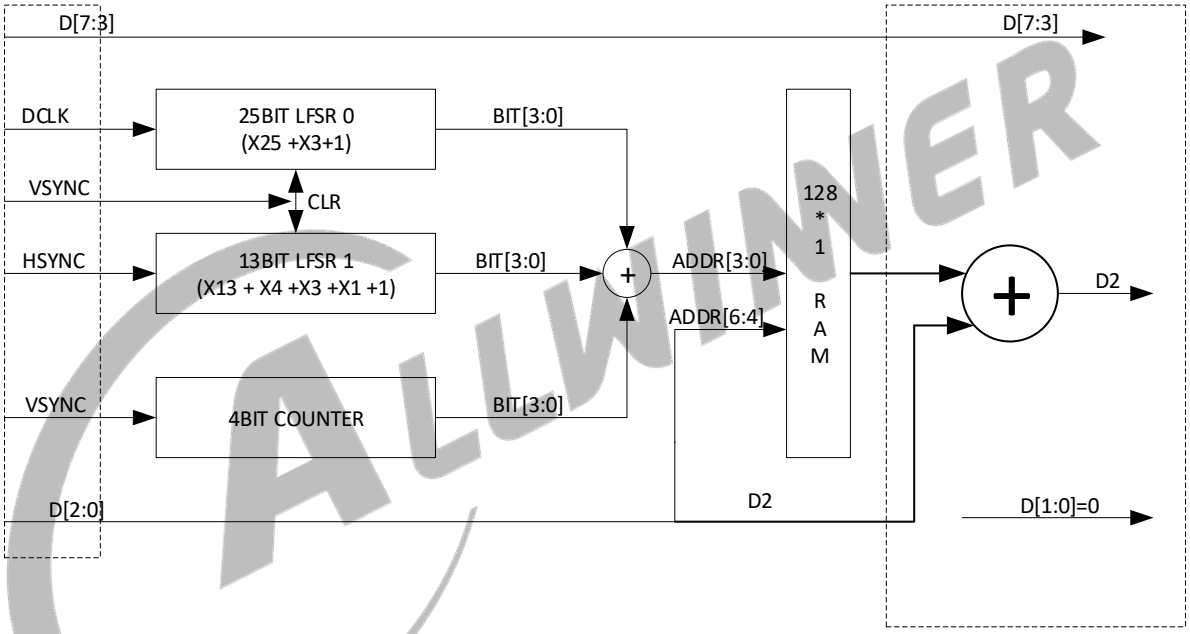
This module is used to map color data from DE.

Every 4 input pixels are a unit. A unit is divided into 12 bytes. Output byte can select one of those 12 bytes. Note that even line and odd line can be different, and output can be 12 bytes (4 pixels) or reduce to 6 bytes (2 pixels).

Figure 6-5 CMAP Module



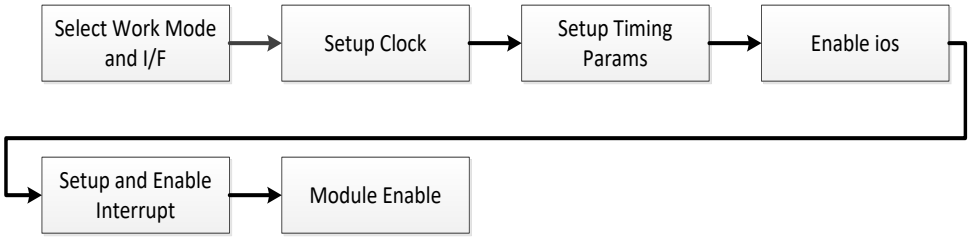
6.1.3.9 Frame Rate Modulation (FRM) Module



6.1.4 Programming Guidelines

6.1.4.1 HV Mode Configuration Process

Figure 6-6 HV Mode Initial Process



1) Parallel RGB

Step 1 Select HV interface type

Configure LCD_CTL_REG[LCD_IF] (reg0x40) to 0 to select HV (Sync+DE) mode, and configure LCD_HV_IF_REG[HV_MODE] (reg0x58) to 0 to select 24bit/1cycle parallel mode.

lcd_dev[sel]->lcd_ctl.lcd_if = HV(Sync+DE);

lcd_dev[sel]->lcd_ctl.src_sel = src; //src = DE/color/grayscale/...

lcd_dev[sel]->lcd_hv_ctl.hv_mode = 24bit/1cycle parallel mode;

Step 2 Clock configuration

NOTE

a. In parallel RGB mode, the displayed pixel clock (pixel_CLK) is required to be consistent with the DCLK, the pixel_clk (pixel_clk=Ht*Vt*frame rate) is decided by external LCD.

b. When using phase adjustment function, the LCD_IO_POL_REG.DCLK_SEL (reg0x88) selects dclk0–2 of different phase, and LCD_IO_POL_REG.IO2_INV can achieve 180°phase delay.

Configure corresponding frequency by setting PLL_VIDEO0/1 register, and configure TCON LCD0 Clock register.

Configure internal frequency division of TCON_LCD. Based on clock source of TCON and DCLK clock ratio, configure LCD_DCLK_REG[LCD_DCLK_DIV]. If using phase adjustment function, LCD_DCLK_REG[LCD_DCLK_EN] needs be set, usually is 0xf. When the dclk1 and dclk2 in LCD_DCLK_REG[LCD_DCLK_EN] are used, the value of LCD_DCLK_REG[LCD_DCLK_DIV] needs no less than 6.

lcd_dev[sel]->lcd_dclk. dclk_en = en;

lcd_dev[sel]->lcd_dclk. dclk_div = div;

Step 3 Set sequence parameters

The sequence parameters include x, ht, hbp, hspw, y, vt, vbp, vspw, and correspond to LCD_BASE_REG from reg0x48 to reg 0x54. Note that hbp includes hspw, and vbp includes vspw. And LCD_BASE2_REG.VT needs be set to the twice of the actual value.

lcd_dev[sel]->lcd_basic0.x = x-1;

lcd_dev[sel]->lcd_basic0.y = y-1;

lcd_dev[sel]->lcd_basic1.ht = ht-1;

lcd_dev[sel]->lcd_basic1.hbp = hbp-1;

lcd_dev[sel]->lcd_basic2.vt = vt*2;

lcd_dev[sel]->lcd_basic2.vbp = vbp-1;

lcd_dev[sel]->lcd_basic3.hspw = hspw-1;

lcd_dev[sel]->lcd_basic3.vspw = vspw-1;

Step 4 Open IO output

Set the corresponding data IO enable and control signal IO enable of LCD_IO_TRI_REG (reg0x8C) to 0 to start enable. Note that except the internal IO of TCON_LCD, the external GPIO mapping needs to be set to LCD mode.

When some control signals require polarity reversal, it can realize by setting LCD_IO_POL_REG.IO0~3_INV (reg0x88).

Step 5 Set and open interrupt function

The LCD_GINT0_REG (reg0x4) controls interrupt mode and flag, and the LCD_GINT1_REG (reg0x8) sets the interrupt line position of Line interrupt mode.

V interrupt:

lcd_dev[sel]->lcd_gint0.vb_en = 1;

Line interrupt:

lcd_dev[sel]->lcd_gint1.lcd_line_int_num = line;

lcd_dev[sel]->lcd_gint0.line_en = 1;

Step 6 Open module enable

Enable LCD_CTL_REG.LCD_EN (reg0x40) and LCD_GCTL_REG.LCD_EN (reg0x00).

lcd_dev[sel]->lcd_ctl.lcd_en = 1;

lcd_dev[sel]->lcd_gctl.lcd_en = 1;

2) Serial RGB

The serial RGB mode is consistent with parallel RGB mode, the main difference is the definition of clock and the sequence of serial data. The difference is as follows.

Step 1 Select HV interface type

Set LCD_CTL_REG.LCD_IF (reg0x40) to 0 to select HV(Sync+DE) mode; set LCD_HV_IF_REG.HV_MODE (reg0x58) to select 8bit/3cycle RGB serial mode (RGB888), 8bit/4cycle Dummy RGB mode (DRGB) or 8bit/4cycle RGB Dummy mode (RGBD).

lcd_dev[sel]->lcd_ctl.lcd_if = HV(Sync+DE);

lcd_dev[sel]->lcd_ctl.src_sel = src; //src = DE/color/grayscale/...

lcd_dev[sel]->lcd_hv_ctl.hv_mode = Serial mode;

Step 2 Set clock and sequence parameters

In serial RGB mode, DCLK is the transfer clock of each byte data. In the same resolution, pixel_clk of serial RGB is three times of its clock in parallel RGB, and ht, hbp, hspw own the same conversion relation. When display is split into odd field and even field, LCD_BASE2_REG.VT needs not to be set to the twice of the actual value.

lcd_dev[sel]->lcd_basic2.vt = vt;

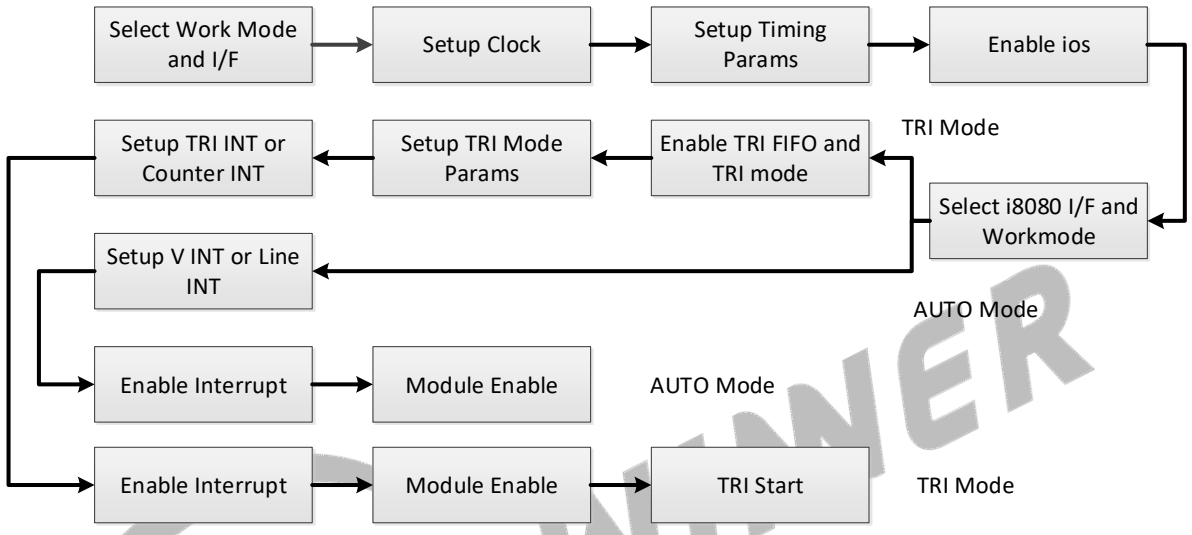
Set LCD_HV_IF_REG.RGB888_ODD_ORDER/LCD_HV_IF_REG.RGB888_ODD_EVEN to select RGB output sequence of the selected odd and even lines.

lcd_dev[lcd_sel]->lcd_hv_ctl.srgb_seq_even = seq_even;

lcd_dev[lcd_sel]->lcd_hv_ctl.srgb_seq_odd = seq_odd;

6.1.4.2 i8080 Mode Configuration Process

Figure 6-7 i8080 Mode Initial Process



Step 1 Select i8080 interface type.

Step 2 The step is the same as HV mode, but pulse adjustment function is invalid.

Step 3 The step is the same as HV mode. When using TRI mode, it is best to configure LCD timing parameters in HV mode, or a handful of functions such as CMAP will not be able to apply.

Step 4 The step is the same as HV mode.

Step 5 Select type and operating mode of i8080, the operating mode includes TRI mode and AUTO mode, and the two operating modes are different.

---------- **For TRI mode**----------

Step 6 Open TRI FIFO switch, and TRI mode function.

Step 7 Set parameters of TRI mode, including block size, block space and block number.

NOTE

a. When output interface is parallel mode, then the setting value of block space parameter is not less than 20.

b. When output interface is 2 cycle serial mode, then the setting value of block space parameter is not less than 40.

c. When output interface is 3 cycle serial mode, then the setting value of block space parameter is not less than 60.

d. When output interface is 4 cycle serial mode, then the setting value of block space parameter is not less than 80.

Step 8 Set the tri interrupt or counter interrupt. When using the two interrupts, mainly in the interrupt service function the tri start operation need be operated (the bit1 of LCD_CPU_IF_REG is set to"1"). If using TE trigger interrupt, you select the external input pin as a trigger signal, the 24-bit for offset 0x8C register is set to "1", to open up input of pad.

Step 9 Open the total switch of interrupt.

Step 10 Open the total enable of interrupt.

Step 11 Operate “tri start” operation (the bit1 of LCD_CPU_IF_REG is set to "1").

---------- **For Auto mode**----------

Step 6 Set and open V interrupt or Line interrupt, the step is the same as HV mode.

Step 7 Open module total enable.

6.1.5 Register List

Module Name	Base Address
TCONLCD0	0x40B41000

Register Name	Offset	Description
LCD_GCTL_REG	0x0000	LCD Global Control Register
LCD_GINT0_REG	0x0004	LCD Global Interrupt Register0
LCD_GINT1_REG	0x0008	LCD Global Interrupt Register1
LCD_FRM_CTL_REG	0x0010	LCD FRM Control Register
LCD_FRM_SEED_REG	0x0014+N*0x0004(N=0~5)	LCD FRM Seed Register
LCD_FRM_TAB_REG	0x002C+N*0x0004(N=0~3)	LCD FRM Table Register
LCD_CTL_REG	0x0040	LCD Control Register
LCD_DCLK_REG	0x0044	LCD Data Clock Register
LCD_BASIC0_REG	0x0048	LCD Basic Timing Register0
LCD_BASIC1_REG	0x004C	LCD Basic Timing Register1
LCD_BASIC2_REG	0x0050	LCD Basic Timing Register2
LCD_BASIC3_REG	0x0054	LCD Basic Timing Register3
LCD_HV_IF_REG	0x0058	LCD Hv Panel Interface Register
LCD_CPU_IF_REG	0x0060	LCD CPU Panel Interface Register
LCD_CPU_WR_REG	0x0064	LCD CPU Panel Write Data Register
LCD_CPU_RD0_REG	0x0068	LCD CPU Panel Read Data Register0
LCD_CPU_RD1_REG	0x006C	LCD CPU Panel Read Data Register1
LCD_IO_POL_REG	0x0088	LCD IO Polarity Register

Register Name	Offset	Description
LCD_IO_TRI_REG	0x008C	LCD IO Control Register
LCD_DEBUG_REG	0x00FC	LCD Debug Register
LCD_CEU_CTL_REG	0x0100	LCD CEU Control Register
LCD_CEU_COEF_MUL_REG0	0x0110+N*0x0004(N=0~2)	LCD CEU Coefficient Register0
LCD_CEU_COEF_MUL_REG1	0x0120+N*0x0004(N=0~2)	LCD CEU Coefficient Register1
LCD_CEU_COEF_MUL_REG2	0x0130+N*0x0004(N=0~2)	LCD CEU Coefficient Register2
LCD_CEU_COEF_ADD_REG	0x011C+N*0x0010(N=0~2)	LCD CEU Coefficient Register3
LCD_CEU_COEF_RANG_REG	0x0140+N*0x0004(N=0~2)	LCD CEU Coefficient Register4
LCD_CPU_TRI0_REG	0x0160	LCD CPU Panel Trigger Register0
LCD_CPU_TRI1_REG	0x0164	LCD CPU Panel Trigger Register1
LCD_CPU_TRI2_REG	0x0168	LCD CPU Panel Trigger Register2
LCD_CPU_TRI3_REG	0x016C	LCD CPU Panel Trigger Register3
LCD_CPU_TRI4_REG	0x0170	LCD CPU Panel Trigger Register4
LCD_CPU_TRI5_REG	0x0174	LCD CPU Panel Trigger Register5
LCD_CMAP_CTL_REG	0x0180	LCD Color Map Control Register
LCD_CMAP_ODD0_REG	0x0190	LCD Color Map Odd Line Register0
LCD_CMAP_ODD1_REG	0x0194	LCD Color Map Odd Line Register1
LCD_CMAP_EVEN0_REG	0x0198	LCD Color Map Even Line Register0
LCD_CMAP_EVEN1_REG	0x019C	LCD Color Map Even Line Register1
LCD_SAFE_PERIOD_REG	0x01F0	LCD Safe Period Register

6.1.6 Register Description

6.1.6.1 0x0000 LCD Global Control Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: LCD_GCTL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	LCD_EN When it is disabled, the module will be reset to idle state. 0: Disable 1: Enable
30: 0	/	/	/

6.1.6.2 0x0004 LCD Global Interrupt Register0 (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: LCD_GINT0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	LCD_VB_INT_EN Enable the Vb interrupt. 0: Disable 1: Enable
30	/	/	/

Offset: 0x0004			Register Name: LCD_GINT0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
29	R/W	0x0	LCD_LINE_INT_EN Enable the line interrupt. 0: Disable 1: Enable
28	/	/	/
27	R/W	0x0	LCD_TRI_FINISH_INT_EN Enable the trigger finish interrupt. 0: Disable 1: Enable
26	R/W	0x0	LCD_TRI_COUNTER_INT_EN Enable the trigger counter interrupt. 0: Disable 1: Enable
25:16	/	/	/
15	R/W0C	0x0	LCD_VB_INT_FLAG It is asserted during vertical no-display period every frame. Write 0 to clear it.
14	/	/	/
13	R/W0C	0x0	LCD_LINE_INT_FLAG Trigger when SY0 match the current LCD scan line. Write 0 to clear it.
12	/	/	/
11	R/W0C	0x0	LCD_TRI_FINISH_INT_FLAG Trigger when cpu trigger mode finish. Write 0 to clear it.
10	R/W0C	0x0	LCD_TRI_COUNTER_INT_FLAG Trigger when tri counter reaches this value Write 0 to clear it.
9:3	/	/	/
2	R/W	0x0	FSYNC_INT_INV Enable the fsync interrupt set signal inverse polarity. When FSYNC is positive, this bit must be 1. And vice versa.
1	R/W	0x0	DE_INT_FLAG Asserted at the first valid line in every frame. Write 0 to clear it.
0	R/W	0x0	FSYNC_INT_FLAG Asserted at the fsync signal in every frame. Write 0 to clear it.

6.1.6.3 0x0008 LCD Global Interrupt Register1 (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: LCD_GINT1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0x0	LCD_LINE_INT_NUM Scan line for LCD line trigger (including inactive lines). Set it for the specified line for trigger0. **Note**: *SY0 is writable only when LINE_TRG0 disable.*
15: 0	/	/	/

6.1.6.4 0x0010 LCD FRM Control Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: LCD_FRM_CTL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	LCD_FRM_EN Enable the dither function. 0: Disable 1: Enable
30:7	/	/	/
6	R/W	0x0	LCD_FRM_MODE_R The R component output bits in dither function. 0: 6bit frm output 1: 5bit frm output
5	R/W	0x0	LCD_FRM_MODE_G The G component output bits in dither function. 0: 6bit frm output 1: 5bit frm output
4	R/W	0x0	LCD_Frm_MODE_B The B component output bits in dither function. 0: 6bit frm output 1: 5bit frm output
3:2	/	/	/
1: 0	R/W	0x0	LCD_FRM_TEST Set the test mode of dither function. 00: FRM 01: Half 5/6bit, half FRM 10: Half 8bit, half FRM 11: Half 8bit, half 5/6bit

6.1.6.5 0x0014+N*0x0004(N=0~5) LCD FRM Seed Register (Default Value: 0x0000_0000)

Offset: 0x0014+N*0x0004(N=0~5)			Register Name: LCD_FRM_SEED_REG
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0014+N*0x0004(N=0~5)			Register Name: LCD_FRM_SEED_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24: 0	R/W	0x0	SEED_VALUE Set the seed used in dither function. N=0: Pixel_Seed_R N=1: Pixel_Seed_G N=2: Pixel_Seed_B N=3: Line_Seed_R N=4: Line_Seed_G N=5: Line_Seed_B **Note**: A*void set it to 0.*

6.1.6.6 0x002C+N*0x0004(N=0~3) LCD FRM Table Register (Default Value: 0x0000_0000)

Offset: 0x002C+N*0x0004(N=0~3)			Register Name: LCD_FRM_TAB_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	FRM_TABLE_VALUE Set the data used in dither function. Usually set as follow: Table 0 = 0x01010000 Table 1 = 0x15151111 Table 2 = 0x57575555 Table 3 = 0x7f7f7777

6.1.6.7 0x0040 LCD Control Register (Default Value: 0x0000_0000)

Offset: 0x0040			Register Name: LCD_CTL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	LCD_EN It executes at the beginning of the first blank line of LCD timing. 0: Disable 1: Enable
30:26	/	/	/
25:24	R/W	0x0	LCD_IF Set the interface type of LCD controller. 00: HV(Sync+DE) 01: 8080 I/F 1x: Reserved
23	R/W	0x0	LCD_RB_SWAP Enable the function to swap red data and blue data in fifo1. 0: Default 1: Swap RED and BLUE data at FIFO1

Offset: 0x0040			Register Name: LCD_CTL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
22	/	/	/
21	R/W	0x0	LCD_FIFO1_RST Writing 1 and 0 in sequence at this bit will reset FIFO 1. **Note**: *1 holding time must more than 1 DCLK*
20	R/W	0x0	LCD_INTERLACE_EN This flag is valid only when LCD_EN == 1 0: Disable 1: Enable
19:9	/	/	/
8:4	R/W	0x0	LCD_START_DLY The unit of delay is T_{line}. **Note**: *valid only when LCD_EN == 1*
3	/	/	/
2: 0	R/W	0x0	LCD_SRC_SEL LCD Source Select 000: DE 001: Color Check 010: Grayscale Check 011: Black by White Check 100: Test Data all 0 101: Test Data all 1 110: Reversed 111: Gridding Check

6.1.6.8 0x0044 LCD Data Clock Register (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: LCD_DCLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0044			Register Name: LCD_DCLK_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	LCD_DCLK_EN LCD clock enable. 0000: dclk_en = 0; dclk1_en = 0; dclk2_en = 0; dclkm2_en = 0; 0001: dclk_en = 1; dclk1_en = 0; dclk2_en = 0; dclkm2_en = 0; 0010: dclk_en = 1; dclk1_en = 0; dclk2_en = 0; dclkm2_en = 1; 0011: dclk_en = 1; dclk1_en = 1; dclk2_en = 0; dclkm2_en = 0; 0101: dclk_en = 1; dclk1_en = 0; dclk2_en = 1; dclkm2_en = 0; 1111: dclk_en = 1; dclk1_en = 1; dclk2_en = 1; dclkm2_en = 1; Others: Reserved
27:7	/	/	/
6: 0	R/W	0x0	LCD_DCLK_DIV Tdclk = Tsclk /DCLKDIV **Note**: *If dclk1&dclk2 used, DCLKDIV >=6* *If dclk only, DCLKDIV >=1*

6.1.6.9 0x0048 LCD Basic Timing Register0 (Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: LCD_BASIC0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0x0	WIDTH_X Panel width is X+1
15:12	/	/	/
11: 0	R/W	0x0	HEIGHT_Y Panel height is Y+1

6.1.6.10 0x004C LCD Basic Timing Register1 (Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: LCD_BASIC1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/

Offset: 0x004C			Register Name: LCD_BASIC1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
28:16	R/W	0x0	HT Thcycle = (HT+1) * Tdclk Computation: 1) parallel:HT = X + BLANK Limitation: 1) parallel: HT >= (HBP +1) + (X+1) +2 2) serial 1: HT >= (HBP +1) + (X+1) *3+2 3) serial 2: HT >= (HBP +1) + (X+1) *3/2+2
15:12	/	/	/
11: 0	R/W	0x0	HBP Horizontal back porch (in dclk) Thbp = (HBP +1) * Tdclk

6.1.6.11 0x0050 LCD Basic Timing Register2 (Default Value: 0x0000_0000)

Offset: 0x0050			Register Name: LCD_BASIC2_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28:16	R/W	0x0	VT TVT = (VT)/2 * Thsync **Note**: *VT/2 >= (VBP+1) + (Y+1) +2*
15:12	/	/	/
11: 0	R/W	0x0	VBP Tvbp = (VBP +1) * Thsync

6.1.6.12 0x0054 LCD Basic Timing Register3 (Default Value: 0x0000_0000)

Offset: 0x0054			Register Name: LCD_BASIC3_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	R/W	0x0	HSPW Thspw = (HSPW+1) * Tdclk **Note**: *HT > (HSPW+1)*
15:10	/	/	/
9: 0	R/W	0x0	VSPW Tvspw = (VSPW+1) * Thsync **Note**: *VT/2 > (VSPW+1)*

6.1.6.13 0x0058 LCD HV Panel Interface Register (Default Value: 0x0000_0000)

Offset: 0x0058			Register Name: LCD_HV_IF_REG
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0058			Register Name: LCD_HV_IF_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	HV_MODE Set the HV mode of LCD controller. 0000: 24bit/1cycle parallel mode 1000: 8bit/3cycle RGB serial mode(RGB888) 1010: 8bit/4cycle Dummy RGB(DRGB) 1011: 8bit/4cycle RGB Dummy(RGBD) 1100: 8bit/2cycle YUV serial mode(CCIR656)
27:26	R/W	0x0	RGB888_ODD_ORDER Serial RGB888 mode Output sequence at odd lines of the panel (line 1, 3, 5, 7…). 00: R→G→B 01: B→R→G 10: G→B→R 11: R→G→B
25:24	R/W	0x0	RGB888_EVEN_ORDER Serial RGB888 mode Output sequence at even lines of the panel (line 2, 4, 6, 8…). 00: R→G→B 01: B→R→G 10: G→B→R 11: R→G→B
23:22	R/W	0x0	YUV_SM Serial YUV mode Output sequence 2-pixel-pair of every scan line. 00: YUYV 01: YVYU 10: UYVY 11: VYUY
21:20	R/W	0x0	YUV_EAV_SAV_F_LINE_DLY Set the delay line mode. 00: F toggle right after active video line 01: Delay 2 line (CCIR PAL) 10: Delay 3 line (CCIR NTSC) 11: Reserved
19	R/W	0x0	CCIR_CSC_DIS Select ‘0’ LCD convert source from RGB to YUV. 0: Enable 1: Disable Only valid when HV mode is “1100”.
18: 0	/	/	/

6.1.6.14 0x0060 LCD CPU Panel Interface Register (Default Value: 0x0000_0000)

Offset: 0x0060			Register Name: LCD_CPU_IF_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:27	/	/	/
26	R/W	0x0	DA Pin A1 value in 8080 mode auto/flash states
25	R/W	0x0	CA Pin A1 value in 8080 mode WR/RD execute
24	/	/	/
23	R	0x0	WR_FLAG The status of write operation. 0: Write operation is finishing 1: Write operation is pending
22	R	0x0	RD_FLAG The status of read operation. 0: Read operation is finishing 1: Read operation is pending
21:18	/	/	/
17	R/W	0x0	AUTO Auto Transfer Mode: If it's 1, all the valid data during this frame are write to panel. **Note**: *This bit is sampled by Vsync.*
16	R/W	0x0	FLUSH Direct transfer mode: If it's enabled, FIFO1 is regardless of the HV timing, pixels data keep being transferred unless the input FIFO was empty. Data output rate control by DCLK.
15:4	/	/	/
3	R/W	0x0	TRI_FIFO_BIST_EN Entry addr is 0xFF8. 0: Disable 1: Enable
2	R/W	0x0	TRI_FIFO_EN Enable the trigger FIFO. 0: Disable 1: Enable
1	R/W	0x0	TRI_START Software must make sure write '1' only when this flag is '0'. Write '1' to start a frame flush, write'0' has no effect. This flag indicated frame flush is running.

Offset: 0x0060			Register Name: LCD_CPU_IF_REG
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	TRI_EN Enable trigger mode. 0: Trigger mode disable 1: Trigger mode enable

6.1.6.15 0x0064 LCD CPU Panel Write Data Register (Default Value: 0x0000_0000)

Offset: 0x0064			Register Name: LCD_CPU_WR_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23: 0	W	0x0	DATA_WR Data write on 8080 bus, launch a write operation on 8080 bus.

6.1.6.16 0x0068 LCD CPU Panel Read Data Register0 (Default Value: 0x0000_0000)

Offset: 0x0068			Register Name: LCD_CPU_RD0_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23: 0	R	0x0	DATA_RD0 Data read on 8080 bus, launch a new read operation on 8080 bus.

6.1.6.17 0x006C LCD CPU Panel Read Data Register1 (Default Value: 0x0000_0000)

Offset: 0x006C			Register Name: LCD_CPU_RD1_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23: 0	R	0x0	DATA_RD1 Data read on 8080 bus, without a new read operation on 8080 bus.

6.1.6.18 0x0088 LCD IO Polarity Register (Default Value: 0x0000_0000)

Offset: 0x0088			Register Name: LCD_IO_POL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	IO_OUTPUT_SEL When set as '1', d[23: 0], io0, io1, io3 sync to dclk. 0: Normal output 1: Register output

Offset: 0x0088			Register Name: LCD_IO_POL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
30:28	R/W	0x0	DCLK_SEL Set the phase offset of clock and data in hv mode. 000: Used DCLK0(normal phase offset) 001: Used DCLK1(1/3 phase offset) 010: Used DCLK2(2/3 phase offset) 101: DCLK0/2 phase 90 100: DCLK0/2 phase 0 Others: Reserved
27	R/W	0x0	IO3_INV Enable invert function of IO3. 0: Not invert 1: Invert
26	R/W	0x0	IO2_INV Enable invert function of IO2. 0: Not invert 1: Invert
25	R/W	0x0	IO1_INV Enable invert function of IO1. 0: Not invert 1: Invert
24	R/W	0x0	IO0_INV Enable invert function of IO0. 0: Not invert 1: Invert
23: 0	R/W	0x0	Data_INV LCD output port D [23: 0] polarity control, with independent bit control. 0: Normal polarity 1: Invert the specify output

6.1.6.19 0x008C LCD IO Control Register (Default Value: 0x0FFF_FFFF)

Offset: 0x008C			Register Name: LCD_IO_TRI_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28	R/W	0x0	RGB_ENDIAN Set the endian of data bits. 0: Normal 1: Bits_invert

Offset: 0x008C			Register Name: LCD_IO_TRI_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
27	R/W	0x1	IO3_OUTPUT_TRI_EN Enable the output of IO3. 1: Disable 0: Enable
26	R/W	0x1	IO2_OUTPUT_TRI_EN Enable the output of IO2. 1: Disable 0: Enable
25	R/W	0x1	IO1_OUTPUR_TRI_EN Enable the output of IO1. 1: Disable 0: Enable
24	R/W	0x1	IO0_OUTPUT_TRI_EN Enable the output of IO0. 1: Disable 0: Enable
23: 0	R/W	0xFFFFFF	DATA_OUTPUT_TRI_EN LCD output port D [23: 0] output enable, with independent bit control. 1: Disable 0: Enable

6.1.6.20 0x00FC LCD Debug Register (Default Value: 0x0000_0000)

Offset: 0x00FC			Register Name: LCD_DEBUG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x0	LCD_FIFO_UNDERFLOW The flag shows whether the fifos in underflow status. 0: Not underflow 1: Underflow
30	/	/	/
29	R	0x1	LCD_FIELD_POL The flag indicates the current field polarity. 0: Second field 1: First field
28	/	/	/
27:16	R	0x0	LCD_CURRENT_LINE The current scan line.
15: 0	/	/	/

6.1.6.21 0x0100 LCD CEU Control Register (Default Value: 0x0000_0000)

Offset: 0x0100			Register Name: LCD_CEU_CTL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CEU_EN Enable CEU function. 0: Bypass 1: Enable
30	R/W	0x0	BT656_F_MASK BT656 F Mask 0: Disable 1: Enable
29	R/W	0x0	BT656_F_MASK_VALUE BT656 F Mask Value 0/1
28: 0	/	/	/

6.1.6.22 0x0110+N*0x0004(N=0~2) LCD CEU Coefficient Register0 (Default Value: 0x0000_0000)

Offset: 0x0110+N*0x0004(N=0~2)			Register Name: LCD_CEU_COEF_MUL_REG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
12: 0	R/W	0x0	CEU_COEF_MUL_VALUE0 Signed 13bit value, range of (-16,16). N=0: Rr N=1: Rg N=2: Rb

6.1.6.23 0x0120+N*0x0004(N=0~2) LCD CEU Coefficient Register1 (Default Value: 0x0000_0000)

Offset: 0x0120+N*0x0004(N=0~2)			Register Name: LCD_CEU_COEF_MUL_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
12: 0	R/W	0x0	CEU_COEF_MUL_VALUE1 Signed 13bit value, range of (-16,16). N=0: Gr N=1: Gg N=2: Gb

6.1.6.24 0x0130+N*0x0004(N=0~2) LCD CEU Coefficient Register2 (Default Value: 0x0000_0000)

Offset: 0x0130+N*0x0004(N=0~2)			Register Name: LCD_CEU_COEF_MUL_REG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/

Offset: 0x0130+N*0x0004(N=0~2)			Register Name: LCD_CEU_COEF_MUL_REG2
Bit	**Read/Write**	**Default/Hex**	**Description**
12: 0	R/W	0x0	CEU_COEF_MUL_VALUE2 Signed 13bit value, range of (-16,16). N=0: Br N=1: Bg N=2: Bb

6.1.6.25 0x011C+N*0x0010(N=0~2) LCD CEU Coefficient Register3 (Default Value: 0x0000_0000)

Offset: 0x011C+N*0x0010(N=0~2)			Register Name: LCD_CEU_COEF_ADD_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:19	/	/	/
18: 0	R/W	0x0	CEU_COEF_ADD_VALUE Signed 19bit value, range of (-16384, 16384). N=0: Rc N=1: Gc N=2: Bc

6.1.6.26 0x0140+N*0x0004(N=0~2) LCD CEU Coefficient Register4 (Default Value: 0x0000_0000)

Offset: 0x0140+N*0x0004(N=0~2)			Register Name: LCD_CEU_COEF_RANG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:16	R/W	0x0	CEU_COEF_RANGE_MIN Unsigned 8bit value, range of [0,255].
15:8	/	/	/
7: 0	R/W	0x0	CEU_COEF_RANGE_MAX Unsigned 8bit value, range of [0,255].

6.1.6.27 0x0160 LCD CPU Panel Trigger Register0 (Default Value: 0x0000_0000)

Offset: 0x0160			Register Name: LCD_CPU_TRI0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0x0	BLOCK_SPACE The spaces between data blocks. It should be set >20*pixel.
15:12	/	/	/
11: 0	R/W	0x0	BLOCK_SIZE The size of data block. It is usually set as X.

6.1.6.28 0x0164 LCD CPU Panel Trigger Register1 (Default Value: 0x0000_0000)

Offset: 0x0164			Register Name: LCD_CPU_TRI1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R	0x0	BLOCK_CURRENT_NUM Shows the current data block transmitting to panel.
15: 0	R/W	0x0	BLOCK_NUM The number of data blocks. It is usually set as Y.

6.1.6.29 0x0168 LCD CPU Panel Trigger Register2 (Default Value: 0x0020_0000)

Offset: 0x0168			Register Name: LCD_CPU_TRI2_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x20	START_DLY T_{dly} = (Start_Delay +1) * be_clk*8.
15	R/W	0x0	TRANS_START_MODE Select the FIFOs used in CPU mode. 0: ECC_FIFO+TRI_FIFO 1: TRI_FIFO
14:13	R/W	0x0	SYNC_MODE Set the sync mode in CPU interface. 0x: Auto 10: 0 11: 1
12: 0	R/W	0x0	TRANS_START_SET Usual set as the length of a line.

6.1.6.30 0x016C LCD CPU Panel Trigger Register3 (Default Value: 0x0000_0000)

Offset: 0x016C			Register Name: LCD_CPU_TRI3_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:28	R/W	0x0	TRI_INT_MODE When set as 01, Tri_Counter_Int occur in cycle of (Count_N+1) × (Count_M+1)×4 dclk. When set as 10 or 11, io0 is map as TE input 00: Disable 01: Counter mode 10: Te rising mode 11: Te falling mode.
27:24	/	/	/
23:8	R/W	0x0	COUNTER_N The value of counter factor.

Offset: 0x016C			Register Name: LCD_CPU_TRI3_REG
Bit	Read/Write	Default/Hex	Description
7: 0	R/W	0x0	COUNTER_M The value of counter factor.

6.1.6.31 0x0170 LCD CPU Panel Trigger Register4 (Default Value: 0x0000_0000)

Offset: 0x0170			Register Name: LCD_CPU_TRI4_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	A1_First_Valid Valid in first Block.
23: 0	R/W	0x0	D23_TO_D0_First_Valid Valid in first Block.

6.1.6.32 0x0174 LCD CPU Panel Trigger Register5 (Default Value: 0x0000_0000)

Offset: 0x0174			Register Name: LCD_CPU_TRI5_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	A1_NON_First_Valid Valid in Block except first.
23: 0	R/W	0x0	D23_TO_D0_NON_First_Valid Valid in Block except first.

6.1.6.33 0x0180 LCD Color Map Control Register (Default Value: 0x0000_0000)

Offset: 0x0180			Register Name: LCD_CMAP_CTL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	COLOR_MAP_EN Enable the color map function. This module only work when X is divided by 4. 0: Bypass 1: Enable
30:1	/	/	/
0	R/W	0x0	OUT_FORMAT Set the pixel output format in color map function. 0: 4 pixel output mode: Out0 -> Out1 -> Out2 -> Out3 1: 2 pixel output mode: Out0 -> Out1

6.1.6.34 0x0190 LCD Color Map Odd Line Register0 (Default Value: 0x0000_0000)

Offset: 0x0190			Register Name: LCD_CMAP_ODD0_REG
Bit	Read/Write	Default/Hex	Description

Offset: 0x0190			Register Name: LCD_CMAP_ODD0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	OUT_ODD1 Indicates the output order of components. bit15-12: Reserved bit11-08: Out_Odd0[23:16] bit07-04: Out_Odd0[15:8] bit03-00: Out_Odd0[7: 0] 0000: in_b0 0001: in_g0 0010: in_r0 0011: Reserved 0100: in_b1 0101: in_g1 0110: in_r1 0111: Reserved 1000: in_b2 1001: in_g2 1010: in_r2 1011: Reserved 1100: in_b3 1101: in_g3 1110: in_r3 1111: Reserved

Offset: 0x0190			Register Name: LCD_CMAP_ODD0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
15: 0	R/W	0x0	OUT_ODD0 Indicates the output order of components. bit15-12: Reserved bit11-08: Out_Odd0[23:16] bit07-04: Out_Odd0[15:8] bit03-00: Out_Odd0[7: 0] 0000: in_b0 0001: in_g0 0010: in_r0 0011: Reserved 0100: in_b1 0101: in_g1 0110: in_r1 0111: Reserved 1000: in_b2 1001: in_g2 1010: in_r2 1011: Reserved 1100: in_b3 1101: in_g3 1110: in_r3 1111: Reserved

6.1.6.35 0x0194 LCD Color Map Odd Line Register1 (Default Value: 0x0000_0000)

Offset: 0x0194			Register Name: LCD_CMAP_ODD1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0194			Register Name: LCD_CMAP_ODD1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	OUT_ODD3 Indicates the output order of components. bit15-12: Reserved bit11-08: Out_Odd0[23:16] bit07-04: Out_Odd0[15:8] bit03-00: Out_Odd0[7: 0] 0000: in_b0 0001: in_g0 0010: in_r0 0011: Reserved 0100: in_b1 0101: in_g1 0110: in_r1 0111: Reserved 1000: in_b2 1001: in_g2 1010: in_r2 1011: Reserved 1100: in_b3 1101: in_g3 1110: in_r3 1111: Reserved

Offset: 0x0194			Register Name: LCD_CMAP_ODD1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
15: 0	R/W	0x0	OUT_ODD2 Indicates the output order of components. bit15-12: Reserved bit11-08: Out_Odd0[23:16] bit07-04: Out_Odd0[15:8] bit03-00: Out_Odd0[7: 0] 0000: in_b0 0001: in_g0 0010: in_r0 0011: Reserved 0100: in_b1 0101: in_g1 0110: in_r1 0111: Reserved 1000: in_b2 1001: in_g2 1010: in_r2 1011: Reserved 1100: in_b3 1101: in_g3 1110: in_r3 1111: Reserved

6.1.6.36 0x0198 LCD Color Map Even Line Register0 (Default Value: 0x0000_0000)

Offset: 0x0198			Register Name: LCD_CMAP_EVEN0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0198			Register Name: LCD_CMAP_EVEN0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	OUT_EVEN1 Indicates the output order of components. bit15-12: Reserved bit11-08: Out_Odd0[23:16] bit07-04: Out_Odd0[15:8] bit03-00: Out_Odd0[7: 0] 0000: in_b0 0001: in_g0 0010: in_r0 0011: Reserved 0100: in_b1 0101: in_g1 0110: in_r1 0111: Reserved 1000: in_b2 1001: in_g2 1010: in_r2 1011: Reserved 1100: in_b3 1101: in_g3 1110: in_r3 1111: Reserved

Offset: 0x0198			Register Name: LCD_CMAP_EVEN0_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
15: 0	R/W	0x0	OUT_EVEN0 Indicates the output order of components. bit15-12: Reserved bit11-08: Out_Odd0[23:16] bit07-04: Out_Odd0[15:8] bit03-00: Out_Odd0[7: 0] 0000: in_b0 0001: in_g0 0010: in_r0 0011: Reserved 0100: in_b1 0101: in_g1 0110: in_r1 0111: Reserved 1000: in_b2 1001: in_g2 1010: in_r2 1011: Reserved 1100: in_b3 1101: in_g3 1110: in_r3 1111: Reserved

6.1.6.37 0x019C LCD Color Map Even Line Register1 (Default Value: 0x0000_0000)

Offset: 0x019C			Register Name: LCD_CMAP_EVEN1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x019C			Register Name: LCD_CMAP_EVEN1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	OUT_EVEN3 Indicates the output order of components. bit15-12: Reserved bit11-08: Out_Odd0[23:16] bit07-04: Out_Odd0[15:8] bit03-00: Out_Odd0[7: 0] 0000: in_b0 0001: in_g0 0010: in_r0 0011: Reserved 0100: in_b1 0101: in_g1 0110: in_r1 0111: Reserved 1000: in_b2 1001: in_g2 1010: in_r2 1011: Reserved 1100: in_b3 1101: in_g3 1110: in_r3 1111: Reserved

Offset: 0x019C			Register Name: LCD_CMAP_EVEN1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
15: 0	R/W	0x0	OUT_EVEN2 Indicates the output order of components. bit15-12: Reserved bit11-08: Out_Odd0[23:16] bit07-04: Out_Odd0[15:8] bit03-00: Out_Odd0[7: 0] 0000: in_b0 0001: in_g0 0010: in_r0 0011: Reserved 0100: in_b1 0101: in_g1 0110: in_r1 0111: Reserved 1000: in_b2 1001: in_g2 1010: in_r2 1011: Reserved 1100: in_b3 1101: in_g3 1110: in_r3 1111: Reserved

6.1.6.38 0x01F0 LCD Safe Period Register (Default Value: 0x0000_0000)

Offset: 0x01F0			Register Name: LCD_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	/	/	/

Offset: 0x01F0			Register Name: LCD_SAFE_PERIOD_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
2: 0	R/W	0x0	SAFE_PERIOD_MODE Select the save mode 000: unsafe 001: safe 011: safe at 2 and safe at sync active 100: safe at line

Contents

Figures

Tables

7 Audio Interfaces

7.1 I2S/PCM

7.1.1 Overview

The I2S/PCM Controller is designed to transfer streaming audio-data between the system memory and the codec chip. The controller supports standard I2S format, left-justified mode format, right-justified mode format, PCM mode format and TDM mode format.

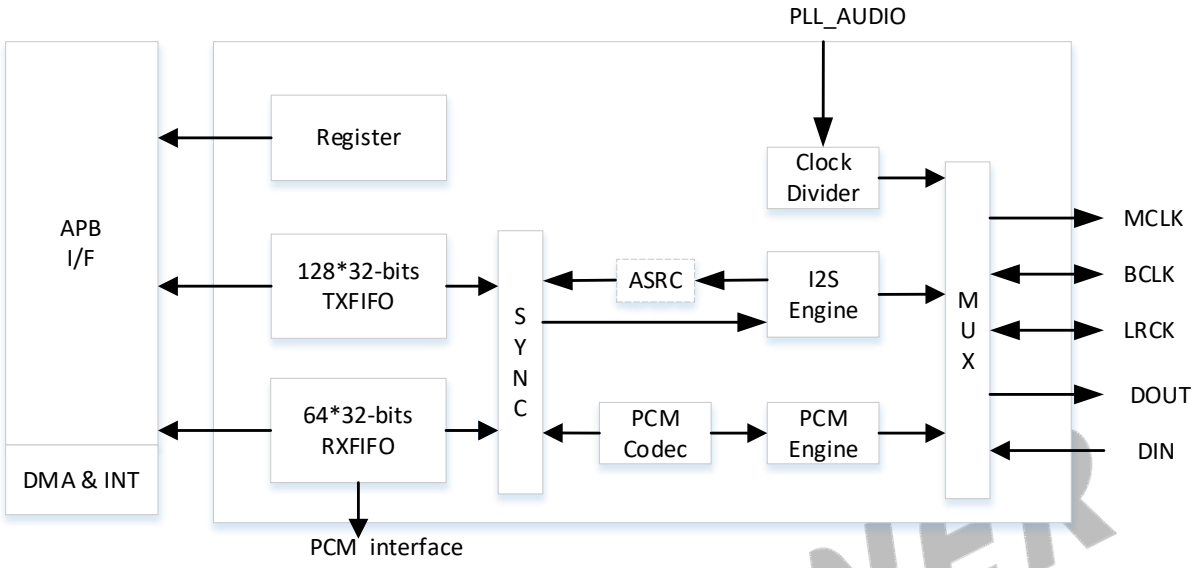
The I2S/PCM controller includes the following features:

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

7.1.2 Block Diagram

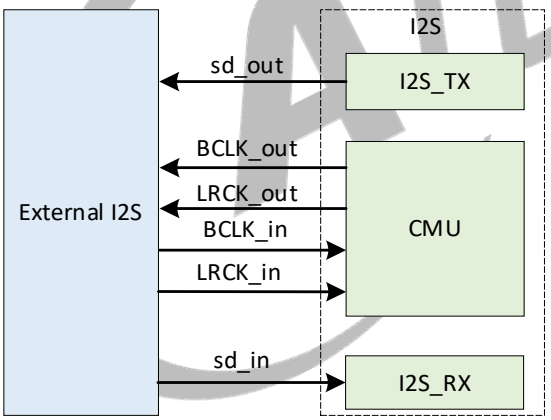
The following Figure shows a block diagram of the I2S/PCM.

Figure 7-1 I2S/PCM System Block Diagram



The following figure shows the typical application of the I2S/PCM interface.

Figure 7-2 Typical Application of I2S/PCM Interface



The I2S/PCM interface system integrates one I2S_TX and one I2S_RX.

- The I2S_TX is for playing music in I2S or PCM format.
- The I2S_RX is for receiving data in I2S or PCM format.
- When the I2S works in the slave mode, the external I2S module provides clocks BCLK_in and LRCK_in for the clock management unit (CMU), and the I2S_TX and I2S_RX work with the two external clocks.
- When the I2S works in the master mode, the CMU provides clocks BCLK_out and LRCK_out for the external I2S module, and the I2S_TX and I2S_RX work with the internal clocks.

7.1.3 Functional Description

7.1.3.1 External Signals

The following table describes the external signals of the I2S/PCM interface.

LRCK and BCLK are bidirectional I/O. When the I2S/PCM interface works in the Master mode, LRCK and BCLK are output pins. When the I2S/PCM interface works in the Slave mode, LRCK and BCLK are input pins.

MCLK is an output pin for external devices. DOUT are the serial data output pins and DIN are the serial data input pins. For details about General Purpose I/O port, refer to section 9.7 "GPIO".

Table 7-1 I2S/PCM External Signals

Signal	Description	Type
I2S_MCLK	I2S/PCM Master Clock	O
I2S_LRCLK	I2S/PCM Sample Rate Clock/Sync	I/O
I2S_BCLK	I2S/PCM Serial Clock	I/O
I2S_DOUT	I2S/PCM Serial Data Output	O
I2S_DIN	I2S/PCM Serial Data Input	I

7.1.3.2 Clock Sources

The following table describes the clock sources for I2S/PCM. For clock setting, configurations, and gating information, refer to the section "CCU" and "CCU_AON".

Table 7-2 I2S/PCM Clock Sources

Clock Name	Description
CCU_AON	APB bus clock. For more details, refer to "CCU" and "CCU_AON".
PLL_AUDIO	24.576MHz or 22.5792MHz generated by PLL_AUDIO to produce 48kHz or 44.1kHz sample frequency.
DPLL	ASRC mclk produced from DPLL.

7.1.3.3 Timing Diagram

The I2S/PCM supports standard I2S mode, Left-justified I2S mode, Right-justified I2S mode, PCM mode, and TDM mode. The software can select the modes by setting I2S/PCM_CTL. The following figures describe the waveforms for SYNC, BCLK, DOUT, and DIN in different modes.

Each sampling period contains an LRCK. The low level of LRCK is the left channel corresponding to the even slots, and the high level is the right channel corresponding to the odd slots. Each slot is the sampling point of a mono channel. The sampling period can support the transmission of 2/4/8/16 slots. The BCLK corresponds to the serial data bit.

Figure 7-3 Standard Mode Timing

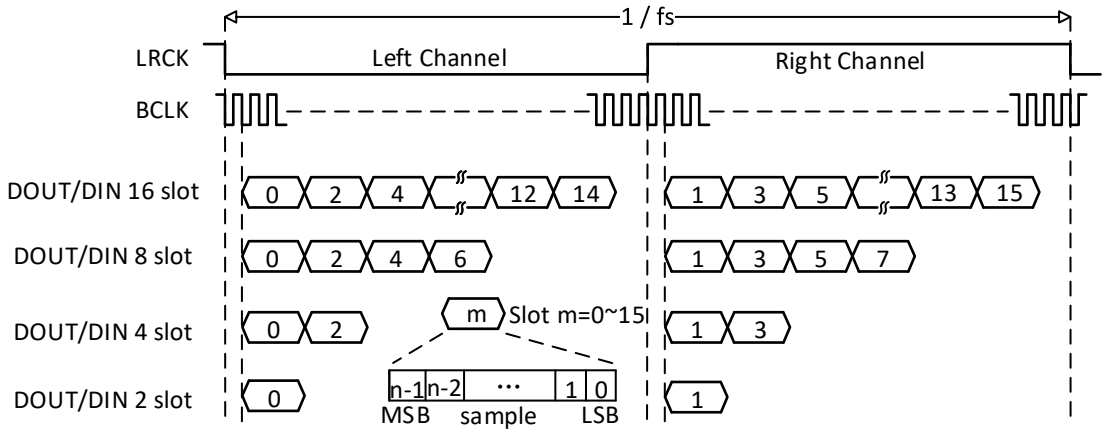


Figure 7-4 Left-Justified Mode Timing

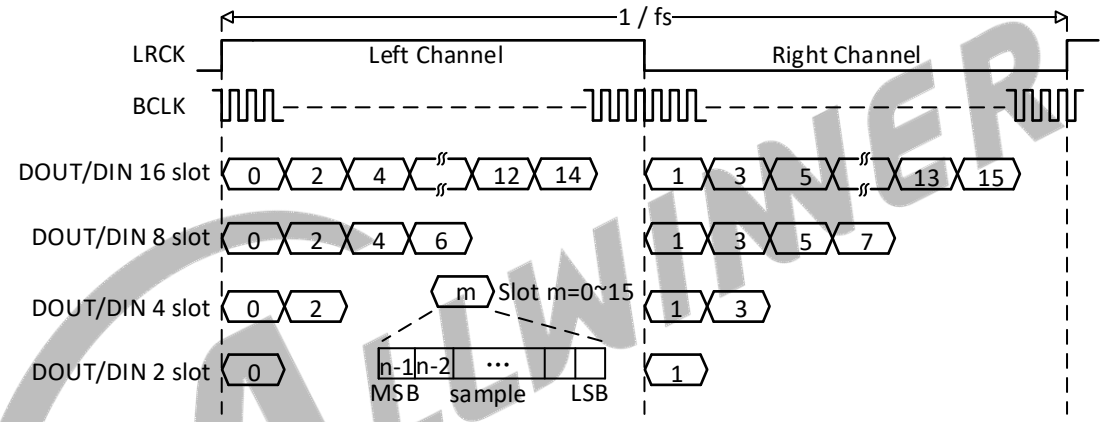


Figure 7-5 Right-Justified Mode Timing

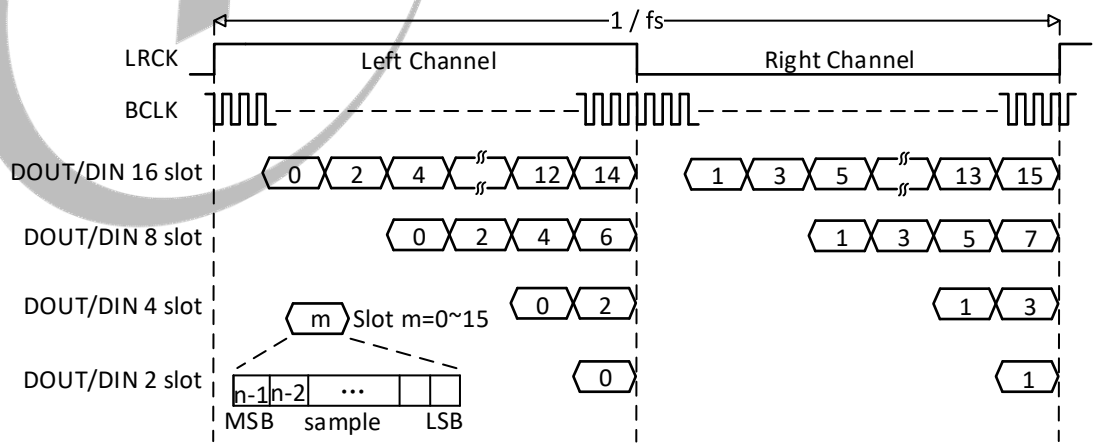


Figure 7-6 PCM Long Frame Mode Timing

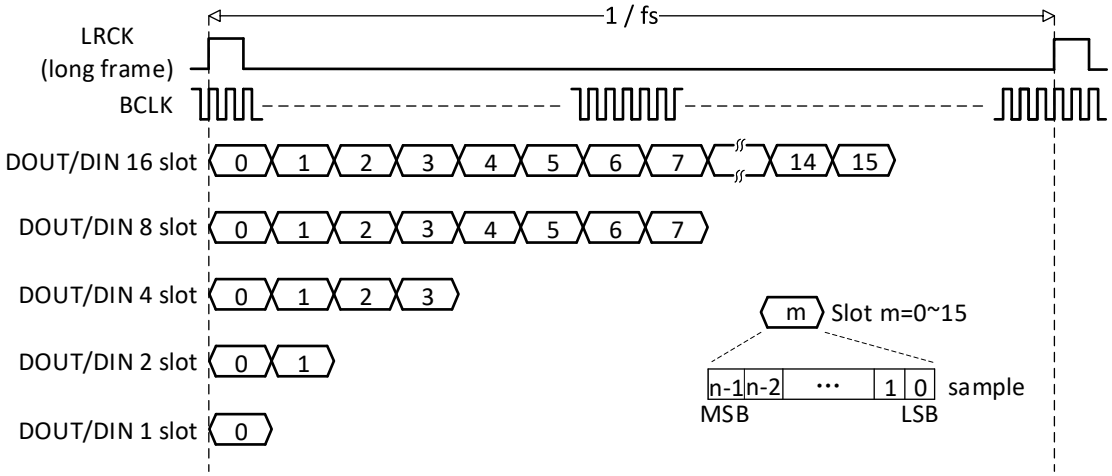
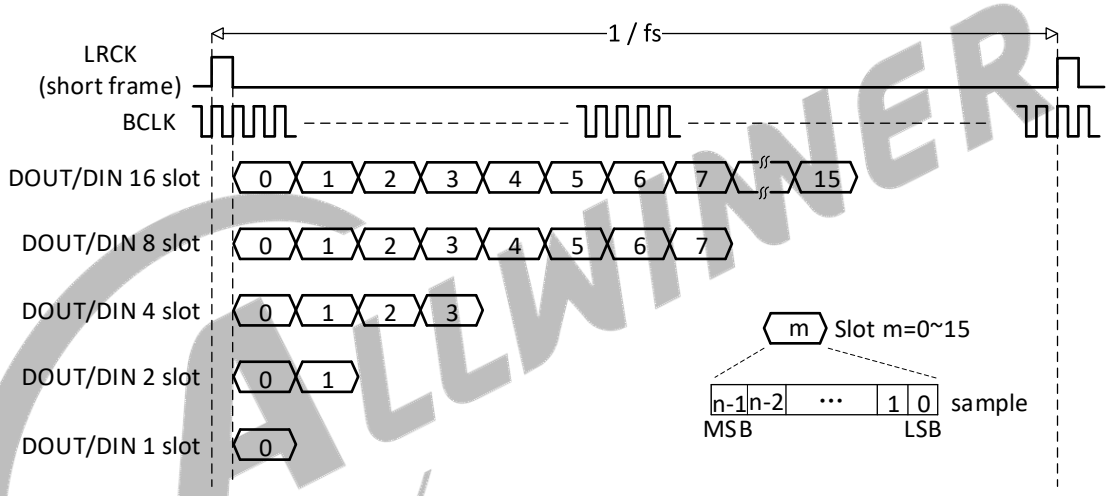


Figure 7-7 PCM Short Frame Mode Timing



7.1.3.4 Asynchronous Sample Rate Coverter (ASRC)

The ASRC module supports sampling rate conversion between the up-sampling and down-sampling. The ASRC also supports sampling rate conversion between the dual-channel audio data, and the size of the sampling data is up to 24 bits.

The ASRC module has the following features:

- Typical THD + N: -130 dB (Range: -125 dB to -139 dB)
- Supportss sampling rate conversion between the up-sampling and down-sampling to implement the sampling rate conversion for stereo data
  - The up-sampling ratio ranges from 1 to 7.5x
  - The down-sampling ratio ranges from 8 to 1x
- Supportss sampling rate conversion between two identical frequencies
- Sampling rate for both the input and output range from 8 kHz to 192 kHz and can be decimal

- Sampling rate can be configured manually or via adaptive generation
- The ASRC input is connected to I2S RX_FIFO_WDATA [31:8], and the input data is 24-bit MSB big-endian. For the input data that is less than 24 bits, use zeros to pad out the values at the low bits instead of high bits
- The ASRC needs some time to calculate the result. The output sample a/b will keep 0 during the calculation, and then change to the valid value when the result comes out.

Calculating the ASRC Latency

Calculate the ASRC up-sampling and down-sampling latency according to the following formulas.

$$\text{Upsampling Latency} = \text{Phase Delay} + \text{FIFO Delay} = 32 + 16 = 48 \text{ Input Sample Periods}$$

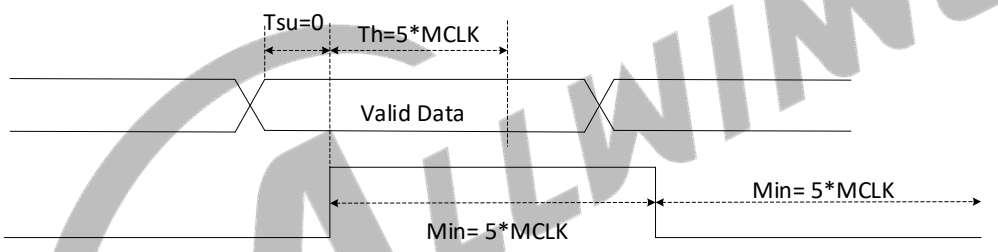
$$\text{Downsampling Latency} = \text{Phase Delay} + \text{FIFO Delay} = (32 * f_{sout}/f_{sin}) + 16 \text{ Input Sample Periods}$$

ASRC Timing

The MCLK samples the input clock CLKIN to generate pulse signals.

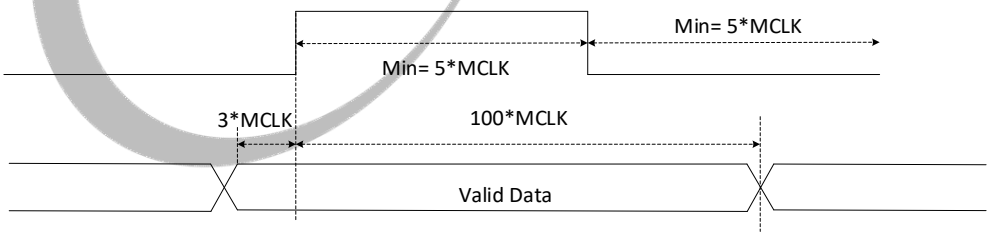
The following figure shows the timing requirements for the inputs.

Figure 7-8 Timing Requirements for Inputs



The following figure shows the timing requirements for the outputs.

Figure 7-9 Timing Requirements for Outputs



For the up-sampling, $F_{MCLK} = F_{sout} * 1350$.

For the down-sampling, $F_{MCLK} = F_{sin} * 0.30 + F_{sout} * 295$.

The following table provides the proper values of MCLK in MHz with different Fsin and Fsout in kHz.

Table 7-3 Proper MCLK Values with Different Fsin and Fsout

Fsin \ Fsout	32	44.1	48	88.2	96	144	192
32	45	60	65	120	130	195	260

44.1	55	60	65	120	130	195	260
48	60	65	65	120	130	195	260
88.2	105	105	110	120	130	195	260
96	110	115	115	125	130	195	260
144	160	165	165	175	180	195	260
192	210	215	215	225	230	245	260

NOTE

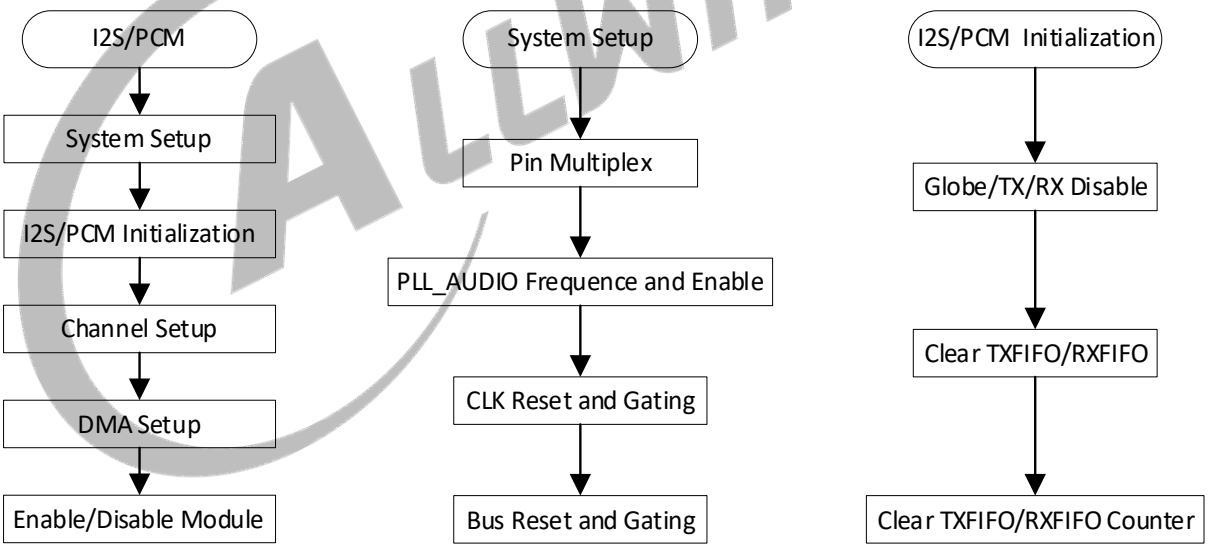
The units for Fsin and Fsout are kHz and MCLK is MHz.

7.1.4 Programming Guidelines

The software operation of the I2S/PCM is divided into five steps: system setup, I2S/PCM initialization, the channel setup, DMA setup, and Enable/Disable module.

The following figure shows the whole operation flow of I2S/PCM.

Figure 7-10 I2S/PCM Operation Flow



7.1.4.1 System Setup and I2S/PCM Initialization

The specific operations of system setup and I2S/PCM Initialization are as follows. (The clock source for the I2S/PCM should be followed.)

Step 1 Disable the AUDIO_PLL though the AUDIO_PLL_CTRL[PLL_ENABLE] in the CCMU. Secondly, set up the frequency of the AUDIO_PLL in the AUDIO_PLL_CTRL.

Step 2 Enable the I2S/PCM gating though the I2S/PCM_CLKD when you checkout that the AUDIO_PLL_CTRL[LOCK] become 1.

Step 3 Reset and enable the I2S/PCM bus gating in the MOD_RST_CTRL1[1].

Step 4 Initialize the I2S/PCM. You should close the globe enable bit(I2S/PCM_CTL[0]), Transmitter Block Enable bit(I2S/PCM_CTL[2]) and Receiver Block enable bit(I2S/PCM_CTL[1]) by writing 0.

Step 5 Clear the TX/RX FIFO by writing 0 to the I2S/PCM_FCTL[25:24].

Step 6 Clear the TX/RX FIFO counter by writing 0 to I2S/PCM_TXCNT and I2S/PCM_RXCNT.

7.1.4.2 Channel Setup and DMA Setup

Step 1 Set up the I2S/PCM of master and slave. The configuration can be referred to the protocol of I2S/PCM.

Step 2 Set up the translation mode, the sample resolution, the wide of the slot, the channel slot number, and the trigger level, and so on through register configuration.

Step 3 The I2S/PCM supports two methods to transfer the data. The most common way is DMA, the setup of DMA can be found in the "DMA". In this module, you just enable the DRQ.

7.1.4.3 Enableing and Disabling I2S/PCM

Step 1 Enable TX/RX by writing I2S/PCM_CTL[TXEN]/ I2S/PCM_CTL[RXEN].

Step 2 Enable I2S/PCM by writing 1 to I2S/PCM_CTL[Globe Enable].

Step 3 Write 0 to the Globe Enable bit to disable I2S/PCM.

7.1.4.4 Application Example of Processing ASRC Input and Output Data

The following example shows a typical application of ASRC: The input data is 24-bit valid, and the output data is a 32-bit data whose highest 24 bits are valid output and the lowest eight bits are padded out with zeros.

To implement the application, configure the sample resolution and slot width as 32 bits. Follow the steps below:

Step 1 For the input register: 0x04 [6:4] sample_res = 3`h7, 0x04 [2: 0] slot_width = 3`h7.
The format of the input data: 32'hXXXXXXXX, where, bit[31] is the MSB and X is the valid data bit.

Step 2 For the output register: 0x04 [6:4] sample_res = 3`h7, 0x04 [2: 0] slot_width = 3`h7
The format of the output data: 32'hXXXXXX00, where, bit[31] is the MSB, X is the valid data bit, and bit[7: 0] are the padded zeros.

7.1.4.5 Converting the Sampling Rate with ASRC

Converting the sampling rate from 48 kHz to 16 kHz is the most common scenario in actual applications. Follow the steps below to convert the sampling rate from 48 kHz to 16 kHz for the 32-bit data.

Step 1 **Configure the AUDIO_PLL Register**

a. Configure AUDIO_PLL_CTRL[31: 0] as 0x8814AB01. That is, AUDIO_PLL = 24*(171+1)/(1+1)/(1+0)/(1+20) = 98.286 MHz. According to the relationship among the Fsin, Fsout, and MCLK, the MCLK should be greater than 60 MHz. In the simulation phase, the HOSC frequency is 25 MHz, so the output frequency of PLL_AUDIO0 should be 25*(171+1)/(1+1)/(1+0)/(1+20) = 102.381 MHz. In the IC test phase, configure the frequency of PLL_AUDIO0 according to its actual output frequency.

b. It is suggested that you configure the ASRC MCLK as an equal-duty-cycle signal. You can specify an odd number for bit[21:16] (PLL_POST_DIV_P) of AUDIO_PLL_CTRL to get an equal-duty-cycle output clock of PLL_AUDIO0.

c. Configure bit[25:24] of I2S_ASRC_CLK_CTRL as 0x00 to select the PLL_AUDIO0(4X).

Step 2 Configure the I2S Registers

a. Configure bit[7:4] (BCLKDIV) of I2S/PCM_CLKD as 4`h9, that is, the frequency of BCLK will be 98.286 MHz/32 = 3.072 MHz.

b. Configure bit[17:8] (LRCK_PERIOD) of I2S/PCM_FMT0 as 10`h1F. That is, the LRCK_PERIOD width is configured as 32 BLCKs and can generate the ASRC CLKIN with a 48 kHz sampling rate.

c. Configure bit[6:4] (Sample Resolution bits) of I2S/PCM_FMT0 as 3`h7 to specify the sample resolution as 32-bit.

d. Configure bit[2: 0] (Slot Width bits) of I2S/PCM_FMT0 as 3`h7 to specify the slot width as 32-bit.

Step 3 Configure the ASRC Registers

a. Configure bit[16] (clock gate) of MCLKCFG as 1`h1 to open the clock gating.

b. Configure bit[3: 0] (division factor) of MCLKCFG as 1`h1 to specify the division factor as 1.

c. Configure bit[20] (clock gate) of FSOUTCFG as 1`h1 to open the clock gating.

d. Configure bit[19:16] (clock select) of FSOUTCFG as 4`h0 to select I2S0_ASRC_CLK as the clock source.

e. Configure bit[7:4] (the first division factor) of FSOUTCFG as 16`h13 to configure the first division factor as 128.

f. Configure bit[3: 0] (the second division factor) of FSOUTCFG as 16`h10 to configure the second division factor as 48.

g. Configure the ASRC ratio.

To configure the ASRC ratio manually, configure the ASRCMANCFG[31] as 1`h1 to enable the manual configuration of ASRC ratio. Configure bit[25: 0] of ASRCMANCFG as 26`h155555 to specify the ratio value as 0x155555. The calculation formula for the ratio value: Dec2Hex (Fsout/Fsin)*222). In this example, Fsout/Fsin = 16 kHz/48 kHz =1/3, then the ratio is 0x155555.

To configure the ASRC ratio automatically, configure the ASRCMANCFG[31] as 1`h0 to enable the automatic configuration of ASRC ratio. Then the system will automatically calculate the ratio value based on the MCLK, Fsout, and Fsin.

7.1.5 Register List

Module Name	Base Address
I2S	0x40045800

Register Name	Offset	Description
I2S/PCM_CTL	0x0000	I2S/PCM Control Register
I2S/PCM_FMT0	0x0004	I2S/PCM Format Register 0
I2S/PCM_FMT1	0x0008	I2S/PCM Format Register 1
I2S/PCM_ISTA	0x000C	I2S/PCM Interrupt Status Register
I2S/PCM_RXFIFO	0x0010	I2S/PCM RXFIFO Register
I2S/PCM_FCTL	0x0014	I2S/PCM FIFO Control Register
I2S/PCM_FSTA	0x0018	I2S/PCM FIFO Status Register
I2S/PCM_INT	0x001C	I2S/PCM DMA And Interrupt Control Register
I2S/PCM_TXFIFO	0x0020	I2S/PCM TXFIFO Register
I2S/PCM_CLKD	0x0024	I2S/PCM Clock Divide Register
I2S/PCM_TXCNT	0x0028	I2S/PCM TX Sample Counter Register
I2S/PCM_RXCNT	0x002C	I2S/PCM RX Sample Counter Register
I2S/PCM_CHCFG	0x0030	I2S/PCM Channel Configuration Register
I2S/PCM_TX0CHSEL	0x0034	I2S/PCM TX0 Channel Select Register
I2S/PCM_TX0CHMAP0	0x0044	I2S/PCM TX0 Channel Mapping Register0
I2S/PCM_TX0CHMAP1	0x0048	I2S/PCM TX0 Channel Mapping Register1
I2S/PCM_RXCHSEL	0x0064	I2S/PCM RX Channel Select Register
I2S/PCM_RXCHMAP0	0x0068	I2S/PCM RX Channel Mapping Register0
I2S/PCM_RXCHMAP1	0x006C	I2S/PCM RX Channel Mapping Register1
I2S/PCM_RXCHMAP2	0x0070	I2S/PCM RX Channel Mapping Register2
I2S/PCM_RXCHMAP3	0x0074	I2S/PCM RX Channel Mapping Register3
I2S/PCM_DBG	0x0078	I2S/PCM DBG Register
MCLKCFG	0x0080	I2S/PCM ASRC MCLK Configure Register
FsoutCFG	0x0084	I2S/PCM ASRC Out Sample Configure Register
FsinEXTCFG	0x0088	I2S/PCM In Sample Pulse Extend Configure Register
ASRCEN	0x008C	I2S/PCM ASRC Enable Configure Register
ASRCMANCFG	0x0090	I2S/PCM ASRC Manual Configure Register
ASRCRATIOSTAT	0x0094	I2S/PCM ASRC Ratio State Configure Register
ASRCFIFOSTAT	0x0098	I2S/PCM ASRC FIFO State Configure Register
ASRCMBISTCFG	0x009C	I2S/PCM ASRC MBIST Test Configure Register
ASRCMBISTSTA	0x00A0	I2S/PCM ASRC MBIST Test State Configure Register

7.1.6 Register Description

7.1.6.1 0x0000 I2S/PCM Control Register (Default Value: 0x0006_0000)

Offset: 0x0000			Register Name: I2S/PCM_CTL
Bit	Read/Write	Default/Hex	Description
31:22	/	/	/

Offset: 0x0000			Register Name: I2S/PCM_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
21	R/W	0x0	RX_SYNC_EN_START Only if RX_SYNC_EN set 1, RX_SYNC_EN_START can take effect. 0: Disable 1: Enable
20	R/W	0x0	RX_SYNC_EN I2S RX Synchronize Enable 0: Disable 1: Enable
19	/	/	/
18	R/W	0x1	BCLK_OUT Bit Clock Direction Select 0: Input 1: Output
17	R/W	0x1	LRCK_OUT LR Clock Direction Select 0: Input 1: Output
16:9	/	/	/
8	R/W	0x0	DOUT0_EN Data0 Output Enable 0: Disable, Hi-Z State 1: Enable
7	R/W	0x0	mad_data_en 0: Disable 1: Enable
6	R/W	0x0	DOUT_MUTE_EN Data Output Mute Enable 0: Normal Transfer 1: Force DOUT to Output 0
5:4	R/W	0x0	MODE_SEL Mode Selection 0: PCM Mode (offset 0: Long Frame; offset 1: Short Frame) 1: Left Mode (offset 0: LJ Mode; offset 1: I2S Mode) 2: Right-Justified Mode 3: Reserved
3	R/W	0x0	LOOP Loop Back Test 0: Normal Mode 1: Loop Back Test When Set '1', Connecting the DOUT with the DIN.

Offset: 0x0000			Register Name: I2S/PCM_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
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

7.1.6.2 0x0004 I2S/PCM Format Register 0 (Default Value: 0x0000_0033)

Offset: 0x0004			Register Name: I2S/PCM_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	/	/	/

Offset: 0x0004			Register Name: I2S/PCM_FMT0
Bit	**Read/Write**	**Default/Hex**	**Description**
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). N+1 For example: N = 7: 8 BCLKs width … N = 1023: 1024 BCLKs width
7	R/W	0x0	BCLK_POLARITY BCLK Polarity Select 0: Normal Mode. DOUT Drive Data at Negative Edge 1: Invert Mode. DOUT Drive Data at Positive Edge
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 Edge Transfer 0: DOUT Drive Data and DIN Sample Data at the Different BCLK Edge 1: DOUT Drive Data and DIN Sample Data at the Same BCLK Edge BCLK_POLARITY = 0, EDGE_TRANSFER = 0, DIN Sample Data at Positive Edge; BCLK_POLARITY = 0, EDGE_TRANSFER = 1, DIN Sample Data at Negative Edge; BCLK_POLARITY = 1, EDGE_TRANSFER = 0, DIN Sample Data at Negative Edge; BCLK_POLARITY = 1, EDGE_TRANSFER = 1, DIN Sample Data at Positive Edge.

Offset: 0x0004			Register Name: I2S/PCM_FMT0
Bit	**Read/Write**	**Default/Hex**	**Description**
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

7.1.6.3 0x0008 I2S/PCM Format Register 1 (Default Value: 0x0000_0030)

Offset: 0x0008			Register Name: I2S/PCM_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 < Slot 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-bits u-law 11: 8-bits A-law
1: 0	R/W	0x0	TX_PDM PCM Data Mode 00: Linear PCM 01: Reserved 10: 8-bits u-law 11: 8-bits A-law

7.1.6.4 0x000C I2S/PCM Interrupt Status Register (Default Value: 0x0000_0010)

Offset: 0x000C			Register Name: I2S/PCM_ISTA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6	R/W1C	0x0	TXU_INT TXFIFO Underrun Pending Interrupt 0: No Pending Interrupt 1: TXFIFO Underrun Pending Interrupt Write '1' to clear this interrupt.
5	R/W1C	0x0	TXO_INT TXFIFO Overrun Pending Interrupt 0: No Pending Interrupt 1: TXFIFO Overrun Pending Interrupt Write '1' to clear this interrupt.
4	R	0x1	TXE_INT TXFIFO Empty Pending Interrupt 0: No Pending IRQ 1: TXFIFO Empty Pending Interrupt When Data in TXFIFO are Less than TX Trigger Level
3	/	/	/
2	R/W1C	0x0	RXU_INT RXFIFO Underrun Pending Interrupt 0: No Pending Interrupt 1: RXFIFO Underrun Pending Interrupt Write '1' to clear this interrupt.
1	R/W1C	0x0	RXO_INT RXFIFO Overrun Pending Interrupt 0: No Pending IRQ 1: RXFIFO Overrun Pending IRQ Write '1' to clear this interrupt.
0	R	0x0	RXA_INT RXFIFO Data Available Pending Interrupt 0: No Pending IRQ 1: Data Available Pending IRQ When Data in RXFIFO are More than RX Trigger Level

7.1.6.5 0x0010 I2S/PCM RXFIFO Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: I2S/PCM_RXFIFO
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0010			Register Name: I2S/PCM_RXFIFO
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	RX_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.

7.1.6.6 0x0014 I2S/PCM FIFO Control Register (Default Value: 0x0004_00F0)

Offset: 0x0014			Register Name: I2S/PCM_FCTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	HUB_EN Audio Hub Enable Only if TXEN set 1, HUB_EN can take effect. 0: Disable 1: Enable
30:26	/	/	/
25	R/WAC	0x0	FTX Flush TX FIFO Write '1' to flush TXFIFO, self clear to '0'.
24	R/WAC	0x0	FRX Flush RX FIFO Write '1' to flush RXFIFO, self clear to '0'.
23:19	/	/	/
18:12	R/W	0x40	TXTL TXFIFO Empty Trigger Level Interrupt and DMA request trigger level for TXFIFO normal condition. Trigger Level = TXTL
11:10	/	/	/
9:4	R/W	0xF	RXTL RXFIFO Trigger Level Interrupt and DMA request trigger level for RXFIFO normal condition. Trigger Level = RXTL + 1
3	/	/	/
2	R/W	0x0	TXIM TXFIFO 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-bits Transmitted Audio Sample: Mode 0: TXFIFO[31: 0] = {APB_WDATA[31:12], 12'h0} Mode 1: TXFIFO[31: 0] = {APB_WDATA[19: 0], 12'h0}

Offset: 0x0014			Register Name: I2S/PCM_FCTL
Bit	**Read/Write**	**Default/Hex**	**Description**
1: 0	R/W	0x0	RXOM RXFIFO Output Mode (Mode 0, 1, 2, 3) 00: Expanding '0' at LSB of RXFIFO Register 01: Expanding Received Sample Sign Bit at MSB of RXFIFO Register 10: Truncating Received Samples at High Half-word of RXFIFO Register and Low Half-word of RXFIFO Register is Filled by '0' 11: Truncating Received Samples at Low Half-word of RXFIFO Register and High Half-word of RXFIFO Register is Expanded by Its Sign Bit Example for 20-bits Received Audio Sample: Mode 0: APB_RDATA[31: 0] = {RXFIFO[31:12], 12'h0} Mode 1: APB_RDATA[31: 0] = {12{RXFIFO[31]}, RXFIFO[31:12]} Mode 2: APB_RDATA [31: 0] = {RXFIFO[31:16], 16'h0} Mode 3: APB_RDATA[31: 0] = {16{RXFIFO[31], RXFIFO[31:16]}

7.1.6.7 0x0018 I2S/PCM FIFO Status Register (Default Value: 0x1080_0080)

Offset: 0x0018			Register Name: I2S/PCM_FSTA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28	R	0x1	TXE TXFIFO Empty 0: No Room for New Sample in TXFIFO 1: More than One Room for New Sample in TXFIFO (>= 1 Word)
27:24	/	/	/
23:16	R	0x80	TXE_CNT TXFIFO Empty Space Word Counter
15:9	/	/	/
8	R	0x0	RXA RXFIFO Available 0: No Available Data in RXFIFO 1: More than One Sample in RXFIFO (>= 1 Word)
7	R	0x1	mad_data_align 0: misalign 1: Align When the mad_data output is changed to apb output, the software will send the order of apb reading data in the case of mad_data_align = 1.
6: 0	R	0x0	RXA_CNT RXFIFO Available Sample Word Counter

7.1.6.8 0x001C I2S/PCM DMA and Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: I2S/PCM_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 When set to '1', an interrupt happens when writing new audio data if TXFIFO is full.
4	R/W	0x0	TXEI_EN TXFIFO Empty Interrupt Enable 0: Disable 1: Enable
3	R/W	0x0	RX_DRQ RXFIFO 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	RXUI_EN RXFIFO Underrun Interrupt Enable 0: Disable 1: Enable
1	R/W	0x0	RXOI_EN RXFIFO Overrun Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	RXAI_EN RXFIFO Data Available Interrupt Enable 0: Disable 1: Enable

7.1.6.9 -0x0020 I2S/PCM TXFIFO Register (Default Value: 0x0000_0000)

Offset: 0x0020	Register Name: I2S/PCM_TXFIFO

Bit	Read/Write	Default/Hex	Description
31: 0	W	0x0	TX_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.

7.1.6.10 0x0024 I2S/PCM Clock Divide Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: I2S/PCM_CLKD
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8	R/W	0x0	MCLKO_EN MCLK Out Enable 0: Disable MCLK Output 1: Enable MCLK Output **Note**: *Whether in Slave or Master mode, when this bit is set to '1', MCLK should be 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

Offset: 0x0024			Register Name: I2S/PCM_CLKD
Bit	**Read/Write**	**Default/Hex**	**Description**
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

7.1.6.11 0x0028 I2S/PCM TX Sample Counter Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: I2S/PCM_TXCNT
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 value at any time. After been updated by the initial value, the counter register should count on base of this initial value.

7.1.6.12 0x002C I2S/PCM RX Sample Counter Register (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: I2S/PCM_RXCNT
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x002C			Register Name: I2S/PCM_RXCNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	RX_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 value at any time. After been updated by the initial value, the counter register should count on base of this initial value.

7.1.6.13 0x0030 I2S/PCM Channel Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: I2S/PCM_CHCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9	R/W	0x0	TX_SLOT_HIZ TX Slot Value in Half Cycle of BCLK 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 The state of transmission line When there is No Transmission 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_SLOT_NUM RX Channel/Slot Number Which between CPU/DMA and RXFIFO 0: 1 Channel or Slot ... 7: 8 Channels or Slots 8: 9 Channels or Slots ... 15:16 Channels or Slots
3: 0	R/W	0x0	TX_SLOT_NUM TX Channel/Slot Number Which between CPU/DMA and TXFIFO 0: 1 Channel or Slot ... 7: 8 Channels or Slots 8: 9 Channels or Slots ... 15:16 Channels or Slots

7.1.6.14 0x0034 I2S/PCM TX0 Channel Select Register (Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: I2S/PCM_TX0CHSEL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21:20	R/W	0x0	TX0_OFFSET TX0 offset Tune, TX0 Data offset to LRCK 0: No offset n: Data is offset by n BCLKs to LRCK
19:16	R/W	0x0	TX0_CHSEL TX0 Channel (Slot) Number Select for each Output 0: 1 Channel / Slot … 7: 8 Channels / Slots 8: 9 Channels / Slots … 15: 16 Channels / Slots
15: 0	R/W	0x0	TX0_CHEN TX0 Channel (Slot) Enable, bit[15: 0] Refer to Slot [15: 0]. When One or More Slot(s) is(are) Disabled, the Affected Slot(s) is(are) Set to Disable State. 0: Disable 1: Enable

7.1.6.15 0x0044 I2S/PCM TX0 Channel Mapping Register0 (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: I2S/PCM_TX0CHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	TX0_CH15_MAP TX0 Channel 15 Mapping 0: 1st Sample … 7: 8th Sample 8: 9th Sample … 15: 16th Sample
27:24	R/W	0x0	TX0_CH14_MAP TX0 Channel 14 Mapping 0: 1st Sample … 7: 8th Sample 8: 9th Sample … 15: 16th sample

Offset: 0x0044			Register Name: I2S/PCM_TX0CHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
23:20	R/W	0x0	TX0_CH13_MAP TX0 Channel 13 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample
19:16	R/W	0x0	TX0_CH12_MAP TX0 Channel 12 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample
15:12	R/W	0x0	TX0_CH11_MAP TX0 Channel 11 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th sample
11:8	R/W	0x0	TX0_CH10_MAP TX0 Channel 10 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample
7:4	R/W	0x0	TX0_CH9_MAP TX0 Channel 9 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample

Offset: 0x0044			Register Name: I2S/PCM_TX0CHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
3: 0	R/W	0x0	TX0_CH8_MAP TX0 Channel 8 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample

7.1.6.16 0x0048 I2S/PCM TX0 Channel Mapping Register1 (Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: I2S/PCM_TX0CHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	TX0_CH7_MAP TX0 Channel 7 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample
27:24	R/W	0x0	TX0_CH6_MAP TX0 Channel 6 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample
23:20	R/W	0x0	TX0_CH5_MAP TX0 Channel 5 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample

Offset: 0x0048			Register Name: I2S/PCM_TX0CHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
19:16	R/W	0x0	TX0_CH4_MAP TX0 Channel 4 Mapping 0: 1st Sample … 7: 8th Sample 8: 9th Sample … 15: 16th Sample
15:12	R/W	0x0	TX0_CH3_MAP TX0 Channel 3 Mapping 0: 1st Sample … 7: 8th Sample 8: 9th Sample … 15: 16th Sample
11:8	R/W	0x0	TX0_CH2_MAP TX0 Channel 2 Mapping 0: 1st Sample … 7: 8th Sample 8: 9th Sample … 15: 16th Sample
7:4	R/W	0x0	TX0_CH1_MAP TX0 Channel 1 Mapping 0: 1st Sample … 7: 8th Sample 8: 9th Sample … 15: 16th Sample
3: 0	R/W	0x0	TX0_CH0_MAP TX0 Channel 0 Mapping 0: 1st Sample … 7: 8th Sample 8: 9th Sample … 15: 16th Sample

7.1.6.17 0x0064 I2S/PCM RX Channel Select Register (Default Value: 0x0000_0000)

Offset: 0x0064			Register Name: I2S/PCM_RXCHSEL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21:20	R/W	0x0	RX_OFFSET RX offset Tune, RX Data offset to LRCK 0: No offset n: Data is offset by n BCLKs to LRCK
19:16	R/W	0x0	RX_CHSEL RX Channel (Slot) Number Select for Input 0: 1 Channel / Slot … 7: 8 Channels / Slots 8: 9 Channels / Slots ... 15: 16 Channels / Slots
15: 0	/	/	/

7.1.6.18 0x0068 I2S/PCM RX Channel Mapping Register0 (Default Value: 0x0000_0000)

Offset: 0x0068			Register Name: I2S/PCM_RXCHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:28	R/W	0x0	RX_CH15_SELECT Rx channel 15 select 00: SDI0
27:24	R/W	0x0	RX_CH15_MAP RX Channel 15 Mapping 0: 1st Sample … 7: 8th Sample 8: 9th Sample ... 15: 16th Sample
23:22	/	/	/
21:20	R/W	0x0	RX_CH14_SELECT Rx channel 14 select 00: SDI0

Offset: 0x0068			Register Name: I2S/PCM_RXCHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
19:16	R/W	0x0	RX_CH14_MAP RX Channel 14 Mapping 0: 1st Sample … 7: 8th Sample 8: 9th Sample … 15: 16th Sample
15:14	/	/	/
13:12	R/W	0x0	RX_CH13_SELECT Rx channel 13 select 00: SDI0
11:8	R/W	0x0	RX_CH13_MAP RX Channel 13 Mapping 0: 1st Sample … 7: 8th Sample 8: 9th Sample … 15: 16th Sample
7:6	/	/	/
5:4	R/W	0x0	RX_CH12_SELECT Rx channel 12 select 00: SDI0
3: 0	R/W	0x0	RX_CH12_MAP RX Channel 12 Mapping 0: 1st Sample … 7: 8th Sample 8: 9th Sample … 15: 16th Sample

7.1.6.19 0x006C I2S/PCM RX Channel Mapping Register1 (Default Value: 0x0000_0000)

Offset: 0x006C			Register Name: I2S/PCM_RXCHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:28	R/W	0x0	RX_CH11_SELECT Rx channel 11 select 00: SDI0

Offset: 0x006C			Register Name: I2S/PCM_RXCHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
27:24	R/W	0x0	RX_CH11_MAP RX Channel 11 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample
23:22	/	/	/
21:20	R/W	0x0	RX_CH10_SELECT Rx channel 10 select 00: SDI0
19:16	R/W	0x0	RX_CH10_MAP RX Channel 10 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample
15:14	/	/	/
13:12	R/W	0x0	RX_CH9_SELECT Rx channel 9 select 00: SDI0
11:8	R/W	0x0	RX_CH9_MAP RX Channel 9 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample
7:6	/	/	/
5:4	R/W	0x0	RX_CH8_SELECT Rx channel 8 select 00: SDI0

Offset: 0x006C			Register Name: I2S/PCM_RXCHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
3: 0	R/W	0x0	RX_CH8_MAP RX Channel 8 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample

7.1.6.20 0x0070 I2S/PCM RX Channel Mapping Register2 (Default Value: 0x0000_0000)

Offset: 0x0070			Register Name: I2S/PCM_RXCHMAP2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:28	R/W	0x0	RX_CH7_SELECT Rx channel 7 select 00: SDI0
27:24	R/W	0x0	RX_CH7_MAP RX Channel 7 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample
23:22	/	/	/
21:20	R/W	0x0	RX_CH6_SELECT Rx channel 6 select 00: SDI0
19:16	R/W	0x0	RX_CH6_MAP RX Channel 6 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample
15:14	/	/	/
13:12	R/W	0x0	RX_CH5_SELECT Rx channel 5 select 00: SDI0

Offset: 0x0070			Register Name: I2S/PCM_RXCHMAP2
Bit	**Read/Write**	**Default/Hex**	**Description**
11: 8	R/W	0x0	RX_CH5_MAP RX Channel 5 Mapping 0: 1st Sample … 7: 8th Sample 8: 9th Sample … 15: 16th Sample
7:6	/	/	/
5:4	R/W	0x0	RX_CH4_SELECT Rx channel 4 select 00: SDI0
3: 0	R/W	0x0	RX_CH4_MAP RX Channel 4 Mapping 0: 1st Sample … 7: 8th Sample 8: 9th Sample … 15: 16th Sample

7.1.6.21 0x0074 I2S/PCM RX Channel Mapping Register3 (Default Value: 0x0000_0000)

Offset: 0x0074			Register Name: I2S/PCM_RXCHMAP3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:28	R/W	0x0	RX_CH3_SELECT Rx channel 4 select 00: SDI0
27:24	R/W	0x0	RX_CH3_MAP RX Channel 3 Mapping 0: 1st Sample … 7: 8th Sample 8: 9th Sample … 15: 16th Sample
23:22	/	/	/
21:20	R/W	0x0	RX_CH2_SELECT Rx channel 2 select 00: SDI0

Offset: 0x0074			Register Name: I2S/PCM_RXCHMAP3
Bit	**Read/Write**	**Default/Hex**	**Description**
19:16	R/W	0x0	RX_CH2_MAP RX Channel 2 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample
15:14	/	/	/
13:12	R/W	0x0	RX_CH1_SELECT Rx channel 1 select 00: SDI0
11:8	R/W	0x0	RX_CH1_MAP RX Channel 1 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample
7:6	/	/	/
5:4	R/W	0x0	RX_CH0_SELECT Rx channel 0 select 00: SDI0
3: 0	R/W	0x0	RX_CH0_MAP RX Channel 0 Mapping 0: 1st Sample ... 7: 8th Sample 8: 9th Sample ... 15: 16th Sample

7.1.6.22 0x0078 I2S/PCM DBG Register (Default Value: 0x0000_0050)

Offset: 0x0078			Register Name: I2S/PCM_DBG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	RXTX_FIFO_TEN RXFIFO & TXFIFO Test Enable 0: Disable 1: Enable
30:8	/	/	/

Offset: 0x0078			Register Name: I2S/PCM_DBG
Bit	Read/Write	Default/Hex	Description
7: 0	R	0x50	PLACE_HOLDER NO Meaning

7.1.6.23 0x0080 I2S/PCM ASRC MCLK Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0080			Register Name: MCLKCFG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	ASRC_MCLK_GATE ASRC Clock Gate En Control Clock gate, 1=non-gated, 0= gated
15:4	/	/	/
3: 0	R/W	0x0	ASRC_MCLK_FREQ_DIV_COE Frequency Division Coefficient 4'd0=res (Clock not output), 4'd1=1x, 4'd2=1/2x, 4'd3=1/4x, 4'd4=1/6x, 4'd5=1/8x, 4'd6=1/12x,4'd7=1/16x,4'd8=1/24x, 4'd9=1/32x, 4'd10=1/48, 4'd11=1/64x, 4'd12=1/96x, 4'd13=1/128x, 4'd14=1/176x, 4'd15=1/192x, others =res.

7.1.6.24 0x0084 I2S/PCM ASRC Out Sample Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0084			Register Name: FsoutCFG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20	R/W	0x0	FSOUT_GATE fsout Clock Gate En Control clock gate，0=gated，1=non-gated
19:16	R/W	0x0	FSOUT_CLK_SRC_SEL fsout clock source sel Clock source select, 01=aclk, 10=aclkm, 11=bclk, 00=i2s0_asrc_clk, others res.
15:8	/	/	/
7:4	R/W	0x0	FSOUT_CLK_FREQ_DIV_COE1 fsout Frequency Division Coefficient 1 There are two frequency dividers in cascade. The first frequency division coefficient is controlled by bit[7:4], and the second is controlled by bit[3: 0]. 4'd0=res (Clock not output), 4'd1=1x, 4'd2=1/2x, 4'd3=1/4x, 4'd4=1/6x, 4'd5=1/8x, 4'd6=1/12x,4'd7=1/16x,4'd8=1/24x, 4'd9=1/32x, 4'd10=1/48, 4'd11=1/64x, 4'd12=1/96x, 4'd13=1/128x, 4'd14=1/176x, 4'd15=1/192x.

Offset: 0x0084			Register Name: FsoutCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
3: 0	R/W	0x0	FSOUT_CLK_FREQ_DIV_COE2 fsout Frequency Division Coefficient 2 There are two frequency dividers in cascade. The first frequency division coefficient is controlled by bit[7:4], and the second is controlled by bit[3: 0]. 4'd0=res (Clock not output), 4'd1=1x, 4'd2=1/2x, 4'd3=1/4x, 4'd4=1/6x, 4'd5=1/8x, 4'd6=1/12x,4'd7=1/16x,4'd8=1/24x, 4'd9=1/32x, 4'd10=1/48, 4'd11=1/64x, 4'd12=1/96x, 4'd13=1/128x, 4'd14=1/176x, 4'd15=1/192x.

7.1.6.25 0x0088 I2S/PCM In Sample Pulse Extend Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0088			Register Name: FsinEXTCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	EXTEND_EN Extend Enable Extend Enable，1=enable,0=disable. EXTEND_EN should be enabled in the actual application of ASRC.
15: 0	R/W	0x0	EXTEND_VALUE Extend value The extended cycle is 1 bclk at least.

7.1.6.26 0x008C I2S/PCM ASRC Enable Configuration Register (Default Value: 0x0000_0000)

Offset: 0x008C			Register Name: ASRCEN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	ASRC_EN ASRC Function Enable ASRC Function Enable，1=enable,0=disable.

7.1.6.27 0x0090 I2S/PCM ASRC Manual Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0090			Register Name: ASRCMANCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	ASRC_RATIO_MANUAL_EN ASRC Ratio manual enable ASRC Ratio manual enable,1=enable,0=disable.
30:26	/	/	/

Offset: 0x0090			Register Name: ASRCMANCFG
Bit	Read/Write	Default/Hex	Description
25: 0	R/W	0x0	ASRC_RATIO_VALUE_MANUAL_CFG ASRC Ratio Value Manual Config ASRC manually configures the value of ratio(Fin/Fout). The value uses the data format of 4.22 without 26bit symbol. The high bit4 is integral digit while the low bit22 is fractional digit.

7.1.6.28 0x0094 I2S/PCM ASRC Ratio State Configuration Register (Default Value: 0x0040_0000)

Offset: 0x0094			Register Name: ASRCRATIOSTAT
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29	R	0x0	ASRC_BUF_OVERFLOW_STA ASRC Buffer Overflow State Adaptively receive buffer overflow, and control mute together with lock. 1: Overflow 0: Not overflow
28	R	0x0	ADAPT_COMPUT_LOCK Adaptive Computational Locking 1: Locked 2: Unlocked
27:26	/	/	/
25: 0	R	0x400000	ADAPT_COMPUT_VALUE Adaptive Computational Value of Ratio

7.1.6.29 0x0098 I2S/PCM ASRC FIFO State Configure Register (Default Value: 0x0000_0000)

Offset: 0x0098			Register Name: ASRCFIFOSTAT
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8: 0	R	0x0	ASRC_RX_FIFO_FULL_LEVAL ASRC RXFIFO Full Level The fifo fill level of received data when manually configuring ratio.

7.1.6.30 0x009C I2S/PCM ASRC MBIST Test Configure Register (Default Value: 0x0000_0000)

Offset: 0x009C			Register Name: ASRCMBISTCFG
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/

Offset: 0x009C			Register Name: ASRCMBISTCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
8	R/W	0x0	ASRC_RAM_BIST_EN ASRC RAM BIST Enable ram bist enable.
7:1	/	/	/
0	R/W	0x0	ASRC_ROM_BIST_EN ASRC ROM BIST Enable rom bist enable.

7.1.6.31 0x00A0 I2S/PCM ASRC MBIST Test State Configure Register (Default Value: 0x0000_0002)

Offset: 0x00A0			Register Name: ASRCMBISTSTA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:19	/	/	/
18	R	0x0	ROM_BIST_ERROR_XOR ROM Bist error xor
17	R	0x0	ROM_BIST_ERROR_SUM ROM Bist error sum
16	R	0x0	ROM_BUSY_STATUS ROM BUSY STATUS 1: Rom busy 0: Rom idle
15:8	/	/	/
7	R	0x0	RAM_BIST_ERR_STATUS RAM Bist err status 1: Error 0: No effect
6:4	R	0x0	RAM_BIST_ERROR_PATTERN. RAM Bist error pattern.
3:2	R	0x0	RAM_BIST_ERROR_CYCLE RAM Bist error cycle.
1	R	0x1	RAM_STOP_STATUS RAM STOP STATUS 1: Stop 0: Running
0	R	0x0	RAM_BUSY_STATUS RAM BUSY STATUS 1: Ram busy 0: Ram idle.

7.2 DMIC

7.2.1 Overview

The digital microphone (DMIC) controller supports one 8-channel digital microphone interface and can output 128 fs or 64 fs (fs = ADC sample rate).

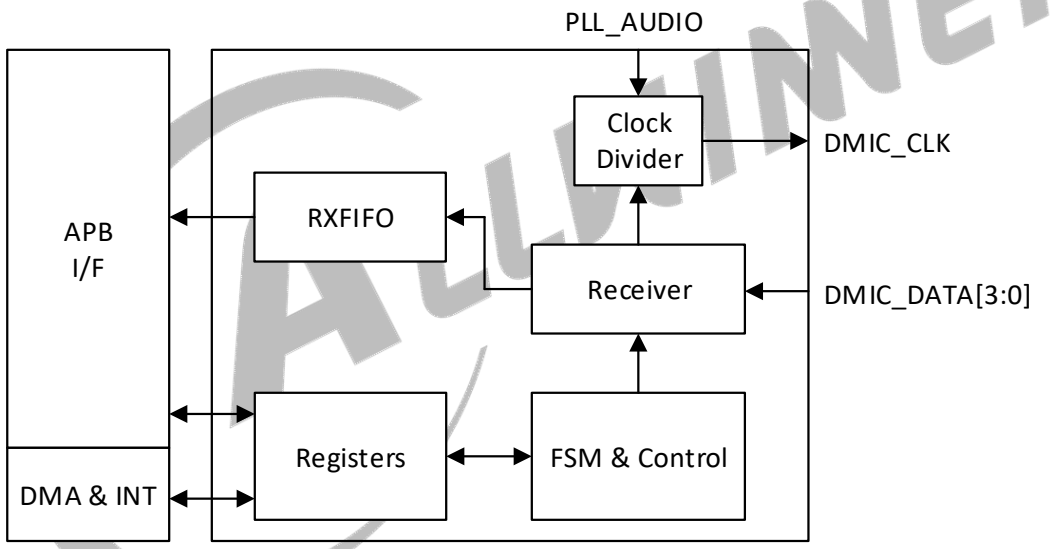
The DMIC controller includes the following features:

- Supports up to 8 channels
- Sample rate from 8 kHz to 48 kHz

7.2.2 Block Diagram

The following figure shows a block diagram of the DMIC.

Figure 7-11 DMIC Block Diagram



7.2.3 Functional Description

7.2.3.1 External Signals

The following table describes the external signals of DMIC.

Table 7-4 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

7.2.3.2 Clock Sources

The following table describes the clock source for DMIC. For clock setting, configurations, and gating information, refer to the section "CCU" and "CCU_AON".

Table 7-5 DMIC Clock Sources

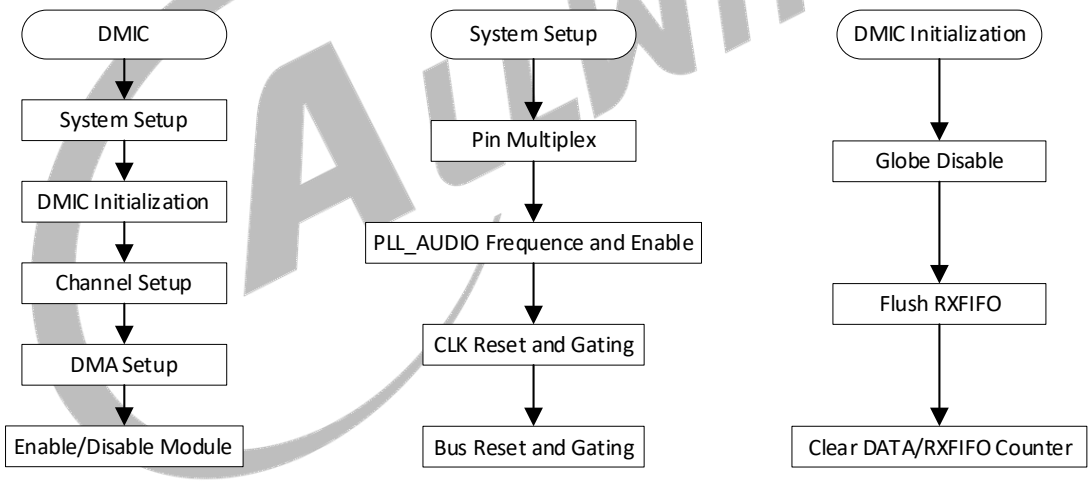
Clock Sources	Description
CCU_AON	APB bus clock. Its frequency must be greater than 24MHz to write RFIFO data.
PLL_AUDIO	DMIC_CLK, 24.576MHz or 22.5792MHz generated by PLL_AUDIO to produce 48kHz or 44.1kHz serial frequency.
RCO_HF_DIV	Supply 8.192MHz clock for DMIC_CLK in low power situation.

7.2.4 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.

The following figure shows the flow chart of the whole operation, the system setup, and the DMIC initialization.

Figure 7-12 DMIC Operation Mode



a. 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. For functions of the multiplex pins, refer to the pin multiplex specification.

Perform the following steps for the clock source. Firstly, disable the PLL_AUDIO through AUDIO_PLL_CTRL [PLL_ENABLE]. Secondly, set up the frequency of the PLL_AUDIO in AUDIO_PLL_CTRL. Then enable PLL_AUDIO. After that, enable the DMIC gating through AUDIO_CLK_CTRL[28] when you check out that the LOCK bit of AUDIO_PLL_CTRL becomes 1. At last, reset and enable the DMIC bus gating by MOD_RST_CTRL[3].

After the system setup, the register of DMIC can be setup. Firstly, 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, 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 and DMIC_CNT.

b. 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, the RXFIFO trigger level, and so on. The setup of the 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 section 3.9 "DMAC". In this module, you just enable the DRQ.

c. Enable and Disable DMIC

To enable the function, you can enable the data channel enable bit (DMIC_EN[7: 0]) by writing 1 to it. After that, enable DMIC by writing 1 to the Globe Enable bit (DMIC_EN[8]). Write 0 to DMIC_EN[8] to disable DMIC.

7.2.5 Register List

Module Name	Base Address
DMIC	0x4004C000

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	DMIC 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

7.2.6 Register Description

7.2.6.1 0x0000 DMIC Enable Control Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: DMIC_EN
Bit	Read/Write	Default/Hex	Description

Offset: 0x0000			Register Name: DMIC_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	mad_data_en 0: Disable 1: Enable
30	/	/	/
29	R/W	0x0	RX_SYNC_EN_START Audio Subsys RX Synchronize Enable Start. Only if RX_SYNC_EN set 1, RX_SYNC_EN_START can take effect. 0: Disable 1: Enable
28	R/W	0x0	RX_SYNC_EN DMIC RX Synchronize Enable 0: Disable 1: Enable
27:9	/	/	/
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

Offset: 0x0000			Register Name: DMIC_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
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

7.2.6.2 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: 48kHz 010: 24kHz 100: 12kHz 110: Reserved 001: 32kHz 011: 16kHz 101: 8kHz 111: Reserved 44.1 kHz/22.05 kHz/11.025 kHz can be supported by PLL_Audio Configure Bit.

7.2.6.3 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

Offset: 0x0008			Register Name: DMIC_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
7	R/W	0x0	DATA3_LR_SWAP_EN DATA3 Left Data and Right Data Swap Enable 0: Disable 1: Enable
6	R/W	0x0	DATA2_LR_SWAP_EN DATA2 Left Data and Right Data Swap Enable 0: Disable 1: Enable
5	R/W	0x0	DATA1_LR_SWAP_EN DATA1 Left Data and Right Data Swap Enable 0: Disable 1: Enable
4	R/W	0x0	DATA0_LR_SWAP_EN DATA0 Left Data and Right Data Swap Enable 0: Disable 1: Enable
3:1	/	/	/
0	R/W	0x0	DMIC_OS_Rate DMIC Oversample Rate 1: 64 (Supports 16 kHz~48 kHz) 0: 128 (Supports 8 kHz~24 kHz)

7.2.6.4 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

7.2.6.5 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

Offset: 0x0014			Register Name: DMIC_INTC
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	DATA_IRQ_EN DMIC RXFIFO Data Available IRQ Enable 0: Disable 1: Enable

7.2.6.6 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 Write '1' to clear this interrupt
0	R/W1C	0x0	RXFIFO_DATA_IRQ_PENDING DMIC RXFIFO Data Available Pending Interrupt 0: No Pending IRQ 1: Data Available Pending IRQ Write '1' to clear this interrupt or automatic clear if fifodeep >= rftlvl (RXFIFO Trigger Level).

7.2.6.7 0x001C DMIC RXFIFO Control Register (Default Value: 0x0000_0040)

Offset: 0x001C			Register Name: DMIC_RXFIFO_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W1C	0x0	DMIC_RXFIFO_FLUSH DMIC RXFIFO Flush Write '1' to flush RXFIFO, self clear to '0'.
30:10	/	/	/
9	R/W	0x0	RXFIFO_MODE RXFIFO Output Mode Correspond to Mode 0 And Mode 1. 0: Expanding '0' at LSB of RXFIFO Register 1: Expanding Received Sample Sign Bit at MSB of RXFIFO Register For 24-bits Received Audio Sample: Mode 0: APB_RDATA[31: 0] = {RXFIFO[20: 0],11'h0} Mode 1: APB_RDATA[31: 0] = {8{RXFIFO[20]}, RXFIFO[20: 0],3'h0} For 16-bits Received Audio Sample: Mode 0: APB_RDATA[31: 0] = {RXFIFO[20:5], 16'h0} Mode 1: APB_RDATA[31: 0] = {16{RXFIFO[20]}, RXFIFO[20:5]}

Offset: 0x001C			Register Name: DMIC_RXFIFO_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
8	R/W	0x0	SR Sample Resolution 0: 16 Bit 1: 24 Bit
7: 0	R/W	0x40	RXFIFO_TRG_LEVEL RXFIFO Trigger Level. Correspond to TRLV[7: 0]. Interrupt and DMA Request Trigger Level for DMIC RXFIFO Normal Condition IRQ/DRQ Generated When WLEVEL > TRLV[7: 0]). **Note**: *WLEVEL represents the number of valid samples in the DMIC RXFIFO.*

7.2.6.8 0x0020 DMIC RXFIFO Status Register (Default Value: 0x0000_0100)

Offset: 0x0020			Register Name: DMIC_RXFIFO_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8	R	0x1	mad_data_align 0: Misalign 1: Align When mad data output is changed to apb output, the order of apb reading data should be sent out in the case of mad_data_align = 1.
7: 0	R	0x0	DMIC_DATA_CNT DMIC RXFIFO Available Sample Word Counter

7.2.6.9 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. The actual value is N+1.

7.2.6.10 0x0028 DMIC Channel Mapping Register (Default Value: 0x7654_3210)

Offset: 0x0028			Register Name: DMIC_CH_MAP
Bit	**Read/Write**	**Default/Hex**	**Description**

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

Offset: 0x0028			Register Name: DMIC_CH_MAP
Bit	**Read/Write**	**Default/Hex**	**Description**
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 Channel 0 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

7.2.6.11 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 reading from RXFIFO. When one sample is read by DMA or by host IO (such as MAD), 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 count on base of this initial value. **Note**: *It is used for Audio/Video Synchronization.*

7.2.6.12 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 Data1 Left Channel Volume Control. (-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 Data1 Right Channel Volume Control. (-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

Offset: 0x0030			Register Name: Data0_DATA1_VOL_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
15:8	R/W	0xA0	DATA0L_VOL Data0 Left Channel Volume Control. (-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 Data0 Right Channel Volume Control. (-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

7.2.6.13 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 Data3 Left Channel Volume Control. (-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

Offset: 0x0034			Register Name: Data2_DATA3_VOL_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
23:16	R/W	0xA0	DATA3R_VOL Data3 Right Channel Volume Control. (-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 Data2 Left Channel Volume Control. (-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 Data2 Right Channel Volume Control. (-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

7.2.6.14 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	/	/	/

Offset: 0x0038			Register Name: HPF_EN_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
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

7.2.6.15 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	HPF_COE High Pass Filter Coefficient

7.2.6.16 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	HPF_GAIN High Pass Filter Gain

7.3 One Wire Audio (OWA)

7.3.1 Overview

The One Wire Audio (OWA) provides a serial bus interface for audio data. This interface is widely used for consumer audio.

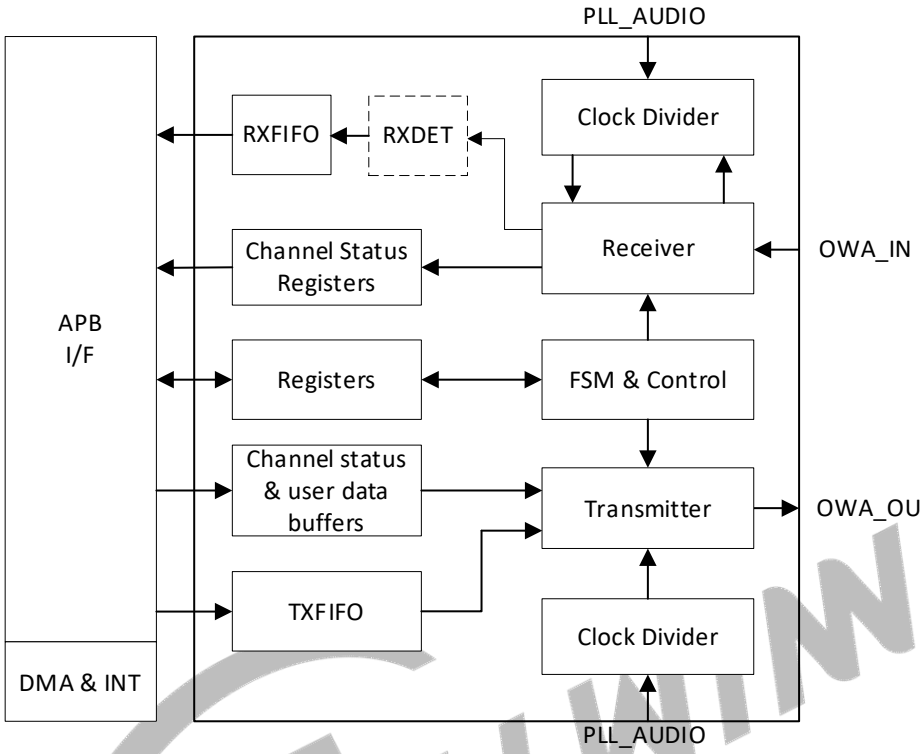
The OWA includes the following features:

- Complies with S/PDIF Interface
- Compatible with standard IEC-60958 and IEC-61937
  - IEC-60958 supports 16-bit, 20-bit and 24-bit data formats
  - IEC-61937 uses the IEC-60958 series for the conveying of non-linear PCM bit streams, each sub-frame transmits 16-bit
- TXFIFO and RXFIFO
  - One 128×24bits TXFIFO and one 64×24bits RXFIFO for audio data transmission
  - Programmable FIFO thresholds
- Supports TX/RX DMA slave interface
- Supports multiple function clock
  - Separate clock for OWA TX and OWA RX
  - The TX function clock supports the frequency of 24.576 MHz and 22.579 MHz
  - The RX function clock supports the frequency of 60MHz and 240MHz, which can realize the sampling rate of 8 kHz to 96 kHz and 32 kHz to 192 kHz respectively
- Supports hardware parity on TX/RX
  - Hardware parity generation on the transmitter
  - Hardware parity checking on the receiver
- Supports channel status capture for the receiver
- Supports channel sample rate capture on the receiver
- Supports insertion detection for the receiver
- Supports channel status insertion for the transmitter
- Supports interrupts and DMA

7.3.2 Block Diagram

The following figure shows the OWA block diagram.

Figure 7-13 OWA Block Diagram



OWA contains the following sub-blocks:

Table 7-6 OWA Sub-blocks

Sub-block	Description
Registers	Analyze the configuration parameter, DMA requests, and IRQ feedbacks.
Receiver	Parses the frame header and receives the data.
Transmitter	Sends the data
FSM	Finite state machine
Clock Divider	Clock divider circuit

7.3.3 Functional Description

7.3.3.1 External Signals

The OWA is a Biphase-Mark Encoding Digital Audio Transfer protocol. In this protocol, the CLK signal and data signal are transferred in the same line. The following figure describes the external signals of OWA. OWA-OUT is the output pin for the output CLK and DATA, and OWA-IN is the input pin for the input CLK and DATA.

Table 7-7 OWA External Signals

Signal Name	Description	Type
OWA_OUT	OWA output	O
OWA_IN	OWA input	I

7.3.3.2 Clock Sources

The OWA has separate clock for OWA_TX and OWA_RX. The following tables describe the clock sources for OWA_TX and OWA_RX. For clock setting, configurations and gating information, refer to the section "CCU" and "CCU_AON".

Table 7-8 OWA Clock Sources

Clock Name	Description
CCU_AON	APB bus clock. For more details, refer to "CCU" and "CCU_AON".
PLL_AUDIO	OWA TX serial clock 24.576MHz or 22.5792MHz.
DPLL	OWA RX serial clock ⩾200MHz.

7.3.3.3 Biphase-Mark Code

In the 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. The following figure and table show how data is encoded to the BMC format.

The frequency of the clock is twice the data bit rate, as shown in the following figure. Also, the clock is always programmed to 128 x fs, where fs is the sample rate. The device receiving in the OWA format can recover the clock and frame information from the BMC signal.

Figure 7-14 OWA Biphase-Mark Code

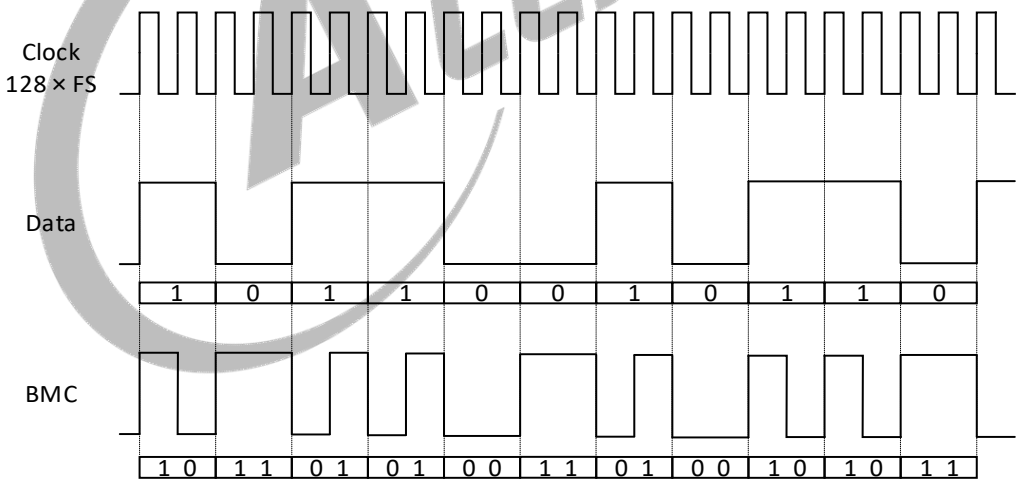
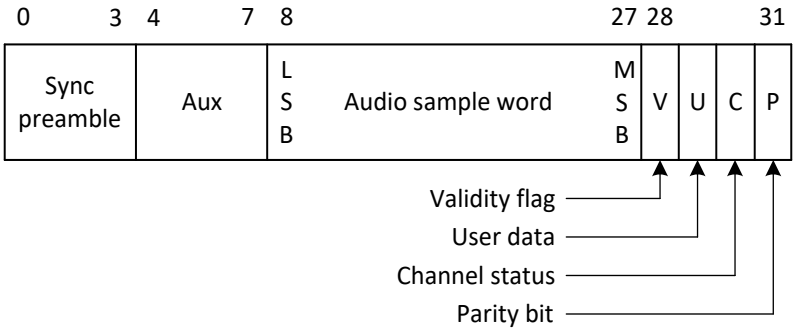


Table 7-9 Biphase-Mark Encoder

Data	Previous State	BMC
0	0	11
0	1	00
1	0	10
1	1	01

Figure 7-15 OWA Sub-Frame Format



Bits 0-3 carry one of the four permitted preambles to signify the type of audio sample in the current sub-frame. 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.

Bits 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 sub-frame.

Bit 29 carries the user data channel (U) associated with the main data field in the sub-frame.

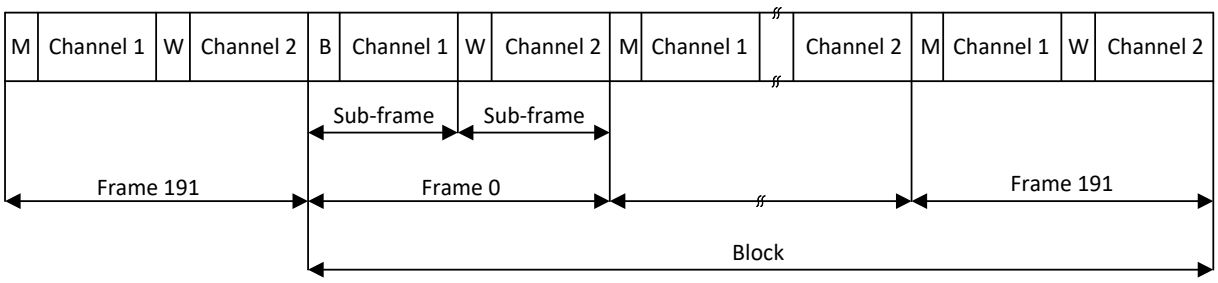
Bit 30 carries the channel status information (C) associated with the main data field in the sub-frame. The channel status indicates if the data in the sub-frame is a digital audio or some other type of data.

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 the following table, the preambles (bit 0-3) are also defined with even parity.

Table 7-10 Preamble Codes

Preamble Code	Previous Logical State	Logical State	Description
B (or Z)	0	1110 1000	Start of a block and sub-frame 1
M (or X)	0	1110 0010	Sub-frame 1
W (or Y)	0	1110 0100	Sub-frame 2

Figure 7-16 OWA Frame/Block Format



7.3.3.4 IEC61937 Transmit Format

IEC 61937 applies to the digital audio interface by using the IEC 60958 series for the conveying of non-linear PCM encoded audio bitstreams. The non-linear PCM encoded audio bitstream is transferred by using the basic 16-bit data area of the IEC 60958 subframes, i.e. in time-slots 12 to 27. Because the non-linear PCM encoded audio bitstream to be transported is at a lower data rate than that supported by the IEC 60958 interface, the audio bitstream is broken into a sequence of discrete data-bursts, and stuffing between the data-bursts is necessary.

IEC 60958 Data Burst

The method of placing the data into the IEC 60958 bitstream is to format the data to be transmitted into data-bursts and to send each data-burst in a continuous sequence of IEC 60958 frames.

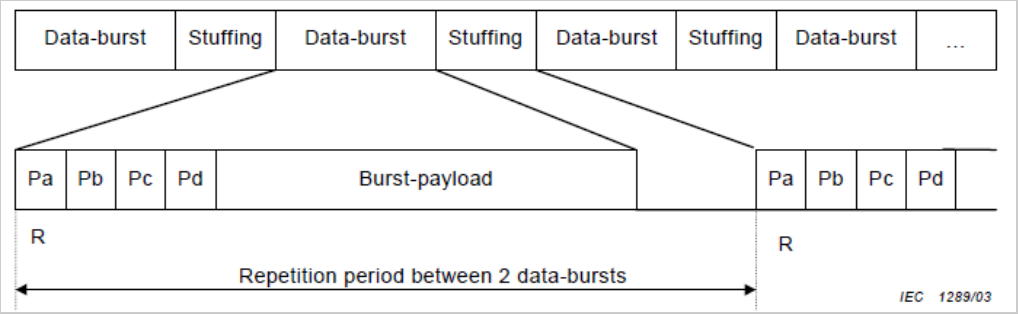
Table 7-11 Bit Allocation of Data-Burst in IEC 60958 Subframes

Subframe	Bit of subframes				
	MSB b27	b26	b25 ..... b14	b13	LSB b12
Frame 0; subframe B or M	0	1		14	15
Frame 0; subframe W	16	17		30	31
Frame 1; subframe B or M	32	33		46	47
Frame 1; subframe W	48	49		62	63
Frame 2; subframe B or M	64	65		78	79
----------			----------		
Last subframe B or M of data-burst	n-32	n-31		n-18	n-17
Last subframe W of data-burst	n-16	n-15		n-2	n-1

Data Burst Format

Each data-burst contains a burst-preamble consisting of four 16-bit words (Pa, Pb, Pc and Pd) followed by the burst-payload which contains data of an encoded audio frame.

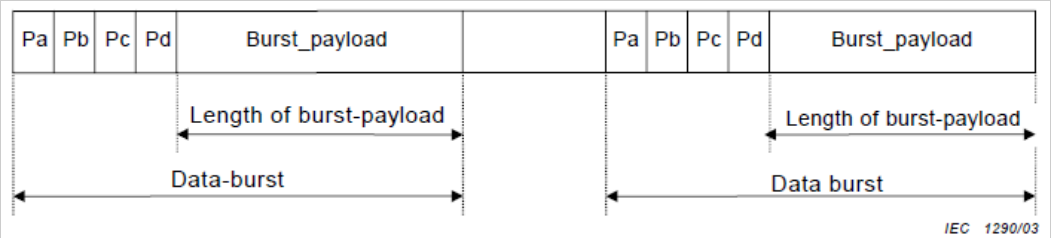
Figure 7-17 Data-Burst Format



a.Burst-preamble

The burst-preamble consists of four mandatory fields. Pa and Pb represent a synchronization word. Pc gives information about the type of data, and some information/control for the receiver. Pd gives the length of the burst-payload, and is limited to 65535 bits in the case of Pd representing bit length, or is limited to 65535 bytes in the case of Pd representing bytes length.

Figure 7-18 Data-burst Preamble



The four preamble words are contained in two sequential IEC 60958 frames. The frame beginning the data-burst contains preamble word Pa in subframe 1, and Pb in subframe 2. The next frame contains Pc in subframe 1 and Pd in subframe 2. When placed into an IEC 60958 subframe, the MSB of a 16-bit burst-preamble word is placed into time-slot 27 and the LSB is placed into time-slot 12.

Figure 7-19 Data-burst Preamble words

Preamble word	Length of field	Contents	Value MSB..LSB
Pa	16-bit	Sync word 1	F872h
Pb	16-bit	Sync word 2	4E1Fh
Pc	16-bit	Burst-info	Table 5
Pd	16-bit	Length-code	Number of bits or number of bytes according to data-type

b.Burst-information

The 16-bit burst-information contains information about which bit of data will be found in the data-burst.

Figure 7-20 Fields of Burst-information

Bits of Pc	Value	Contents	Remark
0 – 6		Data-type	See IEC 61937-2
7	0	Error-flag indicating a valid burst-payload	
	1	Error-flag indicating that the burst-payload may contain errors	
8 – 12		Data-type-dependent info	
13 – 15	0	Bitstream-number	
NOTE The repetition period of pause data-bursts depends on the application in which IEC 60958 is used to convey encoded audio bitstreams.			

The 7-bit data-type is defined in bits 0-6 of the burst-preamble Pc, the bit 6 is the MSB. This data-type field indicates the format of the burst-payload, which will be conveyed in the data-burst. Typical properties of a data-type are the reference point and repetition period of the burst, which is the number of sampling periods of the audio between the reference point of the current data-burst and the reference point of the next data-burst. The reference point is inherently defined for each data-type.

The error-flag bit is available to indicate if the contents of the data-burst contain data errors. If a data-burst is thought to be error-free, or if the data source does not know if the data contains errors, then the value of this bit is set to a '0'. If the data source does know that a particular data-burst contains some errors this bit may be set to a '1'. The usage of this bit by receiver is optional.

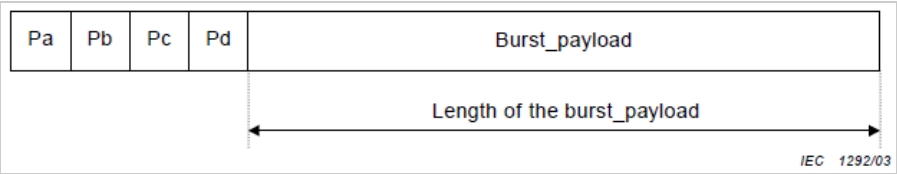
The meaning of the 5-bit data-type-dependent info depends on the value of the data-type.

The 3-bit bitstream-number indicates to which bitstream the data-burst belongs. Eight codes (0-7) are available so that up to eight independent bitstreams may be multiplexed in one bitstream in a time multiplex. Each independent bitstream shall use a unique bit-stream number.

c.Length-code

The length-code indicates the number of bits or bytes according to data-type within the data burst, from 0 to 65535. The size of the Pa, Pb, Pc and Pd is not counted in the value of the length-code. In other words, the length-code indicates the number of bits of the burst-payload in bits, plus the conditional length of Pe and Pf, or the number of bytes of the burst-payload in bytes, plus the conditional length of Pe and Pf if exists.

Figure 7-21 Length of the Burst-Payload Specified by Pd



7.3.3.5 Audio Sample Ratio Detection

The sampling rate is calculated according to the data pulse back-stepping method. In the first phase lock of the CDR, find 1 Frame period, count by using the high-speed sampling clock, and read the counting value of the pulse, then the sampling rate can be calculated.

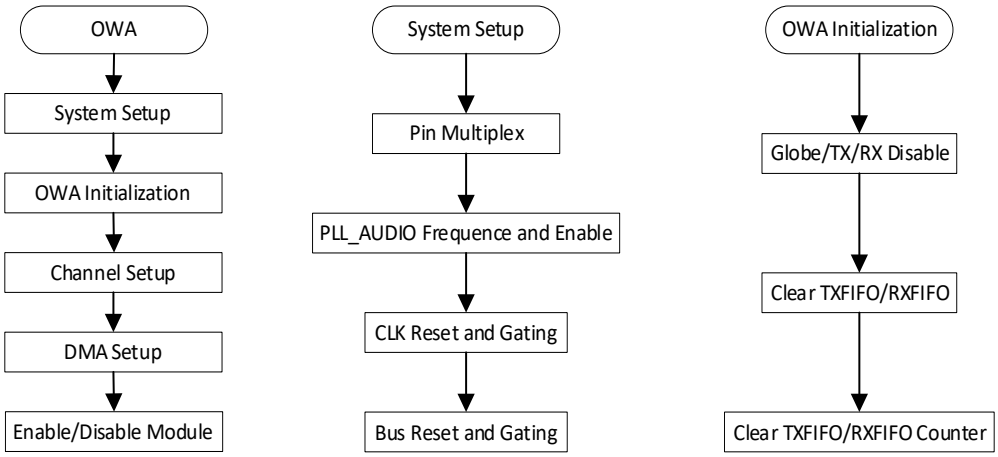
Table 7-12 Corresponding Relation between Different System Clock and Sample Ratio

	Sample Clock Cycles	
TX Sample Rate(kHz)	**196.608 MHz-SysClk**	**200 MHz-SysClk**
22.05	8916(±5)	9070(±5)
24	8192(±5)	8333(±5)
32	6144(±5)	6250(±5)
44.1	4458(±5)	4535(±5)
48	4096(±5)	4166(±5)
96	2048(±5)	2083(±5)
176.4	1114(±5)	1133(±5)
192	1024(±5)	1041(±5)

7.3.4 Operation Mode

The software operation of the OWA is divided into five steps: system setup, OWA initialization, channel setup, DMA setup and enable/disable module. The following sections describe these five steps.

Figure 7-22 OWA Operation Flow



a. 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 section "GPIO".

The clock source for the OWA should be followed. Firstly, reset the audio PLL in AUDIO_PLL_CTRL. Secondly, set up the frequency of the Audio PLL in the AUDIO_PLL_CTRL. After that, enable the OWA gating. Lastly, enable the OWA bus gating.

After the system setup, the register of OWA can be set up. Firstly, reset the OWA by writing 1 to OWA_CTL[0] and clear the TX/RX FIFO by writing 1 to OWA_FCTL[17:16]. After that, enable the globe enable bit by writing 1 to OWA_CTL[1] and clear the interrupt and TX/RX counter by setting OWA_ISTA and OWA_TX_CNT/OWA_RX_CNT.

b. Channel Setup and DMA Setup

You can set up the audio type, clock divider ratio, the sample format, and the trigger level, and so on. The setup of the register can be found in the specification.

The OWA supports two methods to transfer the data. The most common way is DMA, the configuration of DMA can be found in section 3.9 "DMAC". In this module, you just enable the DRQ in OWA_INT[7].

c. Enable and Disable OWA

To enable the function, you can enable TX/RX by writing OWA_TX_CFIG[31]/OWA_RX_CFIG[0]. After that, enable OWA by writing 1 to OWA_CTL[1]. Writing 0 to OWA_CTL[1] to disable process.

7.3.5 Register List

Module Name	Base Address
OWA	0x40045C00

Register Name	Offset	Description
OWA_GEN_CTL	0x0000	OWA General Control Register
OWA_TX_CFIG	0x0004	OWA TX Configuration Register
OWA_RX_CFIG	0x0008	OWA RX Configuration Register

Register Name	Offset	Description
OWA_ISTA	0x000C	OWA Interrupt Status Register
OWA_RXFIFO	0x0010	OWA RXFIFO 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_RX_CNT	0x0028	OWA RX Counter Register
OWA_TX_CHSTA0	0x002C	OWA TX Channel Status Register0
OWA_TX_CHSTA1	0x0030	OWA TX Channel Status Register1
OWA_RXCHSTA0	0x0034	OWA RX Channel Status Register0
OWA_RXCHSTA1	0x0038	OWA RX Channel Status Register1
OWA_EXP_CTL	0x0040	OWA Expand Control Register
OWA_EXP_ISTA	0x0044	OWA Expand Interrupt Status Register
OWA_EXP_INFO_0	0x0048	OWA Expand Information Register0
OWA_EXP_INFO_1	0x004C	OWA Expand Information Register1
OWA_EXP_DBG_0	0x0050	OWA Expand Debug Register0
OWA_EXP_DBG_1	0x0054	OWA Expand Debug Register1

7.3.6 Register Description

7.3.6.1 0x0000 OWA General Control Register (Default Value: 0x0000_0080)

Offset: 0x0000			Register Name: OWA_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
12	R/WAC	0x0	RST_RX Reset RX 0: Normal 1: Reset Self clear to '0'.
11:8	/	/	/
7	R	0x1	Reserved
6:3	/	/	/
2	R/W	0x0	LOOP Loopback Test 0: Normal Mode 1: Loopback Test When the bit is set to '1', the DOUT and DIN need be connected.

Offset: 0x0000			Register Name: OWA_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x0	GEN Global Enable Disabling this bit overrides the operations of enabling and flushing all FIFOs by any other blocks or channels. 0: Disabled 1: Enabled
0	R/W	0x0	RST_TX Reset TX 0: Normal 1: Reset Self clear to 0.

7.3.6.2 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: Disabled 1: Enabled
30:18	/	/	/
17	R/W	0x0	ASS Audio Sample Select when TX FIFO Underrun 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

Offset: 0x0004			Register Name: OWA_TX_CFIG
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x0	TX_CHM CHSTMODE 0: Channel status A and B set to 0 1: Channel status A and B generated from TX_CHSTA
0	R/W	0x0	TXEN TX Enable 0: Disabled 1: Enabled

7.3.6.3 0x0008 OWA RX Configure Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: OWA_RX_CFIG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4	R	0x0	RX_LOCK_FLAG RX Lock Flag 0: Unlocked 1: Locked
3	R/W	0x0	RX_CHST_SRC RX Channel State Source Select 0: RX_CH_STA register holds status from Channel A 1: RX_CH_STA register holds status from Channel B
2	/	/	/
1	R/W	0x0	CHST_CP Channel Status Capture 0: Idle or Capture End 1: Capture Channel Status Start The field must be set to 1 at each operation (such as recording). When set to '1', the system starts to capture the channel status. When finished, the bit will automatically turn to '0'.
0	R/W	0x0	RXEN 0: Disabled 1: Enabled

7.3.6.4 0x000C OWA Interrupt Status Register (Default Value: 0x0000_0010)

Offset: 0x000C			Register Name: OWA_ISTA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:19	/	/	/

Offset: 0x000C			Register Name: OWA_ISTA
Bit	**Read/Write**	**Default/Hex**	**Description**
18	R/W1C	0x0	RX_LOCK_INT RX Lock Interrupt 0: No Pending IRQ 1: RX Lock Pending Interrupt (RX_LOCK_FLAG turns from 0 to 1) Write '1' to clear this interrupt.
17	R/W1C	0x0	RX_UNLOCK_INT RX Unlock Pending Interrupt 0: No Pending IRQ 1: RX Unlock Pending Interrupt (RX_LOCK_FLAG turns from 0 to 1) Write 1 to clear this interrupt.
16	R/W1C	0x0	RX_PARERRI_INT RX Parity Error Pending Interrupt 0: No Pending IRQ 1: RX Parity Error Pending Interrupt Write "1" to clear this interrupt.
15:7	/	/	/
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.
3:2	/	/	/
1	R/W1C	0x0	RXO_INT RXFIFO Overrun Pending Interrupt 0: RXFIFO Overrun Pending Write '1' to clear this interrupt.

Offset: 0x000C			Register Name: OWA_ISTA
Bit	Read/Write	Default/Hex	Description
0	R/W1C	0x0	RXA_INT RXFIFO Available Pending Interrupt 0: No Pending IRQ 1: Data Available Pending IRQ Write '1' to clear this interrupt or automatically clear if the interrupt condition fails.

7.3.6.5 0x0010 OWA RXFIFO Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: OWA_RXFIFO
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x0	RX_DATA The host can get one sample by reading this register, A channel data is first, and then the B channel data.

7.3.6.6 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 The bit takes effect only when the TXEN is set to 1. Audio codec/I2S0/I2S1/I2S2/OWA TXFIFO Hub Enable. 0: Disabled 1: Enabled
30	R/W1C	0x0	FTX Write '1' to flush TXFIFO, self clear to '0'.
29	R/W1C	0x0	FRX Write '1' to flush RXFIFO, self clear to '0'.
28:22	/	/	/
21	R/W	0x0	RX_SYNC_EN_START The bit takes effect only when the RX_SYNC_EN is set to 1. Audio Codec/I2S0/I2S1/I2S2/DMIC/OWA RX Synchronize Enable Start. 0: Disabled 1: Enabled
20	R/W	0x0	RX_SYNC_EN OWA RX Synchronize Enable 0: Disabled 1: Enabled

Offset: 0x0014			Register Name: OWA_FCTL
Bit	**Read/Write**	**Default/Hex**	**Description**
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	/	/	/
10:4	R/W	0x20	RXTL RX FIFO Trigger Level Interrupt and DMA request trigger level for RX FIFO normal condition. Trigger Level = RXTL + 1
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	R/W	0x0	RXOM RXFIFO Output Mode(Mode 0,1,2,3) 00: Expanding '0' at LSB of RXFIFO Register 01: Expanding received sample sign bit at MSB of RXFIFO Register 10: Truncating received samples at high half-word of RXFIFO Register and low half-word of RXFIFO Register is filled by '0' 11: Truncating received samples at low half-word of RXFIFO Register and high half-word of RXFIFO Register is expanded by its sigh bit Mode 0: APB_RDATA[31: 0] = {RXFIFO[23: 0], 8'h0} Mode 1: APB_RDATA[31: 0] = {8'RXFIFO[23], RXFIFO[23: 0]} Mode 2: APB_RDATA[31: 0] = {RXFIFO[23:8], 16'h0} Mode 3: APB_RDATA[31: 0] = {16'RXFIFO[23], RXFIFO[23:8]}

7.3.6.7 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 the TXFIFO is not full) 0: No room for new sample in TXFIFO 1: More than one room for new sample in TXFIFO (>= 1 Word)

Offset: 0x0018			Register Name: OWA_FSTA
Bit	**Read/Write**	**Default/Hex**	**Description**
30:24	/	/	/
23:16	R	0x80	TXE_CNT TXFIFO Empty Space Word Counter
15	R	0x0	RXA RXFIFO Available 0: No available data in RXFIFO 1: More than one sample in RXFIFO (>= 1 Word)
14:7	/	/	/
6: 0	R	0x0	RXA_CNT RXFIFO Available Sample Word Counter

7.3.6.8 0x001C OWA Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: OWA_INT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:19	/	/	/
18	R/W	0x0	RX_LOCKI_EN RX LOCK Interrupt Enable 0: Disabled 1: Enabled
17	R/W	0x0	RX_UNLOCKI_EN RX UNLOCK Interrupt Enable 0: Disabled 1: Enabled
16	R/W	0x0	RX_PARERRI_EN RX PARITY ERORR Interrupt Enable 0: Disabled 1: Enabled
15:8	/	/	/
7	R/W	0x0	TX_DRQ TXFIFO Empty DRQ Enable 0: Disabled 1: Enabled
6	R/W	0x0	TXUI_EN TXFIFO Underrun Interrupt Enable 0: Disabled 1: Enabled
5	R/W	0x0	TXOI_EN TXFIFO Overrun Interrupt Enable 0: Disabled 1: Enabled

Offset: 0x001C			Register Name: OWA_INT
Bit	**Read/Write**	**Default/Hex**	**Description**
4	R/W	0x0	TXEI_EN TXFIFO Empty Interrupt Enable 0: Disabled 1: Enabled
3	/	/	/
2	R/W	0x0	RX_DRQ RXFIFO Data Available DRQ Enable When set to '1', RXFIFO DMA Request is asserted if data is available in RXFIFO. 0: Disabled 1: Enabled
1	R/W	0x0	RXOI_EN RXFIFO Overrun Interrupt Enable 0: Disabled 1: Enabled
0	R/W	0x0	RXAI_EN RXFIFO Data Available Interrupt Enable 0: Disabled 1: Enabled

7.3.6.9 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. A channel data is first, and then the B channel data.

7.3.6.10 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 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 updated by the initial value, the counter register should count on the base of this initial value.

7.3.6.11 0x0028 OWA RX Counter Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: OWA_RX_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	RX_CNT RX Sample Counter The audio sample number of writing into RXFIFO. When one sample is written by Codec, the RX sample counter register increases by one. The RX counter register can be set to any initial value at any time. After being updated by the initial value, the counter register should count on the base of this value.

7.3.6.12 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 Sampling 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

Offset: 0x002C			Register Name: OWA_TX_CHSTA0
Bit	**Read/Write**	**Default/Hex**	**Description**
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 to 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 to 111: Reserved For bit 1 = “1”, other than Linear PCM applications: 000: Default state 001 to 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”.

7.3.6.13 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	/	/	/

Offset: 0x0030			Register Name: OWA_TX_CHSTA1
Bit	**Read/Write**	**Default/Hex**	**Description**
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 bits 010: 18 bits 100: 19 bits 101: 20 bits 110: 17 bits 111: Reserved For bit 0 = “1”: 000: Not indicated 001: 20 bits 010: 22 bits 100: 23 bits 101: 24 bits 110: 21 bits 111: Reserved

Offset: 0x0030			Register Name: OWA_TX_CHSTA1
Bit	Read/Write	Default/Hex	Description
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

7.3.6.14 0x0034 OWA RX Channel Status Register0 (Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: OWA_RX_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 Sampling 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.

Offset: 0x0034			Register Name: OWA_RX_CHSTA0
Bit	**Read/Write**	**Default/Hex**	**Description**
7:6	R/W	0x0	MODE Mode 00: Default mode 01 to 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 to 111: Reserved For bit 1 = '1', Other than Linear PCM applications: 000: Default state 001 to 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

7.3.6.15 0x0038 OWA RX Channel Status Register1 (Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: OWA_RX_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 is not be used 11: No copying is permitted

Offset: 0x0038			Register Name: OWA_RX_CHSTA1
Bit	**Read/Write**	**Default/Hex**	**Description**
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 bits 010: 18 bits 100: 19 bits 101: 20 bits 110: 17 bits 111: Reserved For bit 0 = '1': 000: Not indicated 001: 20 bits 010: 22 bits 100: 23 bits 101: 24 bits 110: 21 bits 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

7.3.6.16 0x0040 OWA Expand Control Register (Default Value: 0x0000_000F)

Offset: 0x0040			Register Name: OWA_EXP_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R/W	0x0	BURST_DATAOUT_SELECT Burst data output select 0: Burst preamble and payload 1: Burst payload
29:16	R/W	0x0	REPEAT_PERIOD_OF_FR_NUM The number for the repetition period of the burst frame Configure this field according to RX data. A mismatch between the configuration data and the received data will result in an error interrupt.
15	R/W	0x0	UNIT_SELECT Unit Select Configure this field according to RX data type 0: In units of 16-bit 1: In units of 2-byte
14	R/W	0x0	OWA_RX_MODE_MAN OWA RX Protocol Select 0: IEC60958 1: IEC61937
13	R/W	0x0	OWA_RX_MODE OWA RX Mode Select 0: Manual Ctrl. Configure by OWA RX_MODE_MAN 1: Auto Ctrl. Configure by the channel status values resolved by hardware
12	R/W	0x0	AUDIO_DATA_BITORDER_EN Audio data bit order enable 0: The audio data received by RX is stored directly into FIFO 1: The audio data received by RX is reversed high and low bits, then stored into FIFO
11	R/W	0x0	DATA_LENGTH_BITORDER_EN Data length bit order enable 0: The received PD data is as the length of the valid audio data 1: The received PD data is reversed high and low bits, then as the length of the valid audio data
10	R/W	0x0	DATA_TYPE_BITORDER_EN Data type bit order enable 0: The received PC data is as the data length of the valid audio 1: The received PC data is reversed high and low bits, then as the length of the valid audio data

Offset: 0x0040			Register Name: OWA_EXP_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
9	R/W	0x0	SYNCW_BITORDER_EN Syncw_bitorder_en 0: Pa/Pb is the sync code of audio data 1: Pa/Pb reversed high and low bits is the sync code of audio data
8	R/W	0x0	INSERT_DETECTION_ENABLE Insert detection enable 0: Disable 1: Enable
7: 0	R/W	0xF	INSERT_DETECTION_NUM Insert detection number Configure how many jumping edges are detected to generate an insertion interrupt

7.3.6.17 0x0044 OWA Expand Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: OWA_EXP_ISTA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	PD_CHANGE_INT_EN PD_LENGTH_CHANGE Interrupt Enable 0: Disable 1: Enable
23	R/W	0x0	PC_PAUSE_STOP_INT PC_PAUSE_BURSTS_STOP Interrupt Enable 0: Disable 1: Enable
22	R/W	0x0	PC_BITSTRM_CHANGE_INT_EN PC_BITSTREAM_CHANGE Interrupt Enable 0: Disable 1: Enable
21	R/W	0x0	PC_ERR_FLAG_INT PC_ERROR_FLAG Interrupt Enable 0: Disable 1: Enable
20	R/W	0x0	PC_DTYPE_CHANGE_INT_EN PC_DATATYPE_CHANGE Interrupt Enable 0: Disable 1: Enable

Offset: 0x0044			Register Name: OWA_EXP_ISTA
Bit	**Read/Write**	**Default/Hex**	**Description**
19	R/W	0x0	RPDB_ERR_INT_EN RPDB_ERROR Interrupt Enable 0: Disable 1: Enable
18	R/W	0x0	PCPD_CAP_INT_EN PCPD_CAP Interrupt Enable 0: Disable 1: Enable
17	R/W	0x0	PAPB_CAP_INT_EN PAPB_CAP Interrupt Enable 0: Disable 1: Enable
16	R/W	0x0	INSERT_INT_EN INSERT Interrupt Enable 0: Disable 1: Enable
15:9	/	/	/
8	R/W1C	0x0	PD_CHANGE_INT PD CHANGE INT 0: No Pending IRQ 1: PD Data length information is change. (except Pause/Null data burst type) Write ‘1’ to clear this interrupt.
7	R/W1C	0x0	PC_PAUSE_STOP_INT Audio bit stream is interrupted. When stopped, the interface becomes idle. 0: No Pending IRQ 1: PC Pause burst Stop, frame sequence discontinued. Transmitters may optionally use the STOP value to indicate that the transmission of the current encoded Write ‘1’ to clear this interrupt.
6	R/W1C	0x0	PC_BITSTRM_CHANGE_INT PC BITSTRM CHANGE INT 0: No Pending IRQ 1: PC Bit Stream Number is change. Bit stream Number indicates which bit stream the data burst belongs. (except Pause/Null data bursts type) Write ‘1’ to clear this interrupt.

Offset: 0x0044			Register Name: OWA_EXP_ISTA
Bit	**Read/Write**	**Default/Hex**	**Description**
5	R/W1C	0x0	PC_ERR_FLAG_INT PC ERR FLAG INT 0: No Pending IRQ 1: PC Error-flag is available to indicate if the contents of the data-burst contain data errors (except Pause/Null data bursts type). The using of this bit by receivers is optional. Write '1' to clear this interrupt.
4	R/W1C	0x0	PC_DTYPE_CHANGE_INT PC DTYPE CHANGE INT 0: No Pending IRQ 1: PC Datatype (except Pause/Null data type) information is change Write '1' to clear this interrupt.
3	R/W1C	0x0	RPDB_ERR_INT RPDB ERR INT 0: No Pending IRQ 1: Hardware counts the repetition period of the burst frame is different from register configuration number Write '1' to clear this interrupt.
2	R/W1C	0x0	PCPD_CAP_INT PCPD CAP INT 0: No Pending IRQ 1: IEC61937 mode captures PC and PD Write '1' to clear this interrupt.
1	R/W1C	0x0	PAPB_CAP_INT PAPB CAP INT 0: No Pending IRQ 1: IEC61937 mode captures PA and PB Write '1' to clear this interrupt.
0	R/W1C	0x0	INSERT_INT INSERT INT 0: No Pending IRQ 1: OWA RX detects device insertion Write '1' to clear this interrupt.

7.3.6.18 0x0048 OWA Expand Information Register 0 (Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: OWA_EXP_INFO_0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R	0x0	PC_DATA PC Data information

Offset: 0x0048			Register Name: OWA_EXP_INFO_0
Bit	Read/Write	Default/Hex	Description
15: 0	R	0x0	PD_DATA PD Data information

7.3.6.19 0x004C OWA Expand Information Register 1 (Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: OWA_EXP_INFO_1
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:16	R	0x0	REPET_PERIOD_OF_FR_VALUE Repetition period of the burst frame value Check whether the repetition period of the burst frame calculated by hardware is consistent with the configuration value.
15: 0	R	0x0	SR_VALUE Sample Rate Value Read this value after RX_LOCK.

7.3.6.20 0x0050 OWA Expand Debug Register 0 (Default Value: 0x0000_0000)

Offset: 0x0050			Register Name: OWA_EXP_DBG_0
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/
18:16	R	0x0	IEC61937_DATA_CAP_FSM IEC61937 Data Captures State Machine 000: IDLE 001: SYNC_PA 010: SYNC_PB 011: DTYPE_PC 100: DLEN_PD 101: RX_ACTIVE
15: 0	R	0x0	DATA_CAP_NUM Remains Data Counter Value See the value of the sampled valid data in real time.

7.3.6.21 0x0054 OWA Expand Debug Register 1 (Default Value: 0x0000_0000)

Offset: 0x0054			Register Name: OWA_EXP_DBG_1
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:16	R	0x0	REPET_PERIOD_OF_FR_CNT Repetition period of the burst frame counter See the value of repetition period counter in real time.

Offset: 0x0054			Register Name: OWA_EXP_DBG_1
Bit	**Read/Write**	**Default/Hex**	**Description**
15: 0	R	0x0	SR_CNT Sample Rate Counter See the value of audio sample ratio in real time.

7.4 Audio Codec

7.4.1 Overview

The Audio Codec is an audio encoder and decoder module that supports the recording and playing of audio. This module includes tri-channel analog digital converter (ADC), dual-channel analog digital converter (DAC) and digital audio processor (DAP).

The Audio Codec has the following features:

- HiFi Audio ADC
  - 3-channel ADCs @ 24-bit
  - Up to 98 dB SNR during ADC recording path (signal through PGA and ADC with A-weighted filter)
  - 3 fully-differential analog microphone inputs with 0 dB~30 dB amplifier gain
  - Supports sample rates ranging from 8 kHz to 96 kHz
  - Digital volume control with 0.5 dB step
  - Digital high-pass filter
  - 128x24-bit FIFO for recording received data
- HiFi Audio DAC
  - 2-channel DACs @ 24-bit (for R128-S1 and R128-S2)
  - 1-channel DAC @ 24-bit (for R128-S3)
  - Up to 119 dB SNR in the DAC playback path (signal through DAC and lineout with A-weighted filter)
  - Stereo lineout driver
  - Supports sample rates ranging from 8 kHz to 192 kHz
  - Digital volume control with 0.5 dB step
  - 20-band Biquads filter for EQ
  - 3-band dynamic range control
  - 1-band dynamic range control
  - 5-band Biquads filter for EQ
  - 128x24-bit FIFO for playing transmitted data
- Three differential microphone inputs: MICIN1P/N, MICIN2P/N, MICIN3P/N.
- Two stereo LINEOUT outputs: LINEOUTLP/N and LINEOUTRP/N (for R128 S1 and R128 S2)

- One single-end LINEOUT output: LINEOUTLP/N (for R128 S3)
- Built-in audio PLL with flexible clocking scheme
- DMA and interrupt support both receiving and transmitting
- Integrated ALDO for analog part
- One low-noise analog microphone bias output three audio inputs

7.4.2 Block Diagram

The following figure shows the block diagram of Audio Codec.

Figure 7-23 Block Diagram of Audio Codec for R128-S1/S2

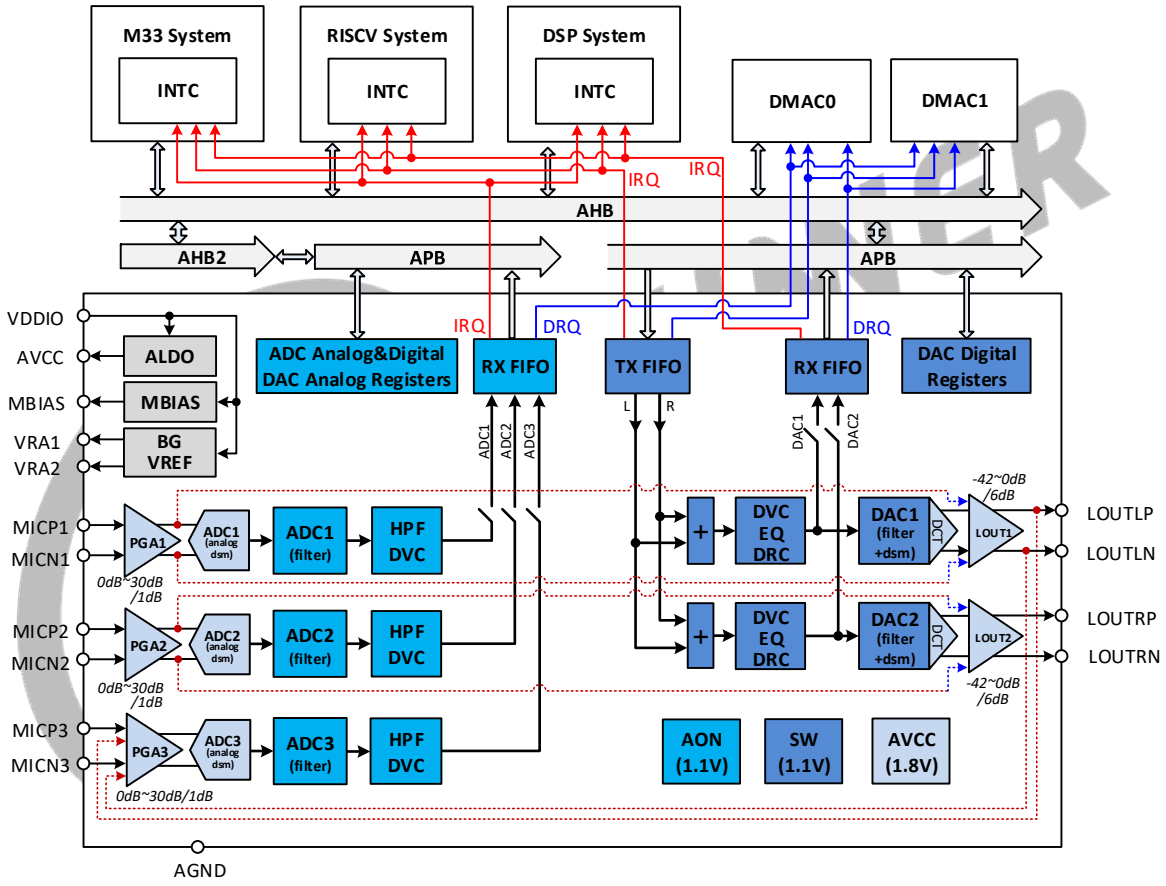
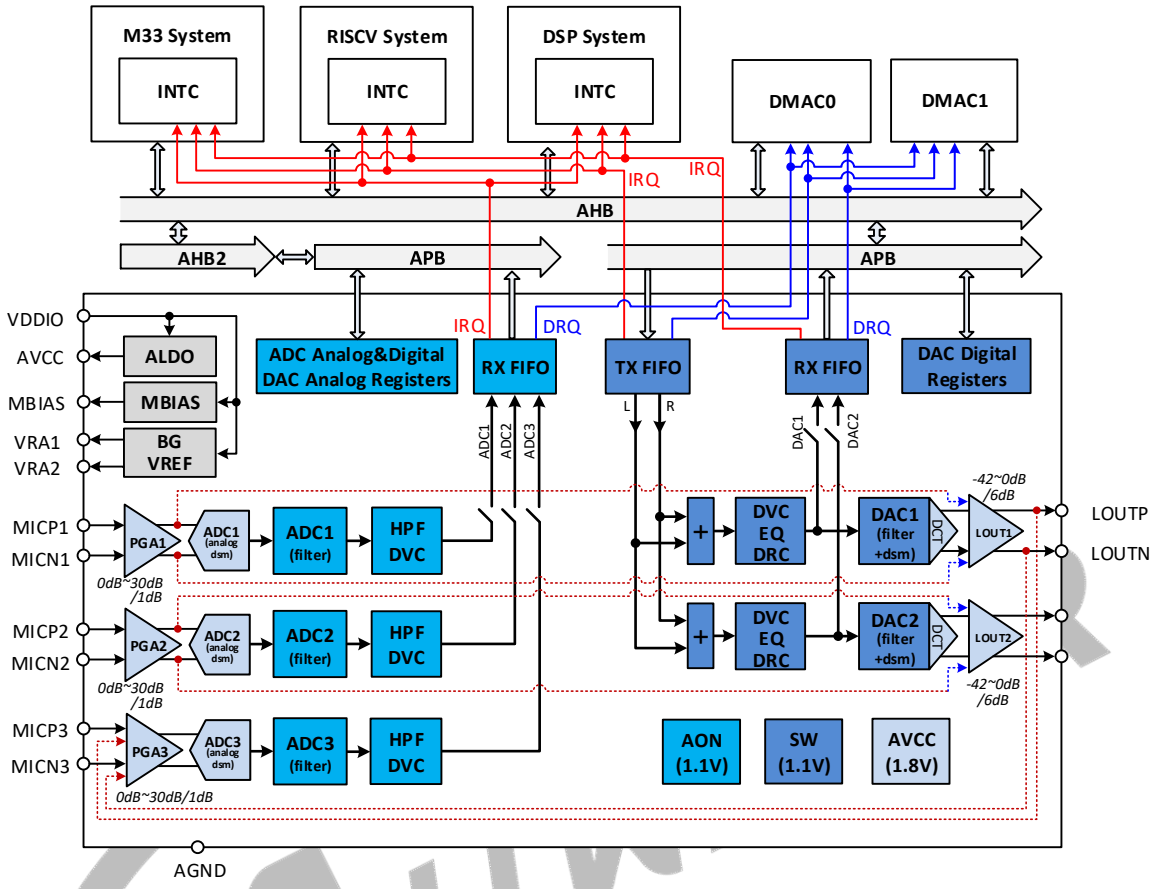


Figure 7-24 Block Diagram of Audio Codec for R128-S3



As shown in the above figure, Interrupt request (IRQ) is mainly used to send requests to DAP when an exception occurs in the course of data receiving and transmitting. DMA request (DRQ) is mainly used for the audio data transmission between buffer cache and FIFO.

7.4.3 Functional Description

7.4.3.1 External Signals

Table 7-13 External Signals of Audio Codec

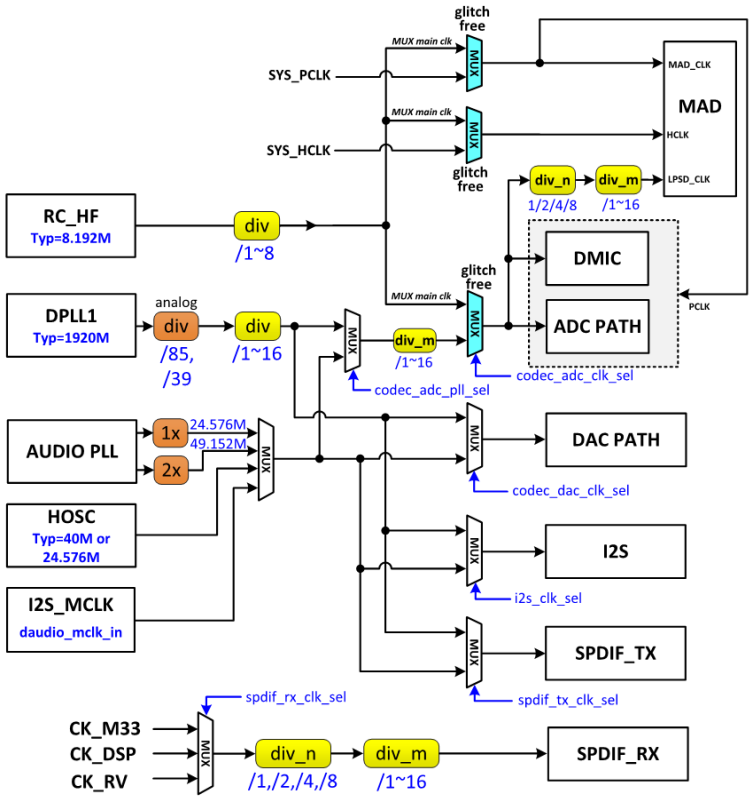
Signal	Description	Type
Analog I/O		
MICIN1P	Positive differential input for MIC1	AI
MICIN1N	Negative differential input for MIC1	AI
MICIN2P	Positive differential input for MIC2	AI
MICIN2N	Negative differential input for MIC2	AI
MICIN3P	Positive differential input for MIC3	AI
MICIN3N	Negative differential input for MIC3	AI
LOUTLP	Line-out amplifier differential Positive Channel 1	AO
LOUTLN	Line-out amplifier differential Negative Channel 1	AO

Signal	Description	Type
LOUTRP	Line-out amplifier differential Positive Channel 2	AO
LOUTRN	Line-out amplifier differential Negative Channel 2	AO
Reference		
VRA1	Internal reference voltage 1	AO
VRA2	Internal reference voltage 2	AO
MBIAS	Microphone bias voltage output	AO
Power/Ground		
AVCC	The ALDO Power output 1.8V for analog part	P
AGND	Analog Ground	G

7.4.3.2 Clock Sources

System configures the control register of CODEC and reads/writes FIFO data with the APB bus. ADC and DAC have the independent control bits of bus clocks, which should be enabled before DAP accesses registers. As ADC path is at the AON domain, it still can normally work when the whole system is in the standby status. As DAC path is at the SW domain, it still can normally work when the whole system is in the standby status.

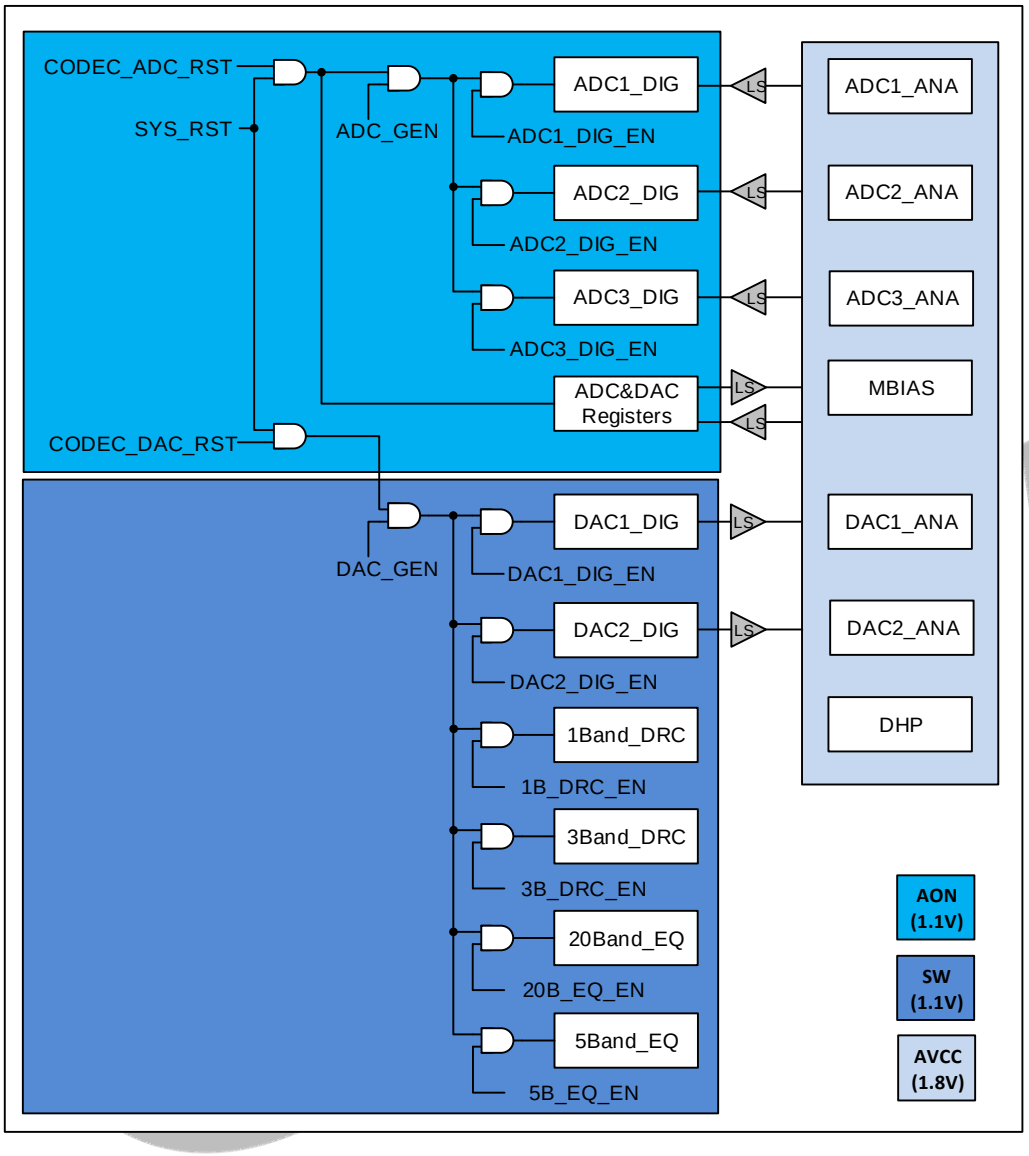
Figure 7-25 Clock Diagram of Audio Codec



7.4.3.3 Reset System

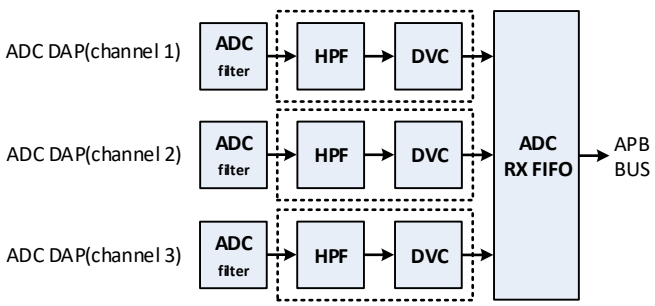
The following figure shows the reset system of the audio codec.

Figure 7-26 Reset System of Audio Codec



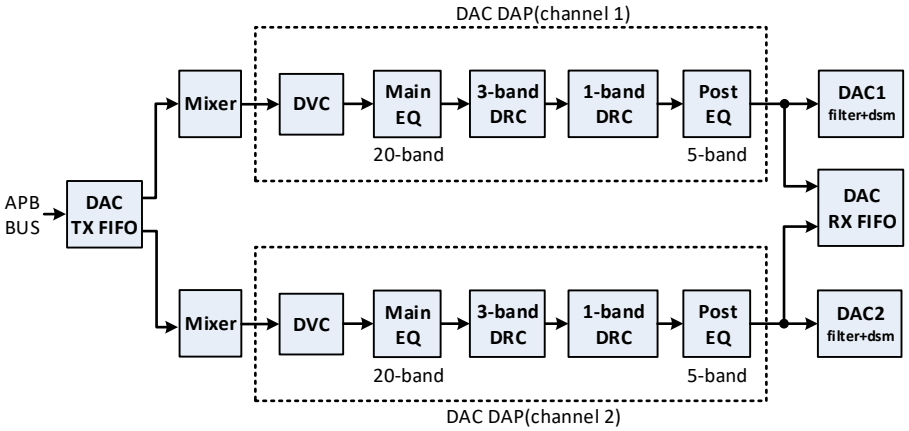
7.4.3.4 ADC Data Path

The ADC data path supports independently dealing dual-channel audio data. High-pass filter (HPF) is enabled by default and can be bypassed through register configuration.



7.4.3.5 DAC Data Path

The DAC data path supports independently dealing dual-channel audio data. ALL data path sub modules can be selected or bypassed.



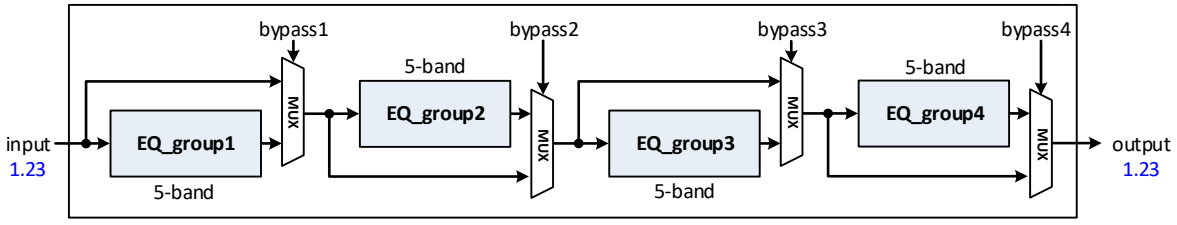
Digital Volume Control

Digital Volume Control (DVC) supports the gain adjustment range (-64dB ~ +63dB) and 0.5dB adjustment accuracy, and has the zero-crossing detection to prevent introducing pop-click noise. The default gain of DVC is 0dB.

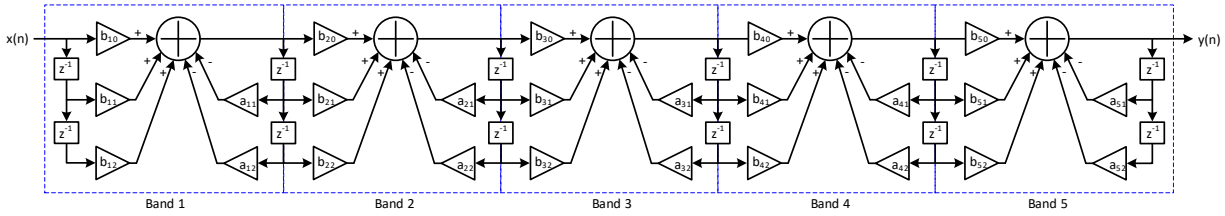
Desired Gain(dB)	DVC.Reg(Dec)	DVC.Reg(Hex)
+63	255	FF
+62.5	254	FE
…	…	…
+0.5	130	82
0	129	81
-0.5	128	80
…	…	…
-63.5	2	2
-64	1	1
mute	0	0

Main Equalizer

The main equalizer (EQ) is a cascade of 20 biquad IIR filters, and the parameters of each filter can be configured by registers. The filters consist of 4 subfilter groups. Each of them can be selected or bypassed to meet various application scenarios.



The block diagram of each subfilter group is as follows:

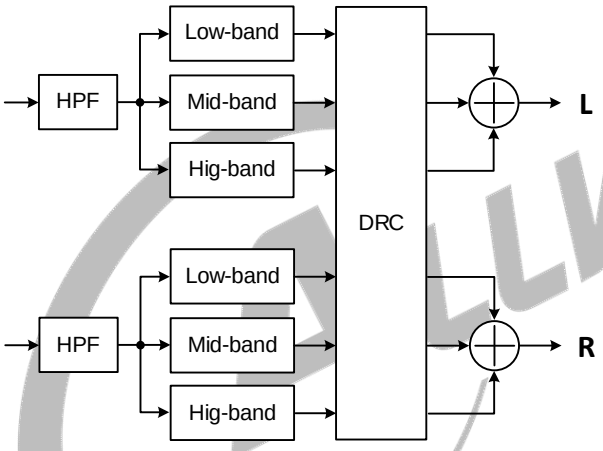


The equation of transmitting each biquad IIR filter is as follows:

$$H(z) = \frac{b0 + b1 \cdot z^{-1} + b2 \cdot z^{-2}}{1 + a1 \cdot z^{-1} + a2 \cdot z^{-2}}$$

3-band Dynamic Range Controller

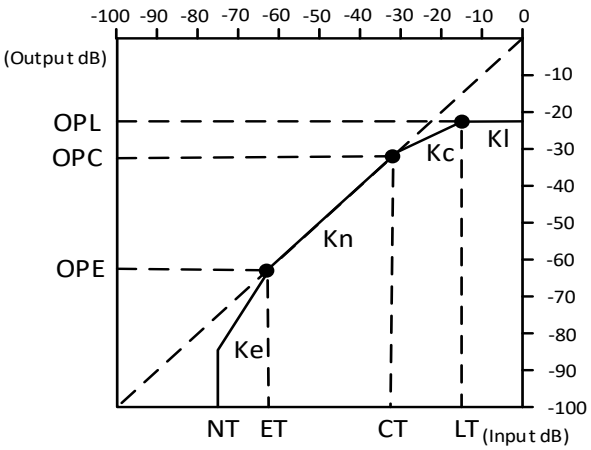
3-band dynamic range controller (DRC) is equipped with Linkwitz-Riley crossover filter, which supports outputting low, medium, and high frequency range data. Each frequency range has independent dynamic range control and weighted output. The filter coefficient can be obtained by specifying the number of dividing frequencies, the intersection frequencies and their attenuation slope.



1-band Dynamic Range Controller

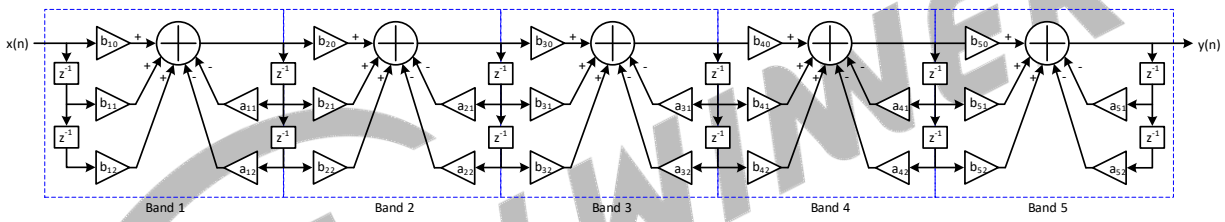
1-band DRC can automatically adjust the dynamic range of signals and divide the input signal amplitude into four segments of expansion area, linear area, compression zone and limited pressure zone. The expansion area is mainly used to distinguish relatively low sound and noise, which can greatly reduce the small environmental noise and improve the recording effect. The compression zone compresses sound from a larger loudness range into a smaller loudness range, which is suitable for quiet environments or earphones to avoid hearing discomfort caused by excessive loudness jump. The voltage limiting area mainly limits the output amplitude to a certain range.

Figure 7-27 DRC Static Curve Parameters



Post Equalizer

The post equalizer is a cascade of 5 biquad IIR filters.



7.4.4 Register List

Module name	Base Address
CODEC_ADC	0x4004B000
CODEC_DAC	0x4003D000

Register name	Offset Address	Descriptions
CODEC ADC PATH		
AC_POWER_CTRL	0x0000	Audio Codec Power Control Register
AC_MBIAS_CTRL	0x0008	MBIAS Control Register
AC_ADC_ANA_CTRL1	0x000C	ADC Analog Control 1 Register
AC_ADC_ANA_CTRL2	0x0010	ADC Analog Control 2 Register
AC_ADC_ANA_CTRL3	0x0014	ADC Analog Control 3 Register
AC_DAC_ANA_CTRL	0x0018	DAC Analog Control Register
AC_DHP_ANA_CTRL	0x001C	DHP Analog Control Register
AC_ADC_DIG_CTRL	0x0100	ADC Digital Control Register
AC_ADC_HPF_CTRL1	0x0104	ADC HPF Control 1 Register
AC_ADC_HPF_CTRL2	0x0108	ADC HPF Control 2 Register
AC_ADC_DIG_VOL	0x010C	ADC Digital Volume Control Register
AC_ADC_RXFIFO_CTRL	0x0200	ADC RX FIFO Control Register

Register name	Offset Address	Descriptions
AC_ADC_RXFIFO_STA	0x0204	ADC RX FIFO Status Register
AC_ADC_RXFIFO	0x0208	ADC RX DATA Register
AC_ADC_RXCNT	0x020C	ADC RX Write Counter Register
AC_ADC_DEBUG	0x0300	ADC Debug Control Register
CODEC DAC PATH		
AC_DAC_DIG_CTRL	0x0000	DAC Digital Control Register
AC_DAC_DIG_VOL	0x0004	DAC Digital Volume Control Register
AC_DAC_DHP_GAIN	0x0008	DAC DHP Gain Control Register
AC_DAC_TXFIFO_CTRL	0x0010	DAC TX FIFO Control Register
AC_DAC_TXFIFO_STA	0x0014	DAC TX FIFO Status Register
AC_DAC_TXFIFO	0x0018	DAC TX DATA Register
AC_DAC_TXCNT	0x001C	DAC TX Write Counter Register
AC_DAC_LBFIFO_CTRL	0x0020	DAC LOOP BACK FIFO Control Register
AC_DAC_LBFIFO_STA	0x0024	DAC LOOP BACK FIFO Status Register
AC_DAC_LBFIFO	0x0028	DAC RX(LOOP BACK) DATA Register
AC_DAC_LBCNT	0x002C	DAC RX(LOOP BACK) Write Counter Register
AC_DAC_DEBUG	0x0040	DAC Debug Control Register
MAIN_EQ_EN_CTRL	0x0100	Main LR EQ Enable Control Register
MAIN_EQL_BQn_B0(n)	0x0104+0x14*(n-1)	Main EQ Left BQn B0 Coefficient Register
MAIN_EQL_BQn_B1(n)	0x0108+0x14*(n-1)	Main EQ Left BQn B1 Coefficient Register
MAIN_EQL_BQn_B2(n)	0x010C+0x14*(n-1)	Main EQ Left BQn B2 Coefficient Register
MAIN_EQL_BQn_A1(n)	0x0110+0x14*(n-1)	Main EQ Left BQn A1 Coefficient Register
MAIN_EQL_BQn_A2(n)	0x0114+0x14*(n-1)	Main EQ Left BQn A2 Coefficient Register
MAIN_EQR_BQn_B0(n)	0x0300+0x14*(n-1)	Main EQ Right BQn B0 Coefficient Register
MAIN_EQR_BQn_B1(n)	0x0304+0x14*(n-1)	Main EQ Right BQn B1 Coefficient Register
MAIN_EQR_BQn_B2(n)	0x0308+0x14*(n-1)	Main EQ Right BQn B2 Coefficient Register
MAIN_EQR_BQn_A1(n)	0x030C+0x14*(n-1)	Main EQ Right BQn A1 Coefficient Register
MAIN_EQR_BQn_A2(n)	0x0310+0x14*(n-1)	Main EQ Right BQn A2 Coefficient Register
POST_EQ_EN_CTRL	0x0500	Post LR EQ Enable Control Register
POST_EQ_BQn_B0(n)	0x0504+0x14*(n-1)	Post EQ BQn B0 Coefficient Register
POST_EQ_BQn_B1(n)	0x0508+0x14*(n-1)	Post EQ BQn B1 Coefficient Register
POST_EQ_BQn_B2(n)	0x050C+0x14*(n-1)	Post EQ BQn B2 Coefficient Register
POST_EQ_BQn_A1(n)	0x0510+0x14*(n-1)	Post EQ BQn A1 Coefficient Register
POST_EQ_BQn_A2(n)	0x0514+0x14*(n-1)	Post EQ BQn A2 Coefficient Register
_1B_DRC_EN_CTRL	0x0600	1-Band DRC Enable Control Register
1B_DRC_HPFC	0x0604	1-Band DRC HPF Coef Register
1B_DRC_CTRL	0x0608	1-Band DRC Control Register
1B_DRC_LPFAT	0x060C	1-Band DRC Left Peak Filter Attack Time Coef Register
1B_DRC_RPFAT	0x0610	1-Band DRC Right Peak Filter Attack Time Coef Register
1B_DRC_LPFRT	0x0614	1-Band DRC Left Peak Filter Release Time Coef Register
1B_DRC_RPFRT	0x0618	1-Band DRC Right Peak Filter Release Time Coef

Register name	Offset Address	Descriptions
		Register
1B_DRC_LRMSAT	0x061C	1-Band DRC Left RMS Filter Average Time Coef Register
1B_DRC_RRMSAT	0x0620	1-Band DRC Right RMS Filter Average Time Coef Register
1B_DRC_CT	0x0624	1-Band DRC Compressor Threshold Setting Register
1B_DRC_KC	0x0628	1-Band DRC Compressor Slope Setting Register
1B_DRC_OPC	0x062C	1-Band DRC Compressor Output at Compressor Threshold Register
1B_DRC_LT	0x0630	1-Band DRC Limiter Threshold Setting Register
1B_DRC_KL	0x0634	1-Band DRC Limiter Slope Setting Register
1B_DRC_OPL	0x0638	1-Band DRC Limiter Output at Limiter Threshold
1B_DRC_ET	0x063C	1-Band DRC Expander Threshold Setting Register
1B_DRC_KE	0x0640	1-Band DRC Expander Slope Setting Register
1B_DRC_OPE	0x0644	1-Band DRC Expander Output at Expander Threshold
1B_DRC_KN	0x0648	1-Band DRC Linear Slope Setting Register
1B_DRC_SFAT	0x064C	1-Band DRC Smooth filter Gain Attack Time Coef Register
1B_DRC_SFRT	0x0650	1-Band DRC Smooth filter Gain Release Time Coef Register
1B_DRC_MXGS	0x0654	1-Band DRC MAX Gain Setting Register
1B_DRC_MNGS	0x0658	1-Band DRC MIN Gain Setting Register
1B_DRC_EPSC	0x065C	1-Band DRC Expander Smooth Time Coef Register
3B_DRC_EN_CTRL	0x0700	3-Band DRC Enable Control Register
3B_DRC_CTRL	0x0704	3-Band DRC Control Register
3B_DRC_DCF_COE	0x0708	3-Band DRC DC Filter Coefficient Register
3B_DRC_IIR_B1(n=0~7)	0x0710 + 0x0004*n	3-Band DRC IIR Filter B1 Coefficient Register
3B_DRC_IIR_B2(n=0~7)	0x0730 + 0x0004*n	3-Band DRC IIR Filter B2 Coefficient Register
3B_DRC_IIR_B3(n=0~7)	0x0750 + 0x0004*n	3-Band DRC IIR Filter B3 Coefficient Register
3B_DRC_IIR_A2(n=0~7)	0x0770 + 0x0004*n	3-Band DRC IIR Filter A2 Coefficient Register
3B_DRC_IIR_A3(n=0~7)	0x0790 + 0x0004*n	3-Band DRC IIR Filter A3 Coefficient Register
3B_DRC_IIR_CFG	0x07B0	3-Band DRC IIR Filter Configuration Register
3B_DRC0_RMS_COE	0x07B4	3-Band DRC0 Combine Control Register
3B_DRC0_RMS_COE	0x07B8	3-Band DRC0 RMS Filter Coefficient Register
3B_DRC0_CURVE_CFG0	0x07BC	3-Band DRC0 Curve Configuration 0 Register
3B_DRC0_CURVE_CFG1	0x07C0	3-Band DRC0 Curve Configuration 1 Register
3B_DRC0_CURVE_CFG2	0x07C4	3-Band DRC0 Curve Configuration 2 Register
3B_DRC0_CURVE_CFG3	0x07C8	3-Band DRC0 Curve Configuration 3 Register
3B_DRC0_CURVE_CFG4	0x07CC	3-Band DRC0 Curve Configuration 4 Register
3B_DRC0_CURVE_CFG5	0x07D0	3-Band DRC0 Curve Configuration 5 Register
3B_DRC0_SMOTH_CFG0	0x07D4	3-Band DRC0 Smooth Configuration 0 Register
3B_DRC0_SMOTH_CFG1	0x07D8	3-Band DRC0 Smooth Configuration 1 Register
3B_DRC0_SMOTH_CFG2	0x07DC	3-Band DRC0 Smooth Configuration 2 Register

Register name	Offset Address	Descriptions
3B_DRC1_RMS_COE	0x07E0	3-Band DRC1 Combine Control Register
3B_DRC1_RMS_COE	0x07E4	3-Band DRC1 RMS Filter Coefficient Register
3B_DRC1_CURVE_CFG0	0x07E8	3-Band DRC1 Curve Configuration 0 Register
3B_DRC1_CURVE_CFG1	0x07EC	3-Band DRC1 Curve Configuration 1 Register
3B_DRC1_CURVE_CFG2	0x07F0	3-Band DRC1 Curve Configuration 2 Register
3B_DRC1_CURVE_CFG3	0x07F4	3-Band DRC1 Curve Configuration 3 Register
3B_DRC1_CURVE_CFG4	0x07F8	3-Band DRC1 Curve Configuration 4 Register
3B_DRC1_CURVE_CFG5	0x07FC	3-Band DRC1 Curve Configuration 5 Register
3B_DRC1_SMOTH_CFG0	0x0800	3-Band DRC1 Smooth Configuration 0 Register
3B_DRC1_SMOTH_CFG1	0x0804	3-Band DRC1 Smooth Configuration 1 Register
3B_DRC1_SMOTH_CFG2	0x0808	3-Band DRC1 Smooth Configuration 2 Register
3B_DRC1_RMS_COE	0x080C	3-Band DRC2 Combine Control Register
3B_DRC1_RMS_COE	0x0810	3-Band DRC2 RMS Filter Coefficient Register
3B_DRC1_CURVE_CFG0	0x0814	3-Band DRC2 Curve Configuration 0 Register
3B_DRC1_CURVE_CFG1	0x0818	3-Band DRC2 Curve Configuration 1 Register
3B_DRC1_CURVE_CFG2	0x081C	3-Band DRC2 Curve Configuration 2 Register
3B_DRC1_CURVE_CFG3	0x0820	3-Band DRC2 Curve Configuration 3 Register
3B_DRC1_CURVE_CFG4	0x0824	3-Band DRC2 Curve Configuration 4 Register
3B_DRC1_CURVE_CFG5	0x0828	3-Band DRC2 Curve Configuration 5 Register
3B_DRC1_SMOTH_CFG0	0x082C	3-Band DRC2 Smooth Configuration 0 Register
3B_DRC1_SMOTH_CFG1	0x0830	3-Band DRC2 Smooth Configuration 1 Register
3B_DRC1_SMOTH_CFG2	0x0834	3-Band DRC2 Smooth Configuration 2 Register

7.4.5 CODEC ADC PATH Register Description

7.4.5.1 0x0000 Audio Codec Power Control Register (Default Value: 0x0010_182A)

Offset: 0x0000			Register Name: AC_POWER_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	ADDA_BIAS_EN ADDA Bias Current Enable 0: Disable 1: Enable
30:22	/	/	/
21:20	R/W	0x1	VRA1_SPEEDUP_TIME(@clk32k) 00: 640 cycles 01: 1120 cycles 10: 1540 cycles 11: 1960 cycles
19	R	0x0	VRA1_SPEEDUP_STA (delay 20ms after VRA1_EN) 0: Speed up not done 1: Speed up done

Offset: 0x0000			Register Name: AC_POWER_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
18	R/W	0x0	ALDO_EN ADLDO Enable 0: Disable 1: Enable
17:15	R/W	0x0	ALDO_VCTRL ALDO Output Voltage Control 000: 1.8V 001: 1.6V 010: 1.65V 011: 1.7V 100: 1.75V 101: 1.85V 110: 1.9V 111: 1.95V
14	R/W	0x0	ALDO_BYPASS ALDO Bypass Mode Enable 0: ALDO in normal mode 1: ALDO in bypass mode
13	R/W	0x0	VRA1_EN (BG_EN) Enable VRA1/BG 0: Disable 1: Enable
12:11	R/W	0x3	VRA1_VCTRL (BG_VSEL) VRA1 Output Voltage Control 00: VRA1=1.4V 01: VRA1=1.25V 10: VRA1=1.1V 11: VRA1=0.9V
10:7	R/W	0x0	BG_TRIM Trim of Output Voltage of VRA1/BG 0000: Min 1111: Max
6:5	R/W	0x1	IOPVRS OPVR Bias Current Control 00: 6 uA 01: 7 uA 10: 8 uA 11: 9 uA

Offset: 0x0000			Register Name: AC_POWER_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
4:3	R/W	0x1	IOPDACS DAC OP Bias Current Control 00: 6 uA 01: 7 uA 10: 8 uA 11: 9 uA
2:1	R/W	0x1	IDHPS DHP OP Bias Current Control 00: 6 uA 01: 7 uA 10: 8 uA 11: 9 uA
0	R/W	0x0	ALDO_LQ

7.4.5.2 0x0008 MBIAS Control Register (Default Value: 0x0000_0030)

Offset: 0x0008			Register Name: AC_MBIAS_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	MBIAS_EN Microphone Bias Enable 0: Disable 1: Enable
6:5	R/W	0x1	MBIAS_VOL_SEL MMICBIAS Voltage Level Select 00: 1.88V 01: 2.09V 10: 2.33V 11: 2.50V
4	R/W	0x1	MBIAS chopper enable 0: Disable 1: Enable
3:2	R/W	0x0	MBIAS chopper clock select 00: 250 kHz 01: 500 kHz 10: 1 MHz 11: 2 MHz
1: 0	R/W	0x0	Reserved

7.4.5.3 0x000C ADC Analog Control 1 Register (Default Value: 0x0003_C000)

Offset: 0x000C	Register Name: AC_ADC_ANA_CTRL1

Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	ADC1_EN ADC1 Channel Global Enable 0: ADC (PGA&DSM) Disable 1: ADC (PGA) Enable
30	R/W	0x0	ADC1/2/3_PGA_CHOPPER_EDGE ADC1/2/3 PGA Chopper Clock Edge Control 0: In phase with ckadc 1: Out of phase with ckadc
29	R/W	0x0	ADC1_DSM_DIS ADC1 DSM Disable Control, enable/disable Only DSM 0: Normal operation 1: Disable DSM modulator
28	R/W	0x0	ADC1_DSM_DEMOFF ADC1 DEM Control 0: Enable DEM 1: Disable DEM
27	R/W	0x0	ADC1/2/3_DSM_EN_DITHER 0: Dither Off 1: Dither On
26:24	R/W	0x0	ADC1/2/3_DSM_DITHER_LVL Dither Level Control (Dither level is positive related to the ctrl bits) 000: Min Level 111: Max Level
23	R/W	0x0	ADC1_DSM_SEL_OUT_EDGE ADC1 clock edge 0: DSM output is clocked on falling edge of input clock 1: DSM output is clocked on rising edge of input clock
22:20	R/W	0x0	ADC1_DSM_OTA_IB_SEL Control bias current for DSM integrator op-amps 00: 6.0 uA 4: 4.0 uA 01: 6.5 uA 5: 4.5 uA 10: 7.0 uA 6: 5.0 uA 11: 7.5 uA 7: 5.5 uA
19:18	R/W	0x0	ADC1_DSM_OTA_CTRL Op number control for ADC1 DSM 000: Max 011: Min
17:16	R/W	0x3	ADC1_PGA_OP_BIAS_CTRL Control ADC1 PGA op-amp Bias CMOS Size 00: 1 parallel CMOS unit enabled 01: 3 parallel CMOS unit enabled 10: 5 parallel CMOS unit enabled 11: 7 parallel CMOS unit enabled

Offset: 0x000C			Register Name: AC_ADC_ANA_CTRL1
Bit	**Read/Write**	**Default/Hex**	**Description**
15:13	R/W	0x6	ADC1_PGA_OTA_IB_SEL Control bias current of OTA for PGA 00: 1 uA 4: 5 uA 01: 2 uA 5: 6 uA 10: 3 uA 6: 7 uA 11: 4 uA 7: 8 uA
12:8	R/W	0x0	ADC1_PGA_GAIN_CTRL ADC1 PGA Gain Setting for MIC-IN: 0x0: 0 dB 0x10: 21 dB 0x1: 6 dB 0x11: 22 dB 0x2: 6 dB 0x12: 23 dB 0x3: 6 dB 0x13: 24 dB 0x4: 9 dB 0x14: 25 dB 0x5: 10 dB 0x15: 26 dB 0x6: 11 dB 0x16: 27 dB 0x7: 12 dB 0x17: 28 dB 0x8: 13 dB 0x18: 29 dB 0x9: 14 dB 0x19: 30 dB 0xA: 15 dB 0x1A: 31 dB 0xB: 16 dB 0x1B: 32 dB 0xC: 17 dB 0x1C: 33 dB 0xD: 18 dB 0x1D: 34 dB 0xE: 19 dB 0x1E: 35 dB 0xF: 20 dB 0x1F: 36 dB
7	R/W	0x0	ADC1_PGA_CHOPPER_ENABLE ADC1 PGA chopper enable 0: Chopper disable 1: Chopper enable
6	R/W	0x0	ADC1_PGA_CHOPPER_NOL_ENABLE ADC1 PGA chopper non-overlapping clock enable 0: Chopper non-overlapping clock disabled 1: Chopper non-overlapping clock enabled
5:4	R/W	0x0	ADC1_PGA_CHOPPER_CKSET Control ADC1 PGA chopper clock frequency Fs:3.072MHz/2.048MHz 00: 384 kHz/256 kHz 01: 192 kHz/128 kHz 10: 96 kHz/64 kHz 11: 48 kHz/32 kHz
3:2	R/W	0x0	ADC1_PGA_CHOPPER_DELAY_SET Control ADC1 PGA chopper clock delay time after CKADC 00: Min 11: Max

Offset: 0x000C			Register Name: AC_ADC_ANA_CTRL1
Bit	**Read/Write**	**Default/Hex**	**Description**
1: 0	R/W	0x0	ADC1_PGA_CHOPPER_NOL_DELAY_SET Control ADC1 PGA chopper clock non-overlapping time 00: Min when ADC1_PGA_CHOPPER_NOL_ENABLE=1 11: Max when ADC1_PGA_CHOPPER_NOL_ENABLE=1

7.4.5.4 0x0010 ADC Analog Control 2 Register (Default Value: 0x0003_C000)

Offset: 0x0010			Register Name: AC_ADC_ANA_CTRL2
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	ADC2_EN ADC2 Channel Global Enable 0: ADC(PGA&DSM) Disable 1: ADC(PGA) Enable
30	R/W	0x0	Reserved
29	R/W	0x0	ADC2_DSM_DIS ADC2 DSM Disable Control, enable/disable Only DSM 0: Normal operation 1: Disable DSM modulator
28	R/W	0x0	ADC2_DSM_DEMOFF ADC2 DEM Control 0: Enable DEM 1: Disable DEM
27:24	R/W	0x0	Reserved
23	R/W	0x0	ADC2_DSM_SEL_OUT_EDGE ADC2 clock edge 0: DSM output is clocked on falling edge of input clock 1: DSM output is clocked on rising edge of input clock
22:20	R/W	0x0	ADC2_DSM_OTA_IB_SEL Control bias current for DSM integrator op-amps 0: 6.0 uA 4: 4.0 uA 1: 6.5 uA 5: 4.5 uA 2: 7.0 uA 6: 5.0 uA 3: 7.5 uA 7: 5.5 uA
19:18	R/W	0x0	ADC2_DSM_OTA_CTRL Op number control for ADC1 DSM 0: Max 3: Min

Offset: 0x0010			Register Name: AC_ADC_ANA_CTRL2
Bit	**Read/Write**	**Default/Hex**	**Description**
17:16	R/W	0x3	ADC2_PGA_OP_BIAS_CTRL Control ADC2 PGA op-amp Bias CMOS Size 0: 1 parallel CMOS unit enabled 1: 3 parallel CMOS unit enabled 2: 5 parallel CMOS unit enabled 3: 7 parallel CMOS unit enabled
15:13	R/W	0x6	ADC2_PGA_OTA_IB_SEL Control bias current of OTA for PGA 0: 1 uA 4: 5 uA 1: 2 uA 5: 6 uA 2: 3 uA 6: 7 uA 3: 4 uA 7: 8 uA
12:8	R/W	0x0	ADC2_PGA_GAIN_CTRL ADC2 PGA Gain Setting for MIC-IN: 0x0: 0 dB 0x10: 21 dB 0x1: 6 dB 0x11: 22 dB 0x2: 6 dB 0x12: 23 dB 0x3: 6 dB 0x13: 24 dB 0x4: 9 dB 0x14: 25 dB 0x5: 10 dB 0x15: 26 dB 0x6: 11 dB 0x16: 27 dB 0x7: 12 dB 0x17: 28 dB 0x8: 13 dB 0x18: 29 dB 0x9: 14 dB 0x19: 30 dB 0xA: 15 dB 0x1A: 31 dB 0xB: 16 dB 0x1B: 32 dB 0xC: 17 dB 0x1C: 33 dB 0xD: 18 dB 0x1D: 34 dB 0xE: 19 dB 0x1E: 35 dB 0xF: 20 dB 0x1F: 36 dB
7	R/W	0x0	ADC2_PGA_CHOPPER_ENABLE ADC2 PGA chopper enable 0: chopper disable 1: chopper enable
6	R/W	0x0	ADC2_PGA_CHOPPER_NOL_ENABLE ADC2 PGA chopper non-overlapping clock enable 0: chopper non-overlapping clock disabled 1: chopper non-overlapping clock enabled

Offset: 0x0010			Register Name: AC_ADC_ANA_CTRL2
Bit	**Read/Write**	**Default/Hex**	**Description**
5:4	R/W	0x0	ADC2_PGA_CHOPPER_CKSET Control ADC2 PGA chopper clock frequency Fs:3.072MHz/2.048MHz 0=384 kHz/256 kHz 1=192 kHz/128 kHz 2=96 kHz/64 kHz 3=48 kHz/32 kHz
3:2	R/W	0x0	ADC2_PGA_CHOPPER_DELAY_SET Control ADC2 PGA chopper clock delay time after CKADC 0: Min 3: Max
1: 0	R/W	0x0	ADC2_PGA_CHOPPER_NOL_DELAY_SET Control ADC2 PGA chopper clock non-overlapping time 0: Min when ADC2_PGA_CHOPPER_NOL_ENABLE=1 3: Max when ADC2_PGA_CHOPPER_NOL_ENABLE=1

7.4.5.5 0x0014 ADC Analog Control 3 Register (Default Value: 0x0003_C000)

Offset: 0x0014			Register Name: AC_ADC_ANA_CTRL3
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	ADC3_EN ADC3 Channel Global Enable 0: ADC(PGA&DSM) Disable 1: ADC(PGA) Enable
30	R/W	0x0	ADC3_MIC_MIX ADC3 INPUT CHOOSE MIC or DHPL 0: MIC 1: DHP
29	R/W	0x0	ADC3_DSM_DIS ADC3 DSM Disable Control, enable/disable Only DSM 0: Normal operation 1: Disable DSM modulator
28	R/W	0x0	ADC3_DSM_DEMOFF ADC3 DEM Control 0: Enable DEM 1: Disable DEM
27:24	R/W	0x0	Reserved
23	R/W	0x0	ADC3_DSM_SEL_OUT_EDGE ADC3 clock edge 0: DSM output is clocked on falling edge of input clock 1: DSM output is clocked on rising edge of input clock

Offset: 0x0014			Register Name: AC_ADC_ANA_CTRL3
Bit	**Read/Write**	**Default/Hex**	**Description**
22:20	R/W	0x0	ADC3_DSM_OTA_IB_SEL Control bias current for DSM integrator op-amps 0: 6.0 uA 4: 4.0 uA 1: 6.5 uA 5: 4.5 uA 2: 7.0 uA 6: 5.0 uA 3: 7.5 uA 7: 5.5 uA
19:18	R/W	0x0	ADC3_DSM_OTA_CTRL Op number control for ADC1 DSM 0: Max 3: Min
17:16	R/W	0x3	ADC3_PGA_OP_BIAS_CTRL Control ADC3 PGA op-amp Bias CMOS Size 0: 1 parallel CMOS unit enabled 1: 3 parallel CMOS unit enabled 2: 5 parallel CMOS unit enabled 3: 7 parallel CMOS unit enabled
15:13	R/W	0x6	ADC3_PGA_OTA_IB_SEL Control bias current of OTA for PGA 0: 1 uA 4: 5 uA 1: 2 uA 5: 6 uA 2: 3 uA 6: 7 uA 3: 4 uA 7: 8 uA
12:8	R/W	0x0	ADC3_PGA_GAIN_CTRL ADC3 PGA Gain Setting for MIC-IN: 0x0: 0 dB 0x10: 21 dB 0x1: 6 dB 0x11: 22 dB 0x2: 6 dB 0x12: 23 dB 0x3: 6 dB 0x13: 24 dB 0x4: 9 dB 0x14: 25 dB 0x5: 10 dB 0x15: 26 dB 0x6: 11 dB 0x16: 27 dB 0x7: 12 dB 0x17: 28 dB 0x8: 13 dB 0x18: 29 dB 0x9: 14 dB 0x19: 30 dB 0xA: 15 dB 0x1A: 31 dB 0xB: 16 dB 0x1B: 32 dB 0xC: 17 dB 0x1C: 33 dB 0xD: 18 dB 0x1D: 34 dB 0xE: 19 dB 0x1E: 35 dB 0xF: 20 dB 0x1F: 36 dB

Offset: 0x0014			Register Name: AC_ADC_ANA_CTRL3
Bit	**Read/Write**	**Default/Hex**	**Description**
7	R/W	0x0	ADC3_PGA_CHOPPER_ENABLE ADC3 PGA chopper enable 0: chopper disable 1: chopper enable
6	R/W	0x0	ADC3_PGA_CHOPPER_NOL_ENABLE ADC3 PGA chopper non-overlapping clock enable 0: chopper non-overlapping clock disabled 1: chopper non-overlapping clock enabled
5:4	R/W	0x0	ADC3_PGA_CHOPPER_CKSET Control ADC3 PGA chopper clock frequency Fs:3.072MHz/2.048MHz 0=384 kHz/256 kHz 1=192 kHz/128 kHz 2=96 kHz/64 kHz 3=48 kHz/32 kHz
3:2	R/W	0x0	ADC3_PGA_CHOPPER_DELAY_SET Control ADC3 PGA chopper clock delay time after CKADC 0: Min 3: Max
1: 0	R/W	0x0	ADC3_PGA_CHOPPER_NOL_DELAY_SET Control ADC3 PGA chopper clock non-overlapping time 0: Min when ADC3_PGA_CHOPPER_NOL_ENABLE=1 3: Max when ADC3_PGA_CHOPPER_NOL_ENABLE=1

7.4.5.6 0x0018 DAC Analog Control Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: AC_DAC_ANA_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	DACL_EN DACL Enable 0: Disable 1: Enable
30	R/W	0x0	DACR_EN DACR Enable 0: Disable 1: Enable
29:28	R/W	0x0	CKDAC_DELAY_SET clock delay time after CKADC 0: Min 3: Max
27:8	R/W	0x0	/

Offset: 0x0018			Register Name: AC_DAC_ANA_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
7	R/W	0x0	DAC_CHOPPER_ENABLE DAC chopper enable 0: chopper disable 1: chopper enable
6	R/W	0x0	DAC_CHOPPER_NOL_ENABLE DAC chopper non-overlapping clock enable 0: chopper non-overlapping clock disabled 1: chopper non-overlapping clock enabled
5:4	R/W	0x0	DAC_CHOPPER_CKSET DAC chopper clock frequency Fs:3.072MHz/2.048MHz 0=384 kHz/256 kHz 1=192 kHz/128 kHz 2=96 kHz/64 kHz 3=48 kHz/32 kHz
3:2	R/W	0x0	DAC_CHOPPER_DELAY_SET Control DAC chopper clock delay time after CKDAC 0: Min 3: Max
1: 0	R/W	0x0	DAC_CHOPPER_NOL_DELAY_SET Control DAC chopper clock non-overlapping time 0: Min when DAC_CHOPPER_NOL_ENABLE=1 3: Max when DAC_CHOPPER_NOL_ENABLE=1

7.4.5.7 0x001C DHP Analog Control Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: AC_DHP_ANA_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	DHPL_EN Left Differential Lineout Enable 0: Disable 1: Enable
30	R/W	0x0	DHPR_EN Right Differential Lineout Enable 0: Disable 1: Enable
29	R/W	0x0	DHP_SELPGA_EN Differential Lineout Input Select PGA Output 0: Disable 1: Enable
28	R/W	0x0	Reserved

Offset: 0x001C			Register Name: AC_DHP_ANA_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
27:26	R/W	0x0	DHP_OPDRV_CUR Differential Lineout OPDRV output stage current setting 00:6uA 01:7uA 10:8uA 11:9uA
25:24	R/W	0x0	DHP_OI_CTRL Differential Lineout OPDRV output stage Driver Control 00:60I 01:80I 10:100I 11:120I
23:12	/	/	/
11	R/W	0x0	DHP_CH_OUT_EDGE Differential Lineout Chopper Clock Edge Control 0: in phase with ckdac 1: out of phase with ckdac
10:8	R/W	0x0	Reserved
7	R/W	0x0	DHP_CHOPPER_ENABLE DHP chopper enable 0: chopper disable 1: chopper enable
6	R/W	0x0	DHP_CHOPPER_NOL_ENABLE DHP chopper non-overlapping clock enable 0: chopper non-overlapping clock disabled 1: chopper non-overlapping clock enabled
5:4	R/W	0x0	DHP_CHOPPER_CKSET DHP chopper clock frequency Fs:3.072MHz/2.048MHz 0=384kHz/256kHz 1=192kHz/128kHz 2=96kHz/64kHz 3=48kHz/32kHz
3:2	R/W	0x0	DHP_CHOPPER_DELAY_SET Control DHP chopper clock delay time after CKDAC 0: Min 3: Max
1: 0	R/W	0x0	DHP_CHOPPER_NOL_DELAY_SET Control DHP chopper clock non-overlapping time 0: Min when DAC_CHOPPER_NOL_ENABLE=1 3: Max when DAC_CHOPPER_NOL_ENABLE=1

7.4.5.8 0x0100 ADC Digital Control Register (Default Value: 0x0000_0C00)

Offset: 0x0100			Register Name: AC_ADC_DIG_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25	R/W	0x0	MAD_DATA_EN MAD PCM Access Enable 0: Disable 1: Enable
24	R/W	0x0	RX_SYNC_EN_START Audio Subsys RX Synchronize Enable Start. Only if RX_SYNC_EN set 1, RX_SYNC_EN_START can take effect. 0: Disable 1: Enable
23	R/W	0x0	RX_SYNC_EN Audio ADC RX Data Synchronize Enable. 0: Disable 1: Enable
22	R/W	0x0	GEN ADC global enable A disable on this bit overrides any other block or channel enables. 0: Disable 1: Enable
21	R/W	0x0	ADC1_DIG_EN. ADC Channel 1 Digital Part Enable 0: Disable 1: Enable
20	R/W	0x0	ADC2_DIG_EN. ADC Channel 2 Digital Part Enable 0: Disable 1: Enable
19	R/W	0x0	ADC3_DIG_EN. ADC Channel 3 Digital Part Enable 0: Disable 1: Enable
18	R/W	0x0	ADC1_SRC_SEL ADC Channel 1 Digital Decimation Filter Source selection 0: Amic 1: Dmic
17	R/W	0x0	ADC2_SRC_SEL ADC Channel 2 Digital Decimation Filter Source selection 0: Amic 1: Dmic

Offset: 0x0100			Register Name: AC_ADC_DIG_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
16	R/W	0x0	ADC3_SRC_SEL ADC Channel 3 Digital Decimation Filter Source selection 0: Amic 1: Dmic
15	R/W	0x0	MIC_O_SWP_1 MIC1 output data swap with MIC2 output data enable 0: Disable 1: Enable
14	R/W	0x0	MIC_O_SWP_2 MIC3 output data swap with MIC4(amic4=0) output data enable 0: Disable 1: Enable
13	R/W	0x0	DMIC_VOL_FIX Dmic volume up to 6 dB 0: Disable 1: Enable
12:9	R/W	0x6	AD_CLK (128*fs, AudioPLL=24.576MHz) AD Clock Divide 0: div1 1: div2 (96kHz) 2: div3 3: div4 (48kHz) 4: div6 (32kHz) 5: div8 (24kHz) 6: div12 (16kHz) 7: div16 (12kHz) 8: div24 (8kHz) Other: Reserved
8:7	R/W	0x0	ADC OSR 00: 128 01: 64 10: 32 11: Reserved
6:5	R/W	0x0	ADC_PTN_SEL ADC Pattern Select 00: Normal 01: 0x5A5A5A(24-bit) 10: 0x123456 (24-bit) 11: Zero data When this bit selected, the digital path data source will be changed to internal rom data.

Offset: 0x0100			Register Name: AC_ADC_DIG_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
4:3	R/W	0x0	ADC_FD_MODE ADC FIFO Delay Mode for writing Data after GEN&ADC_DIG_EN 00: Not delayed 01: Do not write RXFIFO during the delay 10: Each channel writes 0 to RXFIFO during the delay 11: Reserved
2: 0	R/W	0x0	ADC_FDT ADC FIFO Delay Time for writing Data after GEN&ADC_DIG_EN (Fs=48kHz) 000:1.7 ms 001: 3.3 ms 010: 5 ms 011: 6.7 ms 100: 8.3 ms 101: 10 ms 110: 20 ms 111: 30 ms (Fs=16kHz) 000: 5 ms 001: 10 ms 010: 15 ms 011: 20 ms 100: 25 ms 101: 30 ms 110: 60 ms 111: 90 ms

7.4.5.9 0x0104 ADC HPF Control 1 Register (Default Value: 0x1002_A005)

Offset: 0x0104			Register Name: AC_ADC_HPF_CTRL1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29	R/W	0x0	HPF1 ENABLE 0: Disable 1: Enable
28:15	R/W	0x2005	HPF1 coefficient 1.13 format α = -0.999345715621208 Fpass = 5Hz
14	R/W	0x0	HPF2 ENABLE 0: Disable 1: Enable
13: 0	R/W	0x2005	HPF2 coefficient 1.13 format α = -0.999345715621208 Fpass = 5Hz

7.4.5.10 0x0108 ADC HPF Control 2 Register (Default Value: 0x0000_2005)

Offset: 0x0108			Register Name:AC_ADC_HPF_CTRL2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:15	/	/	/
14	R/W	0x0	HPF3 ENABLE 0: Disable 1: Enable
13: 0	R/W	0x2005	HPF3 coefficient 1.13 format α = -0.999345715621208 Fpass = 5Hz

7.4.5.11 0x010C ADC Digital Volume Control Register (Default Value: 0x0181_8181)

Offset: 0x010C			Register Name: AC_ADC_VOL_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	
24	R/W	0x1	ADC_DVC_ZCD_EN ADC DVC Zero Cross Detect Enable 0: Disable 1: Enable
23:16	R/W	0x81	ADC Ch1 digital volume (-64dB to 63dB, 0.5dB/Step) 0x00: Mute 0x01: -64 dB …… 0x80= -0.5 dB 0x81= 0 dB 0x82= 0.5 dB …… 0xFF= 63 dB
15:8	R/W	0x81	ADC Ch2 digital volume (-64dB to 63dB, 0.5dB/Step) 0x00: Mute 0x01: -64 dB …… 0x80= -0.5 dB 0x81= 0 dB 0x82= 0.5 dB …… 0xFF= 63 dB

Offset: 0x010C			Register Name: AC_ADC_VOL_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
7: 0	R/W	0x81	ADC Ch3 digital volume (-64dB to 63dB, 0.5dB/Step) 0x00: Mute 0x01: -64 dB …… 0x80= -0.5 dB 0x81= 0 dB 0x82= 0.5 dB …… 0xFF= 63 dB

7.4.5.12 0x0200 ADC RX FIFO Control Register (Default Value: 0x0000_0400)

Offset: 0x0200			Register Name: AC_ADC_FIFO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	ADC_RX_FIFO_CH1_EN RX FIFO channel 1 enable 0: Disable 1: Enable
15	R/W	0x0	ADC_RX_FIFO_CH2_EN RX FIFO channel 2 enable 0: Disable 1: Enable
14	R/W	0x0	ADC_RX_FIFO_CH3_EN RX FIFO channel 3 enable 0: Disable 1: Enable
13	R/W	0x0	ADC_RX_FIFO_CH4_EN RX FIFO channel 4 enable (just for channel sync and force ch4 rxdata == 0) 0: Disable 1: Enable
12	R/W	0x0	ADC_RX_SAMPLE_BITS. Receiving Audio Sample Resolution 0: 16-bit 1: 24-bit

Offset: 0x0200			Register Name: AC_ADC_FIFO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
11	R/W	0x0	ADC_RX_FIFO_MODE. ADC RX FIFO Output Mode (Mode 0, 1) 0: Expanding '0' at LSB of RX FIFO register 1: Expanding received sample sign bit at MSB of RX FIFO register For 24-bits received audio sample: Mode 0: RXDATA [31: 0] = {FIFO_O[23: 0], 8'h0} Mode 1: Reserved For 16-bits received audio sample: Mode 0: RXDATA [31: 0] = {FIFO_O[23:8], 16'h0} Mode 1: RXDATA [31: 0] = {16{FIFO_O[23]}, FIFO_O[23:8]}
10:4	R/W	0x40	ADC_RX_FIFO_TRG_LEVEL. ADC RX FIFO Trigger Level (RXTL[6: 0]) Interrupt and DMA request trigger level for RX FIFO normal condition IRQ/DRQ Generated when WLEVEL ＞ RXTL[6: 0] **Note**: *WLEVEL represents the number of valid samples in the RX FIFO.*
3	R/W	0x0	ADC_DRQ_EN. ADC FIFO Data Available DRQ Enable. 0: Disable 1: Enable
2	R/W	0x0	ADC_IRQ_EN. ADC FIFO Data Available IRQ Enable. 0: Disable 1: Enable
1	R/W	0x0	ADC_OVERRUN_IRQ_EN. ADC FIFO Over Run IRQ Enable 0: Disable 1: Enable
0	W1C	0x0	ADC_FIFO_FLUSH. ADC FIFO Flush. Write '1' to flush RX FIFO, self clear to'0'.

7.4.5.13 0x0204 ADC RX FIFO Status Register (Default Value: 0xC000_0000)

Offset: 0x0204			Register Name: AC_ADC_FIFO_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x1	ADC_RX_EMPTY. ADC RX FIFO Empty 0: More than one sample point in RX FIFO (>= 1 word) 1: No anyone sample point in RX FIFO

Offset: 0x0204			Register Name: AC_ADC_FIFO_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
30	R	0x1	mad_data_align 0: misalign 1: Align When the mad data output is changed to apb output, the software will send the order of apb reading data in the case of mad_data_align = 1.
29:16	/	/	/
15:8	R	0x0	RXA_CNT. ADC RX FIFO Available Sample Word Counter
7:2	/	/	/
1	R/WC	0x0	RXA_INT. ADC RX FIFO Data Available Pending Interrupt 0: No Pending IRQ 1: Data Available Pending IRQ Write '1' to clear this interrupt or automatic clear if interrupt condition fails.
0	R/WC	0x0	RXO_INT. ADC RX FIFO Overrun Pending Interrupt 0: No Pending IRQ 1: FIFO Overrun Pending IRQ Write '1' to clear this interrupt

7.4.5.14 0x0208 ADC RX DATA Register (Default Value: 0x0000_0000)

Offset: 0x0208			Register Name: AC_ADC_RXDATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	ADC_RX_DATA. ADC RX Sample Host can get one sample by reading this register. The 1st channel sample data is first and then the 2nd channel sample, etc.

7.4.5.15 0x020C ADC RX Write Counter Register (Default Value: 0x0000_0000)

Offset: 0x020C			Register Name: AC_ADC_RX_WR_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x020C			Register Name: AC_ADC_RX_WR_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	ADC_RX_WR_CNT. ADC RX Write Sample Counter The audio sample number of writing into ADC 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 Synchronization*

7.4.6 CODEC DAC PATH Register Description

7.4.6.1 0x0300 ADC Debug Control Register (Default Value: 0x0000_0000)

Offset: 0x0300			Register Name: AC_ADC_DBG_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	W	0x0	Codec ADC Debug Key Write '0x1883' to enable 'ADC Debug enable' to be writable
15:6	/	/	/
5:4	R/W	0x0	DSM_DITHER_CTRL Control of the input pin: dsm_dither_sign and dsm_dither_data 00: single-stage shaped pseudorandom dither signal 01: original pseudorandom dither signal 10: offset voltage of the positive direction 11: offset voltage of the negative direction
3	R/W	0x0	ADC Debug enable 0: Disable 1: Enable
2:1	R/W	0x0	Reserved
0	R/W	0x0	ADC Debug Mode Select 0: ADC analog debug (sdm_clk input, sdm_dat output) 1: ADC digital debug (sdm_clk output, sdm_dat input)

7.4.6.2 0x0000 DAC Digital Control Register (Default Value: 0x0000_2184)

Offset: 0x0000			Register Name: AC_DAC_DIG_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/

Offset: 0x0000			Register Name: AC_DAC_DIG_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
17	R/W	0x0	GEN DAC global enable A disable on this bit overrides any other block or channel enables. 0: Disable 1: Enable
16:15	R/W	0x0	TXL_MIX_CTRL 00: TXL 01: TXR 10: TXL+TXR 11: (TXL+TXR)/2
14:13	R/W	0x1	TXR_MIX_CTRL 00: TXL 01: TXR 10: TXL+TXR 11: (TXL+TXR)/2
12	R/W	0x0	DAC_L_DIG_EN. DAC Left Channel Digital Part Enable 0: Disable 1: Enable
11	R/W	0x0	DAC_R_DIG_EN. DAC Right Channel Digital Part Enable 0: Disable 1: Enable
10:7	R/W	0x3	DA_CLK (128*fs, AudioPLL=24.576 MHz) DA Clock Divide 0: div1 1: div2 (96 kHz) 2: div3 3: div4 (48 kHz) 4: div6 (32 kHz) 5: div8 (24 kHz) 6: div12 (16 kHz) 7: div16 (12 kHz) 8: div24 (8 kHz) Other: Reserved
6:5	R/W	0x0	DAC OSR 00:128 01:64 10:32 11:reserved

Offset: 0x0000			Register Name: AC_DAC_DIG_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
4:3	R/W	0x0	DAC_PTN_SEL DAC Pattern Select 00: Normal (Audio sample from TX mixer) 01: -6 dB sin wave 10: -60 dB sin wave 11: zero data
2: 0	R/W	0x4	DAC _DSDM_Dither_SGM Dither sigma 000: Reserved 001: 1/2 010: $1/2^2$ 011: $1/2^3$ 100: $1/2^4$ 101: $1/2^5$ 110: $1/2^6$ 111: Reserved

7.4.6.3 0x0004 DAC Digital Volume Control Register (Default Value: 0x0001_8181)

Offset: 0x0004			Register Name: AC_DAC_VOL_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	
16	R/W	0x1	DAC_DVC_ZCD_EN DAC DVC Zero Cross Detect Enable 0: Disable 1: Enable
15:8	R/W	0x81	DAC Left Channel digital volume (-64 dB to 63 dB, 0.5 dB/Step) 0x00: Mute 0x01: -64 dB …… 0x80 = -0.5 dB 0x81 = 0 dB 0x82 = 0.5 dB …… 0xFF = 63 dB

Offset: 0x0004			Register Name: AC_DAC_VOL_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
7: 0	R/W	0x81	DAC Right Channel digital volume (-64dB to 63dB, 0.5dB/Step) 0x00: Mute 0x01: -64 dB …… 0x80 = -0.5 dB 0x81 = 0 dB 0x82 = 0.5 dB …… 0xFF = 63 dB

7.4.6.4 0x0008 DAC DHP Gain Control Register (Default Value: 0x0000_0065)

Offset: 0x0008			Register Name: AC_DAC_DHP_GAIN_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	
10:8	R/W	0x0	DHP_Output Gain DHP Output Gain Control, Total 8 level, from 0 dB to -42 dB, 6 dB/step. 000: 0 dB 001: -6 dB 010: -12 dB 011: -18 dB 100: -24 dB 101: -30 dB 110: -36 dB 111: -42 dB
7:5	R/W	0x3	DAC Playback Mute detection in time: @1/fs 000: 16 001: 32 010: 64 011: 128 100: 256 101: 512 110: 1024 111: 2048
4:3	R/W	0x0	ATT_STEP auto attenuate time step setting 00: 128 samples 01: 32 samples 10: 16 samples 11: 1 samples

Offset: 0x0008			Register Name: AC_DAC_DHP_GAIN_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x1	LOUT_AUTO_ATT Lineout auto attenuate enable 0: Disable 1: Enable
1	R/W	0x0	LOUT_AUTO_MUTE Lineout auto mute enable Only if LOUT_AUTO_ATT set 1, Lineout auto mute enable can take effect 0: Disable 1: Enable
0	R/W	0x1	DHP_GAIN_ZC_DEN DHP Gain Zero Cross Detect Enable 0: Disable 1: Enable

7.4.6.5 0x0010 DAC TX FIFO Control Register (Default Value: 0x0000_2000)

Offset: 0x0010			Register Name: AC_DAC_TX_FIFO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
26	R/W	0x0	SEND_LASAT Audio sample select when DAC TX FIFO under run 0: Sending zero 1: Sending last audio sample
25	R/W	0X0	Reserved
24	R/W	0x0	FIFO_MODE For24-bits transmitted audio sample: 0: FIFO_I[23: 0] = {TXDATA[31:8]} 1: Reserved For 16-bits transmitted audio sample: 0: FIFO_I[23: 0] = {TXDATA[31:16], 8'b0} 1: FIFO_I[23: 0] = {TXDATA[15: 0], 8'b0}
23	R/W	0X0	Reserved
22:21	R/W	0x0	DAC_TX_DRQ_CLR_CNT When DAC TX FIFO available room 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
20:14	/	/	/

Offset: 0x0010			Register Name: AC_DAC_TX_FIFO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
13:7	R/W	0x40	DAC_TX_TRIG_LEVEL DAC TX FIFO Empty Trigger Level (TXTL[6: 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 and is only TXTL[6: 0] valid when TXMODE = 0.*
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	DAC_TX_SAMPLE_BITS Transmitting Audio Sample Resolution 0: 16 bits 1: 24 bits
4	R/W	0x0	DAC_TX_DRQ_EN DAC TX FIFO Empty DRQ Enable 0: Disable 1: Enable
3	R/W	0x0	DAC_TX_IRQ_EN DAC TX FIFO Empty IRQ Enable 0: Disable 1: Enable
2	R/W	0x0	DAC_TX_FIFO_UNDERRUN_IRQ_EN DAC TX FIFO Under Run IRQ Enable 0: Disable 1: Enable
1	R/W	0x0	DAC_TX_FIFO_OVERRUN_IRQ_EN DAC TX FIFO Over Run IRQ Enable 0: Disable 1: Enable
0	R/WC	0x0	DAC_TX_FIFO_FLUSH DAC TX FIFO Flush Write '1' to flush DAC TX FIFO, self clear to '0'

7.4.6.6 0x0014 DAC TX FIFO Status Register (Default Value: 0x0080_8008)

Offset: 0x0014			Register Name: AC_DAC_TX_FIFO_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/

Offset: 0x0014			Register Name: AC_DAC_TX_FIFO_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
23	R	0x1	DAC_TX_FIFO_EMPTY TX FIFO Empty 0: More than one sample point in DAC TX FIFO (>= 1 word) 1: No anyone sample point in DAC TX FIFO
22:16	/	/	/
15:8	R	0x80	TXE_CNT DAC TX FIFO Empty Space Word Counter
7:4	/	/	/
3	R/W1C	0x1	TXE_INT DAC TX FIFO Empty Pending Interrupt 0: No Pending IRQ 1: FIFO Empty Pending Interrupt Write '1' to clear this interrupt or automatic clear if interrupt condition fails.
2	R/W1C	0x0	TXU_INT DAC TX FIFO Under run Pending Interrupt 0: No Pending Interrupt 1: FIFO Under run Pending Interrupt Write '1' to clear this interrupt
1	R/W1C	0x0	TXO_INT DAC TX FIFO Overrun Pending Interrupt 0: No Pending Interrupt 1: FIFO Overrun Pending Interrupt Write'1'to clear this interrupt
0	/	/	/

7.4.6.7 0x0018 DAC TX DATA Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: AC_DAC_TXDATA Default Value: 0x0000 0000
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	DAC_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.

7.4.6.8 0x001C DAC TX Write Counter Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: AC_DAC_TX_WR_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x001C			Register Name: AC_DAC_TX_WR_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	DAC_TX_WR_CNT DAC TX Write Sample Counter The audio sample number of writing into DAC TX FIFO. When one sample is written by Digital Audio Engine, the TX sample counter register increases by one. The TX sample counter register can be set to any initial valve at any time. After been updated by the initial value, the counter register should count on base of this initial value.

7.4.6.9 0x0020 DAC LOOP BACK FIFO Control Register (Default Value: 0x0000_0400)

Offset: 0x0020			Register Name: AC_DAC_LB_FIFO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:15	/	/	/
14	R/W	0x0	DAC_LB_FIFO_CH1_EN DAC Loop Back FIFO channel 1 enable 0: Disable 1: Enable
13	R/W	0x0	DAC_LB_FIFO_CH2_EN DAC Loop Back FIFO channel 2 enable 0: Disable 1: Enable
12	R/W	0x0	DAC_LB_SAMPLE_BITS Receiving Audio Sample Resolution 0: 16-bit 1: 24-bit
11	R/W	0x0	DAC_LB_FIFO_MODE DAC Loop Back FIFO Output Mode (Mode 0, 1) 0: Expanding '0' at LSB of LOOP BACK FIFO register 1: Expanding received sample sign bit at MSB of Loop Back FIFO register For 24-bits received audio sample: Mode 0: RXDATA [31: 0] = {FIFO_O[23: 0], 8'h0} Mode 1: Reserved For 16-bits received audio sample: Mode 0: RXDATA [31: 0] = {FIFO_O[23:8], 16'h0} Mode 1: RXDATA [31: 0] = {16{FIFO_O[23]}, FIFO_O[23:8]}

Offset: 0x0020			Register Name: AC_DAC_LB_FIFO_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
10:4	R/W	0x40	LB_FIFO_TRG_LEVEL DAC Loop Back FIFO Trigger Level (**RXTL[6: 0]**) Interrupt and DMA request trigger level for DAC Loop Back FIFO normal condition IRQ/DRQ Generated when WLEVEL ＞ RXTL[6: 0] **Note**: *WLEVEL represents the number of valid samples in the RX FIFO*
3	R/W	0x0	DAC_LB_DRQ_EN DAC Loop Back FIFO Data Available DRQ Enable. 0: Disable 1: Enable
2	R/W	0x0	DAC_LB_IRQ_EN DAC Loop Back FIFO Data Available IRQ Enable. 0: Disable 1: Enable
1	R/W	0x0	DAC_LB_OVERRUN_IRQ_EN DAC Loop Back FIFO Over Run IRQ Enable 0: Disable 1: Enable
0	W1C	0x0	DAC_LB_FIFO_FLUSH DAC Loop Back FIFO Flush Write ‘1’ to flush DAC LOOP BACK FIFO, self clear to‘0’.

7.4.6.10 0x0024 DAC LOOP BACK FIFO Status Register (Default Value: 0x8000_0000)

Offset: 0x0024			Register Name: AC_DAC_LB_FIFO_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x1	DAC_LB_FIFO_EMPTY DAC Loop Back FIFO Empty 0: More than one sample point in RX FIFO (>= 1 word) 1: No anyone sample point in RX FIFO
30:16	/	/	/
15:8	R	0x0	LBA_CNT DAC Loop Back FIFO Available Sample Word Counter
7:2	/	/	/
1	R/WC	0x0	LBA_INT DAC Loop Back FIFO Data Available Pending Interrupt 0: No Pending IRQ 1: Data Available Pending IRQ Write ‘1’ to clear this interrupt or automatic clear if interrupt condition fails.

Offset: 0x0024			Register Name: AC_DAC_LB_FIFO_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/WC	0x0	LBO_INT DAC Loop Back FIFO Overrun Pending Interrupt 0: No Pending IRQ 1: FIFO Overrun Pending IRQ Write '1' to clear this interrupt

7.4.6.11 0x0028 DAC RX (LOOP BACK) DATA Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: AC_ADC_RXDATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	DAC_RX_DATA DAC RX Sample Host can get one sample by reading this register. The left channel sample data is first and then the right channel sample.

7.4.6.12 0x002C DAC RX (LOOP BACK) Write Counter Register (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: AC_ADC_RX_WR_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	DAC_RX_WR_CNT DAC RX Write Sample Counter The audio sample number of writing into DAC Loop Back FIFO. When one sample is written by Digital Audio Engine, the RX sample counter register increases by one. The RX sample counter register can be set to any initial valve at any time. After been updated by the initial value, the counter register should count on base of this initial value.

7.4.6.13 0x0040 DAC Debug Control Register (Default Value: 0x0000_0000)

Offset: 0x0040			Register Name: AC_ADC_DBG_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	W	0x0	Codec DAC Debug Key Write '0x1883' to enable 'DAC Debug enable' be writable
15:3	/	/	/
2	R/W	0x0	DAC Debug enable 0: Disable 1: Enable
1	R/W	0x0	Reserved
0	R/W	0x0	DAC Debug Mode Select 0: DAC analog debug (dsdm_clk and dsdm data input) 1: DAC digital debug (dsdm_clk and dsdm data output)

7.4.6.14 0x0100 Main LR EQ Enable Control Register (Default Value: 0x0000_0000)

Offset: 0x0100			Register Name: MAIN_LR_EQ_ENA_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:4	R/W	0x0	Main EQ Left Channel enable control 0-disable (and bypass) 1-enable BIT7-Band16~20 BIT6-Band11~15 BIT5-Band6~10 BIT4-Band1~5
3: 0	R/W	0x0	Main EQ Right Channel enable control 0-disable (and bypass) 1-enable BIT3-Band16~20 BIT2-Band11~15 BIT1-Band6~10 BIT0-Band1~5

7.4.6.15 0x0104 + 0x14*(n-1) Main EQ Left BQn B0 Coefficient Register (Default Value: 0x0001_0000)

Offset: 0x0104 + 0x14*(n-1)			Register Name: MAIN_EQL_BQn_B0 (n=1~20)
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20: 0	R/W	0x010000	Main EQ Left Channel Band n, b0 coefficient (2’s complement in 5.16 format)

7.4.6.16 0x0108 + 0x14*(n-1) Main EQ Left BQn B1 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x0108 + 0x14*(n-1)			Register Name: MAIN_EQL_BQn_B1 (n=1~20)
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20: 0	R/W	0x0	Main EQ Left Channel Band n, b1 coefficient (2’s complement in 5.16 format)

7.4.6.17 0x010C + 0x14*(n-1) Main EQ Left BQn B2 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x010C + 0x14*(n-1)			Register Name: MAIN_EQL_BQn_B2 (n=1~20)
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20: 0	R/W	0x0	Main EQ Left Channel Band n, b2 coefficient (2’s complement in 5.16 format)

7.4.6.18 0x0110 + 0x14*(n-1) Main EQ Left BQn A1 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x0110 + 0x14*(n-1)			Register Name: MAIN_EQL_BQn_A1 (n=1~20)
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20: 0	R/W	0x0	Main EQ Left Channel Band n, a1 coefficient (2's complement in 5.16 format)

7.4.6.19 0x0114 + 0x14*(n-1) Main EQ Left BQn A2 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x0114 + 0x14*(n-1)			Register Name: MAIN_EQL_BQn_A2 (n=1~20)
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20: 0	R/W	0x0	Main EQ Left Channel Band n, a2 coefficient (2's complement in 5.16 format)

7.4.6.20 0x0300 + 0x14*(n-1) Main EQ Right BQn B0 Coefficient Register (Default Value: 0x0001_0000)

Offset: 0x0300 + 0x14*(n-1)			Register Name: MAIN_EQR_BQn_B0 (n=1~20)
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20: 0	R/W	0x010000	Main EQ Right Channel Band n, b0 coefficient (2's complement in 5.16 format)

7.4.6.21 0x0304 + 0x14*(n-1) Main EQ Right BQn B1 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x0304 + 0x14*(n-1)			Register Name: MAIN_EQR_BQn_B1 (n=1~20)
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20: 0	R/W	0x0	Main EQ Right Channel Band n, b1 coefficient (2's complement in 5.16 format)

7.4.6.22 0x0308 + 0x14*(n-1) Main EQ Right BQn B2 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x0308 + 0x14*(n-1)			Register Name: MAIN_EQR_BQn_B2 (n=1~20)
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20: 0	R/W	0x0	Main EQ Right Channel Band n, b2 coefficient (2's complement in 5.16 format)

7.4.6.23 0x030C + 0x14*(n-1) Main EQ Right BQn A1 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x030C + 0x14*(n-1)	Register Name: MAIN_EQR_BQn_A1 (n=1~20)

Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20: 0	R/W	0x0	Main EQ Right Channel Band n, a1 coefficient (2's complement in 5.16 format)

7.4.6.24 0x0310 + 0x14*(n-1) Main EQ Right BQn A2 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x0310 + 0x14*(n-1)			Register Name: MAIN_EQR_BQn_A2 (n=1~20)
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20: 0	R/W	0x0	Main EQ Right Channel Band n, a2 coefficient (2's complement in 5.16 format)

7.4.6.25 0x0500 Post LR EQ Enable Control Register (Default Value: 0x0000_0000)

Offset: 0x0500			Register Name: POST_LR_EQ_ENA_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	Post EQ Left and Right Channel enable control 0-disable (and bypass) 1-enable

7.4.6.26 0x0504 + 0x14*(n-1) Post EQ BQn B0 Coefficient Register (Default Value: 0x0001_0000)

Offset: 0x0504 + 0x14*(n-1)			Register Name: POST_EQ_BQn_B0 (n=1~5)
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20: 0	R/W	0x1	Post EQ Band n, b0 coefficient (2's complement in 5.16 format)

7.4.6.27 0x0508 + 0x14*(n-1) Post EQ BQn B1 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x0508 + 0x14*(n-1)			Register Name: POST_EQ_BQn_B1 (n=1~5)
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20: 0	R/W	0x0	Post EQ Band n, b1 coefficient (2's complement in 5.16 format)

7.4.6.28 0x050C + 0x14*(n-1) Post EQ BQn B2 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x050C + 0x14*(n-1)			Register Name: POST_EQ_BQn_B2 (n=1~5)
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20: 0	R/W	0x0	Post EQ Band n, b2 coefficient (2's complement in 5.16 format)

7.4.6.29 0x0510 + 0x14*(n-1) Post EQ BQn A1 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x0510 + 0x14*(n-1)			Register Name: POST_EQ_BQn_A1 (n=1~5)
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20: 0	R/W	0x0	Post EQ Band n, a1 coefficient (2's complement in 5.16 format)

7.4.6.30 0x0514 + 0x14*(n-1) Post EQ BQn A2 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x0514 + 0x14*(n-1)			Register Name: POST_EQ_BQn_A2 (n=1~5)
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20: 0	R/W	0x0	Post EQ Band n, a2 coefficient (2's complement in 5.16 format)

7.4.6.31 0x0600 1-Band DRC Enable Control Register (Default Value: 0x0000_0000)

Offset: 0x0600			Register Name: 1B_DRC_ENA_CTRL
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W	0x0	1B_DRC_CAL_ENA 1-Band DRC Calculation Enable 0: Disable 1: Enable
0	R/W	0x0	1B_DRC_HPF_ENA 1-Band DRC HPF function enable 0: Disable 1: Enable

7.4.6.32 0x0604 1-Band DRC HPF Coef Register (Default Value: 0x00FF_FAC1)

Offset: 0x0604			Register Name: 1B_DRC_HPFC
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26: 0	R/W	0xfffac1	HPF coefficient setting and the data is 3.24 format.

7.4.6.33 0x0608 1-Band DRC Control Register (Default Value: 0x0000_0000)

Offset: 0x0608			Register Name: 1B_DRC_CTRL
Bit	Read/Write	Default/Hex	Description
31:15	/	/	/

Offset: 0x0608			Register Name: 1B_DRC_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
14	R	0x0	1-Band DRC delay buffer data output state when drc delay function is enable and the drc function disable. After disable drc function and this bit go to 0, the user should write the drc delay function bit to 0; 0: Not complete 1: complete
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 disable, the signal delay time is unused.
7	R/W	0x0	The delay buffer use or not when the drc disable 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 function. 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 Select Peak filter, the RMS parameter is unused. (AC_DRC_LRMSHAT/AC_DRC_LRMSLAT/AC_DRC_LRMSHAT/AC_DRC_LRMSLAT) When Signal function Select 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)

Offset: 0x0608			Register Name: 1B_DRC_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x0	Delay function enable 0: Disable 1: Enable When the Delay function enable is disable, the Signal delay time is unused.
1	R/W	0x0	DRC LT enable 0: Disable 1: Enable When the DRC LT is disable the LT, Kl and OPL parameter is unused.
0	R/W	0x0	DRC ET enable 0: Disable 1: Enable When the DRC ET is disable the ET, Ke and OPE parameter is unused.

7.4.6.34 0x060C 1-Band DRC Left Peak Filter Attack Time Coef Register (Default Value: 0x000B_77BF)

Offset: 0x060C			Register Name: 1B_DRC_LPFAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:27	/	/	/
26: 0	R/W	0xb77bf	The left peak filter attack time parameter setting, which determine by the equation that AT = 1-exp(-2.2Ts/ta). The format is 3.24. (1ms)

7.4.6.35 0x0610 1-Band DRC Right Peak Filter Attack Time Coef Register (Default Value: 0x000B_77BF)

Offset: 0x0610			Register Name: 1B_DRC_RPFAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:27	/	/	/
26: 0	R/W	0xb77bf	The right peak filter attack time parameter setting, which determine by the equation that AT = 1-exp(-2.2Ts/ta). The format is 3.24.(1ms)

7.4.6.36 0x0614 1-Band DRC Left Peak Filter Release Time Coef Register (Default Value: 0x00FF_E1F8)

Offset: 0x0614			Register Name: 1B_DRC_LPFRT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:27	/	/	/
26: 0	R/W	0xffe1f8	The left peak filter release time parameter setting, which determine by the equation that RT = exp (-2.2Ts/tr). The format is 3.24.(100ms)

7.4.6.37 0x0618 1-Band DRC Right Peak Filter Release Time Coef Register (Default Value: 0x00FF_E1F8)

Offset: 0x0618			Register Name: 1B_DRC_RPFRT
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26: 0	R/W	0xffe1f8	The right peak filter release time parameter setting, which determine by the equation that RT = exp (-2.2Ts/tr). The format is 3.24.(100ms)

7.4.6.38 0x061C 1-Band DRC Left RMS Filter Average Time Coef Register (Default Value: 0x0001_2BAF)

Offset: 0x061C			Register Name: 1B_DRC_LRMSAT
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26: 0	R/W	0x12baf	The left RMS filter average time parameter setting, which determine by the equation that AT = 1-exp(-2.2Ts/tav). The format is 3.24. (10ms)

7.4.6.39 0x0620 1-Band DRC Right RMS Filter Average Time Coef Register (Default Value: 0x0001_2BAF)

Offset: 0x0620			Register Name: 1B_DRC_RRMSAT
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26: 0	R/W	0x12baf	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)

7.4.6.40 0x0624 1-Band DRC Compressor Threshold Setting Register (Default Value: 0x06A4_D3C0)

Offset: 0x0624			Register Name: 1B_DRC_CT
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x6a4d3c0	The compressor threshold setting, which set by the equation that CTin = -CT/6.0206. The format is 8.24 (-40dB)

7.4.6.41 0x0628 1-Band DRC Compressor Slope Setting Register (Default Value: 0x0080_0000)

Offset: 0x0628			Register Name: 1B_DRC_KC
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x800000	The slope of the compressor which determine by the equation that Kc = 1/R, there, R is the ratio of the compressor, which always is integer. The format is 8.24. (2 : 1)

7.4.6.42 0x062C 1-Band DRC Compressor Output at Compressor Threshold Register (Default Value: 0XF95B_2C3F)

Offset: 0x062C			Register Name: 1B_DRC_OPC
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0xf95b2c3f	The output of the compressor which determine by the equation OPC/6.0206 The format is 8.24 (-40dB)

7.4.6.43 0x0630 1-Band DRC Limiter Threshold Setting Register (Default Value: 0x01A9_34F0)

Offset: 0x0630			Register Name: 1B_DRC_LT
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x01a934f0	The limiter threshold setting, which set by the equation that LTin = -LT/6.0206, The format is 8.24. (-10dB)

7.4.6.44 0x0634 1-Band DRC Limiter Slope Setting Register (Default Value: 0x0005_1EB8)

Offset: 0x0634			Register Name: 1B_DRC_KL
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x00051eb8	The slope of the limiter which determine by the equation that Kl = 1/R, there, R is the ratio of the limiter, which always is integer. The format is 8.24. (50 :1)

7.4.6.45 0x0638 1-Band DRC Limiter Output at Limiter Threshold (Default Value: 0xFBD8_FBA7)

Offset: 0x0638			Register Name: 1B_DRC_OPL
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0xfbd8fba7	The output of the limiter which determine by equation OPT/6.0206. The format is 8.24. (-25dB)

7.4.6.46 0x063C 1-Band DRC Expander Threshold Setting Register (Default Value: 0x0BA0_7291)

Offset: 0x063C			Register Name: 1B_DRC_ET
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0ba07291	The expander threshold setting, which set by the equation that ETin = -ET/6.0206. The format is 8.24. (-70dB)

7.4.6.47 0x0640 1-Band DRC Expander Slope Setting Register (Default Value: 0x0500_0000)

Offset: 0x0640			Register Name: 1B_DRC_KE
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x05000000	The slope of the expander which determine by the equation that Ke = 1/R, there, R is the ratio of the expander, which always is integer and the ke must larger than 50. The format is 8.24. (1:5)

7.4.6.48 0x0644 1-Band DRC Expander Output at Expander Threshold (Default Value: 0xF45F_8D6E)

Offset: 0x0644			Register Name: 1B_DRC_OPE
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0xf45f8d6e	The output of the expander which determine by equation OPE/6.0206. The format is 8.24 (-70dB).

7.4.6.49 0x0648 1-Band DRC Linear Slope Setting Register (Default Value: 0x0100_0000)

Offset: 0x0648			Register Name: 1B_DRC_KN
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x01000000	The slope of the linear which determine by the equation that Kn = 1/R, there, R is the ratio of the linear, which always is integer. The format is 8.24. (1:1)

7.4.6.50 0x064C 1-Band DRC Smooth Filter Gain Attack Time Coef Register (Default Value: 0x0002_5600)

Offset: 0x064C			Register Name: 1B_DRC_SFAT
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26: 0	R/W	0x00025600	The smooth filter attack time parameter setting, which determine by the equation that AT = 1-exp(-2.2Ts/tr). The format is 3.24. (5ms)

7.4.6.51 0x0650 1-Band DRC Smooth Filter Gain Release Time Coef Register (Default Value: 0x0000_0F04)

Offset: 0x0650			Register Name: 1B_DRC_SFRT
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26: 0	R/W	0x00000f04	The gain smooth filter release time parameter setting, which determine by the equation that RT = 1-exp(-2.2Ts/tr). The format is 3.24. (200ms)

7.4.6.52 0x0654 1-Band DRC MAX Gain Setting Register (Default Value: 0xFE56_CB0F)

Offset: 0x0654			Register Name: 1B_DRC_MXGS
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0xfe56cb0f	The max gain setting which determine by equation MXG/6.0206. The format is 8.24 and must -20dB <MXG< 30dB (-10dB)

7.4.6.53 0x0658 1-Band DRC MIN Gain Setting Register (Default Value: 0xF95B_2C3F)

Offset: 0x0658			Register Name: 1B_DRC_MNGS
Bit	Read/Write	Default/Hex	Description

Offset: 0x0658			Register Name: 1B_DRC_MNGS
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0xf95b2c3f	The min gain setting which determine by equation MNG/6.0206. The format is 8.24 and must -60dB ≤MNG ≤ -30dB (-40dB)

7.4.6.54 0x065C 1-Band DRC Expander Smooth Time Coef Register (Default Value: 0x0000_640C)

Offset: 0x065C			Register Name: 1B_DRC_EPSC
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26: 0	R/W	0x0000640C	The gain smooth filter release and attack time parameter setting in expander region, which determine by the equation that RT = 1-exp(-2.2Ts/tr). The format is 3.24. (30ms)

7.4.6.55 0x0700 3-Band DRC Enable Control Register (Default Value: 0x0000_0000)

Offset: 0x0700			Register Name: 3B_DRC_ENA_CTRL
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2	R/W	0x0	Drc delay function enable 0: Disable 1: Enable
1	R/W	0x0	Drc calculation enable 0: Disable 1: Enable
0	R/W	0x0	Dc filter enable 0: Disable 1: Enable

7.4.6.56 0x0704 3-Band DRC Control Register (Default Value: 0x0000_0701)

Offset: 0x0704			Register Name: 3B_DRC_CTRL
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/

Offset: 0x0704			Register Name: 3B_DRC_CTRL
Bit	Read/Write	Default/Hex	Description
7:2	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 disable, the signal delay time is unused.
1: 0	R/W	0x1	The number of frequency division 0: Reserved (1 band) 1: 1 band 2: 2 bands 3: 3 bands

7.4.6.57 0x0708 3-Band DRC DC Filter Coefficient Register (Default Value: 0x2000_4003)

Offset: 0x0708			Register Name: 3B_DRC_DCF_COE
Bit	Read/Write	Default/Hex	Description
31:31	/	/	/
30:15	R/W	0x4000	drc_dcfscale: Dc filter scale,16q14
14: 0	R/W	0x4003	drc_dcfcoe: Dc filter coefficient, 15q14 (-16381/2^14)

7.4.6.58 0x0710 + 0x0004*n 3-Band DRC IIR Filter B1 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x0710 + 0x0004*n			Register Name: 3B_DRC_IIR_B1(n=0~7)
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	coefficient of iir filter of crossover filter, 32q31 n=0: lp1 1st order n=1: lp1 2nd order n=2: lp2 1st order n=3: lp2 2nd order n=4: hp1 1st order n=5: hp1 2nd order n=6: hp2 1st order n=7: hp2 2nd order

7.4.6.59 0x0730 + 0x0004*n 3-Band DRC IIR Filter B2 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x0730 + 0x0004*n	Register Name: 3B_DRC_IIR_B2(n=0~7)

Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	coefficient of iir filter of crossover filter, 32q30

7.4.6.60 0x0750 + 0x0004*n 3-Band DRC IIR Filter B3 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x0750 + 0x0004*n			Register Name: 3B_DRC_IIR_B3(n=0~7)
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	coefficient of iir filter of crossover filter, 32q31

7.4.6.61 0x0770 + 0x0004*n 3-Band DRC IIR Filter A2 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x0770 + 0x0004*n			Register Name: 3B_DRC_IIR_A2(n=0~7)
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	coefficient of iir filter of crossover filter, 32q30

7.4.6.62 0x0790 + 0x0004*n 3-Band DRC IIR Filter A3 Coefficient Register (Default Value: 0x0000_0000)

Offset: 0x0790 + 0x0004*n			Register Name: 3B_DRC_IIR_A3(n=0~7)
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	coefficient of iir filter of crossover filter, 32q31

7.4.6.63 0x07B0 3-Band DRC IIR Filter Configuration Register (Default Value: 0x4000_0003)

Offset: 0x07B0			Register Name: 3B_DRC_IIR_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x4000	scalef: iir filter amplitude attenuation, s16q14
15:2	/	/	/
1	R/W	0x1	phasem2: high band phase compensation 0: -1 1: 1
0	R/W	0x1	phasem1: middle band phase compensation 0: -1 1: 1

7.4.6.64 0x07B4 3-Band DRC0 Combine Control Register (Default Value: 0x0000_0002)

Offset: 0x07B4			Register Name: 3B_DRC0_RMS_COE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1: 0	R/W	0x2	drc_comb: The combine method of left and right channel. 0: left 1: right 2: (lef + right)/2 3: square(left^2 + right^2)

7.4.6.65 0x07B8 3-Band DRC0 RMS Filter Coefficient Register (Default Value: 0x0000_012C)

Offset: 0x07B8			Register Name: 3B_DRC0_RMS_COE
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16: 0	R/W	0x12C	alpha factor for RMS filter, 17q16 (0.0046)

7.4.6.66 0x07BC 3-Band DRC0 Curve Configuration 0 Register (Default Value: 0xB000_D800)

Offset: 0x07BC			Register Name: 3B_DRC0_CURVE_CFG0
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0xB000	xset(1): etMag, expander's threshold, 16q8
15: 0	R/W	0xD800	xset(2): ctMag, compressor's threshold, 16q8

7.4.6.67 0x07C0 3-Band DRC0 Curve Configuration 1 Register (Default Value: 0xEC00_B000)

Offset: 0x07C0			Register Name: 3B_DRC0_CURVE_CFG1
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0xEC00	xset(3): ltMag, limter's threshold, 16q8
15: 0	R/W	0xB000	yset(1): describing the dB-dB DRC curve, (etMag,ySet(1)), 16q8

7.4.6.68 0x07C4 3-Band DRC0 Curve Configuration 2 Register (Default Value: 0xD800_EC00)

Offset: 0x07C4			Register Name: 3B_DRC0_CURVE_CFG2
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0xD800	yset(2): describing the dB-dB DRC curve, (ctMag,ySet(2)), 16q8
15: 0	R/W	0xEC00	yset(3): describing the dB-dB DRC curve, (ltMag,ySet(3)), 16q8

7.4.6.69 0x07C8 3-Band DRC0 Curve Configuration 3 Register (Default Value: 0x0100_0100)

Offset: 0x07C8			Register Name: 3B_DRC0_CURVE_CFG3
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:16	R/W	0x100	slopes for the expander, compressor and limiter! It is a 4*1 vector, corresponding to the slopes for the expander, compressor and limiter regions in the dB-dB DRC curve crset(1), 14q8
15:14	/	/	/
13: 0	R/W	0x100	slopes for the expander, compressor and limiter. crset(2), 14q8

7.4.6.70 0x07CC 3-Band DRC0 Curve Configuration 4 Register (Default Value: 0x0100_0100)

Offset: 0x07CC	Register Name: 3B_DRC0_CURVE_CFG4

Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:16	R/W	0x100	slopes for the expander, compressor and limiter! It is a 4*1 vector, corresponding to the slopes for the expander, compressor and limiter regions in the dB-dB DRC curve crset(3), 14q8
15:14	/	/	/
13: 0	R/W	0x100	slopes for the expander, compressor and limiter. crset(4), 14q8

7.4.6.71 0x07D0 3-Band DRC0 Curve Configuration 5 Register (Default Value: 0x0A00_5800)

Offset: 0x07D0			Register Name: 3B_DRC0_CURVE_CFG5
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30:16	R/W	0xA00	max_gain, upper bound of the supported gains in dB, 15q8, default = 10
15	/	/	/
14: 0	R/W	0x5800	min_gain, lower bound of the supported gains in dB, 15q8,default = -40

7.4.6.72 0x07D4 3-Band DRC0 Smooth Configuration 0 Register (Default Value: 0x0000_0256)

Offset: 0x07D4			Register Name: 3B_DRC0_SMOTH_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16: 0	R/W	0x256	Alpha factor for gain smoothing (GS), relating to GS's attack time property. (converting release time to alpha factor should be done by app layer) sfat: 17q16 (0.0091)

7.4.6.73 0x07D8 3-Band DRC0 Smooth Configuration 1 Register (Default Value: 0x003C_000F)

Offset: 0x07D8			Register Name: 3B_DRC0_SMOTH_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28:17	R/W	0x1E	Num of samples should be held before applying AT or RT filtering the input gains for the hidden gainSmoothing method. cfgHoldSamples: 12q0 (30)
16: 0	R/W	0xF	Alpha factor for gain smoothing, relating to GS's release time property. sfrt: 17q16 (0.0002888)

7.4.6.74 0x07DC 3-Band DRC0 Smooth Configuration 2 Register (Default Value: 0x8000_0010)

Offset: 0x07DC			Register Name: 3B_DRC0_SMOTH_CFG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x1	drc_smthmthd: hidden method for gain smoothing 0: new method 1: old method
30:21	/	/	/
20: 0	R/W	0x10	Threshold for switching attack and release states in gainSmoothing. hardKneeTh: 21q14 (3)

7.4.6.75 0x07E0 3-Band DRC1 Combine Control Register (Default Value: 0x0000_0002)

Offset: 0x07E0			Register Name: 3B_DRC1_RMS_COE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1: 0	R/W	0x2	drc_comb: The combine method of left and right channel. 0: left 1: right 2: (lef + right)/2 3: square(left^2 + right^2)

7.4.6.76 0x07E4 3-Band DRC1 RMS Filter Coefficient Register (Default Value: 0x0000_012C)

Offset: 0x07E4			Register Name: 3B_DRC1_RMS_COE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16: 0	R/W	0x12C	alpha factor for RMS filter, 17q16

7.4.6.77 0x07E8 3-Band DRC1 Curve Configuration 0 Register (Default Value: 0xB000_D800)

Offset: 0x07E8			Register Name: 3B_DRC1_CURVE_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0xB000	xset(1): etMag, expander's threshold, 16q8
15: 0	R/W	0xD800	xset(2): ctMag, compressor's threshold, 16q8

7.4.6.78 0x07EC 3-Band DRC1 Curve Configuration 1 Register (Default Value: 0xEC00_B000)

Offset: 0x07EC			Register Name: 3B_DRC1_CURVE_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0xEC00	xset(3): ltMag, limter's threshold, 16q8
15: 0	R/W	0xB000	yset(1): describing the dB-dB DRC curve, (etMag,ySet(1)), 16q8

7.4.6.79 0x07F0 3-Band DRC1 Curve Configuration 2 Register (Default Value: 0xD800_EC00)

Offset: 0x07F0			Register Name: 3B_DRC1_CURVE_CFG2
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0xD800	yset(2): describing the dB-dB DRC curve, (ctMag,ySet(2)), 16q8
15: 0	R/W	0xEC00	yset(3): describing the dB-dB DRC curve, (ltMag,ySet(3)), 16q8

7.4.6.80 0x07F4 3-Band DRC1 Curve Configuration 3 Register (Default Value: 0x0100_0100)

Offset: 0x07F4			Register Name: 3B_DRC1_CURVE_CFG3
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:16	R/W	0x100	slopes for the expander, compressor and limiter! It is a 4*1 vector, corresponding to the slopes for the expander, compressor and limiter regions in the dB-dB DRC curve crset(1), 14q8
15:14	/	/	/
13: 0	R/W	0x100	slopes for the expander, compressor and limiter. crset(2), 14q8

7.4.6.81 0x07F8 3-Band DRC1 Curve Configuration 4 Register (Default Value: 0x0100_0100)

Offset: 0x07F8			Register Name: 3B_DRC1_CURVE_CFG4
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:16	R/W	0x100	slopes for the expander, compressor and limiter! It is a 4*1 vector, corresponding to the slopes for the expander, compressor and limiter regions in the dB-dB DRC curve crset(3), 14q8
15:14	/	/	/
13: 0	R/W	0x100	slopes for the expander, compressor and limiter. crset(4), 14q8

7.4.6.82 0x07FC 3-Band DRC1 Curve Configuration 5 Register (Default Value: 0x0A00_5800)

Offset: 0x07FC			Register Name: 3B_DRC1_CURVE_CFG5
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30:16	R/W	0xa00	max_gain, upper bound of the supported gains in dB, 15q8
15	/	/	/
14: 0	R/W	0x5800	min_gain, lower bound of the supported gains in dB, 15q8

7.4.6.83 0x0800 3-Band DRC1 Smooth Configuration 0 Register (Default Value: 0x0000_0256)

Offset: 0x0800			Register Name: 3B_DRC1_SMOTH_CFG0
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16: 0	R/W	0x256	Alpha factor for gain smoothing (GS), relating to GS's attack time property. (converting release time to alpha factor should be done by app layer) sfat: 17q16

7.4.6.84 0x0804 3-Band DRC1 Smooth Configuration 1 Register (Default Value: 0x003C_000F)

Offset: 0x0804			Register Name: 3B_DRC1_SMOTH_CFG1
Bit	Read/Write	Default/Hex	Description
31:29	/	/	/
28:17	R/W	0x1E	Num of samples should be held before applying AT or RT filtering the input gains for the hidden gainSmoothing method. cfgHoldSamples: 12q0
16: 0	R/W	0xF	Alpha factor for gain smoothing, relating to GS's release time property. sfrt: 17q16

7.4.6.85 0x0808 3-Band DRC1 Smooth Configuration 2 Register (Default Value: 0x8000_0010)

Offset: 0x0808			Register Name: 3B_DRC1_SMOTH_CFG2
Bit	Read/Write	Default/Hex	Description
31	R/W	0x1	drc_smthmthd: hidden method for gain smoothing 0: new method 1: old method
30:21	/	/	/
20: 0	R/W	0x10	Threshold for switching attack and release states in gainSmoothing. hardKneeTh: 21q14

7.4.6.86 0x080C 3-Band DRC2 Combine Control Register (Default Value: 0x0000_0002)

Offset: 0x080C			Register Name: 3B_DRC1_RMS_COE
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1: 0	R/W	0x2	drc_comb: The combine method of left and right channel. 0: left 1: right 2: (lef + right)/2 3: square(left^2 + right^2)

7.4.6.87 0x0810 3-Band DRC2 RMS Filter Coefficient Register (Default Value: 0x0000_012C)

Offset: 0x0810			Register Name: 3B_DRC1_RMS_COE
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16: 0	R/W	0x12c	alpha factor for RMS filter, 17q16

7.4.6.88 0x0814 3-Band DRC2 Curve Configuration 0 Register (Default Value: 0xB000_D800)

Offset: 0x0814			Register Name: 3B_DRC1_CURVE_CFG0
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0xB000	xset(1): etMag, expander's threshold, 16q8
15: 0	R/W	0xD800	xset(2): ctMag, compressor's threshold, 16q8

7.4.6.89 0x0818 3-Band DRC2 Curve Configuration 1 Register (Default Value: 0xEC00_B000)

Offset: 0x0818			Register Name: 3B_DRC1_CURVE_CFG1
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0xEC00	xset(3): ltMag, limter's threshold, 16q8
15: 0	R/W	0xB000	yset(1): describing the dB-dB DRC curve, (etMag,ySet(1)), 16q8

7.4.6.90 0x081C 3-Band DRC2 Curve Configuration 2 Register (Default Value: 0xD800_EC00)

Offset: 0x081C			Register Name: 3B_DRC1_CURVE_CFG2
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0xD800	yset(2): describing the dB-dB DRC curve, (ctMag,ySet(2)), 16q8
15: 0	R/W	0xEC00	yset(3): describing the dB-dB DRC curve, (ltMag,ySet(3)), 16q8

7.4.6.91 0x0820 3-Band DRC2 Curve Configuration 3 Register (Default Value: 0x0100_0100)

Offset: 0x0820			Register Name: 3B_DRC1_CURVE_CFG3
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:16	R/W	0x100	slopes for the expander, compressor and limiter! It is a 4*1 vector, corresponding to the slopes for the expander, compressor and limiter regions in the dB-dB DRC curve crset (1), 14q8
15:14	/	/	/
13: 0	R/W	0x100	slopes for the expander, compressor and limiter. crset(2), 14q8

7.4.6.92 0x0824 3-Band DRC2 Curve Configuration 4 Register (Default Value: 0x0100_0100)

Offset: 0x0824	Register Name: 3B_DRC1_CURVE_CFG4

Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:16	R/W	0x100	slopes for the expander, compressor and limiter! It is a 4*1 vector, corresponding to the slopes for the expander, compressor and limiter regions in the dB-dB DRC curve crset(3), 14q8
15:14	/	/	/
13: 0	R/W	0x100	slopes for the expander, compressor and limiter. crset(4), 14q8

7.4.6.93 0x0828 3-Band DRC2 Curve Configuration 5 Register (Default Value: 0x0A00_5800)

Offset: 0x0828			Register Name: 3B_DRC1_CURVE_CFG5
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30:16	R/W	0xA00	max_gain, upper bound of the supported gains in dB, 15q8
15	/	/	/
14: 0	R/W	0x5800	min_gain, lower bound of the supported gains in dB, 15q8

7.4.6.94 0x082C 3-Band DRC2 Smooth Configuration 0 Register (Default Value: 0x0000_0256)

Offset: 0x082C			Register Name: 3B_DRC1_SMOTH_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16: 0	R/W	0x256	Alpha factor for gain smoothing (GS), relating to GS's attack time property. (converting release time to alpha factor should be done by app layer) sfat: 17q16

7.4.6.95 0x0830 3-Band DRC2 Smooth Configuration 1 Register (Default Value: 0x003C_000F)

Offset: 0x0830			Register Name: 3B_DRC1_SMOTH_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28:17	R/W	0x1E	The number of samples should be held before applying AT or RT filtering the input gains for the hidden gainSmoothing method. cfgHoldSamples: 12q0
16: 0	R/W	0xF	Alpha factor for gain smoothing, relating to GS's release time property. sfrt: 17q16

7.4.6.96 0x0834 3-Band DRC2 Smooth Configuration 2 Register (Default Value: 0x8000_0010)

Offset: 0x0834	Register Name: 3B_DRC1_SMOTH_CFG2

Bit	Read/Write	Default/Hex	Description
31	R/W	0x1	drc_smthmthd: hidden method for gain smoothing 0: new method 1: old method
30:21	/	/	/
20: 0	R/W	0x10	Threshold for switching attack and release states in gainSmoothing. hardKneeTh: 21q14

Contents

Figures

Tables

8 Memory

8.1 HS_PSRAMC

Ultra High Speed Pseudo Static Radom Access Memory Controller (HS_PSRAMC) initializes PSRAM, configures its registers, and performs reading/writing/refreshing operations. HS_PSRAMC is mounted on the MSI bus through HIF interface and transmits data with PSRAM through PHY.

HS_PSRAMC has the following features:

- R128-S1/R128-S2: SIP 8MB PSRAM
- R128-S3: SIP 32 MB PSRAM
- Supports AP memory PSRAM
- Supports OPI as the interface of PSRAM
- Supports the auto-refreshing and self-refreshing of PSRAM
- Supports up to 800 MHz PSRAM. The ratio of PSRAM controller and PSRAM clock is 1:4.
- Supports indirectly accessing the registers of PSRAM through interface configuration
- Supports selecting various PSRAM clocks to reach capacity expansion.
- Supports caching reading/writing commands through CAM
- Supports out-of-order execution of commands
- Supports prefetching read channel

8.2 LS_PSRAMC

Ultra Low Speed Pseudo Static Random Access Memory Controller (LS_PSRAMC) is mainly used to control the commands receiving/transmitting, data reading/writing and XIP code execution of OPI LS_PSRAMC.

PSRAM Controller transmits wakeup commands, reads manufacturing information, reads/writes PSRAM, and configures PSRAM registers by System Bus. Also, it reads and writes PSRAM through Code Bus. System bus and code bus are AHB interface, and their operations cannot be performed simultaneously.

LS_PSRAMC has the following features:

- R128-S2: SIP 8MB PSRAM
- Supports any frequency ratio of AHB and OPI clock
- Supports CPU/DMA to operate PSRAM through SBUS
- Supports PSRAM Wrap Mode (enter/exit)
- Supports 4 offset address ranges
- Supports CBUS out-of-order reading/writing and XIP (eXecuted In Place)

8.3 Flashc_Enc

8.3.1 Overview

Flashc_Enc consists of flash controller and flash encryption. The flash controller module is the reading and writing controller of Nor-Flash and SQPI-PSRAM, and compatible with Standard SPI/Dual SPI/Quad SPI. It is mainly used for controlling the reception and transmission, reading and writing data, and running codes through XIP (eXecute In Place) of Nor-Flash and SQPI-PSRAM. The flash controller will encrypt the data through AES when writing data to flash, and decode them when reading data.

Flashc_Enc has the following features:

- Supports arbitrary frequency ratio of AHB clock and SPI clock
- Supports 4 segments of offset address range
- Supports receiving and transmitting in 1/2/4-wire SPI
- Supports flash programing and reading by configuring registers (SBUS)
- Supports out-of-order reading CBUS and running codes through XIP
- Supports continuous reading mode (enter/exit) and wrap mode (enter/exit)
- Supports the basic operation of SPI flash
- Supports 8 Mb SIP Nor Flash (for R128-S1) and 16 Mb SIP Nor Flash (for R128-S2)
- Supports real-time AES encryption and decryption when reading and writing data. 6 fields of encryption and decryption range is configurable.

8.3.2 Block Diagram

The following figures show the block diagram of Flashc_Enc, flash encryption, and flash controller.

Figure 8-1 Block Diagram of Flash Encryption

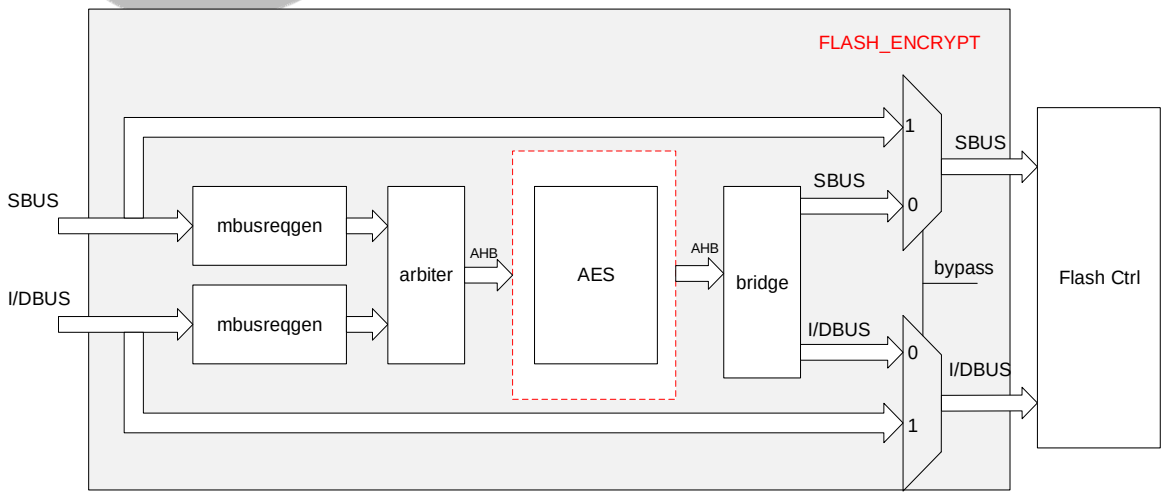
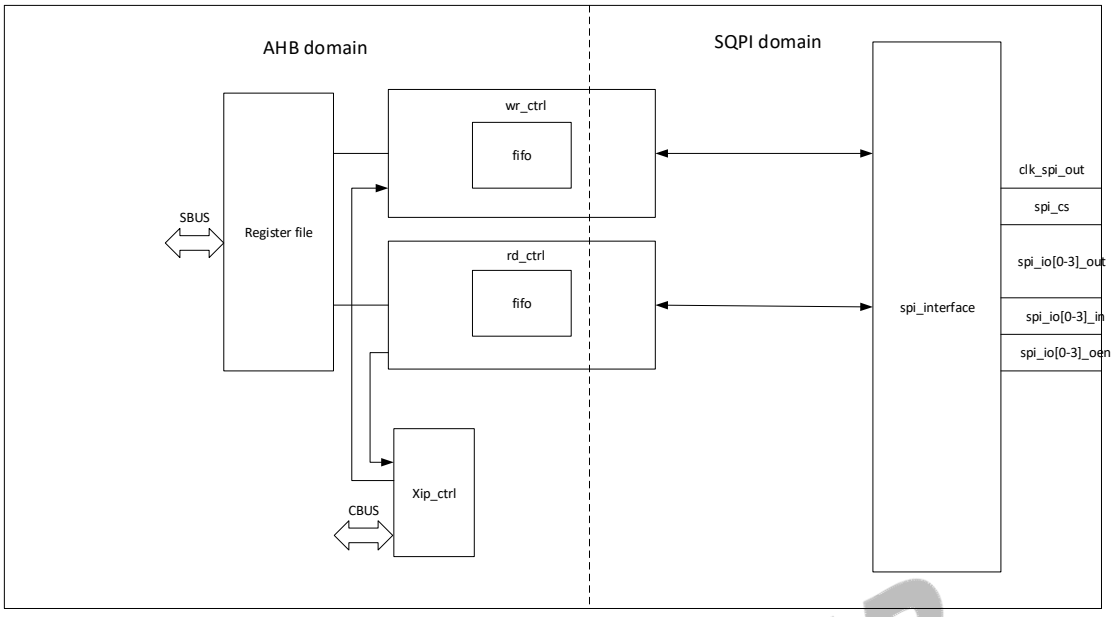


Figure 8-2 Block Diagram of Flash Controller



Flash controller contains the following sub-blocks.

Table 8-1 Flash controller Sub-blocks

Module	Function
Register File	Configure register module
wr_ ctrl	FIFO mode, which is used for the write data transmission among different clock domain.
rd_ ctrl	FIFO mode, which is used for the read data transmission among different clock domain.
spi_interface	SPI protocol control layer

8.3.3 Functional Description

8.3.3.1 Clock Sources

The following table describes the clock sources of Flashc_Enc.

Table 8-2 Flashc_Enc Clock Sources

Clock Sources	Description
hfclk	Controlled by CCU. It is used for SCAN and CLKGATE externally.
clk_spi_flash	Controlled by CCU. It is used for CLKGATE externally, and is used for SCAN internally.
clk_spi_psram	Controlled by CCU. It is used for CLKGATE externally, and is used for SCAN internally.
pclk_flashenc	It is used for SCAN and CLKGATE externally.
test_clk	Test Clock.

8.3.3.2 External Signals

The following table describes the external signals of Flashc_Enc.

Table 8-3 Flashc_Enc External Signals

Signal	Description	Type
SPIF_WP	Flash data 2	I/O
SPIF_HOLD	Flash data 3	I/O
SPIF_MOSI	Flash data 0	I/O
SPIF_MISO	Flash data 1	I/O
SPIF_CS	Flash Chip Select	O
SPIF_CLK	Flash Clock	O

8.3.4 Programming Guidelines

8.3.4.1 Encryption and Decryption

Address range can be enabled by software only after address and KEY are configured well. It should be noticed that the address range being accessed cannot be configured.

8.3.4.2 CBUS Operation

CBUS Read/Write Cacheline Size Configuration

Table 8-4 Read Cacheline Size

CACHE_CFG[1: 0]	Read Cacheline Size
0x0	8 Bytes (64 Bits)
0x1	16 Bytes (128 Bits)
0x2	32 Bytes (256 Bits)
0x3	32 Bytes (256 Bits)

If the CACHE_CFG[1: 0] sends a CBUS read request, the flash controller will send a read instruction (cmd+addr+...+read data). The larger volume of read data contributes to higher efficient continuous read.

Table 8-5 Write Cacheline Size

CACHE_CFG[2]	CACHE_CFG[13:12]	Write Cacheline Size
1	/	Write cacheline based on hsize (namely, the request of AHB access size) or each time when accessing CBUS.
0	0x00	The write cacheline size is 8 Bytes (64 Bits) at max
	0x01	The write cacheline size is 16 Bytes (128 Bits) at max
	0x10	The write cacheline size is 32 Bytes (256 Bits) at max
	0x11	The write cacheline size is 64 Bytes (512 Bits) at max

If the value of CACHE_CFG[2] is 0, flash controller will collect continuous data configured by CACHE_CFG[13:12] before write-once psram. However, it will write data ahead in some special situations to avoid data error. Thus, not every read operation can reach the data volume configured by the CACHE_CFG[13:12].

Filed Bias Configuration

The flash controller contains 6 field bias (field 0-5). The equation of real address is:

$$real\_addr = haddrc + cfg\_addr0\_bias - cfg\_addr0\_start \& 0xfffffff0$$

If the access address of CBUS ranges from START_ADDR0 to END_ADDR0, and the BIAS_ADDR0[31] is set as 1, then calculate the real address based on the above equation, which is exactly the 24 bit accessing address that controller sends to flash/psram.

The address range of Field0-field3 should not be overlapped. If overlapped, the real address will be calculated based on the range with the highest priority (the priority is field0>field1>field2>field3).

Field bias configuration should conform to the rule that it should be the multiples of corresponding address spaces of read cacheline to avoid access error. For example, if the read cacheline size is 32Bytes, the field bias must be the multiples of 0x20; if the read cacheline size is 16Bytes, the bias must be the multiples of 0x10, and so on.

Continuous Read Mode

Enter Continuous Read Mode

Step 1 Switch to SBUS and enter the Continuous Read Mode, where DUMMY DATA can be configured.

Step 2 Configure CBUS read command register as No Send Command.

Step 3 Switch to CBUS to enter the Continuous Read Mode.

Exit Continuous Read Mode

Step 1 Switch to SBUS and exit the Continuous Read Mode, where DUMMY DATA is configured as 0.

Step 2 Configure CBUS read command register as Normal Read and Write.

Step 3 Switch to CBUS.

NOTE

CBUS can be in the Continuous Read Mode when it is read only. CBUS must exit the Continuous Read Mode once SBUS is programmed.

Wrap Mode

Enter Wrap Mode

Step 1 Switch to SBUS and enter the wrap mode, where DUMMY DATA can be configured.

Step 2 Set MEM_COM_CFG [3] as 1 to notice flash controller.

Step 3 Switch to CBUS.

Exit Wrap Mode

Step 1 Switch to SBUS, and exit wrap mode based on the selected command.

Step 2 Set MEM_COM_CFG [3] as 0 to notice Flash Controller.

Step 3 Switch to CBUS.

CBUS Hang

The main causes of CBUS hang are as follows:

- MEM_COM_CFG[4] fails to be opened, and there is SBUS operation when accessing CBUS.
- MEM_COM_CFG[4] is opened, and codes and variable stored on flash are executed through SBUS. CBUS request blocked by arbitration results in deadlock.

8.3.4.3 SBUS Operation

SBUS Register Configuration

Step 1 **Close CBUS Enable**: Write 0 to MEM_COM_CFG [0] to close CBUS enable, and write 1 to MEM_COM_CFG[16] to clear cacheline buffer. Clear CBUS reading cache and write the data not written to CBUS to flash.

Step 2 **Polling State**: Poll whether the bit0 and bit4 of PSRAM_CTRL_COM_CFG is 0. If yes, enter the next step. PSRAM_CTRL_COM_CFG[0] is manually sending the data written to the write FIFO by CBUS as flash controller doesn't support writing data through CBUS.

Step 3 **Send Instructions and Reading/Writing Operation**: Configure SBUS and write 1 to START_SEND_REG. Read S_RDATA_REG or write S_WDATA_REG in reading/writing operation.

Step 4 **Polling State**: Poll whether START_SEND_REG [0] is 0. If yes, repeat the step 3 and step 4 or enter the next step.

Step 5 **Open CBUS Enable**: Write 1 to MEM_COM_CFG [0] to enable CBUS.

DMA Operation

When SBUS performs read/write operation through DMA, the transmission type should be configured as flashctrl and SRAM.

- **Read Operation**: Write 1 to START_SEND_REG and enable DMA to read S_RDATA_REG.
- **Write Operation**: Write 1 to START_SEND_REG and enable DMA to write S_WDATA_REG.

SBUS Hang

The main causes of CBUS hang are as follows:

- The byte number of S_RDATA_REG is larger than that configured by S_RD_NUM in read operation. It is null when reading FIFO.

- The byte number of S_WDATA_REG is lower than that configured by **S_WR_NUM** in write operation. Reading is not finished, but the software mistakenly believes that reading is finished.

8.3.5 Register List

Module Name	Base Address
FLASH ENC	0x4003F800
FLASH CTRL	0x4000B000

Register Name	Offset	Description
Flash_Enc		
ENC_0_ENABLE	0x0000	Encryption and Decryption Field 0 Enable Register
ENC_0_RNG_ST	0x0008	Encryption and Decryption field 0 Start Address Register
ENC_0_RNG_ED	0x000C	Encryption and Decryption Field 0 End Address Register
ENC_0_KEY_RG_0	0x0010	Encryption and Decryption Field 0 Key Register 0
ENC_0_KEY_RG_1	0x0014	Encryption and Decryption Field 0 Key Register 1
ENC_0_KEY_RG_2	0x0018	Encryption and Decryption Field 0 Key Register 2
ENC_0_KEY_RG_3	0x001C	Encryption and Decryption Field 0 Key Register 3
ENC_1_ENABLE	0x0020	Encryption and Decryption Field 1 Enable Register
ENC_1_RNG_ST	0x0028	Encryption and Decryption field 1 Start Address Register
ENC_1_RNG_ED	0x002C	Encryption and Decryption Field 1 End Address Register
ENC_1_KEY_RG_0	0x0030	Encryption and Decryption Field 1 Key Register 0
ENC_1_KEY_RG_1	0x0034	Encryption and Decryption Field 1 Key Register 1
ENC_1_KEY_RG_2	0x0038	Encryption and Decryption Field 1 Key Register 2
ENC_1_KEY_RG_3	0x003C	Encryption and Decryption Field 1 Key Register 3
ENC_2_ENABLE	0x0040	Encryption and Decryption Field 2 Enable Register
ENC_2_RNG_ST	0x0048	Encryption and Decryption field 2 Start Address Register
ENC_2_RNG_ED	0x004C	Encryption and Decryption Field 2 End Address Register
ENC_2_KEY_RG_0	0x0050	Encryption and Decryption Field 2 Key Register 0
ENC_2_KEY_RG_1	0x0054	Encryption and Decryption Field 2 Key Register 1
ENC_2_KEY_RG_2	0x0058	Encryption and Decryption Field 2 Key Register 2
ENC_2_KEY_RG_3	0x005C	Encryption and Decryption Field 2 Key Register 3
ENC_3_ENABLE	0x0060	Encryption and Decryption Field 3 Enable Register
ENC_3_RNG_ST	0x0068	Encryption and Decryption field 3 Start Address Register
ENC_3_RNG_ED	0x006C	Encryption and Decryption Field 3 End Address Register
ENC_3_KEY_RG_0	0x0070	Encryption and Decryption Field 3 Key Register 0
ENC_3_KEY_RG_1	0x0074	Encryption and Decryption Field 3 Key Register 1
ENC_3_KEY_RG_2	0x0078	Encryption and Decryption Field 3 Key Register 2
ENC_3_KEY_RG_3	0x007C	Encryption and Decryption Field 3 Key Register 3
ENC_4_ENABLE	0x0080	Encryption and Decryption Field 4 Enable Register
ENC_4_RNG_ST	0x0088	Encryption and Decryption field 4 Start Address Register
ENC_4_RNG_ED	0x008C	Encryption and Decryption Field 4 End Address Register
ENC_4_KEY_RG_0	0x0090	Encryption and Decryption Field 4 Key Register 0
ENC_4_KEY_RG_1	0x0094	Encryption and Decryption Field 4 Key Register 1

Register Name	Offset	Description
ENC_4_KEY_RG_2	0x0098	Encryption and Decryption Field 4 Key Register 2
ENC_4_KEY_RG_3	0x009C	Encryption and Decryption Field 4 Key Register 3
ENC_5_ENABLE	0x00A0	Encryption and Decryption Field 5 Enable Register
ENC_5_RNG_ST	0x00A8	Encryption and Decryption field 5 Start Address Register
ENC_5_RNG_ED	0x00AC	Encryption and Decryption Field 5 End Address Register
ENC_5_KEY_RG_0	0x00B0	Encryption and Decryption Field 5 Key Register 0
ENC_5_KEY_RG_1	0x00B4	Encryption and Decryption Field 5 Key Register 1
ENC_5_KEY_RG_2	0x00B8	Encryption and Decryption Field 5 Key Register 2
ENC_5_KEY_RG_3	0x00BC	Encryption and Decryption Field 5 Key Register 3
Flash_Ctrl		
MEM_COM_CFG	0x0000	Memory Controller Common Configuration Register
SQPI_COM_CFG	0x0004	SQPI Controller Common Configuration Register
CACHE_CFG	0x0008	Cache Line Size Configuration Register
MEM_AC_CFG	0x000C	Memory AC Character Timing Configuration Register
FLASH_C_READ_CFG	0x0010	Flash Code-bus Read Operation Configuration Register
FLASH_C_WRITE_CFG	0x0014	Flash Code-bus Write Operation Configuration Register
FLASH_C_RD_DUMMY_H	0x0018	Flash Code-bus Read Dummy Data Top Half Register
FLASH_C_RD_DUMMY_L	0x001C	Flash Code-bus Read Dummy Data Bottom Half Register
FLASH_C_WR_DUMMY_H	0x0020	Flash Code-bus Write Dummy Data Top Half Register
FLASH_C_WR_DUMMY_L	0x0024	Flash Code-bus Write Dummy Data Bottom Half Reigister
FLASH_C_IO_SWIT_TIME	0x0028	FLASH Code-bus IO Switch Wait Time Register
S_R/W_CFG	0x002C	System-bus Read/Write Operation Configuration Register
S_ADDR_CFG	0x0030	System-bus Address Configuration Register
S_DUMMY_H	0x0034	System-bus Dummy Data Top Half Register
S_DUMMY_L	0x0038	System-bus Dummy Data Bottom Half Register
S_IO_SWIT_TIME	0x003C	System-bus IO Switch Wait Time Register
S_WR_NUM	0x0040	System-bus Write Byte Number Register
S_RD_NUM	0x0044	System-bus Read Byte Number Register
START_SEND_REG	0x0048	System-bus Start Send Register
FIFO_TRIG_LEV	0x004C	FIFO Trigger Level Register
FIFO_STA_REG	0x0050	FIFO Status Register
INT_EN_REG	0x0054	Interrupt Enable Register
INT_STA_REG	0x0058	Interrupt Status Register
FLASH_MOD_EXE_IND	0x005C	Flash XIP Indication Register
DEBUG_STA	0x0060	Memory Controller Debug State Register
DEBUG_CNT_SBUS_WR	0x0064	Memory Controller SBUS Write Debug Count Register
DEBUG_CNT_SBUS_RD	0x0068	Memory Controller SBUS Read Debug Count Register
Nop_INSTRUCTION	0x006C	Flash Nop Instruction Register
DEBUG_CNT_CBUS_WR	0x0074	Memory Controller CBUS Write Debug Count Register
DEBUG_CNT_CBUS_RD	0x0078	Memory Controller CBUS Read Debug Count Register
START_ADDR0	0x0080	Address Field0 Start Position Register

Register Name	Offset	Description
END_ADDR0	0x0084	Address Field0 End Position Register
BIAS_ADDR0	0x0088	Address Field0 Bias Register
START_ADDR1	0x0090	Address Field1 Start Position Register
END_ADDR1	0x0094	Address Field1 End Position Register
BIAS_ADDR1	0x0098	Address Field1 Bias Register
START_ADDR2	0x00A0	Address Field2 Start Position Register
END_ADDR2	0x00A4	Address Field2 End Position Register
BIAS_ADDR2	0x00A8	Address Field2 Bias Register
START_ADDR3	0x00B0	Address Field3Start Position Register
END_ADDR3	0x00B4	Address Field3 End Position Register
BIAS_ADDR3	0x00B8	Address Field3 Bias Register
START_ADDR4	0x00C0	Address Field4 Start Position Register
END_ADDR4	0x00C4	Address Field4 End Position Register
BIAS_ADDR4	0x00C8	Address Field4 Bias Register
START_ADDR5	0x00D0	Address Field5 Start Position Register
END_ADDR5	0x00D4	Address Field5 End Position Register
BIAS_ADDR5	0x00D8	Address Field5 Bias Register
S_WDATA_REG	0x0100	System-bus Write Data Register
S_RDATA_REG	0x0200	System-bus Read Data Register
PSRAM_ COM_CFG	0x0800	PSRAM Common Configuration Register
PSRAM_CTRL_COM_CFG	0x0804	PSRAM Controller Common Configuration Register
PSRAM_CACHE_CFG	0x0808	PSRAM Cache Configuration Register
PSRAM_C_READ_CFG	0x0810	PSRAM Code-bus Read Operation Configuration Register
PSRAM_C_WRITE_CFG	0x0814	PSRAM Code-bus Write Operation Configuration Register
PSRAM_C_RD_DUMMY_H	0x0818	PSRAM Code-bus Read Dummy Data Top Half Register
PSRAM_C_RD_DUMMY_L	0x081C	PSRAM Code-bus Read Dummy Data Bottom Half Register
PSRAM_C_WR_DUMMY_H	0x0820	PSRAM Code-bus Write Dummy Data Top Half Register
PSRAM_C_WR_DUMMY_L	0x0824	PSRAM Code-bus Write Dummy Data Bottom Half Register
PSRAM_C_IO_SWIT_TIME	0x0828	PSRAM Code-bus IO Switch Wait Time Register
PSRAM_FORCE_CFG	0x086C	PSRAM Force Configure Register Register
PSRAM_COM_CFG	0x0870	PSRAM Common Configure Register Register

8.3.6 Flash Encryption Register Description

In the below, N indicates 6 segments of encryption and decryption fields (0-5).

8.3.6.1 0x0000 + N*0x0020 Encryption and Decryption Field N Enable Register (Default Value: 0x0000_0000)

Offset: 0x0000 + N*0x0020	Register Name: ENC_{N}_ENABLE

Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
0	R/W	0x0	csr_enc_{N}_en Enable the encryption and decryption field {N}, which is valid in high level.

8.3.6.2 0x0008 + N*0x0020 Encryption and Decryption Field N Start Address Register (Default Value: 0x0000_0000)

Offset: 0x0008 + N*0x0020			Register Name: ENC_{N}_RNG_ST
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	R/W	0x0	csr_enc_{N}_start The start address of encryption and decryption field {N}
3:0	/	/	/

8.3.6.3 0x000C + N*0x0020 Encryption and Decryption Field N End Address Register (Default Value: 0x0000_0000)

Offset: 0x000C + N * 0x0020			Register Name: ENC_{N}_RNG_ED
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	R/W	0x0	csr_enc_{N}_end The end address of encryption and decryption field {N}
3: 0	/	/	/

8.3.6.4 0x0010 + N*0x0020 Encryption and Decryption Field N Key Register 0 (Default Value: 0x0000_0000)

Offset: 0x010 + N * 0x20			Register Name: ENC_{N}_KEY_RG_0
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	csr_enc_{N}_key_31_0 The key [31: 0] of encryption and decryption field {N}

8.3.6.5 0x0014 + N*0x0020 Encryption and Decryption Field N Key Register 1 (Default Value: 0x0000_0000)

Offset: 0x0014 + N * 0x0020			Register Name: ENC_{N}_KEY_RG_1
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	csr_enc_{N}_key_63_32 The key [63:32] of encryption and decryption field {N}

8.3.6.6 0x0018 + N*0x0020 Encryption and Decryption Field N Key Register 2 (Default Value: 0x0000_0000)

Offset: 0x0018 + N * 0x0020			Register Name: ENC_{N}_KEY_RG_2
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	csr_enc_{N}_key_95_64 The key [95:64] of encryption and decryption field {N}

8.3.6.7 0x001C + N*0x0020 Encryption and Decryption Field N Key Register 3 (Default Value: 0x0000_0000)

Offset: 0x001C + N * 0x0020			Register Name: ENC_{N}_KEY_RG_3
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	csr_enc_{N}_key_127_96 The key [127:96] of encryption and decryption field {N}

8.3.7 Flash Controller Register Description

8.3.7.1 0x0000 Memory Controller Common Configuration Register (Default Value: 0xB500_0090)

Offset: 0x0000			Register Name: MEM_COM_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	R/W	0x2	Flash Address Size Mode 00: Address Size 8bit. 01: Address Size 16bit. 10: Address Size 24bit. 11: Address Size 32bit.
29:28	R/W	0x3	IO1 Port Vacancy Default Output Value 00: Output 0 01: Output 1 1x: Output z **Note**: *IO1 port vacancy output value when flash is at send state.*
27:26	R/W	0x1	IO2 Port Vacancy Default Output Value 00: Output 0 01: Output 1 1x: Output z **Note**: *IO2 port vacancy output value when flash is at send state.*
25:24	R/W	0x1	IO3 Port Vacancy Default Output Value. 00: Output 0 01: Output 1 1x: Output z **Note**: *IO3 port vacancy output value when flash is at send state.*
23:17	/	/	
16	R/W	0x0	Clear Cacheline Buffer Clear cacheline buffer(256bit) in SPI/AHB domain, usually used before SBUS write. Set 1 by software, clear by hardware when finish.
15:8	R/W	0x0	System-bus HREADY Adjustable Wait Time Out **Note**: *Default value indicates wait time out level =0xFFFF*
7	R/W	0x1	System-bus HREADY Time Out Enable 0: Disable 1: Enable

Offset: 0x0000			Register Name: MEM_COM_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
6	R/W	0x0	Transfer FIFO Reset 0: Auto Clear to 0 1: Reset Transfer FIFO **Note**: Write *1 to this bit will reset the control portion of the transfer FIFO, and auto clear to '0' when completing reset operation, write '0' to this bit has no effect.*
5	R/W	0x0	Receiver FIFO Reset 0: Auto Clear to 0 1: Reset Receiver FIFO **Note**: *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.*
4	R/W	0x1	Bus Arbiter Enable 0: Disable 1: Enable Dual Master Access
3	R/W	0x0	Flash Wrap Around Enable 0: Disable 1: Enable
2	R/W	0x0	Continuous Read Mode Enable 0: Disable 1: Enable **Note**: *Dummy value must be match when CPU enable XIP mode.*
1	/	/	/
0	R/W	0x0	Enable Code-bus Read/Write Operation. 0: Configure code-bus read/write operation. 1: code-bus read/write operation configuration is over. **Note**: *When CPU exits code-bus read/write operation, Clear 0.* *When CPU configures code-bus read/write operation is over, set 1.*

8.3.7.2 0x0004 SQPI Controller Common Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: SQPI_COM_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/
19	R/W	0x0	Single Device Select. It can only take effect when Single Device Enable is 1. 0: Flash 1: PSRAM

Offset: 0x0004			Register Name: SQPI_COM_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
18	R/W	0x0	Single Device Enable. It means CBUS can only operate flash or psram. 0: Single device disable 1: Single device enable
17	R/W	0x1	CLK_SEL_EN When accessing PSRAM, sqpi_clk will be changed from 96MHz to 133MHz by hardware. 0: clock select disable 1: clock select enable
16:15	R/W	0x0	CS Output Select 00: CS0 means flash_cs, CS1 means psram_cs 01: CS0 means psram_cs, CS1 means flash_cs 10: Both of CS0 and CS1 mean flash_cs 11: Both of CS0 and CS1 mean psram_cs
14	/	/	
13:12	R/W	0x0	Flash Wait Half Cycle Number of Receive Data 00: No Delay 01: Delay 1 Half Cycle 10: Delay 2 Half Cycle 11: Delay 3 Half Cycle
11:9	/	/	/
8	R/W	0x0	Flash CS Polarity 0: Enable when CS Signal is Low. 1: Enable when CS Signal is High.
7:5	/	/	/
4	R/W	0x0	First Receive Bit Select 0: MSB First 1: LSB First
3:2	/	/	/
1	R/W	0x0	SPI Clock Polarity Control 0: Active High Polarity 1: Active Low Polarity
0	R/W	0x0	SPI Clock/Data Phase Control 0: Phase 0 (Leading Edge for Sample Data) 1: Phase 1 (Leading Edge for Setup Data)

8.3.7.3 0x0008 Cache Relevant Configuration Register (Default Value: 0x200D_0000)

Offset: 0x0008			Register Name: CACHE_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/

Offset: 0x0008			Register Name: CACHE_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
13:12	R/W	0x2	Write Cache Line Size Configuration 00: 8 Bytes (64 Bits) 01: 16 Bytes (128 Bits) 10: 32 Bytes (256 Bits) 11: 64 Bytes (512 Bits) It means write cache line size.
11:4	R/W	0x0	Code-bus HREADY Adjustable Wait Time Out **Note**: *Default value indicates wait time out level =0xFFFF*
3	R/W	0x1	Code-bus HREADY Time Out Enable 0: Disable 1: Enable
2	R/W	0x1	Code-Bus Write Size Select 0: Code-Bus Write Request Size is Cache-line size 1: Code-Bus Write Request Size is Bus Size (HSIZE) **Note**: *SQPI memory controller gets cache-line and returns data to bus at the same time, so SQPI memory controller doesn't care whether code-bus read request from Cache or CPU.*
1:0	R/W	0x1	Read Cache Line Length Configuration 00: 8 Bytes (64 Bits) 01: 16 Bytes (128 Bits) 10: 32 Bytes (256 Bits) Else: 32 Bytes (256 Bits) It means read cache line size.

8.3.7.4 0x000C Memory AC Character Timing Configuration (Default Value: 0x0100_0505)

Offset: 0x000C			Register Name: MEM_AC_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0x1	SPI Flash CS Enable Wait Cycle (t_SLCH) **Note**: *These bits reperesent CS active to valid CLK edge setup minimum time.*
23:16	R/W	0x0	SPI Flash CS Disable Wait Cycle (t_CHSH) **Note**: *These bits represent valid CLK edge to CS active hold minimum time.*
15:8	R/W	0x5	System-bus CS Deselect Wait Cycle (t_SHSL). **Note**: *These bits represent the CS deselect minimum time. Different values from CS deselect after reading/writing/erasing.*
7: 0	R/W	0x5	Code-bus CS Deselect Wait Cycle (t_SHSL) **Note**: *When selecting flash, these Bits indicates that CS minimum deselecting time after reading. When selecting PSRAM, different value from CS deselects after reading/writing.*

8.3.7.5 0x0010 Flash Code-bus Read Operation Configuration Register (Default Value: 0x0311_0001)

Offset: 0x0010			Register Name: FLASH_C_READ_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0x3	Read Command Send.
23	R/W	0x0	Enable DMA read through I/D bus (flash/psram) 0: Disable. DMA read through SBUS. 1: Enable.
22:20	R/W	0x1	Bit Number of Command Send Every Cycle 000: No Send Command. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: No Send Command. **Note**: *Code-bus command sending mode initial state is single mode. These Bits indicate whether sending command.*
19	/	/	/
18:16	R/W	0x1	Bit Number of Address Send Every Cycle 000: Not Send Address. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send Address. **Note**: *Code-bus address send mode initial state is single mode. These bits indicate whether to send address. For each code-bus operation, sending address is necessary.*
15	/	/	/
14:12	R/W	0x0	Bit Number of DUMMY Send Every Cycle 000: Not Send DUMMY. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send DUMMY. **Note**: *Code-bus dummy send mode initial state is single mode. These bits indicates whether sending dummy.*
11	/	/	/

Offset: 0x0010			Register Name: FLASH_C_READ_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
10:4	R/W	0x0	DUMMY Data Bit Number 0: Send 0 Bytes Dummy. 8: Send 1 Bytes Dummy. 16: Send 2 Bytes Dummy. ..... 64: Send 8 Bytes Dummy. **Note**: *Code-bus send dummy bit number initial state is 0. Include XIP mode bits (M7-M0), WRAP bits (W7-W0). Bit number value must be the multiples of 8 bits (64 Bits at max). Same command may have different dummy numbers at different CLK frequency. If the dummy bit number is more than 8 Bytes, force to 8 Bytes. For example, if dummy has 4 cycles in dual send mode, dummy bits should be configured as 8.*
3	/	/	/
2: 0	R/W	0x1	Bit Number of Data Get Every Cycle 000: Not Get Data. 001: Get 1-Bit Data Every Cycle. 010: Get 2-Bit Data Every Cycle. 011: Get 4-Bit Data Every Cycle. 100: Get 8-Bit Data Every Cycle. Else: Not Get Data. **Note**: *Code-bus get read data mode initial state is single mode. For each code-bus operation, getting data is necessary.*

8.3.7.6 0x0014 Flash Code-bus Write Operation Configuration Register (Default Value: 0x0211_0001)

Offset: 0x0014			Register Name: FLASH_C_WRITE_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0x2	Write Command Send Default is Normal Write
23	/	/	/
22:20	R/W	0x1	Bit Number of Command Send Every Cycle 000: Not Send Command. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send Command. **Note**: *Code-bus command send mode initial state is single mode. These bits indicate whether to send command.*
19	/	/	/

Offset: 0x0014			Register Name: FLASH_C_WRITE_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
18:16	R/W	0x1	Bit Number of Address Send Every Cycle 000: Not Send Address. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send Address. **Note**: *Code-bus address send mode initial state is single mode. These bits indicate whether to send address. For each code-bus operation, sending address is necessary.*
15	/	/	/
14:12	R/W	0x0	Bit Number of DUMMY Send Every Cycle 000: Not Send DUMMY. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send DUMMY. **Note**: *Code-bus dummy send mode initial state is single mode. These bits indicate whether to send dummy.*
11	/	/	/
10:4	R/W	0x0	DUMMY Data Bit Number 0: Send 0 Bytes Dummy. 8: Send 1 Bytes Dummy. 16: Send 2 Bytes Dummy. ..... 64: Send 8 Bytes Dummy. **Note**: *Code-bus send dummy bit number initial state is 0. Bit number value must be the multiples of 8 bits (64 bits at max). Same command may have different dummy numbers at different CLK frequencies. If the dummy bit number is more than 8 bytes, force to 8 Bytes. For example, if dummy has 4 cycles in dual send mode, dummy bits should be configured as 8.*
3	/	/	/
2: 0	R/W	0x1	Bit Number of Data Send Every Cycle. 000: Not Send Data. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send Data. **Note**: *Code-bus send data mode initial state is single mode. For each code-bus operation, getting data is necessary.*

8.3.7.7 0x0018 Flash Code-bus Read Dummy Data Top Half Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: FLASH_C_RD_DUMMY_H
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	Top Half 32 bits Of Code-bus DUMMY Data.

8.3.7.8 0x001C Flash Code-bus Read Dummy Data Bottom Half Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: FLASH_C_RD_DUMMY_L
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	Bottom Half 32 bits Of Code-bus DUMMY Data.

8.3.7.9 0x0020 Flash Code-bus Write Dummy Data Top Half Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: FLASH_C_WR_DUMMY_H
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	Top Half 32 bits Of Code-bus DUMMY Data.

8.3.7.10 0x0024 Flash Code-bus Write Dummy Data Bottom Half Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: FLASH_C_WR_DUMMY_L
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	Bottom Half 32 bits Of Code-bus DUMMY Data.

8.3.7.11 0x0028 Flash Code-bus IO Switch Wait Time Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: FLASH_C_IO_SWIT_TIME
Bit	Read/Write	Default/Hex	Description
31:24	R/W	0x0	FLASH Code-bus Read Latency Wait Cycle
23: 0	R/W	/	/

8.3.7.12 0x002C System-bus Read/Write Operation Configuration Register (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: S_R/W_CFG
Bit	Read/Write	Default/Hex	Description
31:24	R/W	0x0	Read/Write Command Send.
23	R/W	0x0	SBUS Flash or PSRAM Select 0: Select Flash 1: Select PSRAM **Note**: *When selecting flash, code-bus is read only. Memory controller disables code-bus write request.*

Offset: 0x002C			Register Name: S_R/W_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
22:20	R/W	0x0	Bit Number of Command Send Every Cycle. 000: Not Send Command. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send Command. **Note**: *These bits indicate whether to send command.*
19	/	/	/
18:16	R/W	0x0	Bit Number of Address Send Every Cycle. 000: Not Send Address 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send Address **Note**: *These bits indicate whether to send address.*
15	/	/	/
14:12	R/W	0x0	Bit Number of DUMMY Send Every Cycle. 000: Not Send DUMMY. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send DUMMY. **Note**: *These bits indicate whether to send dummy.*
11	/	/	/
10:4	R/W	0x0	DUMMY Data Bit Number. 0: Send 0-Bytes Dummy. 8: Send 1-Bytes Dummy. 16: Send 2-Bytes Dummy. ..... 64: Send 8-Bytes Dummy. **Note**: *System-bus send dummy bit number initial state is 0. Bit number value must be the multiples of 8 bits (64 Bits at max). Same command may have different dummy number at different CLK frequencies. If the dummy bit number is more than 8 bytes, force to 8 Bytes. For example, if dummy has 4 cycles in dual send mode, dummy bits should be configured as 8.*
3	/	/	/

Offset: 0x002C			Register Name: S_R/W_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
2: 0	R/W	0x0	Bit Number of System-bus Data Get/Write Every Cycle. 000: Neither Get nor Send Data Operation. 001: Get/Send 1-Bit Data Every Cycle. 010: Get/Send 2-Bit Data Every Cycle. 011: Get/Send 4-Bit Data Every Cycle. 100: Get/Send 8-Bit Data Every Cycle. Else: Neither Get nor Send Data Operation. **Note**: *These bits indicate whether sending/getting data. Working with write/read number to indicate current operation is sending or getting data state as long as these bits are non-zero.*

8.3.7.13 0x0030 System-bus Address Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: S_ADDR_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	32 Bits Flash Page Address of System-bus Operation. **Note**: *If the flash address size is 24 bits, write Low 24 bits.*

8.3.7.14 0x0034 System-bus Dummy Data Top Half Register (Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: S_DUMMY_H
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	Top Half 32bits Of Dummy Data.

8.3.7.15 0x0038 System-bus Dummy Data Bottom Half Register (Default Value: 0x0000_0000)

Offset: 0x038			Register Name: S_DUMMY_L
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	Bottom Half 32bits Of Dummy Data.

8.3.7.16 0x003C System-bus IO Switch Wait Time Register (Default Value: 0x0000_0000)

Offset: 0x003C			Name: S_IO_SWIT_TIME
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0x0	System-bus Read Latency Wait Cycle.
23: 0	/	/	/

NOTE

When the OPI Port Exist I/O Switch, Port Trans/Get Data Needs Suspend.

8.3.7.17 0x0040 System-bus Write Data Byte Number Register (Default Value: 0x0000_0000)

Offset: 0x0040			Register Name: S_WR_NUM
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	WC	0x0	System-bus Write Number (Byte) 0: Non-Write. 1: Write 1 Byte. 2: Write 2 Bytes. 3: Write 3 Bytes. ..... **Note**: *These bits indicate current operation is writing data to flash. If CPU executes write data operation, the bit number of system-bus data get/write every cycle must be configured as non-zero value. If selecting flash, CPU wants to program more than 256-byte data and divides multiple pages to program. Every time CPU executes page program, the Flash BUSY Bit should be checked.*

8.3.7.18 0x0044 System-bus Read Data Byte Number Register (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: S_RD_NUM
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	WC	0x0	System-bus Read Number (Byte) 0: Non-Read. 1: Read 1 Byte. 2: Read 2 Bytes. 3: Read 3 Bytes. ..... **Note**: *These bits indicate current operation is reading data from flash. If CPU executes read data operation, bit number of system-bus data get/write every cycle must be configured as non-zero value.*

8.3.7.19 0x0048 System-bus Start Send Register (Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: Start_SEND_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/

Offset: 0x0048			Register Name: Start_SEND_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	WC	0x0	Enable System-bus Operation 0: System-bus Operation is Configuring. 1: System-bus Operation Configure Over. **Note**: *If system-bus is configuring write operation, this bit should be configured after all the register which write operation needs complete configuration and before writing data to write FIFO. If system-bus is configuring read operation, this bit should be configured after all the register which read operation needs complete configuration and before reading data from read FIFO. If this bit current value is 1, waiting memory controller clear to 0, CPU can configure next system-bus operation.*

8.3.7.20 0x004C FIFO Trigger Level Register (Default Value: 0x6F0F_6F0F)

Offset: 0x004C			Register Name: FIFO_TRIG_LEV
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0x6F	Write FIFO Full Request Trigger Level (Byte).
23:16	R/W	0x0F	Write FIFO Empty Request Trigger Level (Byte).
15:8	R/W	0x6F	Read FIFO Full Request Trigger Level (Byte).
7: 0	R/W	0x0F	Read FIFO Empty Request Trigger Level (Byte).

8.3.7.21 0x0050 FIFO Status Register (Default Value: 0x0000_0000)

Offset: 0x0050			Register Name: FIFO_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25	R	0x0	Read Buffer Full Flag.
24	R	0x0	Write Buffer Full Flag.
23	R	0x0	Write buffer is writing.
22:20	R	0x0	Write Buffer Counter (Byte) These Bits Indicate Number of Bytes in Write Buffer.
19	R	0x0	Read Buffer is reading.
18:16	R	0x0	Read Buffer Counter (Byte) These Bits Indicate Number of Bytes in Read Buffer.
15:8	R	0x0	Write FIFO Counter (Byte) These Bits Indicate Number of Bytes in Write FIFO. 0: 0 Byte in Write FIFO. 1: 1 Bytes in Write FIFO. … 128: 128 Bytes in Write FIFO.

Offset: 0x0050			Register Name: FIFO_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
7: 0	R	0x0	Read FIFO Counter (Byte) These Bits Indicate Number of Bytes in Read FIFO. 0: 0 Byte in Read FIFO. 1: 1 Byte in Read FIFO. … 128: 128 Bytes in Read FIFO.

8.3.7.22 0x0054 Interrupt Enable Register (Default Value: 0x0000_0000)

Offset: 0x0054			Register Name: INT_EN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R/W	0x0	CBUS requests to access (flash/psram r/w) while start send is high. CBUS will read nop-instruction until next reset. 0: Disable 1: Enable
16	R/W	0x0	CBUS is operating but SBUS send start-send. Start-send will be set to 0 by hw and will not affect CBUS operation. 0: Disable 1: Enable
15	R/W	0x0	When DMA write address was not in the non-cacheable field which were configured in the DCACHE register mr010~mr034, send interrupt to CPU. 0: Disable 1: Enable
14	R/W	0x0	PSRAM read latency Time Out Interrupt Enable 0: Disable 1: Enable
13	R/W	0x0	System-bus HREADY Time Out Interrupt Enable 0: Disable 1: Enable
12	R/W	0x0	Transfer Completed Interrupt Enable 0: Disable 1: Enable
11	R/W	0x0	Write FIFO Underflow Interrupt Enable 0: Disable 1: Enable
10	R/W	0x0	Write FIFO Overflow Interrupt Enable 0: Disable 1: Enable

Offset: 0x0054			Register Name: INT_EN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
9	R/W	0x0	Read FIFO Underflow Interrupt Enable 0: Disable 1: Enable
8	R/W	0x0	Read FIFO Overflow Interrupt Enable 0: Disable 1: Enable
7	/	/	/
6	R/W	0x0	Write FIFO Full Interrupt Enable 0: Disable 1: Enable
5	R/W	0x0	Write FIFO Empty Interrupt Enable 0: Disable 1: Enable
4	R/W	0x0	Write FIFO Ready Request Interrupt Enable 0: Disable 1: Enable
3	/	/	/
2	R/W	0x0	Read FIFO Full Interrupt Enable 0: Disable 1: Enable
1	R/W	0x0	Read FIFO Empty Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	Read FIFO Ready Request Interrupt Enable 0: Disable 1: Enable

8.3.7.23 0x0058 Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x058			Name: INT_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	W1C/R	0x0	CBUS request flash access while start send is high. CBUS will read nop-instruction until next reset. 0: This operation not happened. 1: This operation happened.
16	W1C/R	0x0	CBUS is operating but SBUS send start-send. Start-send will be set to 0 by hardware and will not affect CBUS operation. 0: This operation doesn't happen. 1: This operation happened.

Offset: 0x058			Name: INT_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
15	W1C/R	0x0	When DMA write address is not in the non-cacheable field that were configured in the DCACHE register mr010~mr034, sending interrupt to CPU. 0: This operation doesn't happen. 1: This operation happened.
14	W1C/R	0x0	PSRAM Read Time Out Interrupt Flag. 0: Read time out doesn't happen. 1: Read time out happened.
13	W1C/R	0x0	System-bus HREADY Time Out Interrupt Flag 0: System-bus HREADY Time Out not happened. 1: System-bus HREADY Time Out happened.
12	W1C/R	0x0	Transfer Completed Interrupt Flag 0: Busy. 1: Transfer Completed.
11	W1C/R	0x0	Write FIFO Underflow Interrupt Flag **Note**: *When set, this bit indicates that Write FIFO has underflow. Writing 1 to this bit clears it.*
10	W1C/R	0x0	Write FIFO Overflow Interrupt Flag **Note**: *When set, this bit indicates that Write FIFO has overflowed. Writing 1 to this bit clears it.*
9	W1C/R	0x0	Read FIFO Underflow Interrupt Flag **Note**: *When set, this bit indicates that Read FIFO has underflow. Writing 1 to this bit clears it.*
8	W1C/R	0x0	Read FIFO Overflow Interrupt Flag **Note**: *When set, this bit indicates that Read FIFO has overflow. Writing 1 to this bit clears it.*
7	/	/	/
6	W1C/R	0x0	Write FIFO Full Interrupt Flag 0: Write FIFO is Not Full. 1: Write FIFO is Full. **Note**: *This bit is set when the Write FIFO is full. Writing 1 to this bit clears it.*
5	W1C/R	0x0	Write FIFO Empty Request Interrupt Flag 0: Write FIFO always contains one or more bytes. 1: Write FIFO is empty. **Note**: *This bit is set if the Write FIFO is empty. Writing 1 to this bit clears it.*

Offset: 0x058			Name: INT_STA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
4	W1C/R	0x0	Write FIFO Empty Request Interrupt Flag 0: WF_WL > WF_TRIG_LEVEL. 1: WF_WL <= WF_TRIG_LEVEL has happened. **Note**: *This bit is set at any time if WF_WL <= WF_TRIG_LEVEL. Writing 1 to this bit clears it when WF_WL is the water level of Write FIFO.*
3	/	/	/
2	W1C/R	0x0	Read FIFO Full Interrupt Flag 0: Read FIFO is not Full. 1: Read FIFO is full. **Note**: *This bit is set when Read FIFO is full (>= full_trigger_level). Writing 1 to this bit clears it.*
1	W1C/R	0x0	Read FIFO Empty Interrupt Flag 0: Read FIFO is not empty. 1: Read FIFO is empty. **Note**: *This bit is set if the Read FIFO is empty. Writing 1 to this bit clears it.*
0	W1C/R	0x0	Read FIFO Ready Request Interrupt Flag 0: RF_WL < RX_TRIG_LEVEL. 1: RF_WL >= RX_TRIG_LEVEL has happened. **Note**: *This bit is set at any time if RF_WL >= RF_TRIG_LEVEL. Writing 1 to this bit clears it when RF_WL is the water level of Read FIFO.*

8.3.7.24 0x005C Flash XIP Indication Register (Default Value: 0x0000_0000)

Offset: 0x005C			Register Name: FLASH_MOD_EXE_IND
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R	0x0	Continuous Read Mode(XIP) is executed 0: XIP mode doesn't take effect in memory. 1: XIP mode doesn't take effect in memory.

8.3.7.25 0x0060 Memory Controller Debug State Register (Default Value: 0x0000_0000)

Offset: 0x0060			Register Name: DEBUG_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:15	/	/	/
14	R	0x0	serving_c_wr
13	R	0x0	serving_c_rd
12	R	0x0	serving_s
11:4	/	/	/

Offset: 0x0060			Register Name: DEBUG_STA
Bit	Read/Write	Default/Hex	Description
3:0	R	0x0	Memory Controller State Debug Reg.

8.3.7.26 0x0064 Memory Controller SBUS Write Debug Count Register (Default Value: 0x0000_0000)

Offset: 0x0064			Register Name: DEBUG_CNT_SBUS_WR
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	SBUS Write Count

8.3.7.27 0x0068 Memory Controller SBUS Read Debug Count Register (Default Value: 0x0000_0000)

Offset: 0x0068			Register Name: DEBUG_CNT_SBUS_RD
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	SBUS Read Count

8.3.7.28 0x006C Nop Instruction Register (Default Value: 0xBF00_BF00)

Offset: 0x006C			Register Name: Nop Instruction
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0xbf00_bf00	Nop instruction If CBUS accesses flash while start_send is high, which is forbidden, hrdatac will be permanently set to Nop instruction and hreadyc high. Use for debug.

8.3.7.29 0x0074 CBUS Write Byte Number Register (Default Value: 0x0000_0000)

Offset: 0x0074			Register Name: DEBUG_CNT_CBUS_WR
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x0	CBUS Write Byte Count

8.3.7.30 0x0078 CBUS Read Byte Number Register (Default Value: 0x0000_0000)

Offset: 0x0078			Register Name: DEBUG_CNT_CBUS_RD
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x0	CBUS Read Byte Count

8.3.7.31 0x0080 Address Field0 Start Position Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0080			Register Name: Start_ADDR0
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/

Offset: 0x0080			Register Name: Start_ADDR0
Bit	**Read/Write**	**Default/Hex**	**Description**
29:4	R/W	0x0	Address Field0 Start Position **Note**: A*t the unit of 16 bytes.*
3: 0	/	/	/

8.3.7.32 0x0084 Address Field0 End Position Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0084			Register Name: END_ADDR0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
29:4	R/W	0x0	Address Field0 End Position **Note**: A*t the unit of 16 bytes.*
3: 0	/	/	/

8.3.7.33 0x0088 Address Field0 Bias Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0088			Register Name: BIAS_ADDR0
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	Address Field0 Bias Enable 1: Bias Enabled, add bias when address is between Address Field0 Start Position and End Position. 0: Bias Disabled.
30: 0	R/W	0x0	Address Field0 Bias

8.3.7.34 0x0090 Address Field1 Start Position Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0090			Register Name: Start_ADDR1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
29:4	R/W	0x0	Address Field1 Start Position **Note**: A*t the unit of 16 bytes.*
3: 0	/	/	/

8.3.7.35 0x0094 Address Field1 End Position Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0094			Register Name: END_ADDR1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
29:4	R/W	0x0	Address Field1 End Position **Note**: A*t the unit of 16 bytes.*
3: 0	/	/	/

8.3.7.36 0x0098 Address Field1 Bias Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0098			Register Name: END_ADDR1
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	Address Field1 Bias Enable 1: Bias Enabled, add bias when address is between Address Field1 Start Position and End Position. 0: Bias Disabled.
30: 0	R/W	0x0	Address Field1 Bias

8.3.7.37 0x00A0 Address Field2 Start Position Configuration Register (Default Value: 0x0000_0000)

Offset: 0x00A0			Register Name: Start_ADDR2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
29:4	R/W	0x0	Address Field2 Start Position **Note**: A*t the unit of 16 bytes.*
3: 0	/	/	/

8.3.7.38 0x00A4 Address Field2 End Position Configuration Register (Default Value: 0x0000_0000)

Offset: 0x00A4			Register Name: END_ADDR2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
29:4	R/W	0x0	Address Field2 End Position **Note**: A*t the unit of 16 bytes.*
3:0	/	/	/

8.3.7.39 0x00A8 Address Field2 Bias Configuration Register (Default Value: 0x0000_0000)

Offset: 0x00A8			Register Name: END_ADDR2
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	Address Field2 Bias Enable 1: Bias Enabled, add bias when address is between Address Field2 Start Position and End Position. 0: Bias Disabled.
30:0	R/W	0x0	Address Field2 Bias

8.3.7.40 0x00B0 Address Field3 Start Position Configuration Register (Default Value: 0x0000_0000)

Offset: 0x00B0			Register Name: Start_ADDR3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/

Offset: 0x00B0			Register Name: Start_ADDR3
Bit	**Read/Write**	**Default/Hex**	**Description**
29:4	R/W	0x0	Address Field3 Start Position **Note**: A*t the unit of 16 bytes.*
3: 0	/	/	/

8.3.7.41 0x00B4 Address Field3 End Position Configuration Register (Default Value: 0x0000_0000)

Offset: 0x00B4			Register Name: END_ADDR3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
29:4	R/W	0x0	Address Field3 End Position **Note**: A*t the unit of 16 bytes.*
3: 0	/	/	/

8.3.7.42 0x00B8 Address Field3 Bias Configuration Register (Default Value: 0x0000_0000)

Offset: 0x00B8			Register Name: END_ADDR3
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	Address Field3 Bias Enable 1: Bias Enabled, add bias when address is between Address Field3 Start Position and End Position. 0: Bias Disabled.
30: 0	R/W	0x0	Address Field3 Bias

8.3.7.43 0x00C0 Address Field4 Start Position Configuration Register (Default Value: 0x0000_0000)

Offset: 0x00C0			Register Name: Start_ADDR4
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
29:4	R/W	0x0	Address Field4 Start Position **Note**: A*t the unit of 16 bytes.*
3: 0	/	/	/

8.3.7.44 0x00C4 Address Field4 End Position Configuration Register (Default Value: 0x0000_0000)

Offset: 0x00C4			Register Name: END_ADDR4
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
29:4	R/W	0x0	Address Field4 End Position **Note**: A*t the unit of 16 bytes.*
3: 0	/	/	/

8.3.7.45 0x00C8 Address Field4 Bias Configuration Register (Default Value: 0x0000_0000)

Offset: 0x00C8			Register Name: END_ADDR4
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	Address Field4 Bias Enable 1: Bias Enabled, add bias when address is between Address Field4 Start Position and End Position. 0: Bias Disabled.
30: 0	R/W	0x0	Address Field4 Bias

8.3.7.46 0x00D0 Address Field5 Start Position Configuration Register (Default Value: 0x0000_0000)

Offset: 0x00D0			Register Name: Start_ADDR5
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
29:4	R/W	0x0	Address Field5 Start Position **Note**: A*t the unit of 16 bytes.*
3: 0	/	/	/

8.3.7.47 0x00D4 Address Field5 End Position Configuration Register (Default Value: 0x0000_0000)

Offset: 0x00D4			Register Name: END_ADDR5
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
29:4	R/W	0x0	Address Field5 End Position **Note**: A*t the unit of 16 bytes.*
3: 0	/	/	/

8.3.7.48 0x00D8 Address Field5 Bias Configuration Register (Default Value: 0x0000_0000)

Offset: 0x00D8			Register Name: END_ADDR5
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	Address Field5 Bias Enable 1: Bias Enabled, add bias when address is between Address Field5 Start Position and End Position. 0: Bias Disabled.
30: 0	R/W	0x0	Address Field5 Bias

8.3.7.49 0x0100 System-bus Write Data Register (Default Value: 0x0000_0000)

Offset: 0x0100			Register Name: S_WDATA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	System-bus Write Data Register. **Note**: Write *FIFO Entrance.*

8.3.7.50 0x0200 System-bus Read Data Register (Default Value: 0x0000_0000)

Offset: 0x0200			Register Name: S_RDATA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	System-bus Read Data Register. **Note**: Read *FIFO Exit.*

8.3.7.51 0x0800 PSRAM Common Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0800			Register Name: PSRAM_ COM_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	R/W	0x2	Psram Address Size Mode 00: Address Size 8bit. 01: Address Size 16bit. 10: Address Size 24bit. 11: Address Size 32bit.
[29:4]	/	/	/
3	R/W	0x0	PSRAM Wrap Around Enable. 0: Disable. 1: Enable.
[2: 0]	/	/	/

8.3.7.52 0x0804 PSRAM Controller Common Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0804			Register Name: PSRAM_CTRL_COM_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 17	/	/	/
16	R/W	0x0	When DMA write address is not in the non-cacheable field which is configured in the DCACHE register mr010~mr034, the write operation should not take into effect. 0: Write operation still take into effect. 1: Write operation won't take into effect.
15:5	/	/	/
4	R/W	0x0	Clear Cacheline Buffer Clear Cacheline Buffer(256bit) is in spi/ahb domain, usually used before SBUS write. Set 1 by software, clear by hardware when finish.
3:1	/	/	/
0	R/W	0x0	Force Cache Line Timeout Force to write FIFO data to device (psram), usually used before SBUS read/write. Set 1 by software, clear by HW.

8.3.7.53 0x0808 PSRAM Cache Configuration Register (Default Value: 0x0000_8000)

Offset: 0x0808			Register Name: PSRAM_CACHE_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	Auto send data when not meet cache line size but timeout. Used with bit[30:16]. For example, when cache line length is 128bits, after receive address 0x0, 0x4, 0x8 from upstream, controller didn't get address 0xC data for a long time, then send 0x0, 0x4, 0x8 when this register is configured as 1.
30:16	R/W	0x0	Wait Cache Line data timeout cycle.
15	R/W	0x1	Auto cut sequential word send into two write operation when address cross page boundary. (for PSRAM CBUS write only) For SQPI 16Mb, page size is 512 Bytes. Page size is configured in 0x0870.
14: 0	/	/	/

8.3.7.54 0x0810 PSRAM Code-bus Read Operation Configuration Register (Default Value: 0x0311_0001)

Offset: 0x0810			Register Name: PSRAM_C_READ_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0x3	Read Command Send.
23	/	/	/
22:20	R/W	0x1	Bit Number of Command Send Every Cycle. 000: Not Send Command. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send Command. **Note**: *Code-bus command send mode initial state is single mode. These Bits indicates whether to send command.*
19	/	/	/
18:16	R/W	0x1	Bit Number of Address Send Every Cycle. 000: Not Send Address. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send Address. **Note**: *Code-bus address sends mode initial state is single mode. These bits indicate whether sending address. For each code-bus operation, send address is necessary.*
15	/	/	/

Offset: 0x0810			Register Name: PSRAM_C_READ_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
14:12	R/W	0x0	Bit Number of DUMMY Send Every Cycle. 000: Not Send DUMMY. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send DUMMY. **Note**: *Code-bus dummy send mode initial state is single mode. These bits indicate whether sending dummy.*
11	/	/	/
10:4	R/W	0x0	DUMMY Data Bit Number. 0: Send 0 Bytes Dummy. 8: Send 1 Bytes Dummy. 16: Send 2 Bytes Dummy. ..... 64: Send 8 Bytes Dummy. **Note**: *Code-bus send dummy bit number initial state is 0. Include XIP mode bits (M7-M0), WRAP bits (W7-W0). Bit number value must be the multiples of 8 bits (64 bits at max). Same command may have different dummy numbers at different CLK frequencies. If the dummy bit number is more than 8 Bytes, force to 8 Bytes. For example, if dummy has 4 cycles in dual send mode, dummy bits should be configured as 8.*
3	/	/	/
2: 0	R/W	0x1	Bit Number of Data Get Every Cycle. 000: Not Get Data. 001: Get 1-Bit Data Every Cycle. 010: Get 2-Bit Data Every Cycle. 011: Get 4-Bit Data Every Cycle. 100: Get 8-Bit Data Every Cycle. Else: Not Get Data. **Note**: *Code-bus gets read data mode initial state is single mode. For each code-bus operation, getting data is necessary.*

8.3.7.55 0x0814 PSRAM Code-bus Write Operation Configuration Register (Default Value: 0x0211_0001)

Offset: 0x0814			Register Name: PSRAM_C_WRITE_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0x2	Write Command Send. Default is Normal Write
23	/	/	/

Offset: 0x0814			Register Name: PSRAM_C_WRITE_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
22:20	R/W	0x1	Bit Number of Command Send Every Cycle 000: Not Send Command. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send Command. **Note**: *Code-bus command send mode initial state is single mode. These bits indicate whether to send command.*
19	/	/	/
18:16	R/W	0x1	Bit Number of Address Send Every Cycle 000: Not Send Address. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send Address. **Note**: *Code-bus address send mode initial state is single mode. These bits indicate whether to send address. For each code-bus operation, send address is necessary.*
15	/	/	/
14:12	R/W	0x0	Bit Number of DUMMY Send Every Cycle. 000: Not Send DUMMY. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send DUMMY. **Note**: *Code-bus dummy send mode initial state is single mode. These bits indicate whether to send dummy.*
11	/	/	/
10:4	R/W	0x0	DUMMY Data Bit Number. 0: Send 0 Bytes Dummy. 8: Send 1 Bytes Dummy. 16: Send 2 Bytes Dummy. ..... 64: Send 8 Bytes Dummy. **Note**: *Code-bus send dummy bit number initial state is 0. Bit number value must be the multiples of 8 bits (64 bits at max). Same command may have different dummy numbers at different CLK frequencies. If the dummy bit number is more than 8 Bytes, force to 8 Bytes. For example, if dummy has 4 cycles in dual send mode, dummy bits should be configured as 8.*

Offset: 0x0814			Register Name: PSRAM_C_WRITE_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
3	/	/	/
2: 0	R/W	0x1	Bit Number of Data Send Every Cycle 000: Not Send Data. 001: Send 1-Bit Data Every Cycle. 010: Send 2-Bit Data Every Cycle. 011: Send 4-Bit Data Every Cycle. 100: Send 8-Bit Data Every Cycle. Else: Not Send Data. **Note**: *Code-bus send data mode initial state is single mode. For each code-bus operation, getting data is necessary.*

8.3.7.56 0x0818 PSRAM Code-bus Read Dummy Data Top Half Register (Default Value: 0x0000_0000)

Offset: 0x0818			Register Name: PSRAM_C_RD_DUMMY_H
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	Top Half 32bits Of Code-bus DUMMY Data.

8.3.7.57 0x0081C PSRAM Code-bus Read Dummy Data Bottom Half Register (Default Value: 0x0000_0000)

Offset: 0x0081C			Register Name: PSRAM_C_RD_DUMMY_L
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	Bottom Half 32bits Of Code-bus DUMMY Data.

8.3.7.58 0x0820 PSRAM Code-bus Write Dummy Data Top Half Register (Default Value: 0x0000_0000)

Offset: 0x0820			Register Name: PSRAM_C_WR_DUMMY_H
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	Top Half 32bits Of Code-bus DUMMY Data.

8.3.7.59 0x0824 PSRAM Code-bus Write Dummy Data Bottom Half Register (Default Value: 0x0000_0000)

Offset: 0x0824			Register Name: PSRAM_C_WR_DUMMY_L
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	Bottom Half 32bits Of Code-bus DUMMY Data.

8.3.7.60 0x0828 PSRAM Code-bus IO Switch Wait Time Register (Default Value: 0x0000_0000)

Offset: 0x0828			Register Name: PSRAM_C_IO_SWIT_TIME
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0x0	PSRAM Code-bus Read Latency Wait Cycle
23: 0	R/W	/	/

NOTE

When the OPI Port Exist I/O Switch, Port Trans/Get Data Needs Suspend.

8.3.7.61 0x086C PSRAM Force Configure Register (Default Value: 0x0000_0000)

Offset: 0x086C			Register Name: PSRAM_FORCE_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1: 0	R/W	0x0	Psram Wait Half Cycle Number of Receive Data. 00: No Delay. 01: Delay 1 Half Cycle. 10: Delay 2 Half Cycle. 11: Delay 3 Half Cycle.

8.3.7.62 0x0870 PSRAM Common Configure Register (Default Value: 0x21C0_0000)

Offset: 0x0870			Register Name: PSRAM_COM_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:28	R/W	0x2	PSRAM Page Size Select. 00: 128 bytes 01: 256 bytes 10: 512 bytes 11: 1024 bytes
27:16	R/W	0x01C0	Maximum CE# low cycle number. CE# should not be kept low for more than 4 us. This value is different for various frequency. Default value is for working in 133MHz. When configured 0, it means not checking CE# low width.
15	/	/	/
14	R/W	0x0	Write or read could start on random address. 0: Write or read should start on even address. 1: Write or read could start on odd or even address. When writing or reading mode registers, set this to 1. When memory access, set this to 0.
13: 0	/	/	/

8.4 SD/MMC Host Controller (SMHC)

8.4.1 Overview

The SMHC controls the read/write operations on the secure digital (SD) cards, multimedia cards (MMC), and various extended devices that is based on the secure digital input/output (SDIO) protocol. The processor provides three SMHC interfaces for controlling the SD cards, MMCs, and SDIO devices.

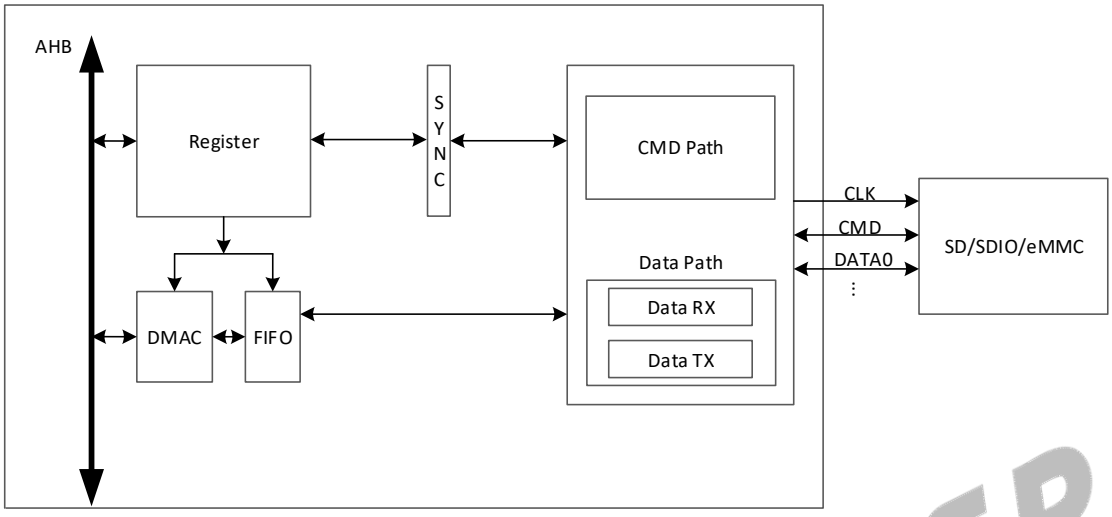
The SMHC has the following features:

- Compatible with Secure Digital Memory (SD mem-version 2.0)
- Compatible with Secure Digital I/O (SDIO-version 3.0)
- Compatible with embedded Multimedia Card (eMMC-version 5.0)
- Supports Card insertion and removal interrupt
- Supports hardware CRC generation and error detection
- Supports programmable baud rate
- Supports SDIO interrupts in 1-bit and 4-bit modes
- Supports block size of 1 to 65535 bytes
- Supports descriptor-based internal DMA controller
- Internal 1024-Bytes RX FIFO and 1024-Bytes TX FIFO
- Supports 1-bit, 4-bit SD and SDIO data bus width
- Supports 1-bit, 4-bit eMMC data bus width

8.4.2 Block Diagram

The following figure shows a block diagram of the SMHC.

Figure 8-3 SMHC Block Diagram



8.4.3 Functional Description

8.4.3.1 External Signals

The following table describes the external signals of SMHC.

Table 8-6 SMHC External Signals

Signal	Type	Description
SDC_CLK	O	Card Clock Output
SDC_CMD	O	Card Command Output
SDC_DATA0	I/O	Card Data Input/Output
SDC_DATA1	I/O	Card Data Input/Output
SDC_DATA2	I/O	Card Data Input/Output
SDC_DATA3	I/O	Card Data Input/Output

8.4.3.2 Clock Sources

The following table describes the clock sources of SMHC.

Table 8-7 SMHC Clock Sources

Clock Sources	Description
CLK_HOSC	Crystal clock. The default value is 24M.
CLK1_DEV	Device1 clock. The default value is 1920M.
CLK3_DEV	Device3 clock. The default value is 1600M.
CLK32K	Crystal clock. The default value is 32K.

8.4.3.3 Data Path

The SMHC and SD/SDIO/eMMC contains the following interface buses: CLK, CMD, and DATA 1⁄4. During one clock cycle, the SMHC can transmit a bit command with one or two bits data in 1-ch DATA mode, or four or eight bits data in 4-ch DATA mode. The CMD is a bidirectional channel for initializing the SD/SDIO/eMMC and transmitting commands. It can work in both the open-drain mode and push-pull mode. The DATA is also a bidirectional channel. It works in the push-pull mode.

Reading Data from the SD/SDIO/eMMC

The register configures the signals for the read operation, and synchronize the signals to the SMHC clock domain. Then the Data RX reads data from the SD/SDIO/eMMC via the CLK/CMD/DATA interface buses and writes the data in the FIFO. After that, the DMAC transfers the data from the FIFO to the memory.

Writing Data to the SD/SDIO/eMMC

The register configures the signals for the write operation, and synchronize the signals to the SMHC clock domain. Then the DMAC reads data from the memory and writes the data to the FIFO. After that, the Data TX reads the data from the FIFO and writes the data to the SD/SDIO/eMMC via the CLK/CMD/DATA interface buses.

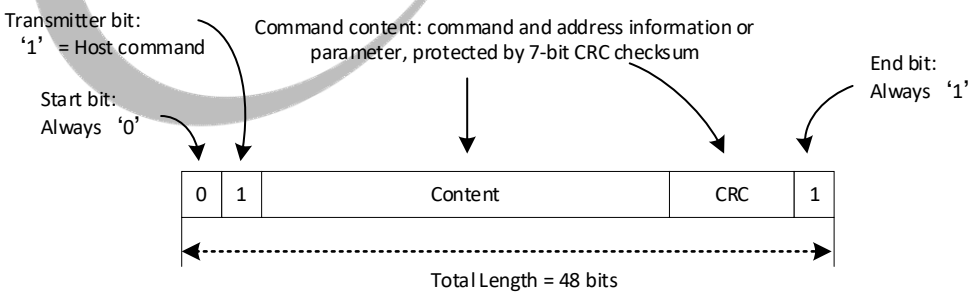
8.4.3.4 Package Format

Data transmission over the SD/eMMC bus is based on command and data bitstreams that are initiated by a start bit and terminated by a stop bit. There are three types of SD/eMMC packets: command token, response token, and data packet.

Command Tokens

The command token starts an operation. A command is sent from the host to a device. It is transferred serially on the CMD line. Command tokens have the following coding scheme:

Figure 8-4 Command Token Format



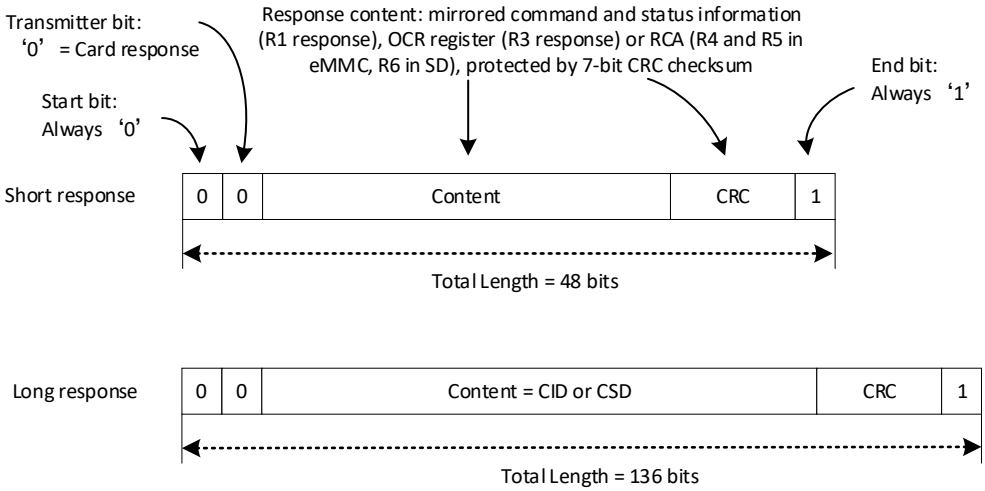
Each command token has 48 bits, preceded by a start bit ('0') and succeeded by an end bit ('1'). To detect transmission errors, each token is protected by CRC bits.

Response Tokens

After receiving a command, the card returns a 48-bit or 136-bit response based on the command type.

A response token is sent from the device to the host as an answer to a previously received command. It is transferred serially on the CMD line.

Figure 8-5 Response Token Format



Data Packets

Data can be transferred from the device to the host or vice versa. Data are transferred via the data lines.

NOTE

R128 does not support the bus width of 8 bits.

Figure 8-6 Data Packet Format for SDR

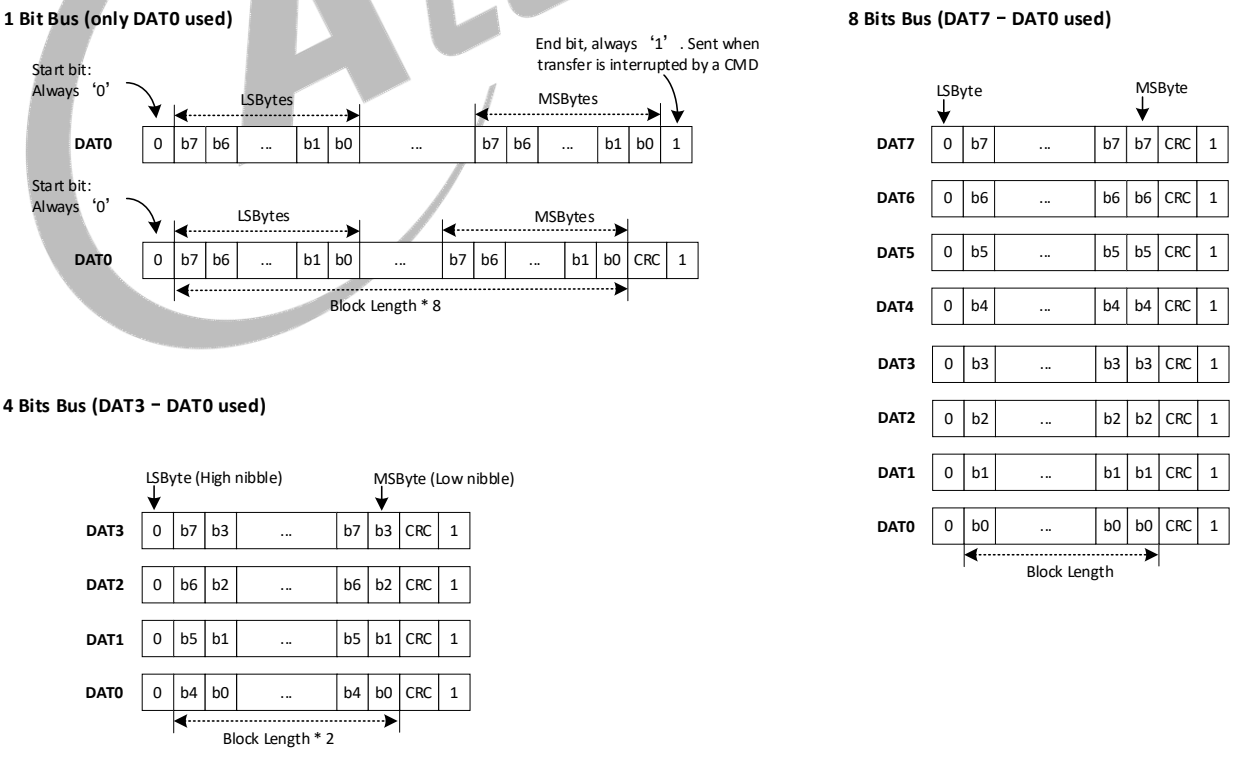
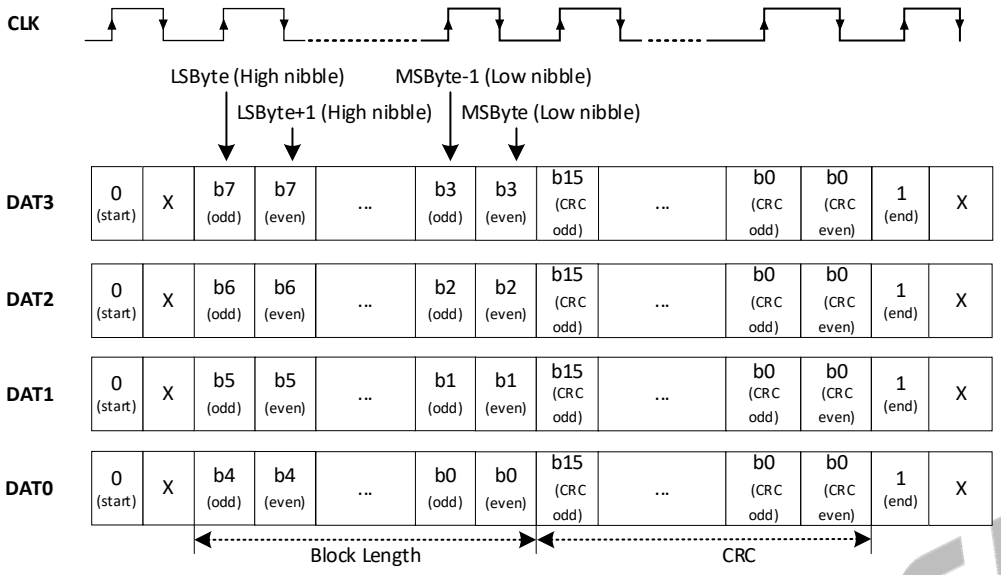
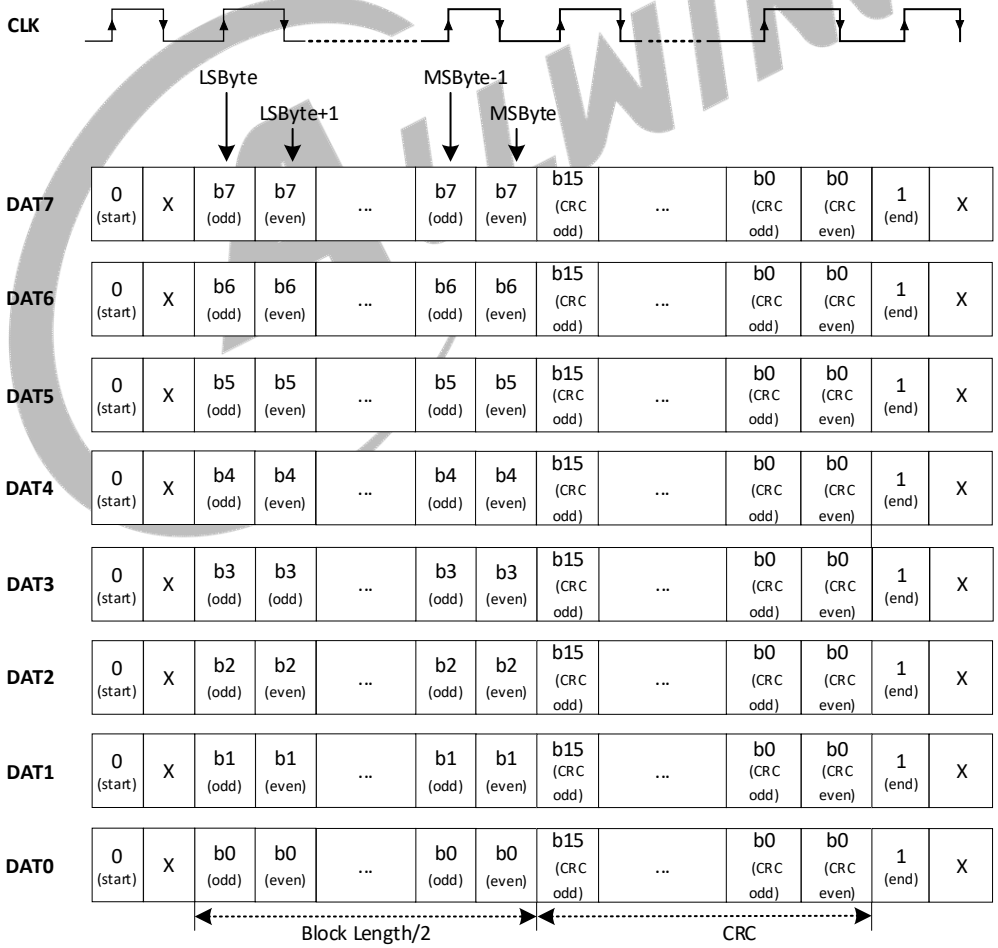


Figure 8-7 Data Packet Format for DDR

4 Bits Bus DDR (DAT3 – DAT0 used)



8 Bits Bus DDR (DAT7 – DAT0 used)



NOTE

a. Bytes data are not interleaved but CRCs are interleaved.

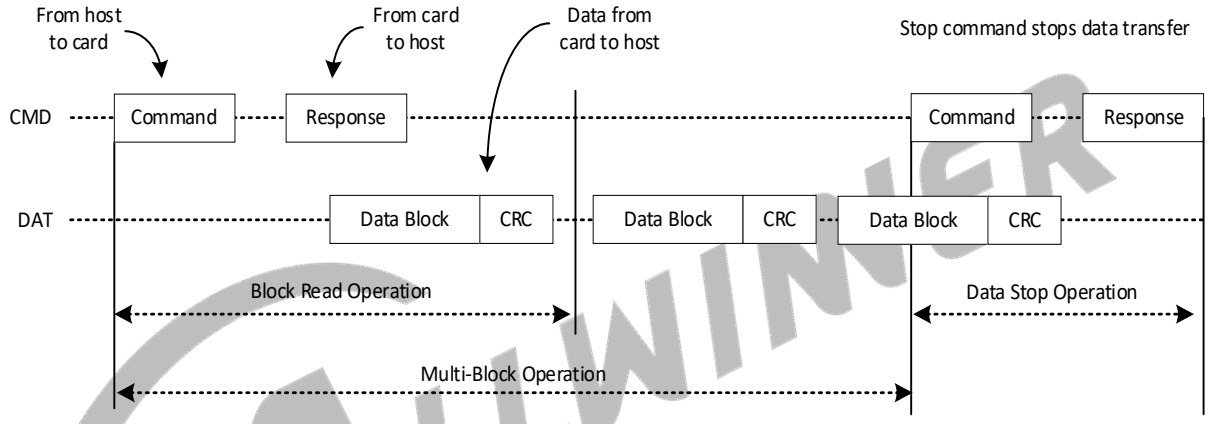
b. Start and end bits are only valid on the rising edge (“X” indicates “undefined”).

8.4.3.5 Data Transfer

Transferring data to or from the SD/eMMC card are done in blocks. Single and multiple block operations are widely used during data transmission.

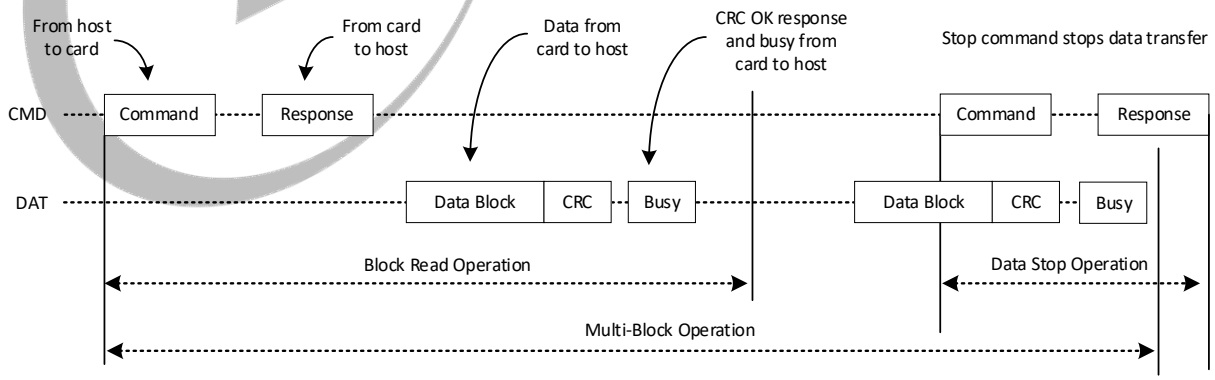
The following figure shows the single-block and multi-block read operation.

Figure 8-8 Single-Block and Multi-Block Read Operation



The following figure shows the single-block and multi-block write operation.

Figure 8-9 Single-Block and Multi-Block Read Operation



8.4.3.6 Internal DMA Controller Description

The SMHC has an internal DMA controller (IDMAC) to transfer data between the host memory and SMHC port. With a descriptor, the IDMAC can efficiently move data from the source to destination by automatically loading the next DMA transfer arguments, which needs less CPU intervention. Before transferring data in the IDMAC, the host driver should construct a descriptor list, configure arguments of every DMA transfer, and then launch the descriptor and start the DMA.

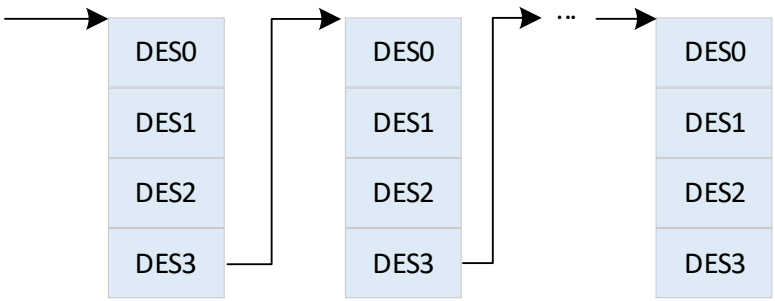
The IDMAC has an interrupt controller. When enabled, it generates an interrupt to the HOST CPU in situations such as data transmission is completed or some error is happened.

IDMAC Descriptor Structure

The IDMAC uses a descriptor with a chain structure, and each descriptor points to a unique buffer and the next descriptor.

The following figure shows the internal formats of a descriptor.

Figure 8-10 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 corresponds to the [31: 0] bits, DES1 corresponds to the [63:32] bits, DES2 corresponds to the [95:64] bits, and DES3 corresponds the [127:96] bits in a descriptor.

The following tables shows the bit definition of DES0, DES1, and DES2.

Table 8-8 DES0 Definition

Bits	Name	Description
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 the transfer is over.
30	ERROR	ERR_FLAG When some errors happen in transfer, this bit will be set to 1.
29:5	/	/
4	Chain Flag	CHAIN_MOD When set to 1, this bit indicates that the second address in the descriptor is the next descriptor address. It must be set to 1.
3	First DES Flag	FIRST_FLAG When set to 1, this bit indicates that this descriptor contains the first buffer of data. It must be set to 1 in the first DES.
2	Last DES Flag	LAST_FLAG When set to 1, this bit indicates that the buffers this descriptor points to are the last data buffer.
1	Disable Interrupt on completion	CUR_TXRX_OVER_INT_DIS When set to 1, this bit will prevent the setting of the TX/RX interrupt

Bits	Name	Description
		bit of the IDMAC status register for data that ends in the buffer the descriptor points to.
0	/	/

Table 8-9 DES1 Definition

Bits	Name	Description
31:13	/	/
12: 0	Buffer size	BUFF_SIZE The bits indicate the data buffer byte size, which must be a multiple of 4 bytes. If this field is 0, the DMA ignores this buffer and proceeds to the next descriptor.

Table 8-10 DES2 Definition

Bits	Name	Descriptor
31: 0	Buffer address pointer	BUFF_ADDR The bits indicate the physical address of the data buffer. It is a word (4 byte) address.

Table 8-11 DES3 Definition

Bits	Name	Descriptor
31: 0	Next descriptor address	NEXT_DESP_ADDR The bits indicate the pointer to the physical memory where the next descriptor is present. It is a word address.

8.4.3.7 Timing Specification

Refer to the following 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)

8.4.3.8 Calibrating the Delay Chain

There are two delay chains in SMHC: Data strobe delay chain and sample delay chain.

Data strobe delay chain: Used to generate delay to make proper timing between Data Strobe and data signals.

Sample delay chain: Used to generate delay to make proper timing between the internal card clock signal 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.

Follow the steps below to calibrate the delay chain:

Step 1 Enable SMHC. In order to calibrate the delay chain by the operation registers in SMHC, the SMHC must be enabled through AHB reset and AHB clock gating control registers.

Step 2 Configure a proper clock for SMHC. The delay chain calibration is based on the clock for SMHC from Clock Control Unit (CCU). The delay chain calibration is an internal function in SMHC and needs no devices. So it is unnecessary to open the clock signal for devices. The recommended clock frequency is 200 MHz.

Step 3 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.

Step 4 Write 0x8000 to **delay control register** to start calibrating the delay chain.

Step 5 Wait until the flag (bit14 in **delay control register**) of calibration done is set. The number of delay cells is shown at bit[14:8] in **delay control register**. The delay time generated by these delay cells is equal to the cycle of the SMHC clock nearly. This value is the result of calibration.

Step 6 Calculate the delay time of one delay cell according to the cycle of the SMHC clock and the result of calibration.

8.4.4 Programming Guidelines

8.4.4.1 Initializing SMHC

Before data and commands are exchanged between a card and the SMHC, the SMHC needs to be initialized. Follow the steps below to initialize the SMHC:

Step 1 Configure the corresponding GPIO register as an SMHC by Port Controller module; reset clock by writing 1 to MOD_RST_CTRL0[SDC0_RST], and open clock gating by writing 1 to SDC0_CLK_CTRL[MCLK_ENABLE]; select clock sources by configuring the SDC0_CLK_CTRL[SRC_SEL] register, and set the division factor by configuring SDC0_CLK_CTRL[CLK_DIV_RATIO_N/M].

Step 2 Configure SMHC_CTRL to reset the FIFO and controller, and enable the global interrupt; configure SMHC_INTMASK to 0xFFCE to enable normal interrupts and error abnormal interrupts, and then register the interrupt function.

Step 3 Configure SMHC_CLKDIV to open clock for devices; configure SMHC_CMD as the change clock command (for example 0x80202000); send the update clock command to deliver clocks to devices.

Step 4 Configure SMHC_CMD as a normal command. Configure SMHC_CMDARG to set command parameters. Configure SMHC_CMD to set parameters like whether to send the response, the response type, and the response length and then send the commands. According to the initialization process in the protocol, you can finish SMHC initialization by sending the corresponding command one by one.

8.4.4.2 Writing a Single Data Block

To write a single data block, follow the steps below:

Step 1 Write 0x1 to SMHC_CTRL[DMA_RST] to reset the internal DMA controller; write 0x82 to SMHC_DMAC to enable the IDMAC interrupt and configure AHB master burst transfers; configure SMHC_IDIE to enable the transfer interrupt, receive interrupt, and abnormal interrupt.

Step 2 Configure SMHC_FIFOTH to determine the burst size and TX/RX trigger level. For example, if SMHC_FIFOTH is configured as 0x300F00F0, it indicates the burst size is 16, TX_TL is 15, and RX_TL is 240. Configure SMHC_DLBA to determine the start address of the DMA descriptor.

Step 3 To write one block data to sector1, configure SMHC_BYTCNT[BYTE_CNT] to 0x200 and configure the descriptor according to the data size; set the data sector address of CMD24 (Single Data Block Write) to 0x1, write 0x80002758 to SMHC_CMD, and send CMD24 command to write data to the device.

Step 4 Check whether SMHC_RINTSTS[CC] is 1. If yes, the command is sent successfully; otherwise, continue to wait until timeout, and then exit the process.

Step 5 Check whether SMHC_IDST[TX_INT] is 1. If yes, the data transmission for writing DMA is completed. Write 0x337 to SMHC_IDST to clear the interrupt flag. Otherwise, continue to wait until timeout, and then exit the process.

Step 6 Check whether SMHC_RINTSTS[DTC] is 1. If yes, the data transmission and CMD24 writing operations are completed. Otherwise, abnormity exists. Read SMHC_RINTSTS and SMHC_STATUS to query the existing abnormity.

Step 7 Send CMD13 command to query whether the device writing operation is completed and returns to the 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, and then check whether the highest bit of SMHC_RESP0 (CMD13 response) is 1. If yes, the device is in the idle status, and the next command can be sent. Otherwise, the device is in the busy status. Continue to send CMD13 to wait for the device to enter the idle status until timeout.

8.4.4.3 Reading a Single Data Block

To read a single data block, follow the steps below:

Step 1 Write 0x1 to SMHC_CTRL[DMA_RST] to reset the internal DMA controller; write 0x82 to SMHC_DMAC to enable the IDMAC interrupt and configure AHB master burst transfers; configure SMHC_IDIE to enable the transfer interrupt, receive interrupt, and abnormal interrupt.

Step 2 Configure SMHC_FIFOTH to determine the burst size and TX/RX trigger level. For example, if SMHC_FIFOTH is configured as 0x300F00F0, it indicates the burst size is 16, TX_TL is 15, and RX_TL is 240. Configure SMHC_DLBA to determine the start address of the DMA descriptor.

Step 3 To read one block data from sector1, configure SMHC_BYTCNT [BYTE_CNT] to 0x200 and configure the descriptor according to the data size; set the data sector address of CMD17 command (Single Data Block Read) to 0x1, write 0x80002351 to SMHC_CMD, and send CMD17 command to read data from the device to DRAM/SRAM.

Step 4 Check whether SMHC_RINTSTS[CC] is 1. If yes, the command is sent successfully; otherwise, continue to wait until timeout, and then exit the process.

Step 5 Check whether SMHC_IDST[RX_INT] is 1. If yes, the data transmission for writing DMA is completed. Write 0x337 to SMHC_IDST to clear the interrupt flag. Otherwise, continue to wait until timeout, and then exit the process.

Step 6 Check whether SMHC_RINTSTS[DTC] is 1. If yes, data transmission and CMD17 reading operation are completed. Otherwise, abnormity exists. Read SMHC_RINTSTS and SMHC_STATUS to query the existing abnormity.

8.4.4.4 Writing Open-Ended Multiple Data Blocks

To write open-ended multiple data blocks, follow the steps below:

Step 1 Write 0x1 to SMHC_CTRL[DMA_RST] to reset the internal DMA controller; write 0x82 to SMHC_DMAC to enable the IDMAC interrupt and configure AHB master burst transfers; configure SMHC_IDIE to enable the transfer interrupt, receive interrupt, and abnormal interrupt.

Step 2 Configure SMHC_FIFOTH to determine the burst size and TX/RX trigger level. For example, if SMHC_FIFOTH is configured as 0x300F00F0, it indicates the burst size is 16, TX_TL is 15, and RX_TL is 240. Configure SMHC_DLBA to determine the start address of the DMA descriptor.

Step 3 To write three blocks of data to sectors begin with sector0, configure SMHC_BYTCNT[BYTE_CNT] to 0x600 and configure the descriptor according to the data size; set the data sector address of CMD25 command (Multiple Data Blocks Write) to 0x0, write 0x80003759 to SMHC_CMD, and send CMD25 command to read data from the device to DRAM/SRAM.

Step 4 Check whether SMHC_RINTSTS[CC] is 1. If yes, the command is sent successfully; otherwise, continue to wait until timeout, and then exit the process.

Step 5 Check whether SMHC_IDST[TX_INT] is 1. If yes, the data transmission for writing DMA is completed. Write 0x337 to SMHC_IDST to clear the interrupt flag. Otherwise, continue to wait until timeout, and then exit the process.

Step 6 Check whether SMHC_RINTSTS[ACD] and SMHC_RINTSTS[DTC] are both 1. If yes, the data transmission, CMD12 transmission, and CMD25 writing operations are completed. Otherwise, abnormity exists. Read SMHC_RINTSTS and SMHC_STATUS to query the existing abnormity.

Step 7 Send CMD13 command to query whether the device writing operation is completed and returns to the idle status. For example, device RCA is 0x1234, first set SMHC_CMDARG to 0x12340000, write 0x8000014D to SMHC_CMD, go to step 4 to ensure command transfer completed, and then check whether the highest bit of SMHC_RESP0 (CMD13 response) is 1. If yes, the device is in the idle status, and the next command can be sent. Otherwise, the device is in the busy status. Continue to send CMD13 to wait for the device to enter the idle status until timeout.

8.4.4.5 Reading Open-Ended Multiple Data Blocks (CMD18 + Auto CMD12)

To read open-ended multiple data blocks, follow the steps below:

Step 1 Write 0x1 to SMHC_CTRL[DMA_RST] to reset the internal DMA controller; write 0x82 to SMHC_DMAC to enable the IDMAC interrupt and configure AHB master burst transfers; configure SMHC_IDIE to enable the transfer interrupt, receive interrupt, and abnormal interrupt.

Step 2 Configure SMHC_FIFOTH to determine the burst size and TX/RX trigger level. For example, if SMHC_FIFOTH is configured as 0x300F00F0, it indicates the burst size is 16, TX_TL is 15, and RX_TL is 240. Configure SMHC_DLBA to determine the start address of the DMA descriptor.

Step 3 To read three blocks of data from sectors begin with sector0, configure SMHC_BYTCNT[BYTE_CNT] to 0x600 and configure the descriptor according to the data size; set the data sector address of CMD18 command (Multiple Data Blocks Read) to 0x0, write 0x80003352 to SMHC_CMD, and send CMD18 command to read data to the device. When the data transmission is completed, CMD12 will be sent automatically.

Step 4 Check whether SMHC_RINTSTS[CC] is 1. If yes, the command is sent successfully; otherwise, continue to wait until timeout, and then exit the process.

Step 5 Check whether SMHC_IDST[RX_INT] is 1. If yes, the data transmission for writing DMA is completed. Write 0x337 to SMHC_IDST to clear the interrupt flag. Otherwise, continue to wait until timeout, and then exit the process.

Step 6 Check whether SMHC_RINTSTS[ACD] and SMHC_RINTSTS[DTC] are both 1. If yes, data transmission, CMD12 transmission, and CMD18 reading operation are completed. Otherwise, abnormity exists. Read SMHC_RINTSTS and SMHC_STATUS to query the existing abnormity.

8.4.4.6 Writing Pre-Defined Multiple Data Blocks (CMD23 + CMD25)

To write pre-defined multiple data blocks, follow the steps below:

Step 1 Write 0x1 to SMHC_CTRL[DMA_RST] to reset the internal DMA controller; write 0x82 to SMHC_DMAC to enable the IDMAC interrupt and configure AHB master burst transfers; configure SMHC_IDIE to enable the transfer interrupt, receive interrupt, and abnormal interrupt.

Step 2 Configure SMHC_FIFOTH to determine the burst size and TX/RX trigger level. For example, if SMHC_FIFOTH is configured as 0x300F00F0, it indicates the burst size is 16, TX_TL is 15, and RX_TL is 240. Configure SMHC_DLBA to determine the start address of the DMA descriptor.

Step 3 To write three blocks of data, configure SMHC_CMDARG to 0x3 to specify the number of data blocks as three. Then write 0x80000157 to SMHC_CMD to send the CMD23 command. Check whether SMHC_RINTSTS[CC] is 1. If yes, the command is sent successful; otherwise, continue to wait until timeout, and then exit the process.

Step 4 Configure SMHC_BYTCNT[BYTE_CNT] to 0x600 and configure the descriptor according to the data size; set the data sector address of CMD25 command (Multiple Data Blocks Write) to 0x0, write 0x80002759 to SMHC_CMD, and send CMD25 command to write data to the device.

Step 5 Check whether SMHC_RINTSTS[CC] is 1. If yes, the command is sent successful; otherwise, continue to wait until timeout, and then exit the process.

Step 6 Check whether SMHC_IDST[TX_INT] is 1. If yes, the data transmission for writing DMA is completed. Write 0x337 to SMHC_IDST to clear the interrupt flag. Otherwise, continue to wait until timeout, and then exit the process.

Step 7 Check whether SMHC_RINTSTS[DTC] is 1. If yes, the data transmission and CMD25 writing operations are completed. Otherwise, abnormity exists. Read SMHC_RINTSTS and SMHC_STATUS to query the existing abnormity.

Step 8 Send CMD13 command to query whether the device writing operation is completed and returns to the idle status. For example, device RCA is 0x1234, first set SMHC_CMDARG to 0x12340000, write 0x8000014D to SMHC_CMD, go to step 4 to ensure command transfer completed, and then check whether the highest bit of SMHC_RESP0 (CMD13 response) is 1. If yes, the device is in the idle status, and the next command can be sent. Otherwise, the device is in the busy status. Continue to send CMD13 to wait for the device to enter the idle status until timeout.

8.4.4.7 Reading Pre-Defined Multiple Data Blocks (CMD23 + CMD18)

To read pre-defined multiple data blocks, follow the steps below:

Step 1 Write 0x1 to SMHC_CTRL[DMA_RST] to reset the internal DMA controller; write 0x82 to SMHC_DMAC to enable the IDMAC interrupt and configure AHB master burst transfers; configure SMHC_IDIE to enable the transfer interrupt, receive interrupt, and abnormal interrupt.

Step 2 Configure SMHC_FIFOTH to determine the burst size and TX/RX trigger level. For example, if SMHC_FIFOTH is configured as 0x300F00F0, it indicates the burst size is 16, TX_TL is 15, and RX_TL is 240. Configure SMHC_DLBA to determine the start address of the DMA descriptor.

Step 3 To read three blocks of data, configure SMHC_CMDARG to 0x3 to specify the number of data blocks as three. Then write 0x80000157 to SMHC_CMD to send the CMD23 command. Check whether SMHC_RINTSTS[CC] is 1. If yes, the command is sent successful; otherwise, continue to wait until timeout, and then exit the process.

Step 4 Configure SMHC_BYTCNT[BYTE_CNT] to 0x600 and configure the descriptor according to the data size; set the data sector address of CMD18 (Multiple Data Blocks Read) to 0x0, write 0x80002352 to SMHC_CMD, and send CMD18 command to read data from device to DRAM/SRAM.

Step 5 Check whether SMHC_RINTSTS[CC] is 1. If yes, the command is sent successful; otherwise, continue to wait until timeout, and then exit the process.

Step 6 Check whether SMHC_IDST[RX_INT] is 1. If yes, the data transmission for writing DMA is completed. Write 0x337 to SMHC_IDST to clear the interrupt flag. Otherwise, continue to wait until timeout, and then exit the process.

Step 7 Check whether SMHC_RINTSTS[DTC] is 1. If yes, the data transmission and CMD18 writing operations are completed. Otherwise, abnormity exists. Read SMHC_RINTSTS and SMHC_STATUS to query the existing abnormity.

8.4.5 Register List

Module Name	Base Address
SDIO	0x40008000

Register Name	Offset	Description
SMHC_CTRL	0x0000	SMHC Global Control Register
SMHC_CLKDIV	0x0004	SMHC Clock Control Register
SMHC_TMOUT	0x0008	SMHC Timeout Register
SMHC_CTYPE	0x000C	SMHC Bus Width Register
SMHC_BLKSIZ	0x0010	SMHC Block Size Register
SMHC_BYTCNT	0x0014	SMHC Byte Count Register
SMHC_CMD	0x0018	SMHC Command Register
SMHC_CMDARG	0x001C	SMHC Command Argument Register
SMHC_RESP0	0x0020	SMHC Response 0 Register
SMHC_RESP1	0x0024	SMHC Response 1 Register
SMHC_RESP2	0x0028	SMHC Response 2 Register
SMHC_RESP3	0x002C	SMHC Response 3 Register
SMHC_INTMASK	0x0030	SMHC Interrupt Mask Register
SMHC_MINTSTS	0x0034	SMHC Masked Interrupt Status Register

Register Name	Offset	Description
SMHC_RINTSTS	0x0038	SMHC Raw Interrupt Status Register
SMHC_STATUS	0x003C	SMHC Status Register
SMHC_FIFOTH	0x0040	SMHC FIFO Water Level Register
SMHC_FUNS	0x0044	SMHC FIFO Function Select Register
SMHC_TBC0	0x0048	SMHC Transferred Byte Count Register 0
SMHC_TBC1	0x004C	SMHC Transferred Byte Count Register 1
SMHC_DBGC	0x0050	SMHC Debug Control Register
SMHC_CSDC	0x0054	SMHC CRC Status Detect Control Register
SMHC_A12A	0x0058	SMHC Auto Command 12 Argument Register
SMHC_NTSR	0x005C	SMHC New Timing Set Register
SMHC_HWRST	0x0078	SMHC Hardware Reset Register
SMHC_DMAC	0x0080	SMHC DMAC Control Register
SMHC_DLBA	0x0084	SMHC Descriptor List Base Address Register
SMHC_IDST	0x0088	SMHC DMAC Status Register
SMHC_IDIE	0x008C	SMHC DMAC Interrupt Enable Register
SMHC_CHDA	0x0090	SMHC Current Host Descriptor Address Register
SMHC_CDBA	0x0094	SMHC Current DMA Buffer Descriptor Address Register
SMHC_THLD	0x0100	SMHC Card Threshold Control Register
SMHC_SFC	0x0104	SMHC Sample FIFO Control Register
SMHC_A23A	0x0108	SMHC Auto Command 23 Argument Register
EMMC_DDR_SBIT_DET	0x010C	SMHC eMMC4.5 DDR Start Bit Detection Control Register
SMHC_EXT_CMD	0x0138	SMHC Extended Command Register
SMHC_EXT_RESP	0x013C	SMHC Extended Response Register
SMHC_DRV_DL	0x0140	SMHC Drive Delay Control Register
SMHC_SAMP_DL	0x0144	SMHC Sample Delay Control Register
SMHC_DS_DL	0x0148	SMHC Data Strobe Delay Control Register
SMHC_FIFO	0x0200	SMHC FIFO Register

8.4.6 Register Description

8.4.6.1 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 1: AHB bus 0: DMA bus
30:13	/	/	/

Offset: 0x0000			Register Name: SMHC_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
12	R/W	0x0	TIME_UNIT_CMD Time unit for command line Time unit 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 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 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.

Offset: 0x0000			Register Name: SMHC_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
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.

8.4.6.2 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_DAT0 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 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)

8.4.6.3 0x0008 SMHC Timeout Register (Default Value: 0xFFFF_FF40)

Offset: 0x0008			Register Name: SMHC_TMOUT
Bit	**Read/Write**	**Default/Hex**	**Description**

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),the 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, this value is no effect.
7: 0	R/W	0x40	RTO_LMT Response Timeout Limit

8.4.6.4 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

8.4.6.5 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

8.4.6.6 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.

8.4.6.7 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 will be set in SMHC_RINTSTS 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, 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	/	/	/

Offset: 0x0018			Register Name: SMHC_CMD
Bit	**Read/Write**	**Default/Hex**	**Description**
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 the current data transmission.(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 transferring 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 transmission 1: Send stop command automatically at end of data transmission If set, the SMHC_RESP1 will record the response of auto CMD12.
11	R/W	0x0	TRANS_MODE Transfer Mode 0: Block data transmission command 1: Stream data transmission command
10	R/W	0x0	TRANS_DIR Transfer Direction 0: Read operation 1: Write operation
9	R/W	0x0	DATA_TRANS Data Transmission 0: Without data transmission 1: With data transmission

Offset: 0x0018			Register Name: SMHC_CMD
Bit	**Read/Write**	**Default/Hex**	**Description**
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

8.4.6.8 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

8.4.6.9 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

8.4.6.10 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[63:32] of response

8.4.6.11 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

8.4.6.12 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

8.4.6.13 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

Offset: 0x0030			Register Name: SMHC_INTMASK
Bit	**Read/Write**	**Default/Hex**	**Description**
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 Transmission 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	/	/	/

8.4.6.14 0x0034 SMHC Masked Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: SMHC_MINTSTS
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x0	M_CARD_REMOVAL_INT Card Removed
30	R	0x0	M_CARD_INSERT Card Inserted
29:17	/	/	/
16	R	0x0	M_SDIO_INT SDIO Interrupt
15	R	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 token or received CRC status token is negative.
14	R	0x0	M_ACD_INT Auto Command Done When set, it means auto stop command(CMD12) completed.
13	R	0x0	M_DSE_BC_INT Data Start Error/busy clear 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 after the last block.
12	R	0x0	M_CB_IW_INT Command Busy and Illegal Write

Offset: 0x0034			Register Name: SMHC_MINTSTS
Bit	**Read/Write**	**Default/Hex**	**Description**
11	R	0x0	M_FU_FO_INT FIFO Underrun/Overflow
10	R	0x0	M_DSTO_VSD_INT Data Starvation Timeout/V1.8 Switch Done
9	R	0x0	M_DTO_BDS_INT Data Timeout/Boot Data Start
8	R	0x0	M_RTO_BACK_INT Response Timeout/Boot ACK Received
7	R	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 token is negative.
6	R	0x0	M_RCE_INT Response CRC Error
5	R	0x0	M_DRR_INT Data Receive Request When set, it means that there are enough data in FIFO during receiving data.
4	R	0x0	M_DTR_INT Data Transmit Request When set, it means that there are enough space in FIFO during transmitting data.
3	R	0x0	M_DTC_INT Data Transmission Complete
2	R	0x0	M_CC_INT Command Complete
1	R	0x0	M_RE_INT Response Error When set, Transmit Bit error or End Bit error or CMD Index error may occurs.
0	/	/	/

8.4.6.15 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.

Offset: 0x0038			Register Name: SMHC_RINTSTS
Bit	**Read/Write**	**Default/Hex**	**Description**
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 token. 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/busy clear 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 after the last block. 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 did 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.

Offset: 0x0038			Register Name: SMHC_RINTSTS
Bit	**Read/Write**	**Default/Hex**	**Description**
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 token 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 is enough space in FIFO during transmitting data. This is write-1-to-clear bits.
3	R/W1C	0x0	DTC Data Transmission Complete 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	/	/	/

8.4.6.16 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

Offset: 0x003C			Register Name: SMHC_STATUS
Bit	**Read/Write**	**Default/Hex**	**Description**
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]; check 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

Offset: 0x003C			Register Name: SMHC_STATUS
Bit	**Read/Write**	**Default/Hex**	**Description**
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 TX Water Level flag 0: FIFO didn't reach receive trigger level 1: FIFO reached receive trigger level

8.4.6.17 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 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) Recommended: MSize = 16, TX_TL = 240, RX_TL = 15 FIFO_DEPTH = 256 FIFO_SIZE = 256 * 32 = 1K
27:24	R	0x0	/

Offset: 0x0040			Register Name: SMHC_FIFOTH
Bit	**Read/Write**	**Default/Hex**	**Description**
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. 15 (means greater than 15)
15:8	R	0x0	/
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) 240(means less than or equal to 240)

8.4.6.18 0x0044 SMHC FIFO 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.

Offset: 0x0044			Register Name: SMHC_FUNS
Bit	**Read/Write**	**Default/Hex**	**Description**
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.

8.4.6.19 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 transmission completes.

8.4.6.20 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 transmission completes.

8.4.6.21 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	/	/	/

Offset: 0x0054			Register Name: SMHC_CSDC
Bit	Read/Write	Default/Hex	Description
3: 0	R/W	0x3	CRC_DET_PARA CRC Detect Para 011: Other speed mode Others: Reserved

8.4.6.22 0x0058 SMHC Auto Command 12 Argument Register (Default Value: 0x0000_FFFF)

Offset: 0x0058			Register Name: SMHC_A12A
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0xFFFF	A12A Auto CMD12 Argument The argument of command 12 automatically send by controller with this field.

8.4.6.23 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_SEL Mode Select 0: Old mode of Sample/Output Timing 1: New mode of Sample/Output Timing Default value : 1
30:25	/	/	/
24	R/W	0x1	CMD_DAT_RX_PHASE_CLR During update clock operation, clear command line's and data lines' input phase. 0: Disable 1: Enable
23	/	/	/
22	R/W	0x1	DAT_CRC_STATUS_RX_PHASE_CLR Before receive CRC status, clear data lines' input phase. 0: Disable 1: Enable
21	R/W	0x1	DAT_TRANS_RX_PHASE_CLR Before transfer data, clear data lines' input phase. 0: Disable 1: Enable
20	R/W	0x1	DAT_RECV_RX_PHASE_CLR Before receive data, clear data lines' input phase clear 0: Disable 1: Enable

Offset: 0x005C			Register Name: SMHC_NTSR
Bit	**Read/Write**	**Default/Hex**	**Description**
19:17	/	/	/
16	R/W	0x1	CMD_SEND_RX_PHASE_CLR Before send command, command rx phase clear 0: Disable 1: Enable
15:10	/	/	/
9:8	R/W	0x0	DAT_SAM_TIM_PHS Data Sample Timing Phase 00: Sample timing phase offset 90^0 01: Sample timing phase offset 180^0 10: Sample timing phase offset 270^0 Default value: 00
7:6	/	/	/
5:4	R/W	0x0	CMD_SAM_TIM_PHS Command 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 Default value: 00
3: 0	/	/	/

8.4.6.24 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 Hardware Reset 1: Active mode 0: Reset These bits cause the cards to enter pre-idle state, which requires them to be re-initialized.

8.4.6.25 0x0080 SMHC DMAC Control Register (Default Value: 0x0000_0000)

Offset: 0x0080			Register Name: SMHC_DMAC
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0080			Register Name: SMHC_DMAC
Bit	**Read/Write**	**Default/Hex**	**Description**
31	W	0x0	DES_LOAD_CTRL When DMAC fetches a descriptor, if the valid bit of a descriptor is not set, DMAC FSM will go to the suspend state. Setting this bit will make DMAC re-fetch descriptor again and do the transfer normally.
30:11	/	/	/
10:8	/	/	/
7	R/W	0x0	IDMAC_ENB IDMAC Enable. When set, the IDMAC is enabled. DE is read/write.
6:2	/	/	/
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.

8.4.6.26 0x0084 SMHC Descriptor List Base Address Register (Default Value: 0x0000_0000)

Offset: 0x0084			Register Name: SMHC_DLBA
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. It is a word(4byte) address.

8.4.6.27 0x0088 SMHC DMAC Status Register (Default Value: 0x0000_0000)

Offset: 0x0088			Register Name: SMHC_IDST
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16:13	/	/	/

Offset: 0x0088			Register Name: SMHC_IDST
Bit	**Read/Write**	**Default/Hex**	**Description**
12:10	R	0x0	DMAC_ERR_STA Error Bits. Indicates the type of error that caused a Bus Error. Valid only with Fatal Bus Error bit (SMHC_IDST[2]) set. This field does not generate an interrupt. 001: Host Abort received during transmission 010: Host Abort received during reception Others: Reserved This bit is read-only.
9	R/W1C	0x0	AIS Abnormal Interrupt Summary. Logical OR of the following: SMHC_IDST[2]: Fatal Bus Interrupt SMHC_IDST[4]: Descriptor unavailable bit Interrupt SMHC_IDST[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	NIS Normal Interrupt Summary. Logical OR of the following: SMHC_IDST[0]: Transmit Interrupt SMHC_IDST[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 RCRC: Response CRC SBE: Start Bit Error DRTO: Data Read Timeout DCRC: Data CRC for Receive RE: Response Error Writing a 1 clears this bit.

Offset: 0x0088			Register Name: SMHC_IDST
Bit	**Read/Write**	**Default/Hex**	**Description**
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 (SMHC_IDST [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.

8.4.6.28 0x008C SMHC DMAC Interrupt Enable Register (Default Value: 0x0000_0000)

Offset: 0x008C			Register Name: SMHC_IDIE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:6	/	/	/
5	R/W	0x0	ERR_SUM_INT_ENB Card Error Summary Interrupt Enable. 1: Enable the Card Error Summary Interrupt. 0: Disable the Card Error Summary Interrupt
4	R/W	0x0	DES_UNAVL_INT_ENB Descriptor Unavailable Interrupt. When set with Abnormal Interrupt Summary Enable, the Descriptor Unavailable interrupt is enabled.
3	/	/	/
2	R/W	0x0	FERR_INT_ENB Fatal Bus Error Enable. When set with Abnormal Interrupt Summary Enable, the Fatal Bus Error Interrupt is enabled. When reset, Fatal Bus Error Enable Interrupt is disabled.

Offset: 0x008C			Register Name: SMHC_IDIE
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x0	RX_INT_ENB Receive Interrupt Enable. When set 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 set with Normal Interrupt Summary Enable, Transmit Interrupt is enabled. When reset, Transmit Interrupt is disabled.

8.4.6.29 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_WR_THLD Card Read/write Threshold Size
15:3	/	/	/
2	R/W	0x0	CARD_WR_THLD_ENB Card Write Threshold Enable 0: Card write threshold disable 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 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 Disable 1: Card Read Threshold Enable Host controller initiates Read Transfer only if CARD_RD_THLD amount of space is available in receive FIFO

8.4.6.30 0x0104 SMHC Sample FIFO Control Register (Default Value: 0x0000_0006)

Offset: 0x0104			Register Name: SMHC_SFC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/

Offset: 0x0104			Register Name: SMHC_SFC
Bit	**Read/Write**	**Default/Hex**	**Description**
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 change between 0x0 and 0xF, but actually the available value and the position of stopping clock must be decided by the actual situation. The value increase one in this field is linked to one cycle(two cycle 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.

8.4.6.31 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.

8.4.6.32 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**
31:1	/	/	/
0	R/W	0x0	HALF_START_BIT Control for start bit detection mechanism inside host controller 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.

8.4.6.33 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 set this bit, send CMD23 automatically before send command specified in SMHC_CMD register. When SOFT_RST set, this field will be cleared.

8.4.6.34 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.

8.4.6.35 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 When 0x5c[31]=0: 0: Data drive phase offset is 90° at SDR mode, 45° at DDR8 mode, 90° at DDR4 mode. 1: Data drive phase offset is 180° at SDR mode, 90° at DDR8 mode, 0° at DDR4 mode. When 0x5c[31]=1: 0: Data drive phase offset is 90° at SDR mode, 45° at DDR mode; 1: Data drive phase offset is 180° at SDR mode, 90° at DDR mode
16	R/W	0x1	CMD_DRV_PH_SEL Command Drive Phase Select When 0x5c[31]=0: 0: Command drive phase offset is 90° at SDR mode, 45° at DDR8 mode, 90° at DDR4 mode. 1: Command drive phase offset is 180° at SDR mode, 90° at DDR8 mode, 180° at DDR4 mode. When 0x5c[31]=1: 0: Command drive phase offset is 90° at SDR mode, 45° at DDR mode. 1: Command drive phase offset is 180° at SDR mode, 90° at DDR mode.
15: 0	/	/	/

8.4.6.36 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 1: Start sample delay chain calibration. 0: Stop 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 1: Enable sample delay specified at SAMP_DL_SW 0: Disable 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.

8.4.6.37 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.

Offset: 0x0148			Register Name: SMHC_DS_DL
Bit	**Read/Write**	**Default/ Hex**	**Description**
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

8.4.6.38 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

9 Peripherals

9.1 TWI

9.1.1 Overview

The two-wire interface (TWI) is designed to be used as an interface between CPU and the serial TWI bus. It supports all the standard TWI transfer, including Slave and Master. The communication of the two-wire bus is carried out by a byte-wise mode based on interrupt or polled handshaking. This TWI controller can be operated in standard mode (up to 100 kbit/s) and fast mode (up to 400 kbit/s). Masters and 10-bit addressing mode are supported for this specified application. General Call Addressing is also supported in Slave mode.

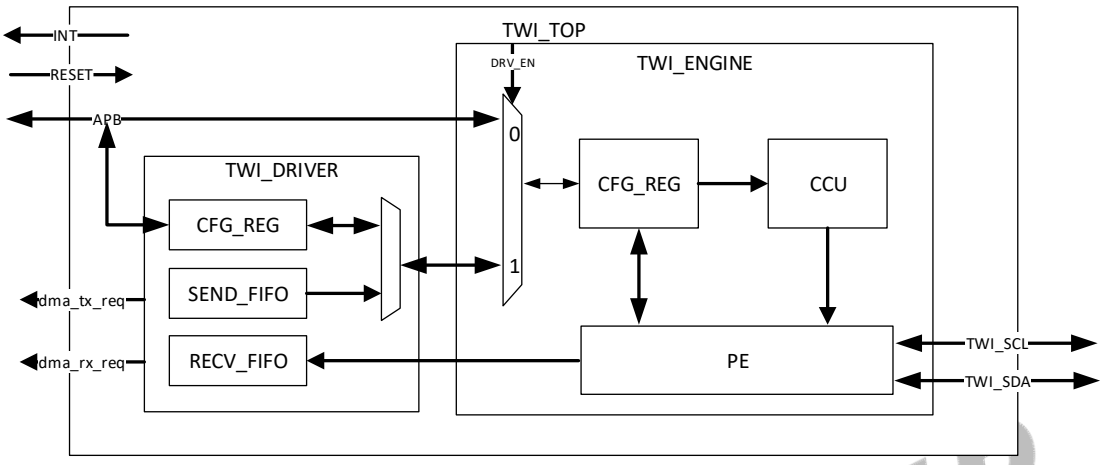
The TWI controller includes the following features:

- Supports 2 TWIs
- Software-programmable for Slave or Master
- Supports Repeated START signal
- Allows 10-bit addressing with TWI bus
- Performs arbitration and clock synchronization
- Owns address and General Call address detection
- Interrupt on address detection
- Supports speeding up to 400 kbit/s in fast mode
- Allows operation from a wide range of input clock frequencies
- TWI Driver supports packet transmission and DMA when TWI works in Master mode

9.1.2 Block Diagram

The following figure shows the block diagram of TWI.

Figure 9-1 TWI Block Diagram



TWI contains the following sub-blocks.

Table 9-1 TWI Sub-blocks

Term	Definition
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 Functional Description

9.1.3.1 External Signals

The following table 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 an output pin. When TWI is configurable as Slave device, TWI_SCK is an 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 chapter3.

Table 9-2 TWI External Signals

Signal	Description	Type
TWI0_SCK	TWI0 Clock Signal	I/O
TWI0_SDA	TWI0 Serial Data	I/O
TWI1_SCK	TWI1 Clock Signal	I/O
TWI1_SDA	TWI1 Serial Data	I/O

9.1.3.2 Clock Sources

Each TWI controller has a fixed clock source. APB2 is the clock source of TWI in CPUX, and APBS is the clock source of R-TWI in CPUS. The APB Bus gets a few clock sources. Users can select one of them to make APB clock. The following table describes the clock sources for TWI. For clock setting, configurations and gating information, refer to the section "CCU" and "CCU_AON".

Table 9-3 TWI Clock Sources

Clock Sources	Description
APBS Bus	R-TWI in CPUS, for details on APBS refer to *PRCM*
APB2 Bus	TWI in CPUX, for details on APB2 refer to *CCU*

After select a proper clock, for using the TWI in CPUX, users must open the gating of TWI and release the reset bit. For using the TWI in CPUS, users also need to open the gating of R-TWI and release the reset bit.

For more details on the gating/reset reg, refer to the CCU or PRCM.

9.1.3.3 Master and Slave Mode of TWI Engine

There are four operation modes on the TWI bus which dictates the communications method. They are Master Transmit, Master Receive, Slave Transmit and Slave Receive. In general, CPU host controls TWI Engine by writing commands and data to its registers. TWI Engine transmits an interrupt to CPU each time when a byte transfer is done or a START/STOP conditions is detected. The CPU host can also poll the status register for current status if the interrupt mechanism is not disabled by the CPU host.

When the CPU host wants to start a bus transfer, it initiates a bus START to enter the master mode by setting IM_STA bit in the 2WIRE_CNTR register to high (before it must be low). The TWI Engine will assert INT line and INT_FLAG to indicate a completion for the START condition and each consequent byte transfer. At each interrupt, the micro-processor needs to check the 2WIRE_STAT register for current status. A transfer has to be concluded with STOP condition by setting M_STP bit high.

In Slave Mode, the TWI Engine also constantly samples the bus and looks for its own slave address during addressing cycles. Once a match is found, it is addressed and interrupt the CPU host with the corresponding status. Upon request, the CPU host should read the status, read/write 2WIRE_DATA data register, and set the 2WIRE_CNTR control register. After each byte transfer, a slave device always halts the operation of remote master by holding the next low pulse on SCL line until the microprocessor responds to the status of previous byte transfer or START condition.

9.1.3.4 TWI Driver

TWI Driver is only supported in master mode. When TWI works in master mode, TWI Driver drives TWI Engine for one or more packet transmission instead of CPU host. Packet transmission is defined as follows: Reg address bytes and Write data bytes are buffered in SEND FIFO; reading data is buffered in RECV FIFO.

Figure 9-2 TWI Driver Write Packet Transmission

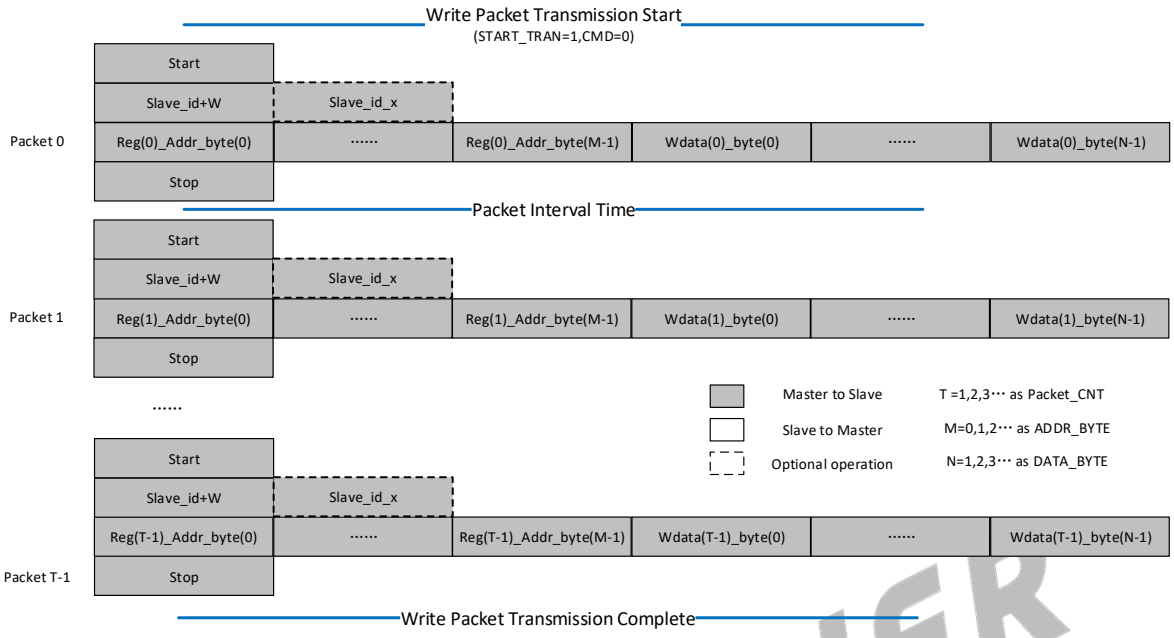
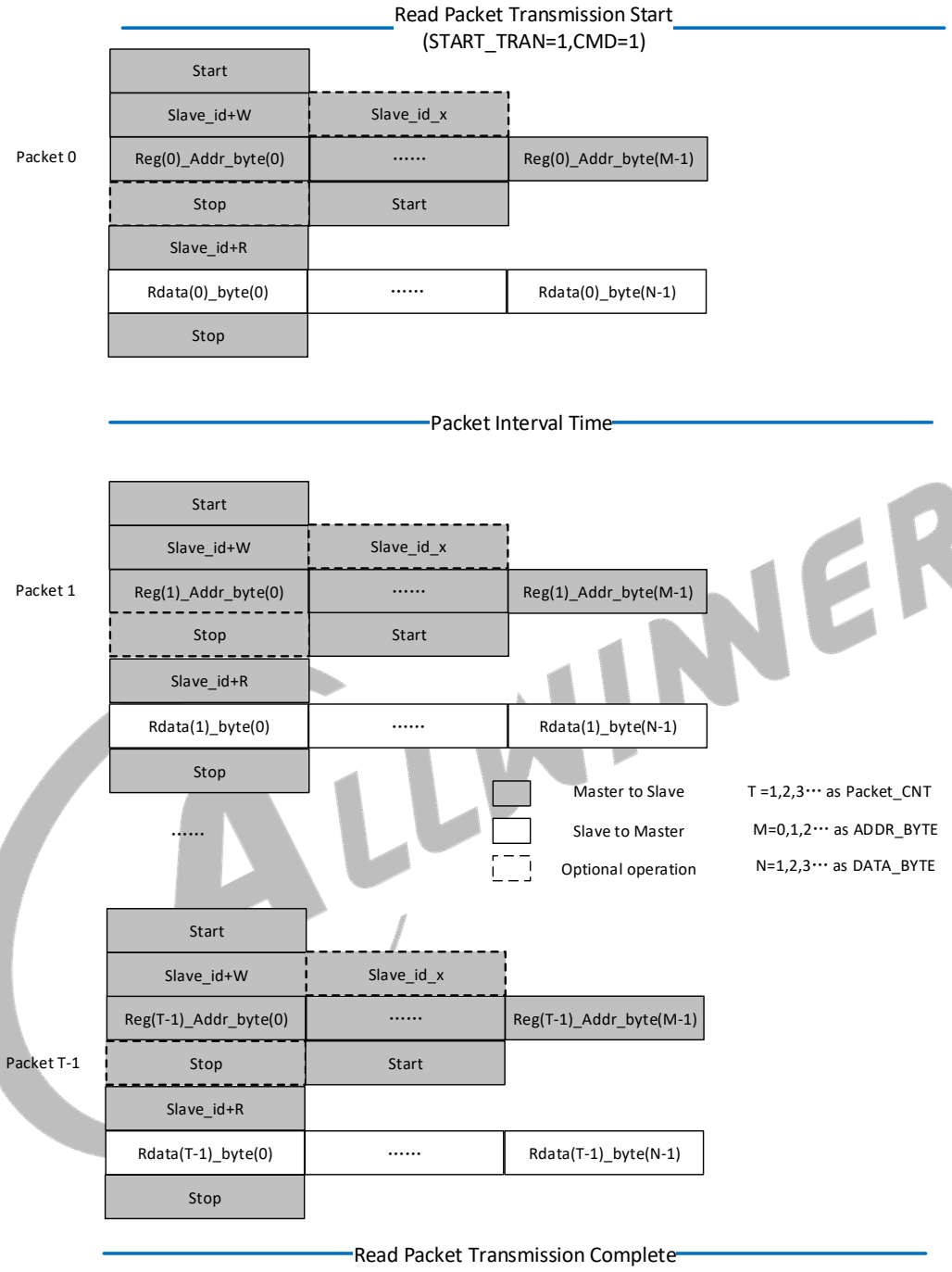


Figure 9-3 TWI Driver Write Packet Transmission

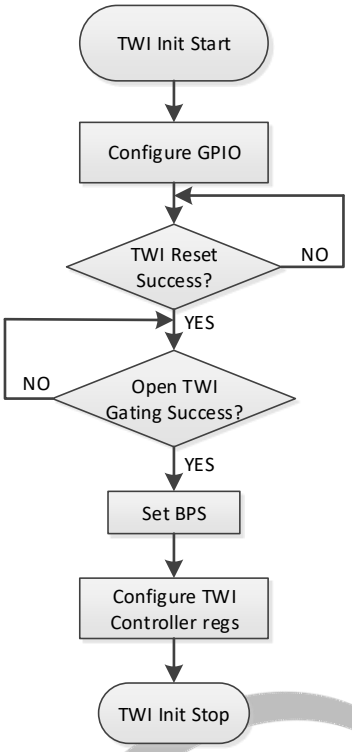


9.1.4 Programming Guidelines

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 sends a start condition. In the addressing formats of 7-bit, TWI sends out an 8-bit 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, refer to TWI_XADDR.

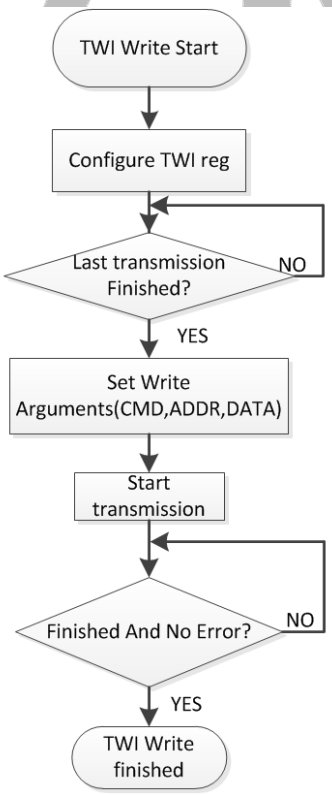
The following figure shows a software operation flow of TWI Initialization.

Figure 9-4 TWI Initialization Flow



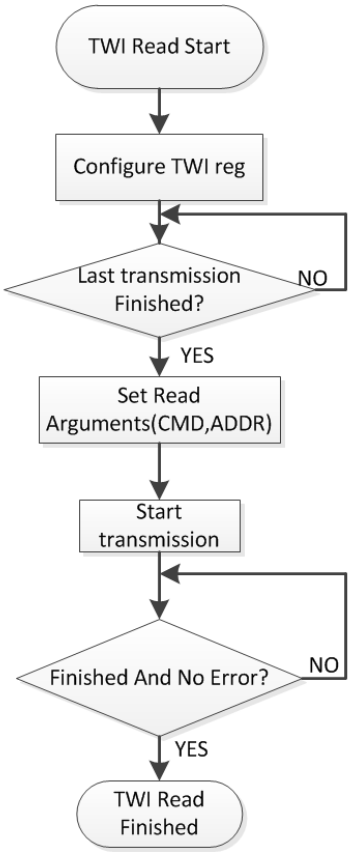
The following figure shows a software operation flow to control TWI Engine write to device.

Figure 9-5 TWI Write Flow



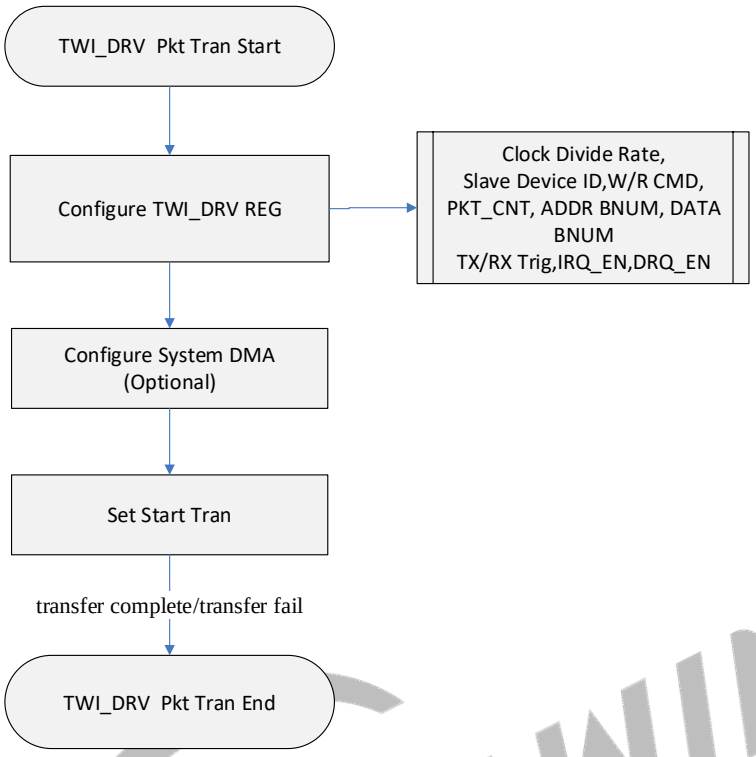
The following figure shows a software operation flow to control TWI Engine read from device.

Figure 9-6 TWI Read Flow



The following figure shows a software operation flow for Packet Transmission by TWI Driver.

Figure 9-7 TWI Driver Packet Transmission Flow



9.1.5 Register List

Module Name	Base Address
TWI0	0x40049000
TWI1	0x40049400

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

Register Name	Offset	Description
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 when in slave mode. When the TWI receives this address after a START condition, it will generate an interrupt and enter slave mode. (SLA6 corresponds to the first bit received from the TWI bus.) If GCE is set to '1', the TWI will also recognize the general call address (00h).

For 10-bit addressing:

When the address received starts with 11110b, the TWI recognizes this as the first part of a 10-bit address and if the next two bits match ADDR [2:1] (i.e. SLAX9 and SLAX8 of the device's extended address), it sends an ACK. (The device does not generate an interrupt at this point.) If the next byte of the address matches the XADDR register (SLAX7 – SLAX0), the TWI generates an interrupt and goes into slave mode.

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 for transmitting 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 inputs ISDA/ISCL are ignored and the TWI Controller will not respond to any address on the bus 1: The TWI will respond to calls to its slave address – and to the general call address if the GCE bit in the ADDR register is set. **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 transmission 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.

Offset: 0x000C			Register Name: TWI_CNTR
Bit	**Read/Write**	**Default/Hex**	**Description**
4	R/WAC	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 behave as if a STOP condition has been received, but no STOP condition will be transmitted on the TWI bus. If both M_STA and M_STP bits are set, the TWI will first transmit the STOP condition (if in master mode) then transmit the START condition. 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 transmission 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.

Offset: 0x000C			Register Name: TWI_CNTR
Bit	**Read/Write**	**Default/Hex**	**Description**
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	/	/	/

Offset: 0x0010			Register Name: TWI_STAT
Bit	**Read/Write**	**Default/Hex**	**Description**
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 = F 0 = 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 = 24Mhz (APB clock input) For 400kHz full speed 2Wire, CLK_N = 1, CLK_M=2 F0 = 24M/2^1=12Mhz, F1= F0/ (10*(2+1)) = 0.4Mhz For 100kHz standard speed 2Wire, CLK_N=1, CLK_M=11 F0=48M/2^1=12Mhz, F1=F0/(10*(11+1)) = 0.1Mhz

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	/	/	/

Offset: 0x001C			Register Name: TWI_EFR
Bit	**Read/Write**	**Default/Hex**	**Description**
0:1	R/W	0x0	DBN Data Byte number follow Read Command Control 0: No Data Byte to be wrote after read command 1: Only 1-byte data to be wrote after read command 2: Bytes data can be written after read command 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), value of this bit decide 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 control 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), value of this bit decide the output level of TWI_SDA 0: Output low level 1: Output high level

Offset: 0x0020			Register Name: TWI_LCR
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	SDA_CTL_EN TWI_SDA line state control enable When this bit is set, the state of TWI_SDA is control 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

Offset: 0x0200			Register Name: TWI_DRV_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
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 did 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**

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 0~255 bytes can be sent as slave device reg address
15: 0	R/W	0x1	DATA_BYTE 1~65535 bytes can be sent/received as data

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=24MHz/2^CLK_N

Offset: 0x0210			Register Name: TWI_DRV_BUS_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
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**

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 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 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 universal asynchronous receiver transmitter (UART) provides an asynchronous serial communication with external devices, modem (data carrier equipment, DCE). It performs serial-to-parallel conversion on the data received from peripherals and transmits the converted data to the internal bus. It also performs parallel-to-serial conversion on the data that is transmitted to peripherals.

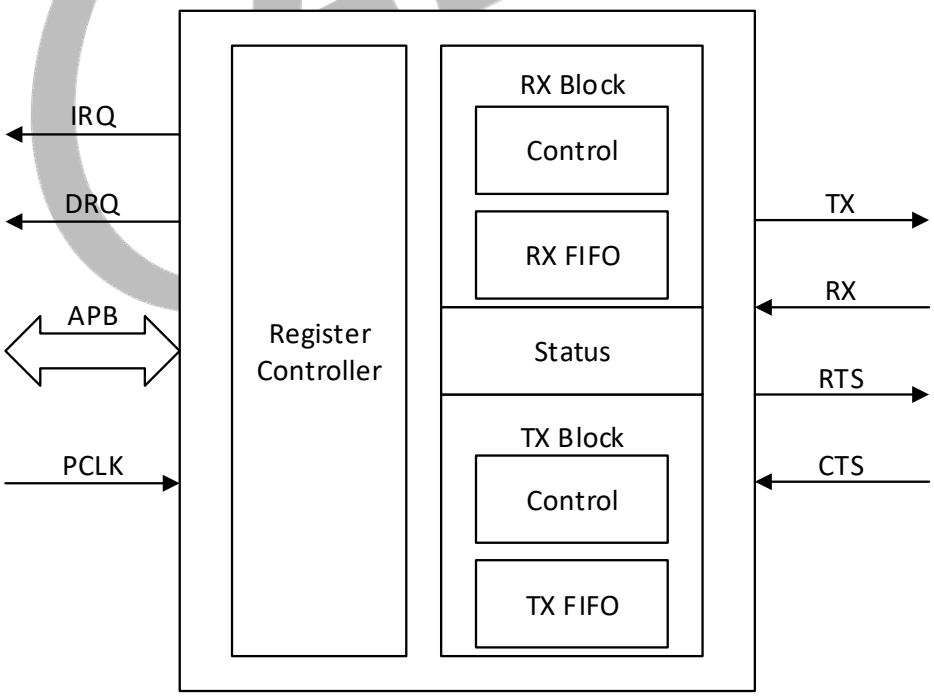
The UART controller includes the following features:

- Compatible with industry-standard 16450/16550 UARTs
- 64-Byte Transmit and receive data FIFOs
- Supports DMA controller interface
- Supports Software/Hardware Flow Control
- Supports IrDA 1.0 SIR
- Supports RS-485 mode

9.2.2 Block Diagram

The following figure shows the block diagram of UART.

Figure 9-8 UART Block Diagram



9.2.3 Functional Description

9.2.3.1 External Signals

Signal	Description	Type
UART0		
UART0_TX	Serial Data Output	O
UART0_RX	Serial Data Input	I
UART1		
UART1_TX	Serial Data Output	O
UART1_RX	Serial Data Input	I
UART1_CTS	Clear to Send	I
UART1_RTS	Request to Send	O
UART2		
UART2_TX	Serial Data Output	O
UART2_RX	Serial Data Input	I
UART2_CTS	Clear to Send	I
UART2_RTS	Request to Send	O

9.2.3.2 Clock Sources

The following table describes the clock sources for UART. For clock setting, configurations and gating information, refer to the section "CCU" and "CCU_AON".

Table 9-4 UART Clock Sources

Clock Sources	Description
APB_CLK	Clock of APB

9.2.3.3 Timing Diagram

The following figures show the UART timing diagram

Figure 9-9 UART Serial Data Timing

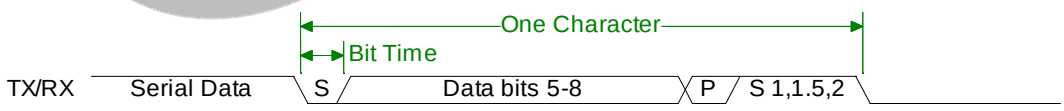
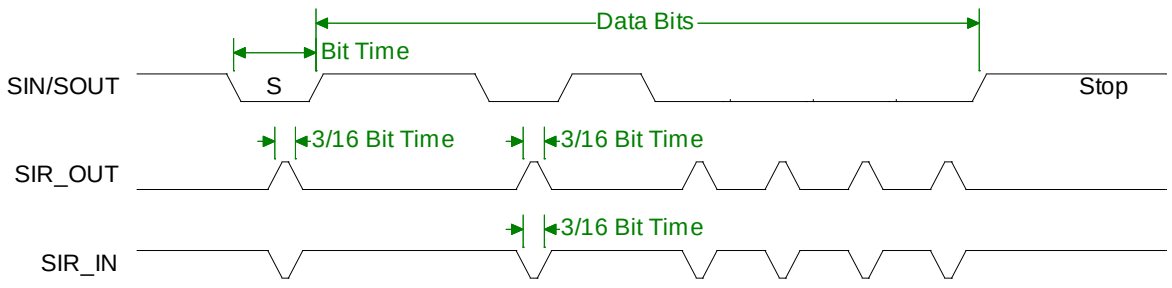


Figure 9-10 UART IrDA Timing



9.2.3.4 Operating Mode

Data Frame Format

The UART_LCR register can set the 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 the UART protocol, the low level of the TXD signal indicates the start of a data frame. When the UART transmits data, the level needs to hold high.
- **Data signal (data bit)**: The data bit width can be configured as 5-bit, 6-bit, 7-bit, and 8-bit through different applications.
- **Parity bit**: It is a 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 the TXD signal indicates the end of a data frame.

Baud and Error Rates

The baud rate is calculated as follows: Baud rate = SCLK/ (16 * divisor).

The SCLK is usually APB and can be set in the section "CCU" and "CCU_AON".

The 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 the different UART mode and the 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
14769231	1	921600	16	0.16
48000000	3	1000000	16	0

Clock source	Divisor	Baud rate	Over sampling	Error (%)
24000000	1	1500000	16	0
48000000	1	3000000	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

DLAB Definition

The DLAB control bit (UART_LCR[7]) is the access control bit of the divisor Latch register.

If DLAB is 0, then the 0x00 offset address is the RX/TX FIFO register, and the 0x04 offset address is the UART_IER register.

If DLAB is 1, then the 0x00 offset address is the UART_DLL register, and the 0x04 offset address is the UART_DLH register.

When the UART initials, the divisor needs to be set. That is, writing 1 to DLAB can access the UART_DLL and UART_DLH register, after finishing the configuration, writing 0 to DLAB can access the RX/TX FIFO register.

CHCFG_AT_BUSY Setting

The functions of the CHCFG_AT_BUSY (UART_HALT[1]) and CHANGE_UPDATE (UART_HALT[2]) are as follows.

CHCFG_AT_BUSY: Enable the bit, the software can also set the UART controller when UART is busy, such as the UART_LCR, UART_DLH, UART_DLL register.

CHANGE_UPDATE: If CHCFG_AT_BUSY is enabled, and CHANGE_UPDATE is written to 1, the configuration of the UART controller can be updated. After completing the update, the bit is cleared to 0 automatically.

Setting divisor performs the following steps:

Step 1 Write 1 to CHCFG_AT_BUSY to enable “configure at busy”.

Step 2 Write 1 to DLAB (UART_LCR[7]) and set the UART_DLH and UART_DLL registers.

Step 3 Write 1 to CHANGE_UPDATE to update the configuration. The bit is cleared to 0 automatically after completing the update.

UART Busy Flag

The UART_USR[0] is a busy flag of the UART controller.

When the TX transmits data, or the RX receives data, or the TX FIFO is not empty, or the RX FIFO is not empty, then the busy flag bit can be set to 1 by hardware, which indicates the UART controller is busy.

UART Baudrate Detection

With this function, the receiving device does not need to set the baud rate in advance during the serial communication, as it can change its baud rate in real time based on the transmitting device’s.

The receiving device determines the transmission rate based on a pre-selected, received character, which should consist of alternating 0bit and 1bit (such as 0x55).

Set UART_BDC[1] as 1 to start UART baudrate detection, and set UART_BDC[1] as 0 to suspend UART baudrate detection.

The detection circuit initiates detection at the falling edge of UART RX, and suspends detection at the rising edge of UART RX. In addition, it calculates the PCLK periods of adjacent rising/falling edges based on the detected duration of 1 bit. The PCLK periods are written to PCLK_NUM by hardware. Thus, the baud ratio can be set by the equation: Baud ratio= pclk_spc/period.

UART baudrate detection includes two modes: Normal mode and compare mode. The two modes can be selected by writing to the UART_BDC[0].

In the normal mode, when the detection circuit is enabled, PCLK periods will be latched and written to PCLK_NUM in the duration of each bit. The baud rate will be calculated only when this mode is at a stable PCLK period and suitable for receiving the data like 0x55.

In the compare mode, when the detection circuit is enabled, PCLK periods will be latched. If PCLK periods are smaller than the latched PCLK_NUM, they will be written to PCLK_NUM. This mode is to obtain the minimum PCLK periods and suits receiving random characters (namely, the correct baud rate will be detected only if a single bit data varies.)

9.2.4 Programming Guidelines

The following figures show the UART DRQ Flow Chart and UART IRQ Flow Chart.

Figure 9-11 UART DRQ Flow Chart

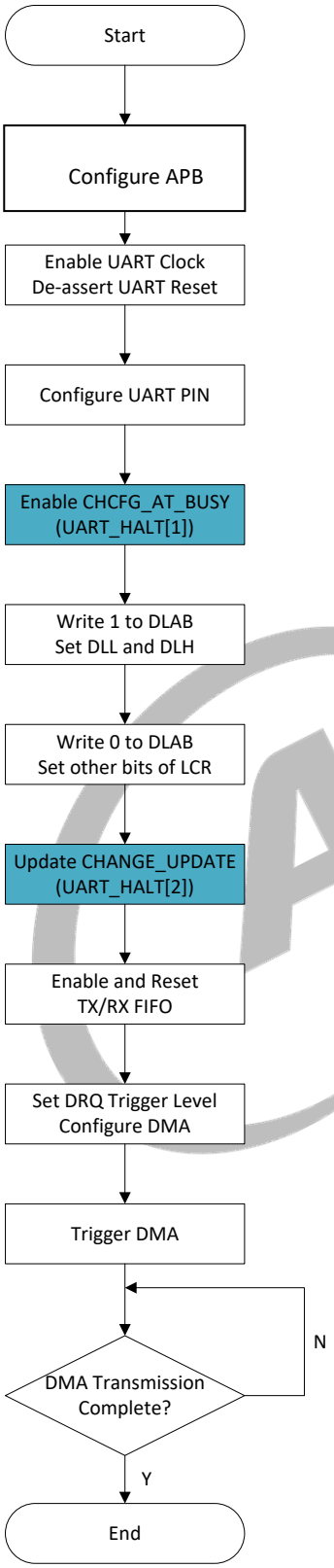
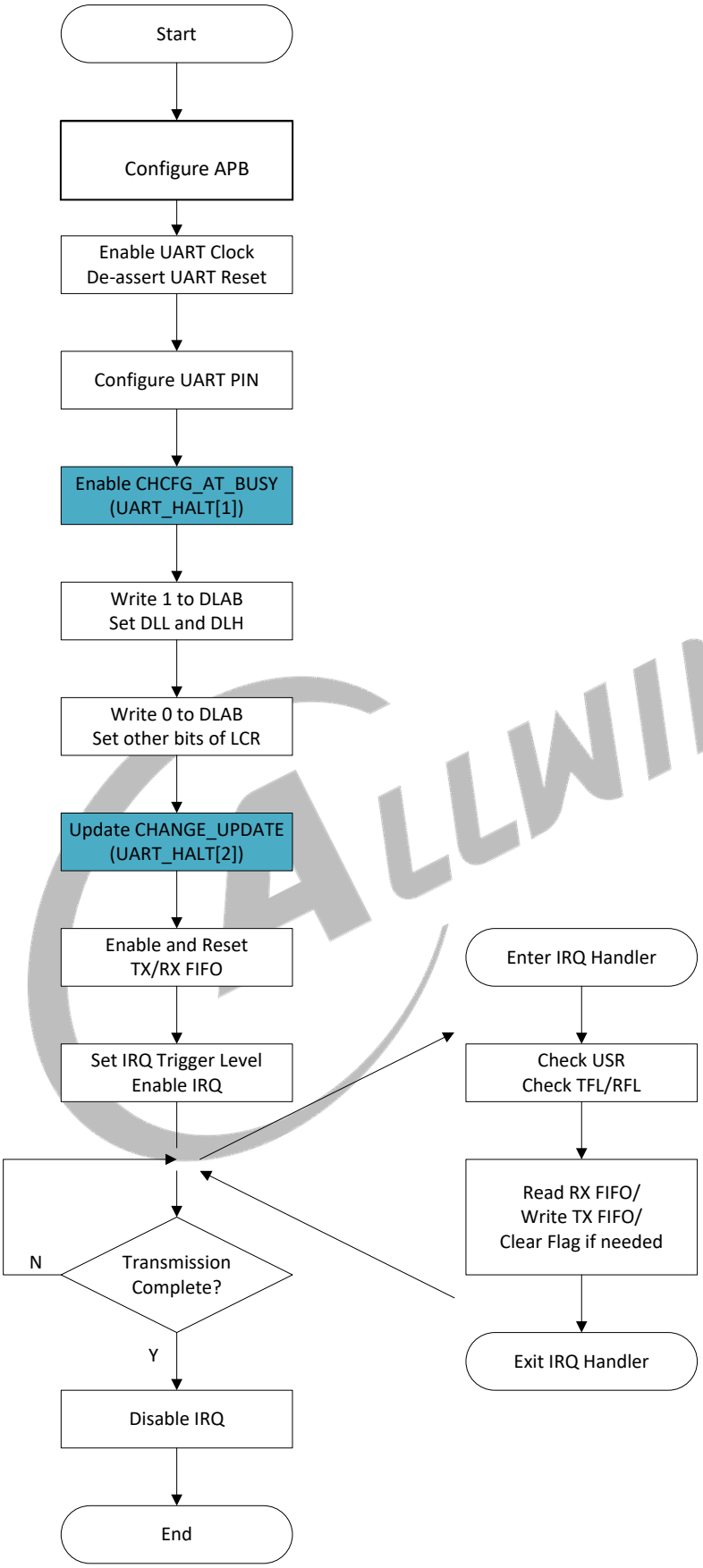


Figure 9-12 UART IRQ Flow Chart



9.2.5 Register List

Module Name	Base Address
UART0	0x40047000
UART1	0x40047400
UART2	0x40047800

Register Name	Offset Address	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_HALT	0x00A4	UART Halt TX Register
TX_DLY	0x00CC	UART TX Delay Register
UART_BDCR	0x00D4	UART Baudrate Detection Control Register
UART_BDCLR	0x00D8	UART Baudrate Detection Counter Low Register
UART_BDCHR	0x00DC	UART Baudrate Detection Counter High 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	/	/	/

Offset: 0x0000			Register Name: UART_RBR
Bit	**Read/Write**	**Default/Hex**	**Description**
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 is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an overrun error occurs.

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 to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, 16 characters of data may be written to the THR before the FIFO is full. Any attempt to write data when the FIFO is full results in the write data being lost.

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	/	/	/

Offset: 0x0000			Register Name: UART_DLL
Bit	**Read/Write**	**Default/Hex**	**Description**
7: 0	R/W	0x0	DLL Divisor Latch Low Lower 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occurs. 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-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the 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	/	/	/

Offset: 0x0004			Register Name: UART_IER
Bit	**Read/Write**	**Default/Hex**	**Description**
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 interrupts. 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 Bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt.

9.2.6.7 UART Interrupt Identity Priority

Interrupt ID	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset
0001	-	None	None	-
0110	Highest	Receiver line status	Overrun/parity/framing errors or break interrupt	Reading the line status register
0011	Second	RS485 Interrupt	In RS485 mode, receive 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)

Interrupt ID	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset
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

9.2.6.8 0x0008 UART FIFO Control Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: UART_FCR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/

Offset: 0x0008			Register Name: UART_FCR
Bit	**Read/Write**	**Default/Hex**	**Description**
7:6	W	0x0	RT RCVR Trigger This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode, it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. 00: 1 character in the FIFO 01: FIFO ¼ full 10: FIFO ½ full 11: FIFO-2 less than full
5:4	W	0x0	TFT TX Empty Trigger Writes have no effect when THRE_MODE_USER = Disabled. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. 00: FIFO empty 01: 2 characters in the FIFO 10: FIFO ¼ full 11: FIFO ½ full
3	W	0x0	DMAM DMA Mode 0: Mode 0 In this mode, if PTE is high and TX FIFO is enable, the TX DMA request will send when TFL is less than or equals 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 equals 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 equals 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 empty. If RFL is equals to or more than FIFO Trigger Level, the RX DRQ will be set 1,in otherwise, it will be set 0.

Offset: 0x0008			Register Name: UART_FCR
Bit	**Read/Write**	**Default/Hex**	**Description**
2	W	0x0	XFIFOR XMIT FIFO Reset This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request. It is 'self-clearing'. It is not necessary to clear this bit.
1	W	0x0	RFIFOR RCVR FIFO Reset This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request. It is 'self-clearing'. It is not necessary to clear this bit.
0	W	0x0	FIFOE Enable FIFOs This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset.

9.2.6.9 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)

Offset: 0x000C			Register Name: UART_LCR
Bit	**Read/Write**	**Default/Hex**	**Description**
6	R/W	0x0	BC Break Control Bit This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE = Enabled and active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.
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 used to reverse the LCR[4]. 00: Odd Parity 01: Even Parity 1X: Reverse LCR[4] In RS485 mode, it is the 9th bit--address bit. 11:9th bit = 1, indicates that this is an address byte 10:9th bit = 0, indicates that this is a data byte **Note**: *When use this function, PEN(LCR[3]) must be 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

Offset: 0x000C			Register Name: UART_LCR
Bit	**Read/Write**	**Default/Hex**	**Description**
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 set to zero, one stop bit is transmitted in the serial data. If set to one and the data bits are set to 5 (LCR[1: 0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected the receiver checks only the first stop bit. 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

9.2.6.10 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: Reversed
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

Offset: 0x0010			Register Name: UART_MCR
Bit	**Read/Write**	**Default/Hex**	**Description**
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] set to zero), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE == Enabled AND active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.
3:2	/	/	/
1	R/W	0x0	RTS Request to Send This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. 0: rts_n de-asserted (logic 1) 1: rts_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input.

Offset: 0x0010			Register Name: UART_MCR
Bit	**Read/Write**	**Default/Hex**	**Description**
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] set to one), the dtr_n output is held inactive high while the value of this location is internally looped back to an input.

9.2.6.11 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 that 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.

Offset: 0x0014			Register Name: UART_LSR
Bit	**Read/Write**	**Default/Hex**	**Description**
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, sin, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits. If in infrared mode (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	R	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.

Offset: 0x0014			Register Name: UART_LSR
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R	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	R	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.

9.2.6.12 0x0018 UART Modem Status Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: UART_MSR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/

Offset: 0x0018			Register Name: UART_MSR
Bit	**Read/Write**	**Default/Hex**	**Description**
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] set to one), 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).

Offset: 0x0018			Register Name: UART_MSR
Bit	**Read/Write**	**Default/Hex**	**Description**
3	R	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	R	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	R	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.*

Offset: 0x0018			Register Name: UART_MSR
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R	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.*

9.2.6.13 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.

9.2.6.14 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.

Offset: 0x007C			Register Name: UART_USR
Bit	**Read/Write**	**Default/Hex**	**Description**
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 0: Idle or inactive 1: Busy

9.2.6.15 0x0080 UART Transmit FIFO Level Register (Default Value: 0x0000_0000)

Offset: 0x0080			Register Name: UART_TFL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6: 0	R	0x0	TFL Transmit FIFO Level This is indicates the number of data entries in the transmit FIFO.

9.2.6.16 0x0084 UART Receive FIFO Level Register (Default Value: 0x0000_0000)

Offset: 0x0084			Register Name: UART_RFL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6: 0	R	0x0	RFL Receive FIFO Level This is indicates the number of data entries in the receive FIFO.

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**

Offset: 0x00A4			Register Name: UART_HALT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	
7	R/W	0x0	DMA_PTE_RX RX_DRQ Transmission In the DMA1 mode, DRQ will be sent in the case of receive timeout or when RFL is greater than or equal to TRIG. In the DMA0 mode, REQ will be sent in the case of DMA_PTE_RX=1 and FIFO enable and when RFL is greater than or equal to TRIG. Otherwise, DRQ will be sent if there is valid data for RX.
6	R/W	0x0	PTE TX_REQ Transmission In the DMA1 mode (FIFO enable, FCR[3]=1), if PTE signal is at high level, the DMA request will be sent when TFL is less than or equal to TRIG; if PTE signal is at low level, the DMA request will be sent when FIFO is null and switched to 0 when FIFO is full. In the DMA0 mode (FIFO disable, FCR[3]=0), if PTE signal is at high level and FIFO is enabled, the DMA request will be sent when TFL is less than or equal to TRIG; DRQ will be switched to 0 when TFL is not less than or equal to TRIG. If PTE signal is at high level and FIFO is disabled, DMA request will be sent when THRE is null; DRQ will be switched to 0 when THRE is not null. DMA request will be sent when PTE signal is at low level and FIFO is null.
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 using HALT[1] to change the baud rate or LCR configuration, write 1 to update the configuration and waiting this bit self-clear to 0 to finish update process. Write 0 to this bit has no effect.
1	R/W	0x0	CHCFG_AT_BUSY This is an enable bit for the user to change LCR register configuration (except for the DLAB bit) and baud rate register (DLH and DLL) when the UART is busy (USB[0] is 1).

Offset: 0x00A4			Register Name: UART_HALT
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	HALT_TX This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 0: Halt TX disabled 1: Halt TX enabled **Note**: *If FIFOs are not enabled, the setting of the halt TX register has no effect on operation.*

9.2.6.18 0x00CC UART TX Delay (Default Value: 0x0000_0000)

Offset: 0x00CC			Register Name: UART_TXDLY
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 Tclk. It is use to control the space between two bytes in TX.

9.2.6.19 0x00D4 UART Baudrate Detection Control Register (Default Value: 0x0000_0000)

Offset: 0x00D4			Register Name: UART_BDC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1	R/W	0x0	Auto Baudrate Detect Enable 1: Enable 0: Disable **Note**: Set *this bit to 1 to start the UART Baudrate Detection. The detection circuit will detect the start bit and count the cycles of PCLK in one-bit duration. The number of the cycles will be written in the PCLK_NUM field. Set this bit to 0 to stop the UART Baudrate Detection.*

Offset: 0x00D4			Register Name: UART_BDC
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	Mode 1: Normal mode 0: Compare mode **Note:** *In normal mode, when the detection circuit is enabled, the number of the PCLK cycles will be latched in each bit duration and be written into the PCLK_NUM field.* *In compare mode, the number of the PCLK cycles will be latched in each one bit duration, but only the value which equals to the value latched in last time will be written into the PCLK_NUM field.*

9.2.6.20 0x00D8 UART Baudrate Detection Counter Low Register (Default Value: 0x0000_0000)

Offset: 0x00D8			Register Name: UART_BDCL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7: 0	R/W	0x0	PCLK_NUM_LOW The low 8 bits of the number of the cycles counted in one bit duration.

9.2.6.21 0x00DC UART Baudrate Detection Counter High Register (Default Value: 0x0000_0000)

Offset: 0x00DC			Register Name: UART_ABCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7: 0	R/W	0x0	PCLK_NUM_HIGH The high 8 bits of the number of the cycles counted in one bit duration.

9.3 LP_UART

9.3.1 Overview

The low power universal asynchronous receiver transmitter (LP_UART) supports 5-bit received continuous data comparison. (2 stop bit should be set) If the comparison succeeds, then LPUART will generates an interrupt, and CPU will be waked up.

9.3.2 Register List

Module Name	Base Address
LPUART0	0x4004A800
LPUART1	0x4004AC00

Register Name	Offset Address	Description
GP_SR_CON	0x0000	LPUART Control Register
LPUART_BAUD_CONFIG_REG	0x0004	LPUART Baudrate Configure Register
LPUART_RX_CONFIG_REG	0x0010	LPUART RX Configure Register
LPUART_RX_DATA_REG	0x0014	LPUART RX Data Register
LPUART_INTR_EN_REG	0x0020	LPUART Interrupt Enable Register
LPUART_INTR_STATUS_REG	0x0024	LPUART Interrupt Status Register
LPUART_INTR_CLR_REG	0x0028	LPUART Interrupt Clear Register
LPUART_RX_CMP_REG1	0x002C	LPUART RX Compare Register1
LPUART_RX_CMP_REG2	0x0030	LPUART RX Compare Register2

9.3.3 Register Description

9.3.3.1 0x0000 LPUART Control Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: GP_SR_CON
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8	W	0x0	cfg_soft_rst_p Soft reset Set to 1 by software, then it will be auto cleared by hardware.
7: 0	/	/	/

9.3.3.2 0x0004 LPUART Baudrate Configure Register (Default Value: 0x0302_0006)

Offset: 0x0004			Register Name: LPUART_BAUD_CONFIG_REG
Bit	Read/Write	Default/Hex	Description

Offset: 0x0004			Register Name: LPUART_BAUD_CONFIG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0x3	cfg_denominator[7: 0] Configure divider. Take an example of 32K clock and 9600 baudrate. With the double-edge design, it will be set as 3 as the divider is 3 if 64K is divided by 9.6K.
23:16	R/W	0x2	cfg_remainder[7: 0] Configure remainder. Take an example of 32K clock and 9600 baudrate. With the double-edge design, it will be set as 2 as the remainder is 2 if 64K is divided by 9.6K.
15: 0	R/W	0x6	cfg_quotient[15: 0] Configure quotient. Take an example of 32K clock and 9600 baudrate. With the double-edge design, it will be set as 6 as the quotient is 6 if 64K is divided by 9.6K.

9.3.3.3 0x0010 LPUART RX Configure Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: LPUART_RX_CONFIG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:13	/	/	/
12	R/W	0x0	cfg_rx_msb_first 0: The serial data received first is used as LSB. 1: The serial data received first is used as MSB.
11	/	/	/
10:8	R/W	0x0	cfg_rx_data_width[2: 0] Select the length of RX data bit 000: 4bit data 001: 5bit data 010: 6bit data 011: 7bit data 100: 8bit data 101: 9bit data others: 8bit data **Note**: The *bit width of RXFIFO is 9bit. If the bit width of RX data is smaller than 9bit, the high bit will be supplemented with 0.*
7	/	/	/

Offset: 0x0010			Register Name: LPUART_RX_CONFIG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
6:4	R/W	0x0	cfg_rx_parity_mode[2: 0] Select parity bit of RX 000: No parity bit. 001: Even. During the whole process of bit transmission, the number of 1 is even, which can be adjusted by parity bit. 010: Odd. During the whole process of bit transmission, the number of 1 is odd, which can be adjusted by parity bit. 011: Space. The parity bit is always 0. 100: Mark. The parity bit is always 1. 101/110/111: No parity bit.
3	/	/	/
2	R	0x0	rx_partiy_err_rpt Report whether the latest RX parity result is correct. 0: The parity result is correct. 1: The parity result is wrong.
1	/	/	/
0	R/W	0x0	cfg_rx_en cfg_rx_en can be used by hardware to enable the clock of RX module. 1: Enable RX module 0: Disable RX module

9.3.3.4 0x0014 LPUART RX Data Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: LPUART_RX_DATA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8: 0	R	0x0	rx_data[8: 0] Reading rx_data triggers reading new data from RXFIFO.

9.3.3.5 0x0020 LPUART Interrupt Enable Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: LPUART_INTR_EN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9	R/W	0x0	cfg_rx_data_int_en RX Receive Data Interrupt Enable Once RX receives data, interrupts will be enabled/disabled. 1: Enable. 0: Disable.
8	/	/	/

Offset: 0x0020			Register Name: LPUART_INTR_EN_REG
Bit	Read/Write	Default/Hex	Description
7	R/W	0x0	cfg_rx_data_cmp_int_en Interrupts will be enabled/disabled after RX data comparison. 1: Enable. 0: Disable.
6: 0	/	/	/

9.3.3.6 0x0024 LPUART Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: LPUART_INTR_STATUS_REG
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9	R	0x0	rx_data_int_flag The interrupt will be flagged once RX receives data.
8	/	/	/
7	R	0x0	rx_data_cmp_int_flag The interrupt will be flagged after successful RX data comparison.
6: 0	/	/	/

9.3.3.7 0x0028 LPUART Interrupt Clear Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: LPUART_INTR_CLR_REG
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9	W	0x0	cfg_rx_data_int_clr The rx_data_int_flag is cleared by writing 1.
8	/	/	/
7	W	0x0	cfg_rx_data_cmp_int_clr The rx_data_cmp_int_flag is cleared by writing 1.
6: 0	/	/	/

9.3.3.8 0x002C LPUART RX Compare Register1 (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: LPUART_RX_CMP_REG1
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:21	R/W	0x0	cfg_rx_cmp2[8: 0] The third register compared with RX Data
20:12	R/W	0x0	cfg_rx_cmp1[8: 0] The second register compared with RX Data
11:3	R/W	0x0	cfg_rx_cmp0[8: 0] The first register compared with RX Data

Offset: 0x002C			Register Name: LPUART_RX_CMP_REG1
Bit	**Read/Write**	**Default/Hex**	**Description**
2: 0	R/W	0x0	cfg_rx_cmp_num[2: 0] Configure the number for comparison. 000: Compare 1 number 001: Compare 1 number 010: Compare 2 numbers 011: Compare 3 numbers 100: Compare 4 numbers 101: Compare 5 numbers 110: Compare 1 number 111: Compare 1 number

9.3.3.9 0x0030 LPUART RX Compare Register2 (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: LPUART_RX_CMP_REG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20:12	R/W	0x0	cfg_rx_cmp4[8: 0] The fifth register compared with RX Data
11:3	R/W	0x0	cfg_rx_cmp3[8: 0] The fourth register compared with RX Data
2: 0	/	/	/

9.4 SPI

9.4.1 Overview

The Serial Peripheral Interface (SPI) is a full-duplex, synchronous, four-wire serial communication interface between a CPU and SPI-compliant external devices. The SPI controller contains a 64 x 8 bits receiver buffer (RXFIFO) and a 64 x 8 bits transmit buffer (TXFIFO). It can work in master mode and slave mode.

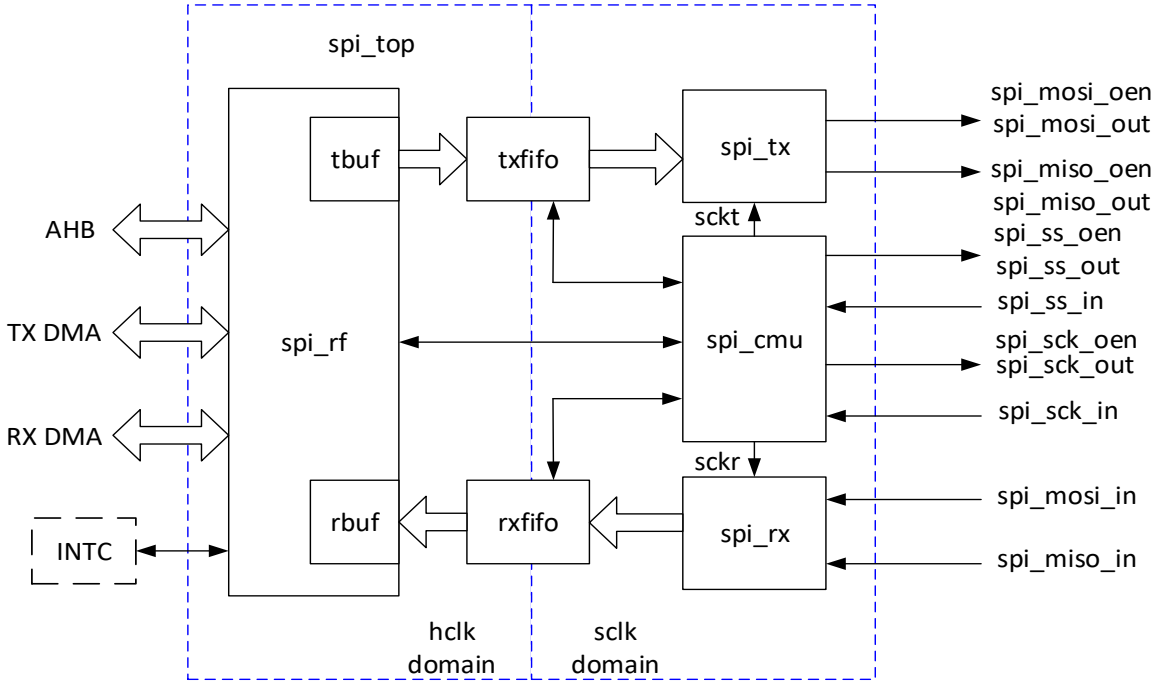
The SPI has the following features:

- Full-duplex synchronous serial interface
- Master/slave configurable
- 8-bit wide by 64-entry FIFO for both transmitting and receiving data
- Polarity and phase of the Chip Select (SPI_SS) and SPI Clock (SPI_SCLK) are configurable. Four chip selects support multiple peripherals.
- Supports interrupts and DMA
- Supports mode0, mode1, mode2, and mode3
- Supports 3-wire/4-wire SPI
- Supports programmable serial data frame length: 0 bit to 32 bits
- Supports standard SPI, dual-output/dual-input SPI, dual I/O SPI, quad-output/quad-input SPI
- Supports maximum IO rate of the mass production: 96 MHz
- Supports 5 clock sources, Interrupt or DMA

9.4.2 Block Diagram

The following figure shows a block diagram of the SPI.

Figure 9-13 SPI Block Diagram



SPI contains the following sub-blocks:

Table 9-8 SPI Sub-blocks

Sub-block	Description
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 8 bits, then the data is written into the RXFIFO.
spi_rbuf	The block is used to convert the RXFIFO data into the reading data length of AHB.
txfifo, rxfifo	The data transmitted from the SPI to the external serial device is written into the TXFIFO; the 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 transmission, 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.

9.4.3 Functional Description

9.4.3.1 External Signals

The following table describes the external signals of SPI. The MOSI and MISO are bidirectional I/O. when SPI acts as a master device, the CLK and CS are the output pin; when SPI acts as a slave device, the CLK and CS are the input pin. When using SPI, the corresponding PADs are selected as SPI function via the section "GPIO".

Table 9-9 SPI External Signals

Signal	Description	Type
SPI0_CS	SPI0 chip select signal, low active When the device is not selected, data will not be accepted via the SI pin, and the SO pin will stop transmission.	I/O
SPI0_CLK	SPI0 clock signal This pin is used to provide a clock to the device and is used to control the flow of data to and from the device.	I/O
SPI0_MOSI	SPI0 master data output, slave data input	I/O
SPI0_MISO	SPI0 master data input, slave data output	I/O
SPI0_WP	Write protection and low active It also can be used for serial data input and output for SPI Quad Input or Quad Output mode.	I/O
SPI0_HOLD	When the device is selected and a serial sequence is underway, the HOLD pin can be used to temporarily pause the serial communication with the master device without deselecting or resetting the serial sequence. While the HOLD pin is asserted, the SO pin is at high impedance, and all transitions on the SCK pin and data on the SI pin are ignored. It also can be used for serial data input and output for SPI Quad Input or Quad Output mode.	I/O

9.4.3.2 Clock Sources

The SPI controller gets 5 different clock sources, users can select one of them to make SPI clock source. The following table describes the clock sources for SPI. For clock setting, configurations and gating information, refer to the section "CCU" and "CCU_AON".

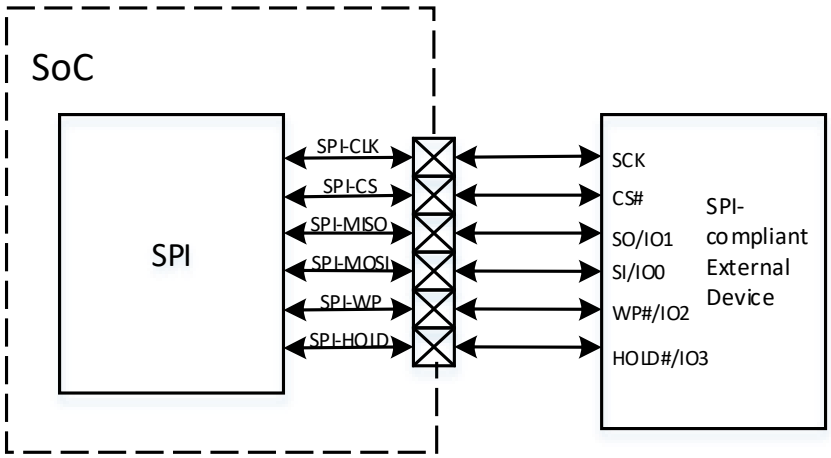
Table 9-10 SPI Clock Sources

Clock Sources	Description
HFCLK	High frequency crystal clock for system (optional frequency: 24M/24.576M/26M/32M/40M)
DEVCLK	General-purpose high frequency crystal clock for system interfaces (frequency range: 0~392M)

9.4.3.3 Typical Application

The following figure shows the application block diagram when the SPI master device is connected to a slave device.

Figure 9-14 SPI Application Block Diagram



9.4.3.4 SPI Transmission Format

The SPI supports 4 different formats for data transmission. The software can select one of the four modes in which the SPI works by setting the bit1 (Polarity) and bit0 (Phase) of SPI_TCR. The SPI controller master uses the SPI_SCLK signal to transfer data with shift register. Data is clocked using any one of four programmable clock phase and polarity combinations.

The CPOL (SPI_TCR[1]) defines the polarity of the clock signal (SPI_SCLK). The SPI_SCLK is a high level when CPOL is '1', and it is a low level when CPOL is '0'. The CPHA (SPI_TCR[0]) decides whether the leading edge of SPI_SCLK is used to setup or sample data. The leading edge is used to setup data when CPHA is '1', and sample data when CPHA is '0'. The following table lists the four modes.

Table 9-11 SPI Transmission Format

Mode	Polarity (CPOL)	Phase (CPHA)	Leading Edge	Trailing Edge
Mode0	0	0	Sample on the rising edge	Setup on the falling edge
Mode1	0	1	Setup on the rising edge	Sample on the falling edge
Mode2	1	0	Sample on the falling edge	Setup on the rising edge
Mode3	1	1	Setup on the falling edge	Sample on the rising edge

The following figures describe four waveforms for SPI_SCLK.

Figure 9-15 SPI Phase 0 Timing Diagram

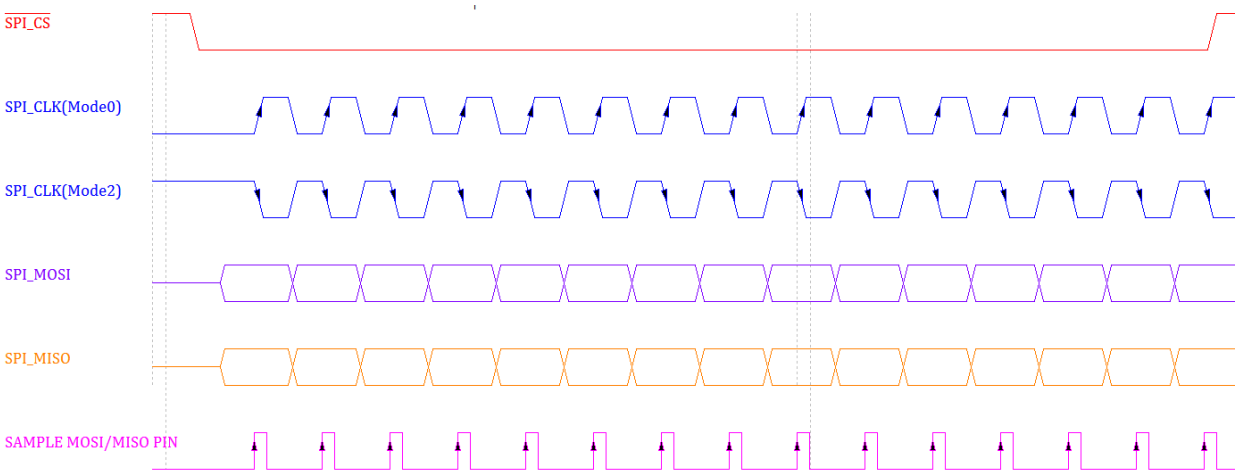
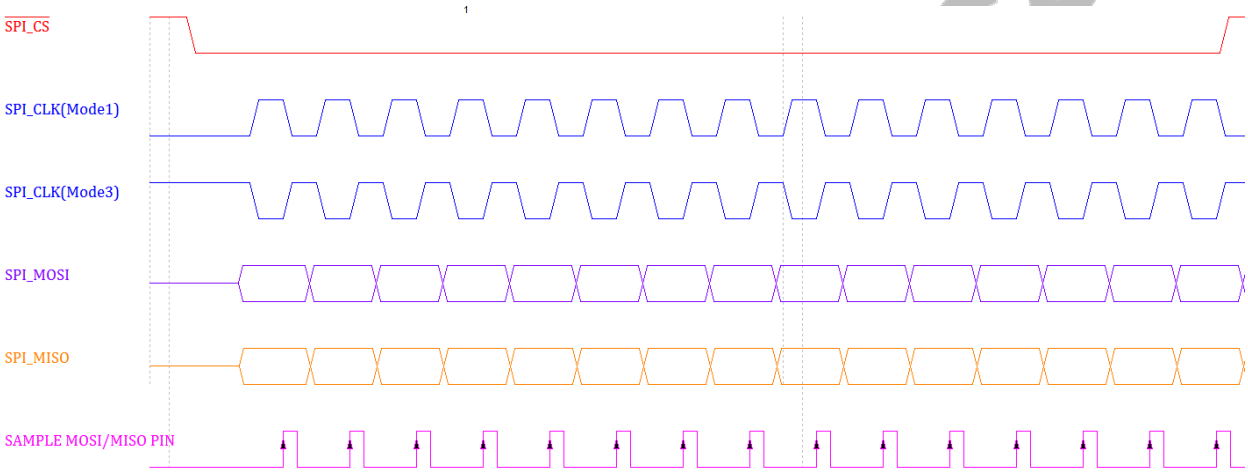


Figure 9-16 SPI Phase 1 Timing Diagram



9.4.3.5 SPI Master and Slave Mode

The SPI controller can be configured to a master or slave device. The master mode is selected by setting the MODE bit (SPI_GCR[1]); the slave mode is selected by clearing the MODE bit.

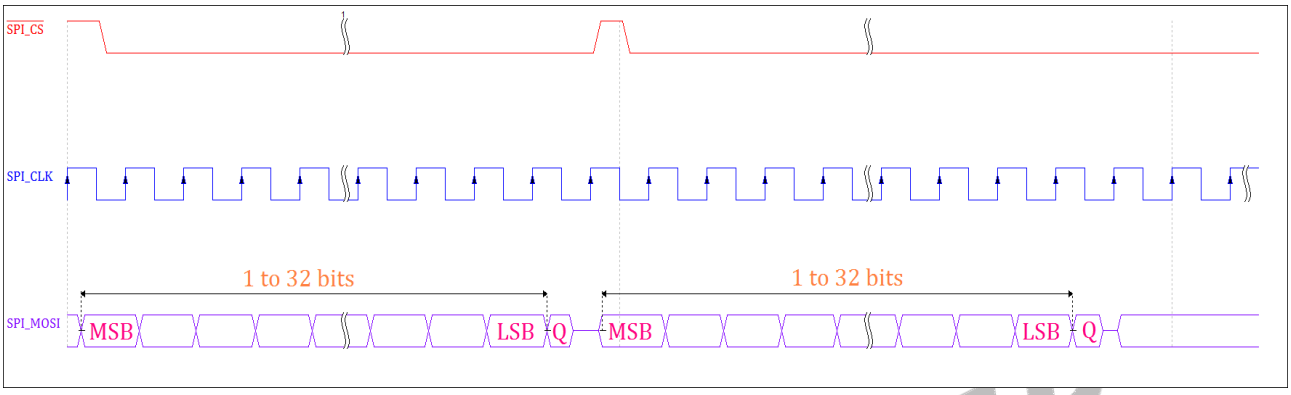
In master mode, the SPI_CLK is generated and transmitted to the external device, and the data from the TX FIFO is transmitted on the MOSI pin, the data from the slave device is received on the MISO pin and sent to RX FIFO. The Chip Select (SPI_SS) is an active low signal, and it must be low before the data are transmitted or received. The SPI_SS can be selected as the auto control mode or software manual control mode. When using the auto control, the SS_OWNER (SPI_TCR [6]) must be cleared (default value is 0); when using the manual control, the SS_OWNER must be set. And the level of SPI_SS is controlled by SS_LEVEL (SPI_TCR [7]).

In slave mode, after the software selects the MODE bit (SPI_GCR[1]) to '0', it waits for master to initiate a transmission. When the master asserts SPI_SS, and SPI_CLK is transmitted to the slave device, the slave data is transmitted from TX FIFO on the MISO pin and the data from the MOSI pin is received in RX FIFO.

9.4.3.6 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 (SPI_BATC[1: 0]) is 0x2. And in the 3-wire mode, the input data and the output data use the same single data line. The following figure describes the 3-wire mode.

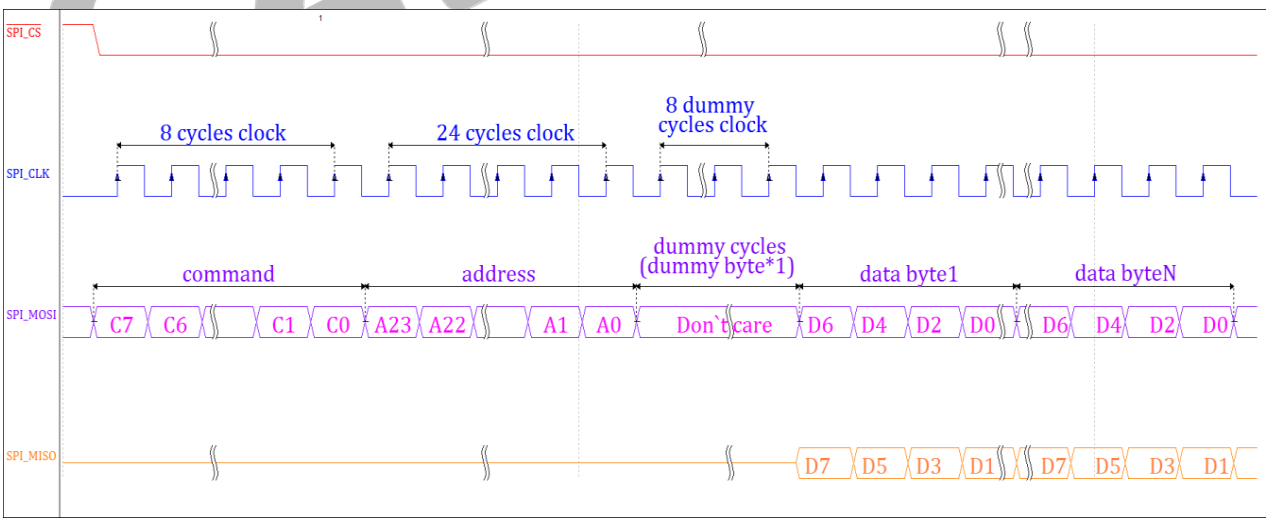
Figure 9-17 SPI 3-Wire Mode



9.4.3.7 SPI Dual Mode

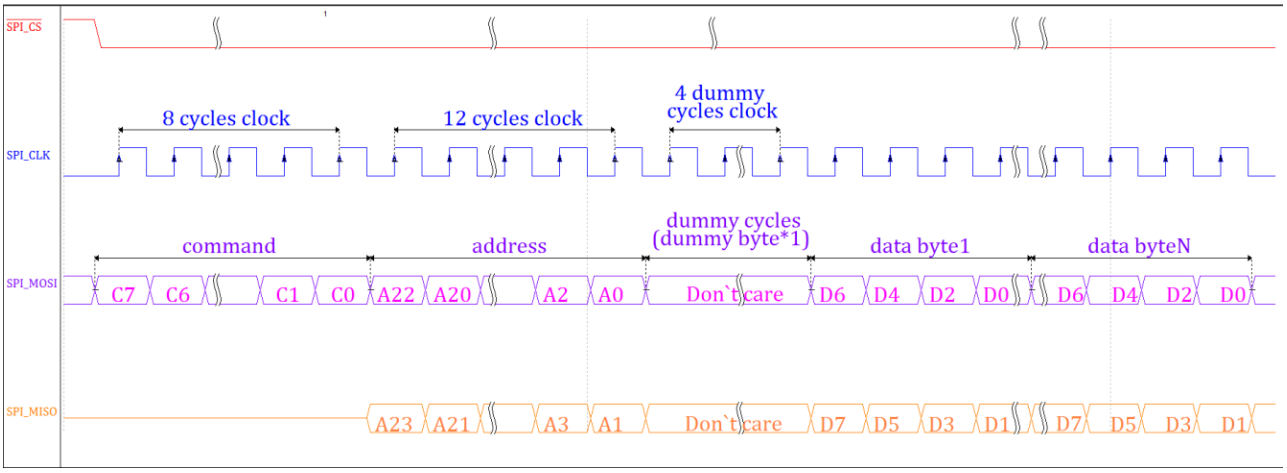
The dual read mode (SPI x2) is selected when the DRM is set in SPI_BCC[28]. Using the dual mode allows data to be transferred to or from the device at double the rate of standard single mode, the data can be read at fast speed using two data bits (MOSI and MISO) at a time. The following two figures describe the dual-input/dual-output SPI and the dual I/O SPI.

Figure 9-18 SPI Dual-Input/Dual-Output Mode



In the dual-input/dual-output SPI mode, the command, address, and the dummy bytes output in a unit of a single bit in serial mode through the SPI_MOSI line, only the data bytes are output (write) and input (read) in a unit of dual bits through the SPI_MOSI and SPI_MISO.

Figure 9-19 SPI Dual I/O Mode

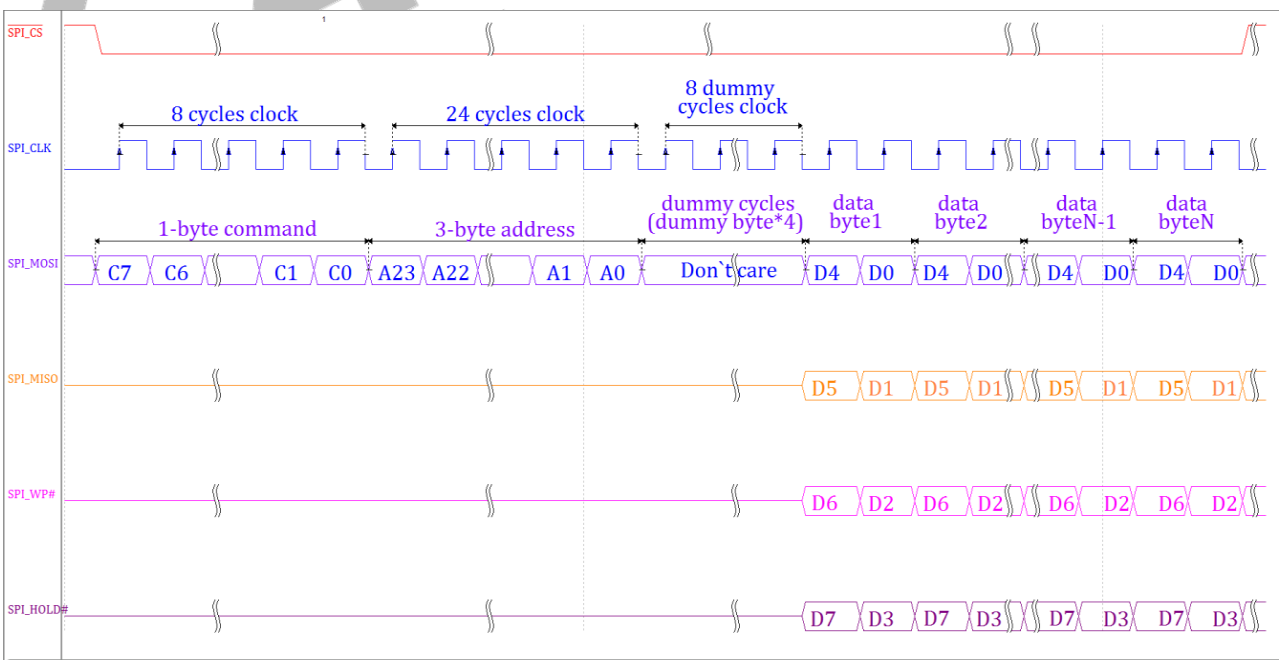


In the dual I/O SPI mode, only the command bytes output in a unit of a single bit in serial mode through the SPI_MOSI line. The address bytes and the dummy bytes output in a unit of dual bits through the SPI_MOSI and SPI_MISO. And the data bytes output (write) and input (read) in a unit of dual bits through the SPI_MOSI and SPI_MISO.

9.4.3.8 SPI Quad Mode

The quad read mode (SPI x4) is selected when the Quad_EN is set in SPI_BCC[29]. Using the quad mode allows data to be transferred to or from the device at 4 times the rate of standard single mode, the 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 9-20 SPI Quad-Input/Quad-Output Mode



In the quad-input/quad-output SPI mode, the command, address, and the dummy bytes output in a unit of a single bit in serial mode through SPI_MOSI line. Only the data bytes output (write) and input (read) in a unit of quad bits through the SPI_MOSI, SPI_MISO, SPI_WP#, and SPI_HOLD#.

9.4.3.9 Transmission/Reception Bursts in Master Mode

In SPI master mode, the transmission and reception bursts (byte in unit) are configured before the SPI transfers the serial data between the processor and external device. The transmission bursts are written in MWTC (bit[23: 0]) of the SPI_MTC. The transmission bursts in single mode before automatically sending dummy bursts are written in STC (bit[23: 0]) of the SPI_BCC. For dummy data, the SPI controller can automatically send before receiving by writing DBC (bit[27:24]) in the. If users do not use the SPI controller to send dummy data automatically, then the dummy bursts are used as the transmission counters to write together in MWTC (bit[23: 0]) of the SPI_MTC. In master mode, the total burst numbers are written in MBC (bit[23: 0]) of the SPI_MBC. When all transmission and reception bursts are transferred, the SPI controller will send a completed interrupt, at the same time, the SPI controller will clear DBC, MWTC, and MBC.

9.4.3.10 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 the master mode. The SPI clock is selected from different clock sources, the SPI must configure different work mode. There are three work modes: Normal sample mode, delay half-cycle sample mode, delay one-cycle sample mode. The Delay half-cycle sample mode is the default mode of the SPI controller. When the SPI runs at 40 MHz or below 40 MHz, the SPI can work at normal sample mode or delay half-cycle sample mode. When the SPI runs over 80 MHz, setting the SDC bit in the SPI_TCR 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 between master and slave. The following tables show the different configurations of the SPI sample mode.

Table 9-12 SPI Old 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	>=80 MHz

CAUTION

The remaining spectrum is not recommended. Because when the output delay of SPI flash (refer to the datasheet of the manufactures for the specific delay time) is the same with the half-cycle time of SPI working clock, the variable edge of the output data for the device bumps into the clock sampling edge of the controller, so setting 1 cycle of sampling delay would cause stability problem.

Table 9-13 SPI New Sample Mode

SPI Sample Mode	SDM (bit13)	SDC (bit11)	SDC1 (bit15)

SPI Sample Mode	SDM (bit13)	SDC (bit11)	SDC1 (bit15)
normal sample	1	0	0
delay half cycle sample	0	0	0
delay one cycle sample	0	1	0
delay 1.5 cycle sample	1	1	0
delay 2 cycle sample	1	0	1
delay 2.5 cycle sample	0	0	1
delay 3 cycle sample	0	1	1

9.4.3.11 SPI Error Conditions

If any error conditions occur, the hardware will set the corresponding status bits in the SPI_ISR and stop the transfer. For the SPI controller, the following error scenarios can happen.

1.TX_FIFO Underrun

The TX_FIFO underrun happens when the CPU/DMA reads data 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_ISR. The SPI controller will generate an interrupt if interrupts are enabled. The software has to clear the error bit and the TF_UDF bit. To start a new transaction, the software has to reset the FIFO by writing to the SRST (soft reset) bit in the SPI_GCR.

2.TX_FIFO Overflow

The TX_FIFO overflow happens when the CPU/DMA writes data 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_ISR. The SPI controller will generate an interrupt if interrupts are enabled. The software has to clear the error bit and the TF_OVF bit. To start a new transaction, the software has to reset the FIFO by writing to the SRST (soft reset) bit in the SPI_GCR.

3.RX_FIFO Underrun

The RX_FIFO underrun happens when the CPU/DMA reads data from RX FIFO when it is empty. In the case, the SPI controller will end the transmission and flag the error bit along with the RF_UDF bit in the SPI_ISR. The SPI controller will generate an interrupt if interrupts are enabled. The software has to clear the error bit and the RF_UDF bit. To start a new transaction, the software has to reset the FIFO by writing to the SRST (soft reset) bit in the SPI_GCR.

4.RX_FIFO Overflow

The RX_FIFO overflow happens when the CPU/DMA writes data into the RX FIFO when it is full. In the case, the SPI controller will end the transmission and flag the error bit along with the RF_OVF bit in the SPI_ISR. The SPI controller will generate an interrupt if interrupts are enabled. The software has to clear the error bit and the RF_OVF bit. To start a new transaction, the software has to reset the FIFO by writing to the SRST (soft reset) bit in the SPI_GCR.

9.4.4 Programming Guidelines

9.4.4.1 Writing/Reading Data Process

The SPI transfers serial data between the processor and the external device. The CPU mode and DMA mode are the two main operational modes for SPI. For each SPI, the data is simultaneously transmitted (shifted out serially) and received (shifted in serially). The SPI has 2 channels, including the TX channel and RX channel. The TX channel has the path from TX FIFO to the external device. The RX channel has the path from the external device to RX FIFO.

Write Data: The CPU or DMA must write data on the SPI_TXD, the data on the register are automatically moved to TX FIFO.

Read Data: To read data from RX FIFO, the CPU or DMA must access the SPI_RXD and the data are automatically sent to the SPI_RXD register.

In CPU or DMA mode, the SPI sends a completed interrupt (SPI_ISR[TC]) to the processor after each transmission is complete.

Figure 9-21 SPI Write/Read Data in CPU Mode

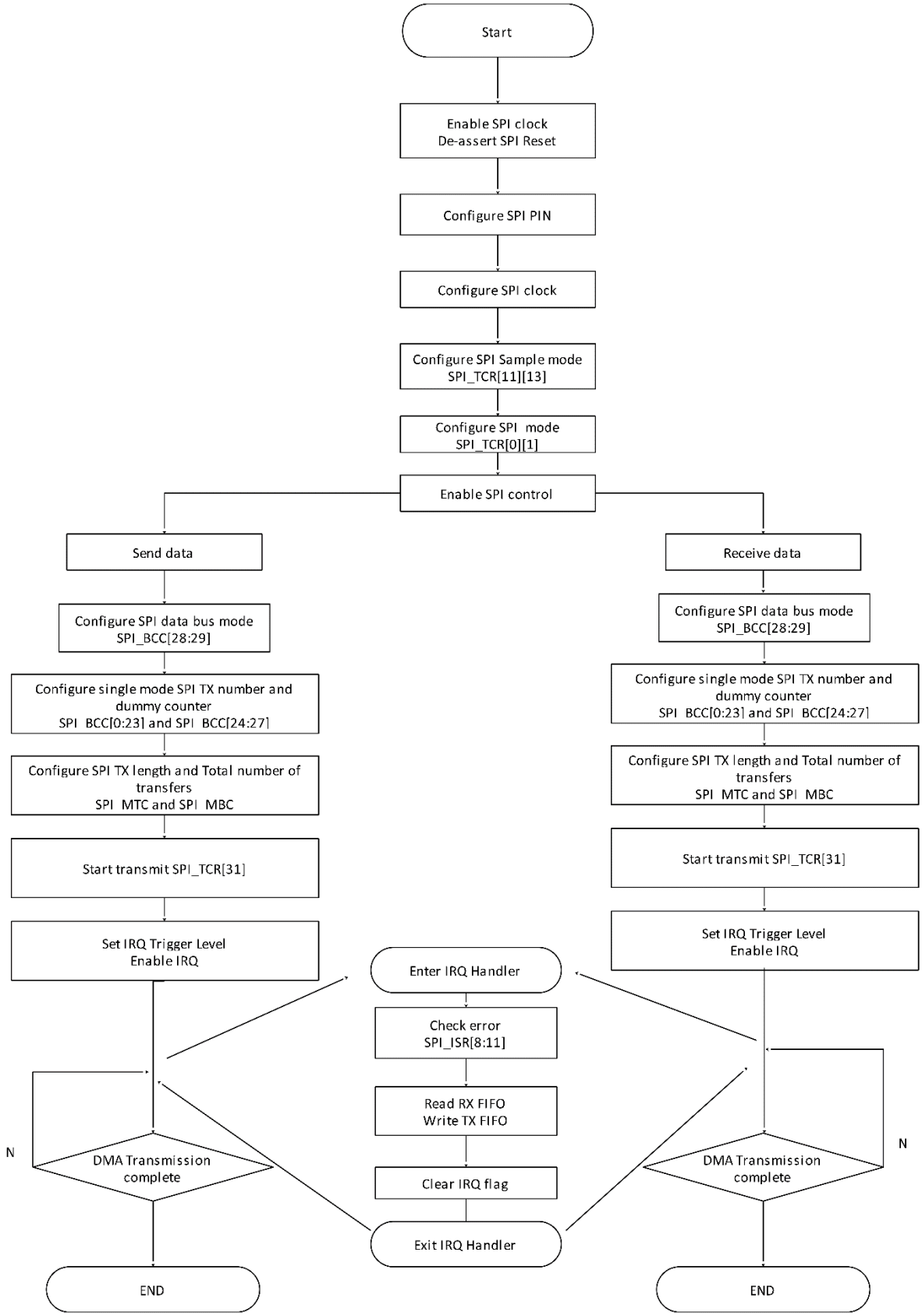
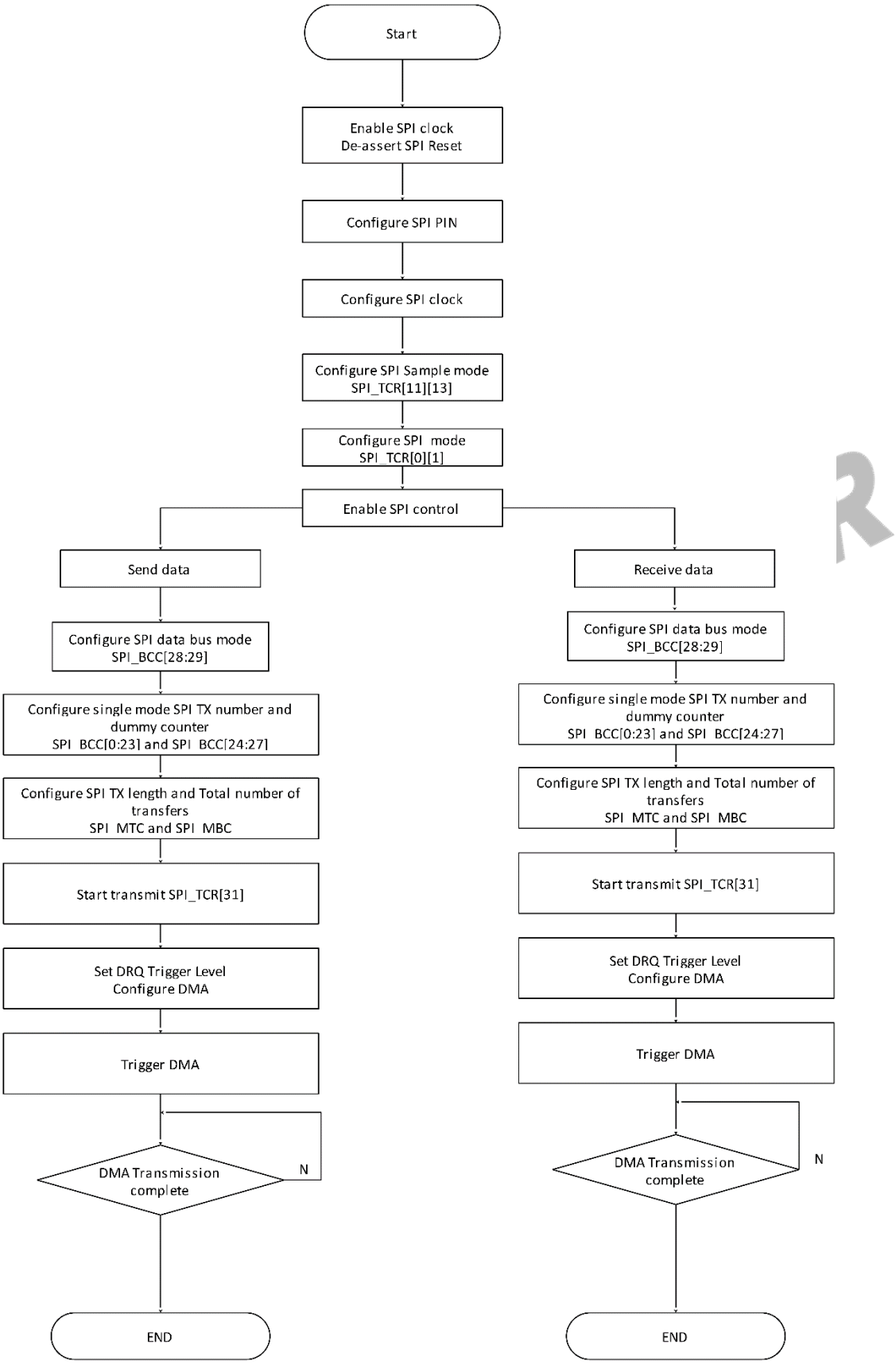


Figure 9-22 SPI Write/Read Data in DMA Mode



9.4.4.2 Calibrate Delay Chain

The SPI has one delay chain which is used to generate delay to make proper timing between the internal SPI clock signal and data signals. Delay chain is made up of 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:

Step 1 Enable SPI. To calibrate the delay chain by the operation registers in SPI, the SPI must be enabled through AHB reset and AHB clock gating control registers.

Step 2 Configure a proper clock for SPI. The calibration delay chain is based on the clock for SPI from CCU.

Step 3 Set proper initial delay value. Write 0xA0 to the SPI_SAMP_DL to set initial delay value 0x20 to delay chain. Then write 0x0 to the SPI_SAMP_DL to clear this value.

Step 4 Write 0x8000 to the SPI_SAMP_DL to start to calibrate the delay chain.

Step 5 Wait until the flag (SPI_SAMP_DL[14]) of calibration done is set. The number of delay cells is shown at the SPI_SAMP_DL[13:8]. The delay time generated by these delay cells is nearly equal to the cycle of the SPI clock. This value is the result of calibration.

Step 6 Calculate the delay time of one delay cell according to the cycle of the SPI clock and the result of calibration.

9.4.5 Register List

Module Name	Base Address
SPI0	0x40009000

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_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 3Wire CLOCK Configuration Register
SPI_TBR	0x0048	SPI TX Bit Register
SPI_RBR	0x004C	SPI RX Bit Register

Register Name	Offset	Description
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

9.4.6 Register Description

9.4.6.1 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 Write '1' to this bit will clear the SPI controller, and auto clear to '0' when reset operation completes Write '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. 1: Stop transmitting data when RXFIFO is full 0: Normal operation, ignore RXFIFO status **Note**: *It cannot be written when XCH=1*
6:3	/	/	/
2	R/W	0x0	MODE_SELEC Sample Timing Mode Select 0: Old mode of Sample Timing 1: New mode of Sample Timing **Note**: *It cannot be written when XCH=1*
1	R/W	0x0	MODE SPI Function Mode Select 0: Slave Mode 1: Master Mode **Note**: *It cannot be written when XCH=1*
0	R/W	0x0	EN SPI Module Enable Control 0: Disable 1: Enable **Note**: *After transforming from bit_mode to byte_mode, it must enable the SPI Module again.*

9.4.6.2 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. Write “1” to this bit will start the SPI burst, and will auto clear after finishing the bursts transfer specified by BC. Write “1” to SRST will also clear this bit. Write ‘0’ to this bit has no effect. **Note**: *It cannot be written when XCH=1.*
30:16	/	/	/
15	R/W	0x0	SDC1 Master Sample Data Control register1 Set this bit to ‘1’ to make the internal read sample point with a delay of half cycle of SPI_CLK. It is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK propagating between master and slave. 0: Normal operation, do not delay internal read sample point 1: Delay internal read sample point **Note**: *It cannot be written when XCH=1.*
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 input/output mode for SPI mode 0.
13	R/W	0x0	SDM Master Sample Data Mode 1: Normal Sample Mode 0: Delay 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.
12	R/W	0x0	FBS First Transmit Bit Select 0: MSB first 1: LSB first **Note**: *It cannot be written when XCH=1.*

Offset: 0x0008			Register Name: SPI_TCR
Bit	**Read/Write**	**Default/Hex**	**Description**
11	R/W	0x0	SDC Master Sample Data Control Set this bit to '1' to make the internal read sample point with a delay of half cycle of SPI_CLK. It is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK propagating between master and slave. 0: Normal operation, do not delay internal read sample point 1: Delay internal read sample point **Note**: *It cannot be written when XCH=1.*
10	R/W	0x0	RPSM Rapids mode select Select Rapids mode for high speed write. 0: Normal write mode 1: Rapids write mode **Note**: *It 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 **Note**: *It 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. **Note**: *It 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 **Note**: *It 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 **Note**: *It cannot be written when XCH=1.*

Offset: 0x0008			Register Name: SPI_TCR
Bit	**Read/Write**	**Default/Hex**	**Description**
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 **Note**: *It 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 **Note**: *It 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) **Note**: *It 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) **Note**: *It 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) **Note**: *It cannot be written when XCH=1.*

9.4.6.3 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

Offset: 0x0010			Register Name: SPI_IER
Bit	**Read/Write**	**Default/Hex**	**Description**
12	R/W	0x0	TC_INT_EN Transfer Completed Interrupt Enable 0: Disable 1: Enable
11	R/W	0x0	TF_UDR_INT_EN TXFIFO under run 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 under run Interrupt Enable 0: Disable 1: Enable
8	R/W	0x0	RF_OVF_INT_EN RX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
7	/	/	/
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

Offset: 0x0010			Register Name: SPI_IER
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	RF_RDY_INT_EN RX FIFO Ready Request Interrupt Enable 0: Disable 1: Enable

9.4.6.4 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 has been exchanged. In other condition, when set, this bit indicates that all the data in TXFIFO has been loaded in the Shift register, and the Shift register has shifted out all the bits. Writing 1 to this bit clears it. 0: Busy 1: Transfer Completed
11	R/W1C	0x0	TF_UDF TXFIFO under run 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 overflowed 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.

Offset: 0x0014			Register Name: SPI_ISR
Bit	**Read/Write**	**Default/Hex**	**Description**
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 has 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. Where TX_WL is the water level of RXFIFO
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. Where RX_WL is the water level of RXFIFO.

9.4.6.5 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 Write '1' to this bit will reset the control portion of the TX FIFO and auto clear to '0' when completing reset operation, write to '0' has no effect.
30	R/W	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, write '1' to this bit will switch TX FIFO read and write function to AHB bus. This bit is used to test the TX FIFO, don't set in normal operation and don't set RF_TEST and TF_TEST at the same time.*
29:25	/	/	/
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 Write '1' to this bit will reset the control portion of the receiver FIFO, and auto clear to '0' when completing reset operation, write '0' to this bit has no effect.
14	R/W	0x0	RF_TEST 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 switches RX FIFO read and writes function to AHB bus. This bit is used to test the RX FIFO. Do not set in normal operation or 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

Offset: 0x0018			Register Name: SPI_FCR
Bit	**Read/Write**	**Default/Hex**	**Description**
7: 0	R/W	0x1	RX_TRIG_LEVEL RX FIFO Ready Request Trigger Level

9.4.6.6 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
27:26	/	/	/
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

9.4.6.7 0x0020 SPI Wait Clock Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: SPI_WCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/

Offset: 0x0020			Register Name: SPI_WCR
Bit	**Read/Write**	**Default/Hex**	**Description**
19:16	R/W	0x0	SWC Dual mode direction switch wait clock counter (for master mode only). 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 starting dual data transmission in dual SPI mode. The SPI module counts SPI_SCLK by SWC for delaying next word data transmission. Cannot be written when XCH=1.*
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 transmission. The SPI module counts SPI_SCLK by WCC for delaying next word data transmission. 0: No wait states inserted N: N SPI_SCLK wait states inserted **Note**: *Cannot be written when XCH=1.*

9.4.6.8 0x0028 SPI Sample Delay Control Register (Default Value: 0x0000_2000)

Offset: 0x0028			Register Name: SPI_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	/	/	/

Offset: 0x0028			Register Name: SPI_SAMP_DL
Bit	**Read/Write**	**Default/Hex**	**Description**
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.

9.4.6.9 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 **Note**: *It cannot be written when XCH=1; Total transfer data, include the TXD, RXD and dummy burst.*

9.4.6.10 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 **Note**: *It cannot be written when XCH=1.*

9.4.6.11 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 **Note**: *It cannot be written when XCH=1; Quad mode include 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 **Note**: *It 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 is don't care by the device. 0: 0 burst 1: 1 burst … N: N bursts **Note**: *It 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 **Note**: *It cannot be written when XCH=1.*

9.4.6.12 0x0040 SPI Bit-Aligned Transfer Configure Register (Default Value: 0x0000_00A0)

Offset: 0x0040			Register Name: SPI_BATC
Bit	**Read/Write**	**Default/Hex**	**Description**

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 t1o transfer the serial bits frame, it is only valid when *Work Mode Select==0x10/0x11.* 0: Idle 1: Initiates transfer Write "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. Write '0' to this bit has no effect.
30	R/W	0x0	MSMS Master Sample Standard 0: Standard Sample Mode 1: Delay 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 **Note**: *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 **Note**: *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 **Note**: *It is only valid when Work Mode Select=0x10/0x11, and can't be written when TCE=1.*
15:14	/	/	/

Offset: 0x0040			Register Name: SPI_BATC
Bit	**Read/Write**	**Default/Hex**	**Description**
13:8	R/W	0x00	Configure the length of serial data frame(burst) of TX 000000: 0bit 000001: 1bit … 100000: 32bits Other values: Reserved **Note**: *It is only valid when Work Mode Select=0x10/0x11, and can't 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 **Note**: *It is only valid when Work Mode Select=0x10/0x11, and only work in Mode0, can't 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 **Note**: *It is only valid when Work Mode Select=0x10/0x11, and only work in Mode0, can't 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) **Note**: *It is only valid when Work Mode Select=0x10/0x11, and only work in Mode0, can't 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 **Note**: *It is only valid when Work Mode Select=0x10/0x11, and only work in Mode0, can't be written when TCE=1.*

Offset: 0x0040			Register Name: SPI_BATC
Bit	Read/Write	Default/Hex	Description
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: Reserve 10: Data frame is bit aligned in 3-Wire SPI 11: Data frame is bit aligned in Standard SPI

9.4.6.13 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.

9.4.6.14 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. **Note**: *In the process of transmission, the LSB is transmitted first.*

NOTE

This register is only valid when Work Mode Select=0x10/0x11.

9.4.6.15 0x004C SPI RX Bit Register (Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: SPI_RBR
Bit	Read/Write	Default/Hex	Description

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. **Note**: *In the process of transmission, the LSB is transmitted first.*

NOTE

This register is only valid when Work Mode Select=0x10/0x11.

9.4.6.16 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	0x5	The delay cycles The counts of hold cycles from DMA last signal high to dma_active high

9.4.6.17 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 increase by 2. In word accessing method, four SPI burst data are written and the TXFIFO depth is increased by 4. **Note**: *This address is writing-only if TF_TEST is '0', and if TF_TEST is set to '1', this address is readable and writable to test the TX FIFO through the AHB bus.*

9.4.6.18 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 decrease by 2. In word accessing method, the four SPI bursts are returned and the RXFIFO depth is decreased by 4. **Note**: *This address is read-only if RF_TEST is '0', and if RF_TEST is set to '1', this address is readable and writable to test the RX FIFO through the AHB bus.*

9.5 SPI_DBI

9.5.1 Overview

R128 provides a 3/4 line SPI display bus interface (SPI_DBI) for video data transmission. It supports DBI mode or SPI mode. The DBI mode is compatible with multiple video data formats at the same time. The SPI mode is used for low-cost display schemes.

The SPI mode 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, and four chips select to support multiple peripherals.
- Supports interrupts and DMA
- Supports mode0, mode1, mode2, and mode3
- Supports 3-wire/4-wire SPI
- Supports programmable serial data frame length: 0 bit to 32 bits
- Supports standard SPI, dual-output/dual-input SPI, dual I/O SPI, quad-output/quad-input SPI
- Supports maximum IO rate of the mass production: 96 MHz
- Supports 5 clock sources, Interrupt or DMA

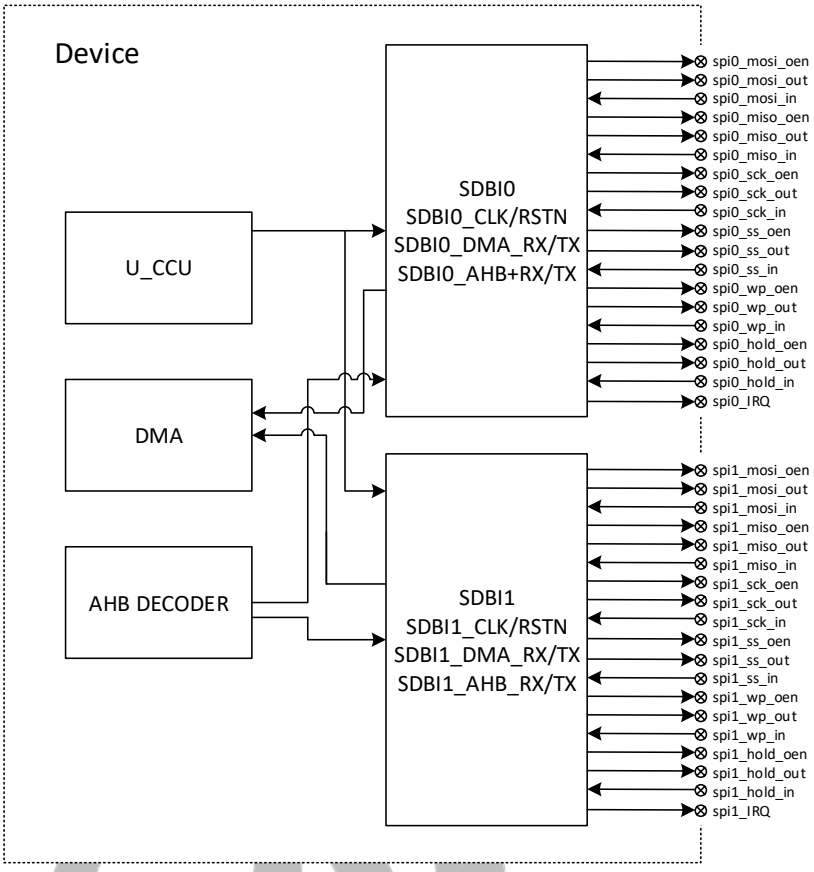
The DBI mode has the following features:

- Supports DBI Type C 3 Line/4 Line Interface Mode
- Supports 2 Data Lane Interface Mode
- Supports data source from CPU or DMA
- Supports RGB111/444/565/666/888 video format
- Maximum resolution of RGB666 240 x 320@30Hz with single data lane
- Maximum resolution of RGB888 240 x 320@60Hz or 320 x 480@30Hz with dual data lane
- Supports Tearing effect
- Supports software flexible control video frame rate

9.5.2 Block Diagram

The following figure shows a block diagram of the SPI_DBI.

Figure 9-23 SPI_DBI Block Diagram



9.5.3 Functional Description

9.5.3.1 External Signals

The following table 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.

Figure 9-24 SPI_DBI External Signals

External Signal		Description	Type
DBI Mode	DBI_CSX	Chip select signal, low active	I/O
	DBI_SCLK	Serial clock signal	I/O
	DBI_SDO	Data output signal	I/O
	DBI_SDI	Data input signal, the data is sampled on the rising edge and the falling edge	I/O
	DBI_TE	Tearing effect input, it is used to capture the external TE signal edge.	I/O

	DBI_DCX	DCX pin is the select output signal of data and command. DCX = 0: register command; DCX = 1: Data or parameter.	I/O
	DBI_WRX	When DBI operates in dual data lane format, the RGB666 format 2 can use WRX to transfer data	I/O
SPI Mode	SPI1_CS	SPI1 chip select signal, low active When the device is not selected, data will not be accepted via the SI pin, and the SO pin will stop transmission.	I/O
	SPI1_CLK	SPI1 clock signal This pin is used to provide a clock to the device and is used to control the flow of data to and from the device.	I/O
	SPI1_MOSI	SPI1 master data out, slave data in	I/O
	SPI1_MISO	SPI1 master data in, slave data out	I/O
	SPI1_WP	Write protection and active low It also can be used for serial data input and output for SPI Quad Input or Quad Output mode.	I/O
	SPI1_HOLD	When the device is selected and a serial sequence is underway, the HOLD pin is can be used to temporarily pause the serial communication with the master device without deselecting or resetting the device serial sequence. While the HOLD pin is asserted, the SO pin is at high impedance, and all transitions on the SCK pin and data on the SI pin are ignored. It also can be used for serial data input and output for SPI Quad Input or Quad Output mode.	I/O

9.5.3.2 Clock Sources

The SPI_DBI controller gets 5 different clock sources, users can select one of them to make SPI_DBI clock source. The following table describes the clock sources for SPI_DBI. The following table describes the clock sources for SPI_DBI. For clock setting, configurations and gating information, refer to the section "CCU" and "CCU_AON".

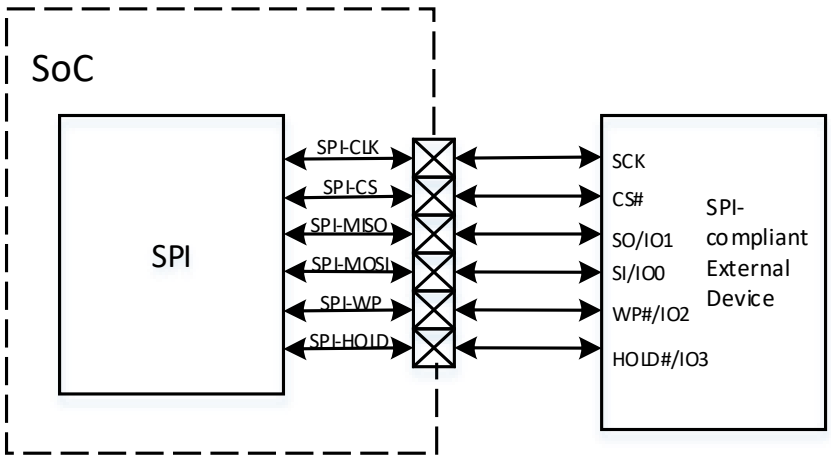
Table 9-14 SPI_DBI Clock Sources

Clock Sources	Description
HFCLK	High frequency crystal clock for system (optional frequency: 24M/24.576M/26M/32M/40M)
DEVCLK	General-purpose high frequency crystal clock for system interfaces (frequency range: 0~392M)

9.5.3.3 Typical Application

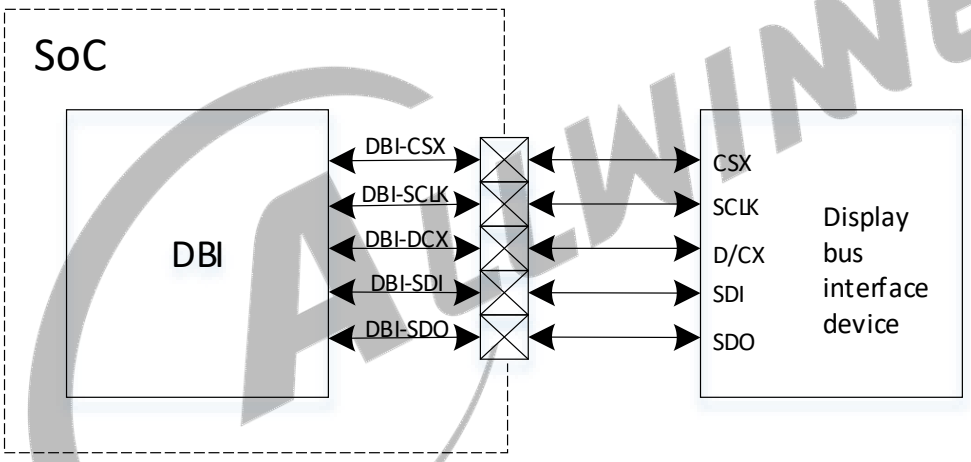
The following figure shows the application block diagram when the SPI master device is connected to a slave device.

Figure 9-25 SPI Application Block Diagram



The following figure shows the application block diagram when the DBI master device is connected to a display bus interface device.

Figure 9-26 DBI Application Block Diagram



9.5.3.4 SPI Transmission Format

The SPI supports 4 different formats for data transmission. The software can select one of the four modes in which the SPI works by setting the bit1 (Polarity) and bit0 (Phase) of SPI_TCR. 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.

The CPOL (SPI_TCR[1]) defines the polarity of the clock signal (SPI_SCLK). The SPI_SCLK is a high level when CPOL is '1' and it is a low level when CPOL is '0'. The CPHA (SPI_TCR[0]) decides whether the leading edge of SPI_SCLK is used to setup or sample data. The leading edge is used to setup data when CPHA is '1', and sample data when CPHA is '0'. The following table lists the four modes.

Table 9-15 SPI Transmission Format

SPI Mode	Polarity (CPOL)	Phase (CPHA)	Leading Edge	Trailing Edge
mode0	0	0	Sample on the rising edge	Setup on the falling edge
mode1	0	1	Setup on the rising edge	Sample on the falling edge

SPI Mode	Polarity (CPOL)	Phase (CPHA)	Leading Edge	Trailing Edge
mode2	1	0	Sample on the falling edge	Setup on the rising edge
mode3	1	1	Setup on the falling edge	Sample on the rising edge

The following two figures describe four waveforms for SPI_SCLK.

Figure 9-27 SPI Phase 0 Timing Diagram

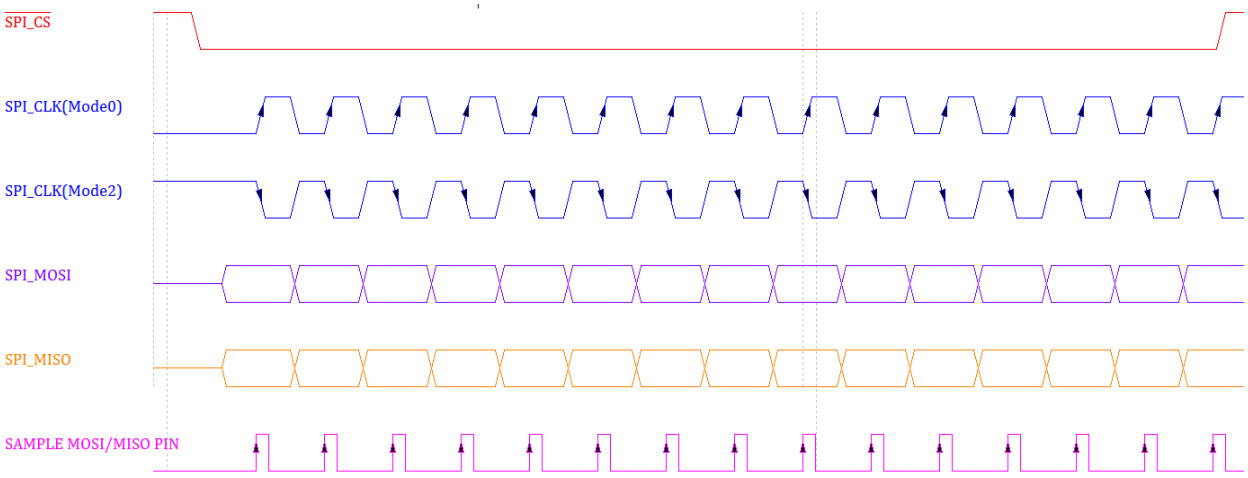
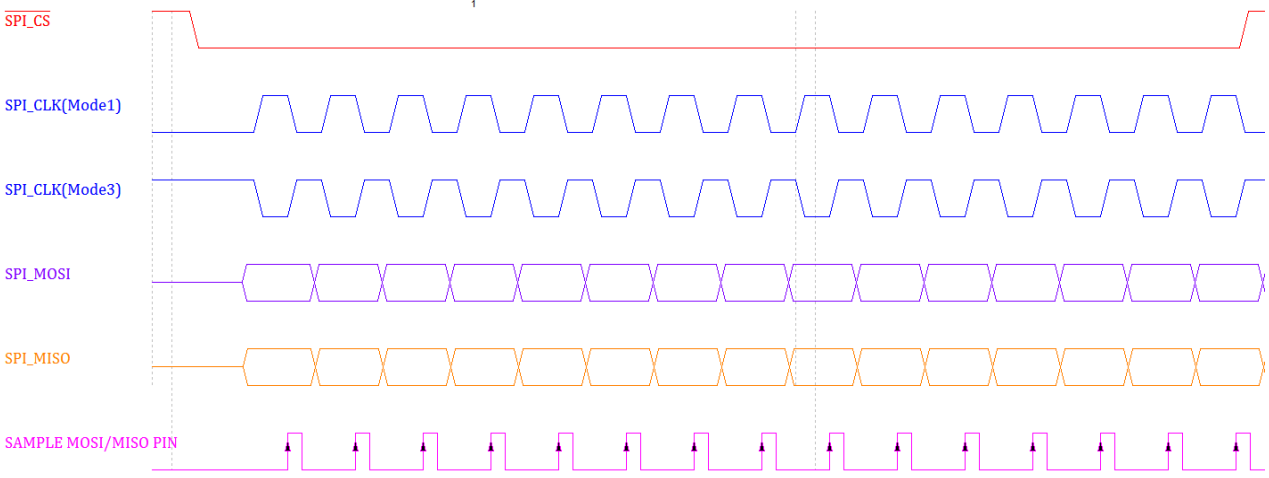


Figure 9-28 SPI Phase 1 Timing Diagram



9.5.3.5 SPI Master and Slave Mode

The SPI controller can be configured to a Master or Slave device. The master mode is selected by setting the MODE bit(**SPI_GCR**[1]); the slave mode is selected by clearing the MODE bit.

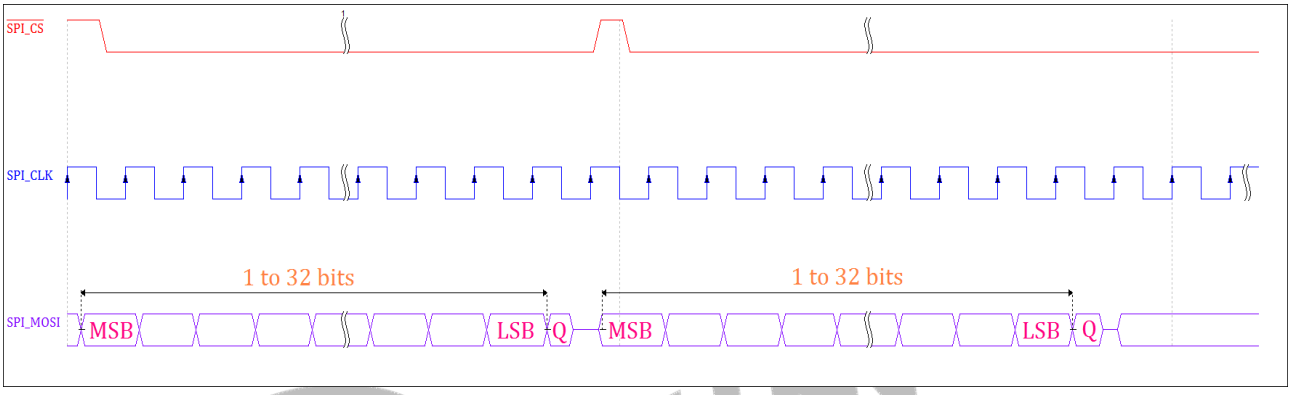
In the master mode, SPI_CLK is generated and transmitted to external device, and the data from the TX FIFO is transmitted on the MOSI pin, the data from slave is received on the MISO pin and sent to RX FIFO. The Chip Select(SPI_SS) is active low signal, and it must be set low before data is transmitted or received. SPI_SS can be selected SPI auto control or software manual control. When using auto control, SS_OWNER(SPI_TCR[6]) must be cleared (the default value is 0); when using manual control, SS_OWNER must be set. The level of SPI_SS is controlled by the SS_LEVEL(SPI_TCR[7]).

In the slave mode, after the software selects the MODE bit(SPI_GCR[1]) 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.

9.5.3.6 SPI 3-Wire Mode

The SPI 3-Wire Mode is only valid when the SPI controller works in the master mode, and is selected when the Mode Select bit (SPI_BATC[1: 0]) is equal to 0x2. 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.

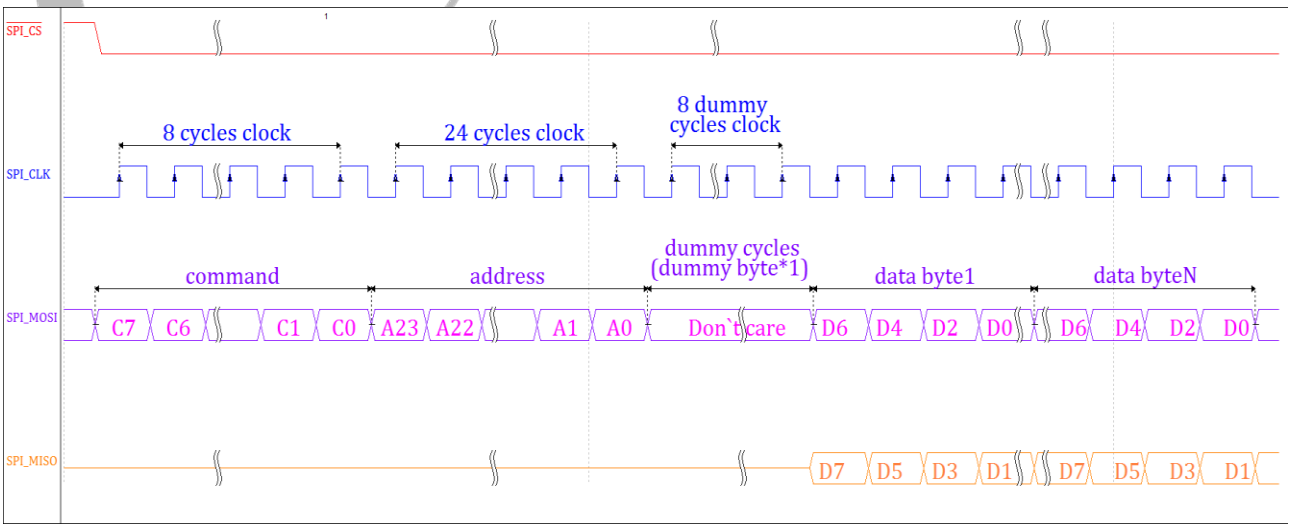
Figure 9-29 SPI 3-Wire Mode



9.5.3.7 SPI Dual Mode

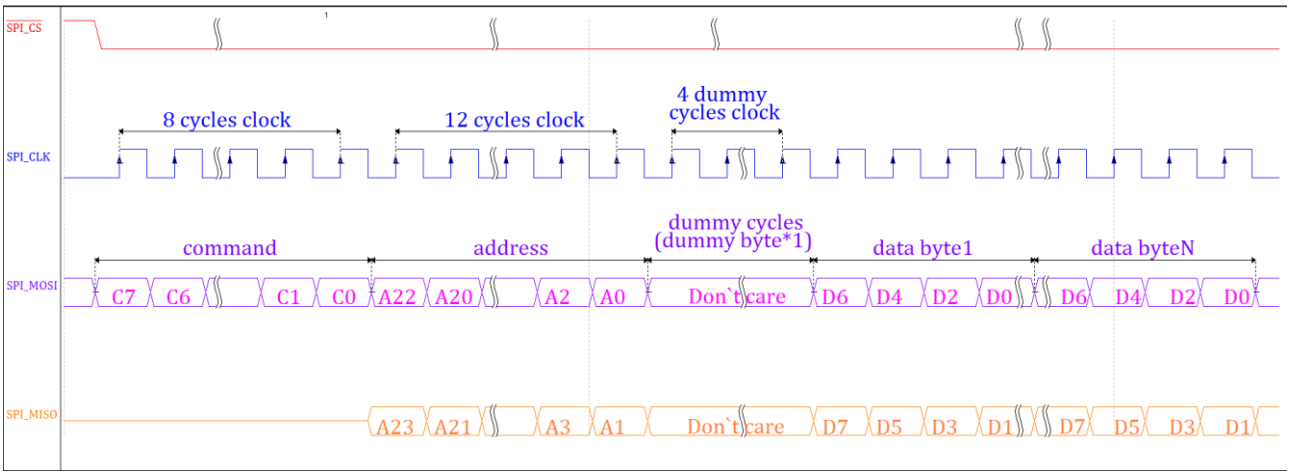
The dual read mode (SPI x2) is selected when the DRM is set in the SPI_BCC[28]. Using the dual mode allows data to be transferred to or from the device at two times the rate of standard single mode SPI devices, data can be read at fast speed using two data bits (MOSI and MISO) at a time. The following figure describes the Dual Input/Dual Output SPI and the Dual I/O SPI.

Figure 9-30 SPI Dual-Input/Dual-Output Mode



In the dual Input/dual 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 dual bits through the SPI_MOSI and SPI_MISO.

Figure 9-31 SPI Dual I/O Mode

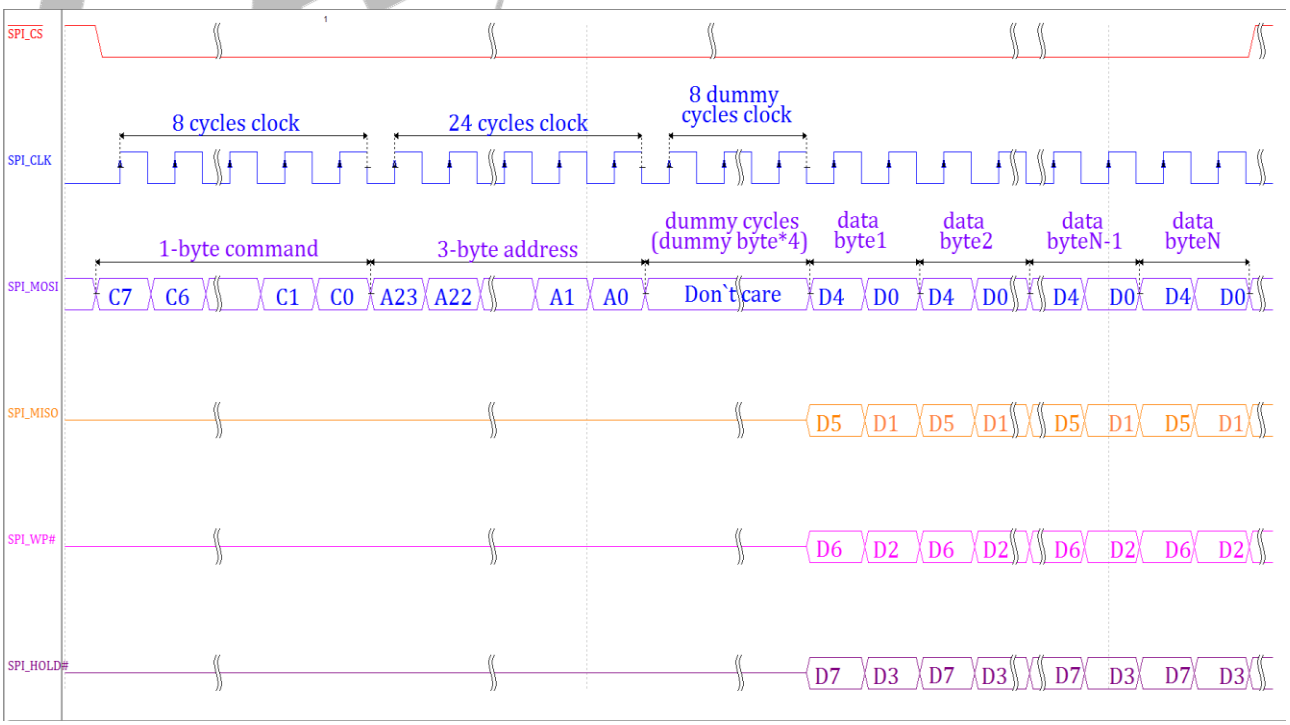


In the Dual I/O SPI, only the command bytes are output in unit of a single bit in serial mode through SPI_MOSI line. The address bytes and the dummy bytes are output in unit of dual bits through the SPI_MOSI and SPI_MISO. And the data bytes are output(write) and input(read) in unit of dual bits through the SPI_MOSI and SPI_MISO.

9.5.3.8 SPI Quad Mode

The Quad read mode (SPI x4) is selected when the Quad_EN is set in the SPI_BCC[29]. 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 9-32 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#.

9.5.3.9 Transmission/Reception Bursts in Master Mode

In SPI Master mode, the transmit and receive bursts (byte in unit) are configured before the SPI transfers serial data between the processor and external device. The transmit burst write in MWTC (bit[23: 0]) of the SPI_MTC. The transmit burst in single mode before automatically sending dummy burst write in STC (bit[23: 0]) of the SPI_BCC. For dummy data, SPI controller can automatically sent before receive by writing in DBC(bit[27:24]) of the SPI_BCC If users don't use SPI controller to send dummy data automatically, then the dummy bursts are used as the transmit counters to write together in MWTC(bit[23: 0]) of the SPI_MTC. In Master mode, the total burst numbers write in MBC(bit[23: 0]) of the SPI_MBC. When all transmit burst and receive burst are transferred, SPI controller will send a completed interrupt, at the same time, SPI controller will clear DBC, MWTC and MBC.

9.5.3.10 SPI Sample Mode and Run Clock Configuration

The SPI Controller runs at 3kHz~100MHz 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 mode: 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 48MHz or below 48MHz, SPI can work at normal sample mode or delay half cycle sample mode. When SPI runs over 48MHz, Set SDC in SPI_TCR to '1' to make the internal read sample point with a delay of half cycle 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 following tables shows the different configuration of SPI sample modes:

Table 9-16 SPI old Sample Mode

SPI Sample Mode	SDM(bit13)	SDC(bit11)	Run Clock
normal sample	1	0	<=24MHz
delay half cycle sample	0	0	<=40MHz
delay one cycle sample	0	1	>=60MHz

Table 9-17 SPI New Sample Mode

SPI Sample Mode	SDM(bit13)	SDC(bit11)	SDC1(bit15)
normal sample	1	0	0
delay half cycle sample	0	0	0
delay one cycle sample	0	1	0
delay 1.5 cycle sample	1	1	0
delay 2 cycle sample	1	0	1
delay 2.5 cycle sample	0	0	1
delay 3 cycle sample	0	1	1

9.5.3.11 DBI 3-Line Interface Writing and Reading Timing

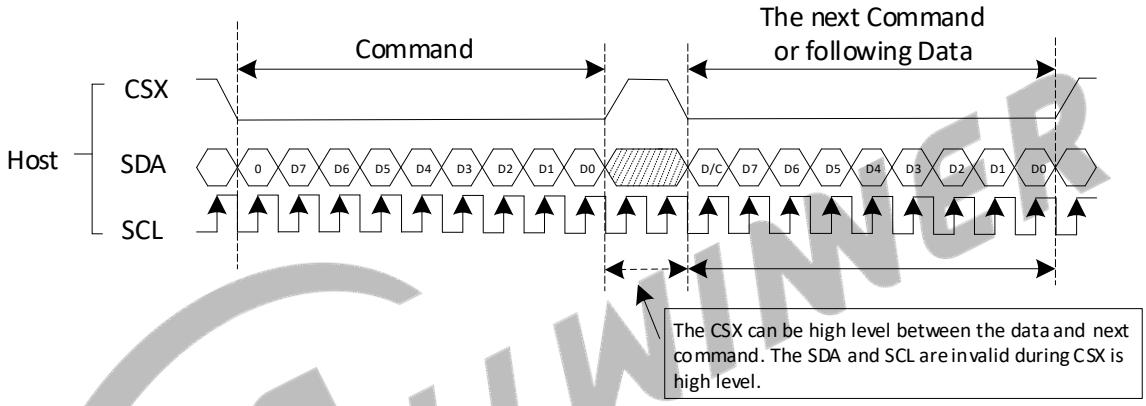
The 3-line DBI Interface I contains CSX, SDA, and SCL, where SDA shares this port for bidirectional port data input and output.

The 3-line DBI Interface II contains CSX, SDA, SCL, and SDI; Data input and output ports are independent of each other.

Since the 3-line display bus mode has no Data/Command data line indicating whether Data or Command is currently being transmitted, an extra bit is added to the data-stream before MSB to indicate whether Data or Command is currently being transmitted. (0: Command, 1: Data)

The following figure shows the writing operation format of 3-line DBI Interface I and Interface II.

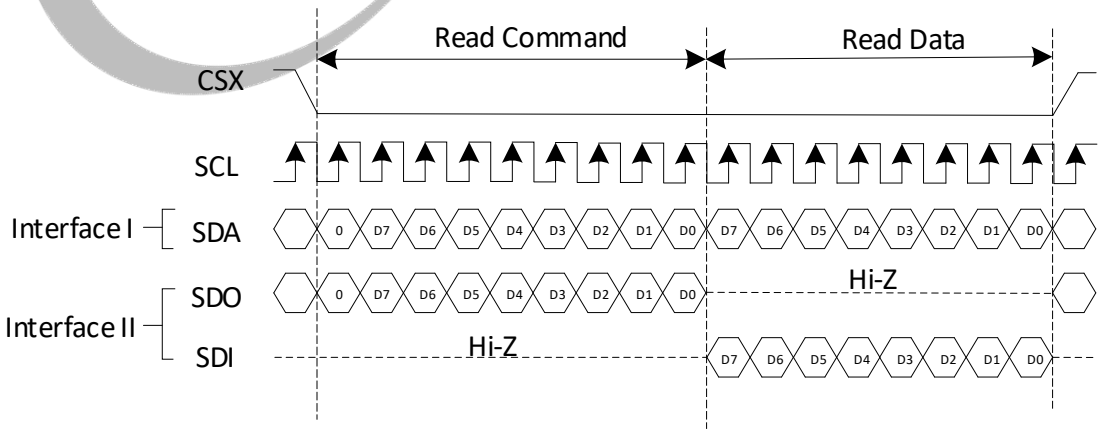
Figure 9-33 DBI 3-Line Display Bus Serial Interface Writing Operation Format



The 3-line DBI Interface I uses the SDA port as bidirectional data input and output port. There are only three cases of data reading volume, 8bits/24bits/32bits, and the first data sampled is high.

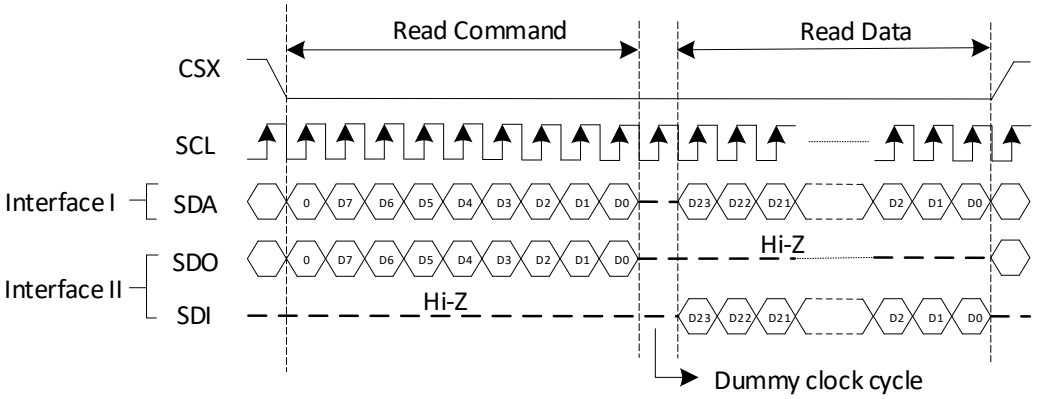
The following figure shows the 8 bits reading operation format of 3-line DBI Interface I and Interface II. After the read command is transmitted, the data is read immediately with on dummy period.

Figure 9-34 DBI 3-Line Display Bus Serial Interface 8-bit Reading Operation Format



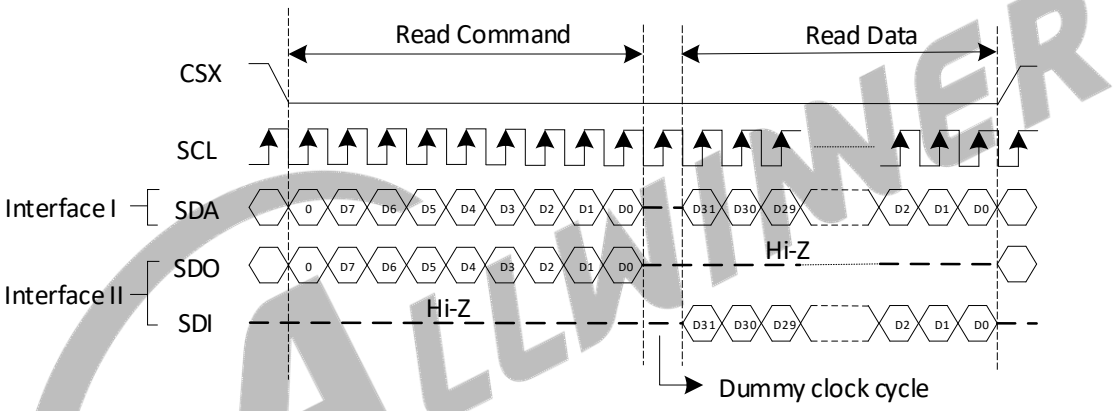
The following figure shows the 24 bits reading operation format of 3-line DBI Interface I and Interface II. After the read command is transmitted, the data is read after waiting for the dummy clock cycle.

Figure 9-35 DBI 3-Line Display Bus Serial Interface 24-bit Reading Operation Format



The following figure shows the 32 bits reading operation format of 3-line DBI Interface I and Interface II. After the read command is transmitted, the data is read after waiting for the dummy clock cycle.

Figure 9-36 DBI 3-Line Display Bus Serial Interface 32-bit Reading Operation Format



9.5.3.12 DBI 4-Line Interface Writing and Reading Timing

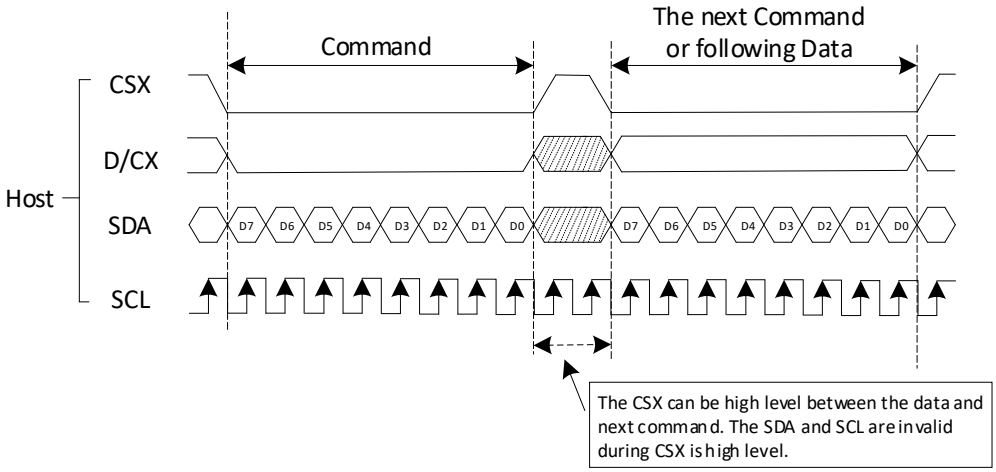
The 4-line DBI Interface I contains CSX, D/CX, SDA, and SCL, where SDA shares this port for bidirectional port data input and output.

The 4-line DBI Interface II contains CSX, D/CX, SDA, SCL, and SDI; Data input and output ports are independent of each other.

Since the 4-line display bus mode has a Data/Command data line indicating whether Data or Command is currently being transmitted (0: Command, 1: Data). So there is no need to add an extra bit to data-stream before MSB like the 3-line DBI.

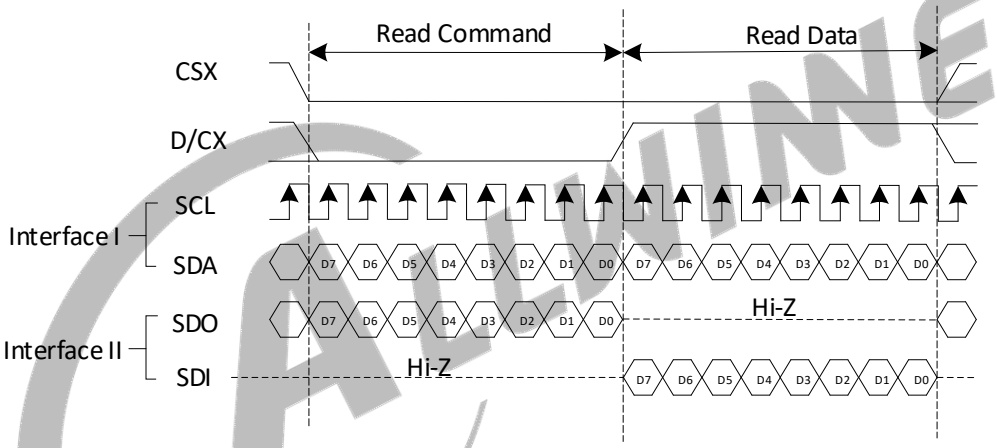
The following figure shows the writing operation format of 4-line DBI Interface I and Interface II.

Figure 9-37 DBI 4-Line Display Bus Serial Interface Writing Operation Format



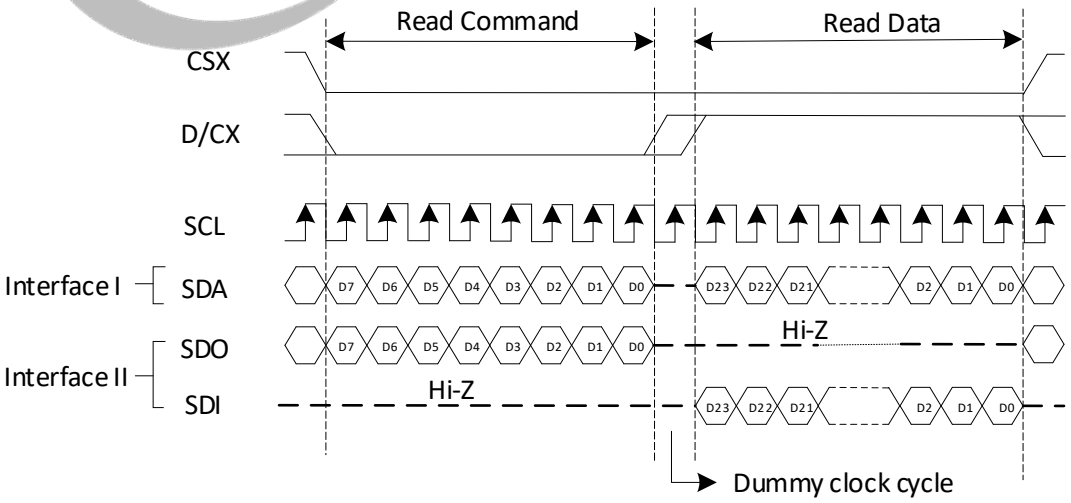
The following figure shows the 8 bits reading operation format of 4-line DBI Interface I and Interface II.

Figure 9-38 DBI 4-Line Display Bus Serial Interface 8-bit Reading Operation Format



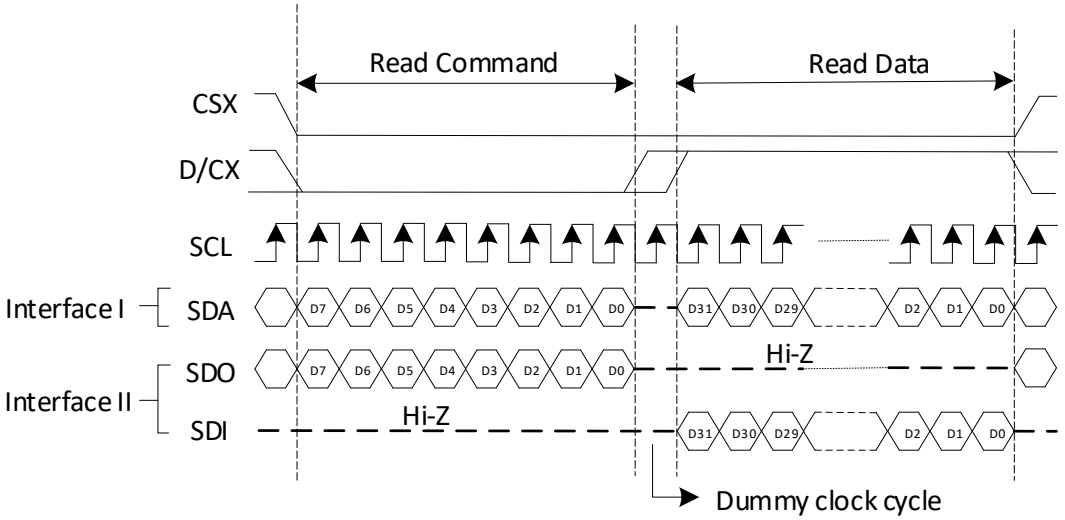
The following figure shows the 24 bits reading operation format of 4-line DBI Interface I and Interface II.

Figure 9-39 DBI 4-Line Display Bus Serial Interface 24-bit Reading Operation Format



The following figure shows the 32 bits reading operation format of 4-line DBI Interface I and Interface II.

Figure 9-40 DBI 4-Line Display Bus Serial Interface 32-bit Reading Operation Format



9.5.3.13 DBI 3-Line Interface Transmit Video Format

Figure 9-41 RGB111 3-Line Interface Transmit Video Format

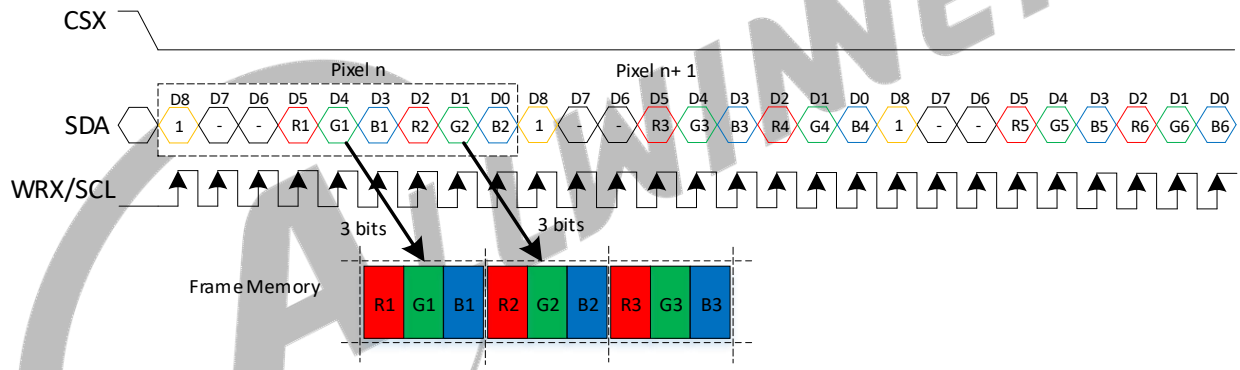
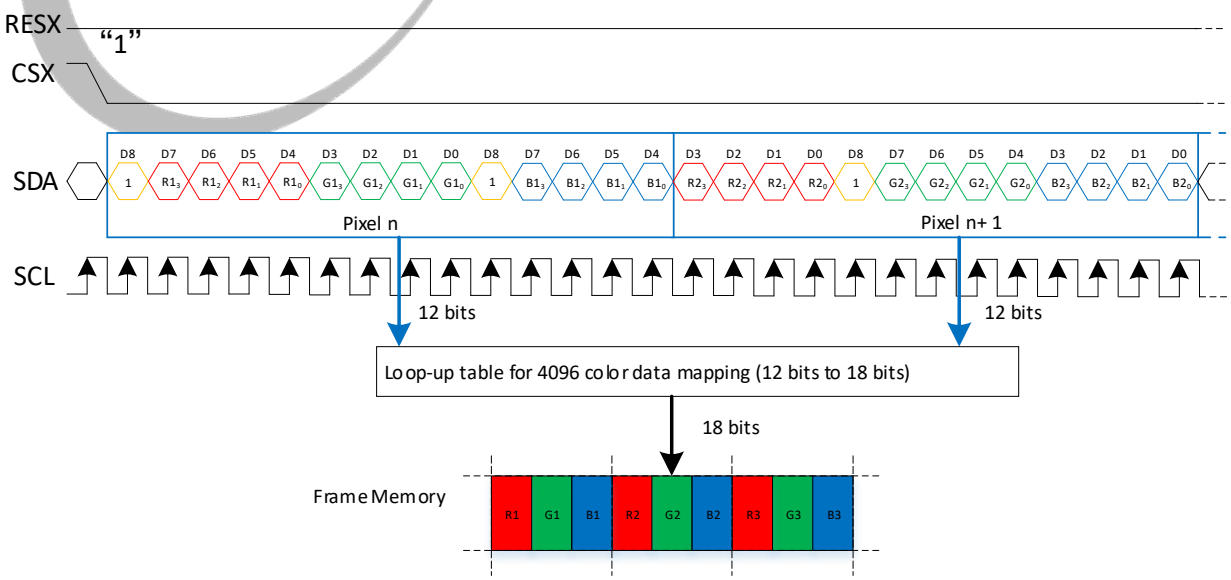
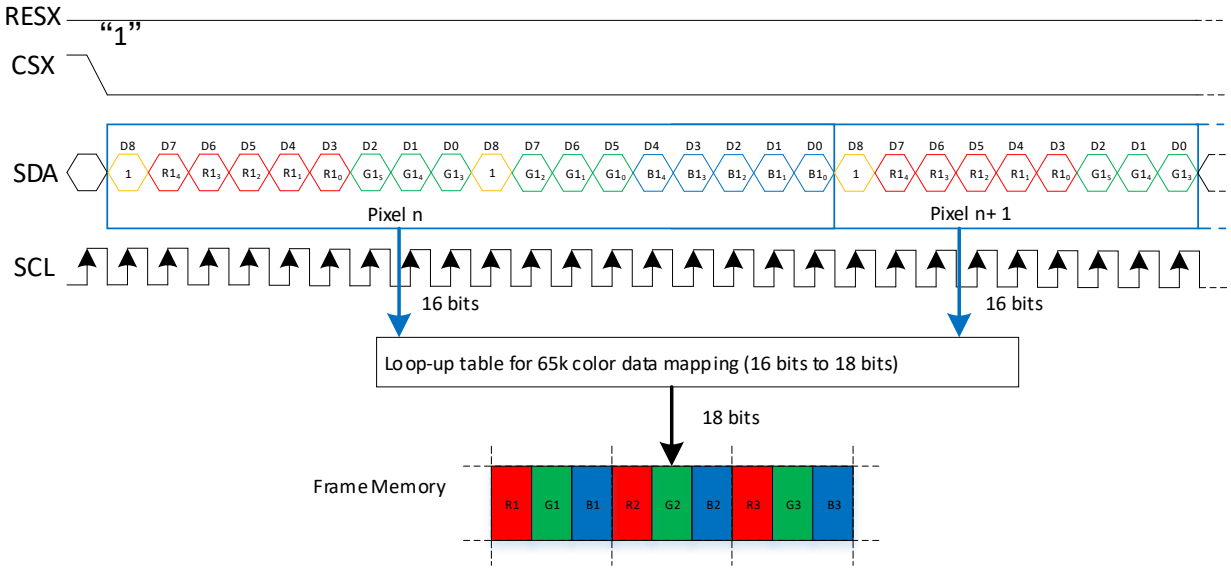


Figure 9-42 RGB444 3-Line Interface Transmit Video Format



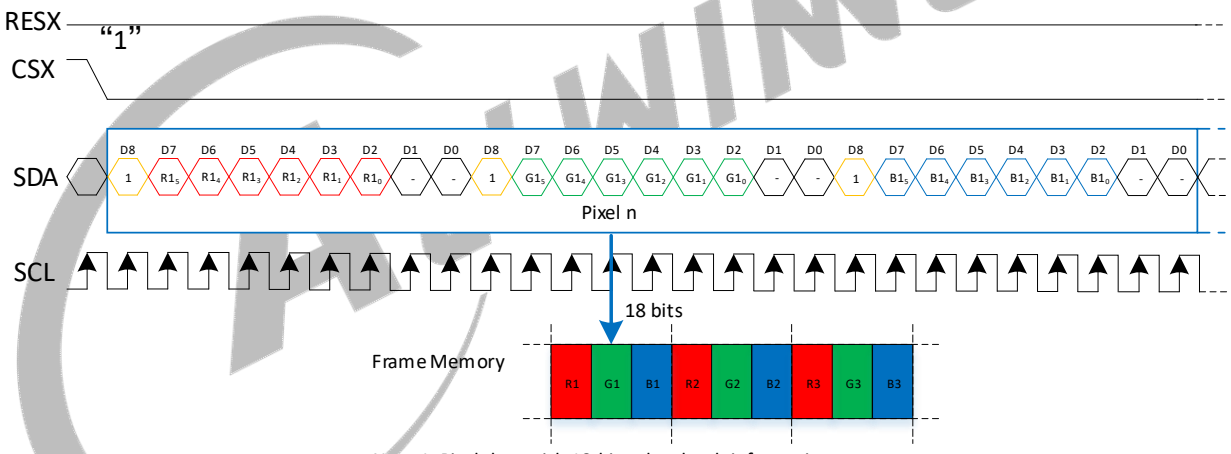
Note 1. Pixel data with 12-bit color depth information
Note 2. The most significant bits are: Rx3, Gx3 and Bx3
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

Figure 9-43 RGB565 3-Line Interface Transmit Video Format



Note 1. Pixel data with 16-bit color depth information
Note 2. The most significant bits are: Rx4, Gx5 and Bx4
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

Figure 9-44 RGB666 3-Line Interface Transmit Video Format



Note 1. Pixel data with 18-bit color depth information
Note 2. The most significant bits are: Rx5, Gx5 and Bx5
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

9.5.3.14 DBI 4-Line Interface Transmit Video Format

Figure 9-45 RGB111 4-Line Interface Transmit Video Format

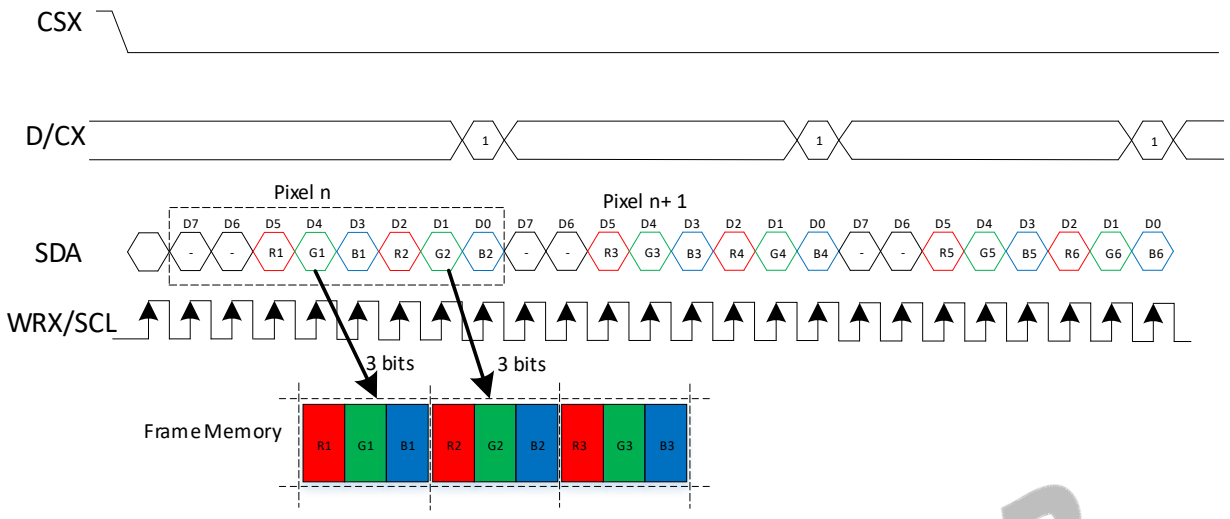
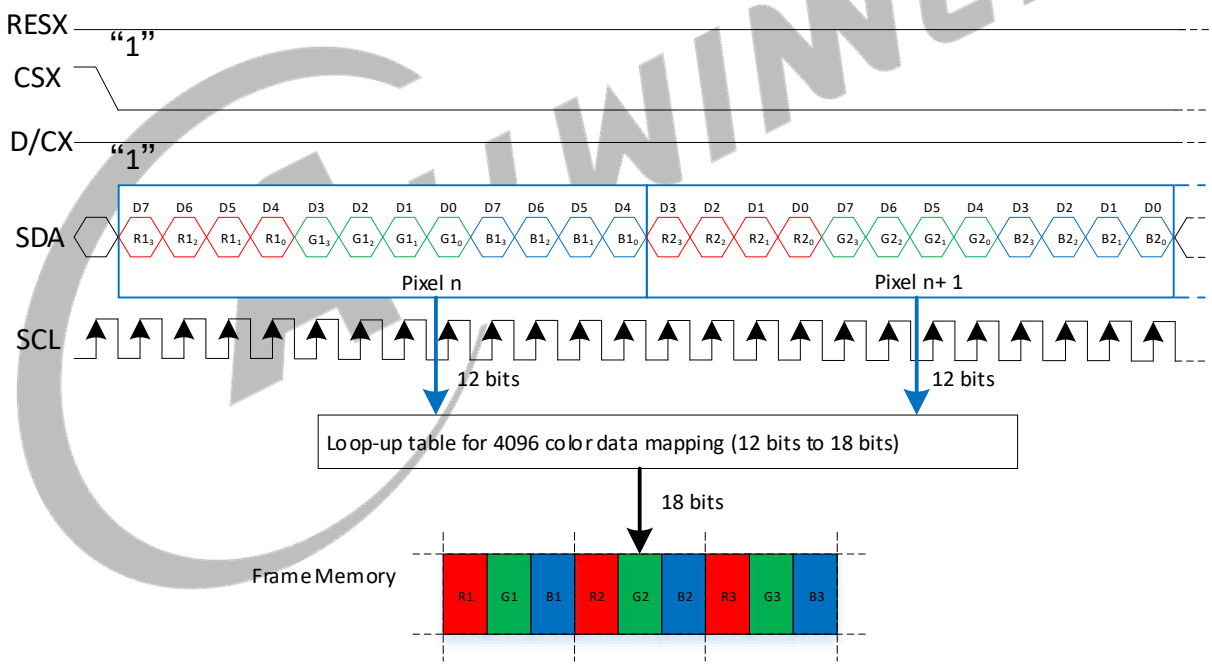
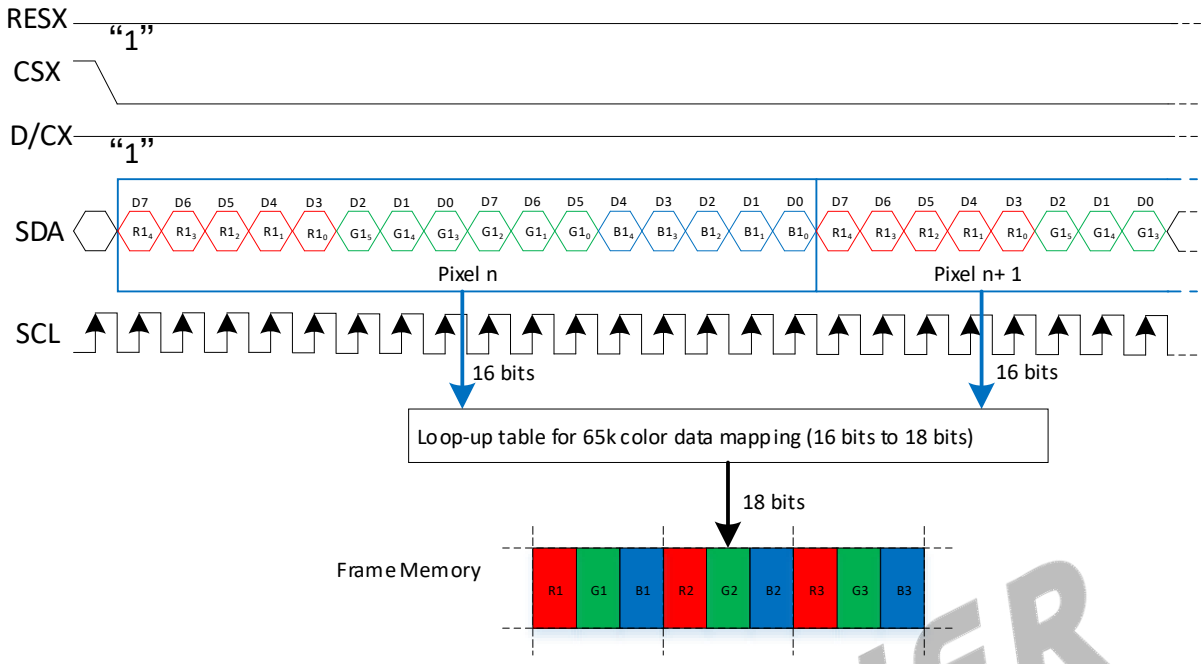


Figure 9-46 RGB444 4-Line Interface Transmit Video Format



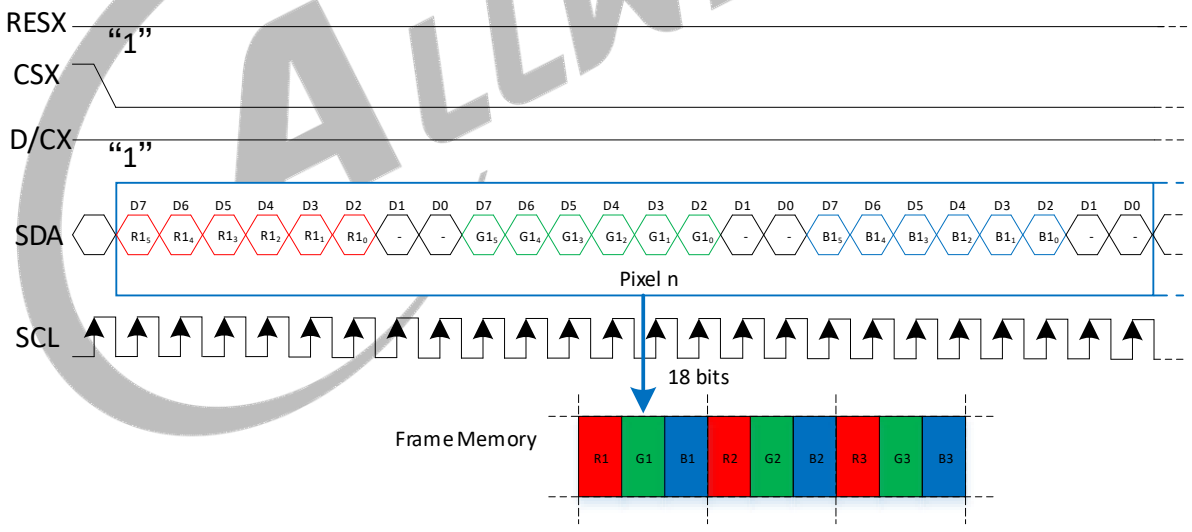
Note 1. Pixel data with 12-bit color depth information
Note 2. The most significant bits are: Rx3, Gx3 and Bx3
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

Figure 9-47 RGB565 4-Line Interface Transmit Video Format



Note 1. Pixel data with 16-bit color depth information
Note 2. The most significant bits are: Rx4, Gx5 and Bx4
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

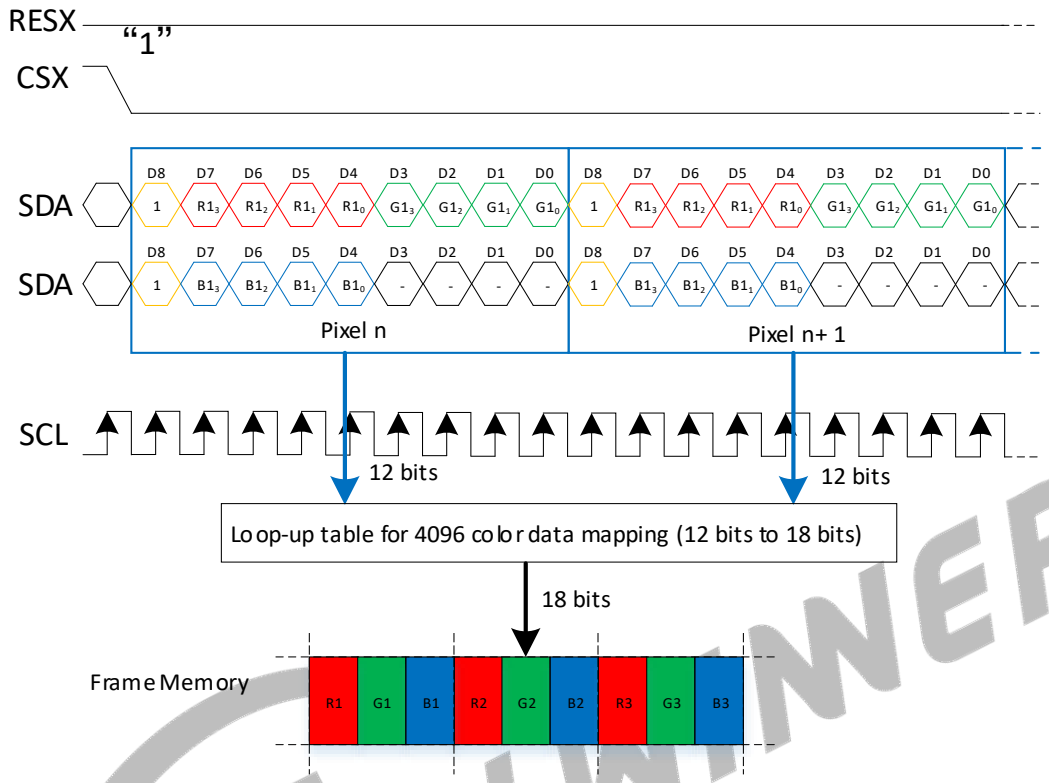
Figure 9-48 RGB666 4-Line Interface Transmit Video Format



Note 1. Pixel data with 18-bit color depth information
Note 2. The most significant bits are: Rx5, Gx5 and Bx5
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

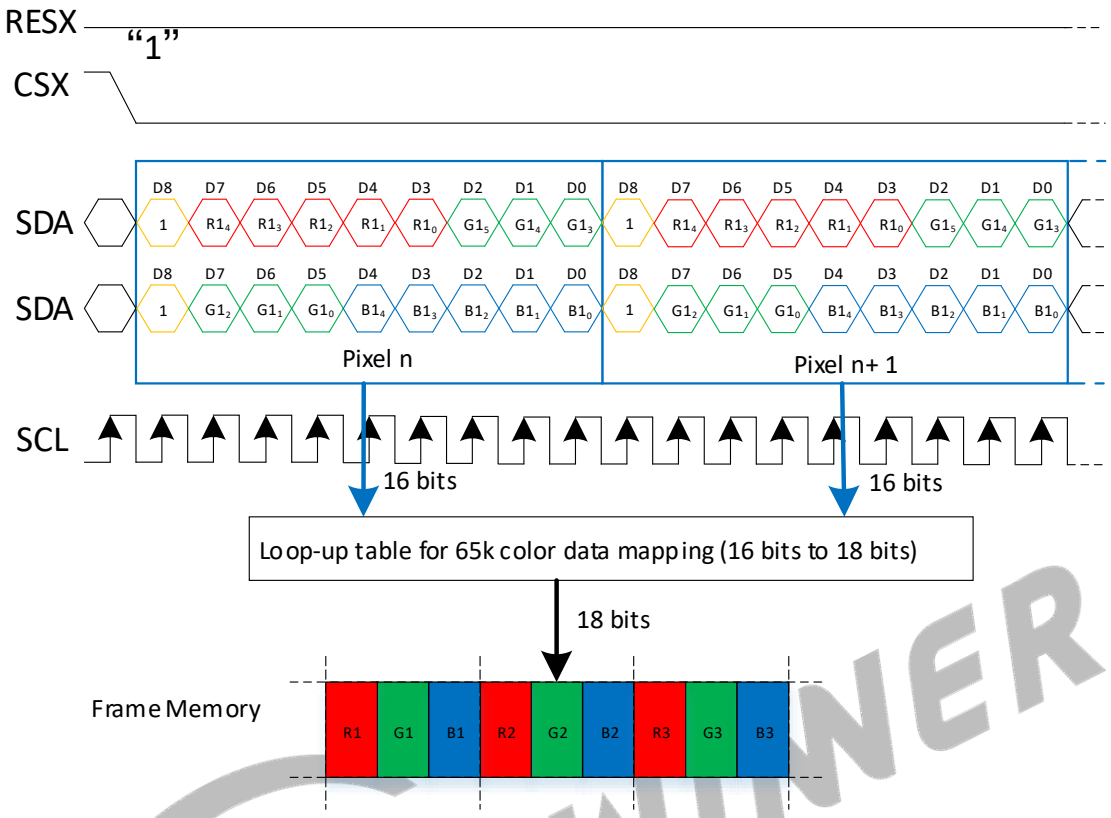
9.5.3.15 DBI 2 Data Lane Interface Transmit Video Format

Figure 9-49 RGB444 2 Data Lane Interface Transmit Video Format



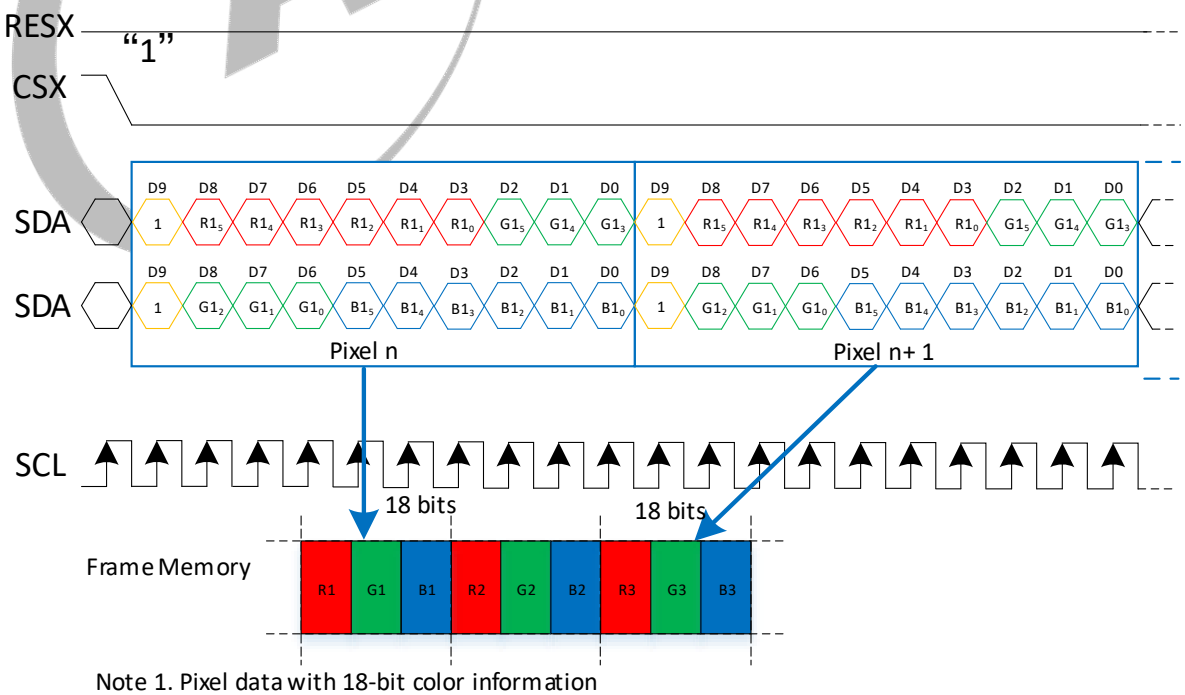
Note 1. Pixel data with 12-bit color information
Note 2. The most significant bits are: Rx3, Gx3 and Bx3
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

Figure 9-50 RGB565 2 Data Lane Interface Transmit Video Format



Note 1. Pixel data with 16-bit color information
Note 2. The most significant bits are: Rx4, Gx5 and Bx4
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

Figure 9-51 RGB666 2 Data Lane Interface Transmit Video Format 0



Note 1. Pixel data with 18-bit color information
Note 2. The most significant bits are: Rx5, Gx5 and Bx5
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

Figure 9-52 RGB666 2 Data Lane Interface Transmit Video Format 1 (ilitek)

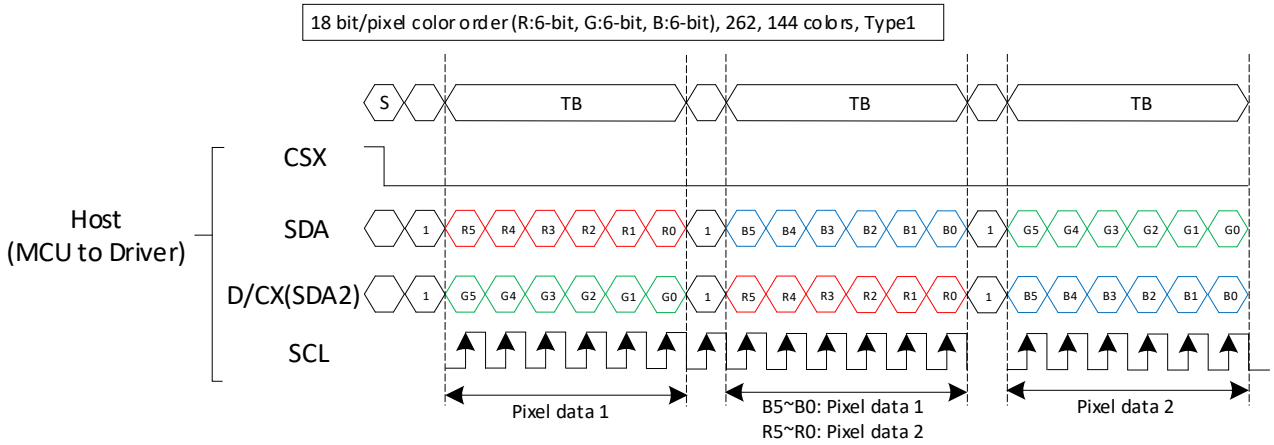


Figure 9-53 RGB666 2 Data Lane Interface Transmit Video Format 2 (New vision)

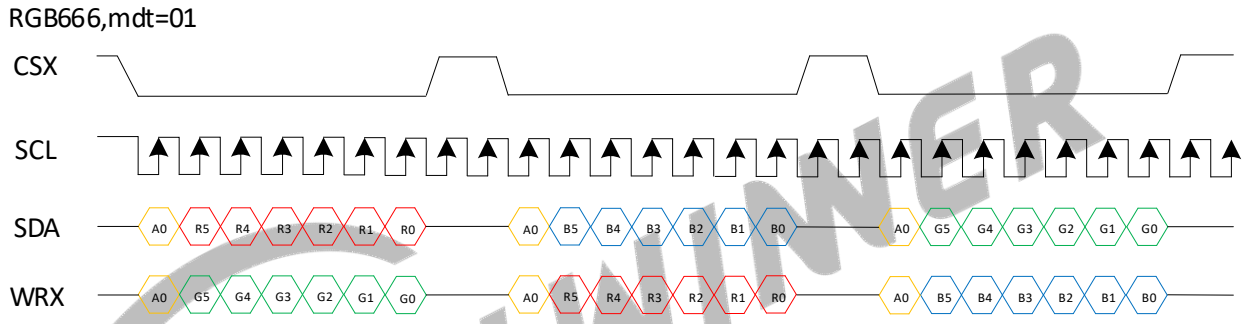
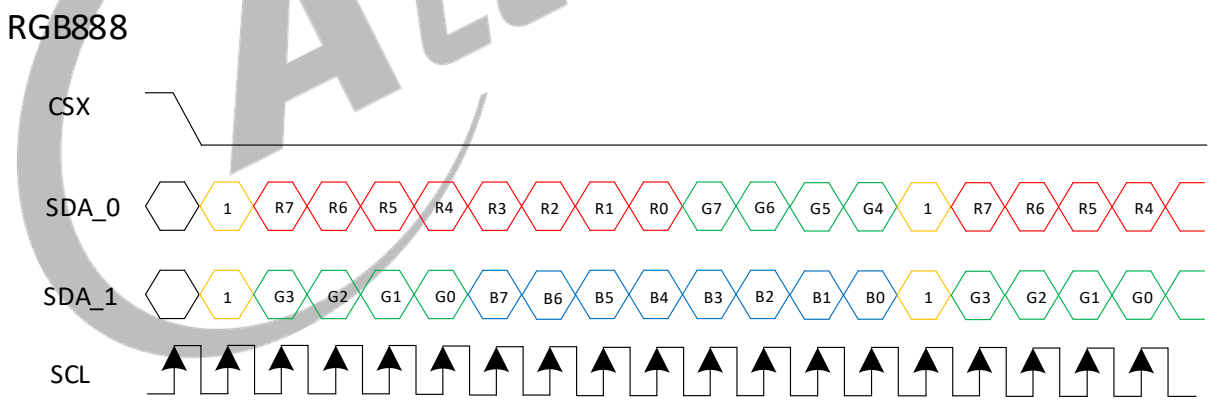


Figure 9-54 RGB888 2 Data Lane Interface Transmit Video Format



Note 1. Pixel data with 24-bit color information
Note 2. The most significant bits are: R7, G7 and B7
Note 3. The least significant bits are: R0, G0 and B0

9.5.4 Programming Guidelines

9.5.4.1 Writing/Reading Data Process in SPI Mode

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 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 a completed interrupt (SPI_ISR[12]) to the processor at the end of each transfer.

Figure 9-55 SPI Write/Read Data in CPU Mode

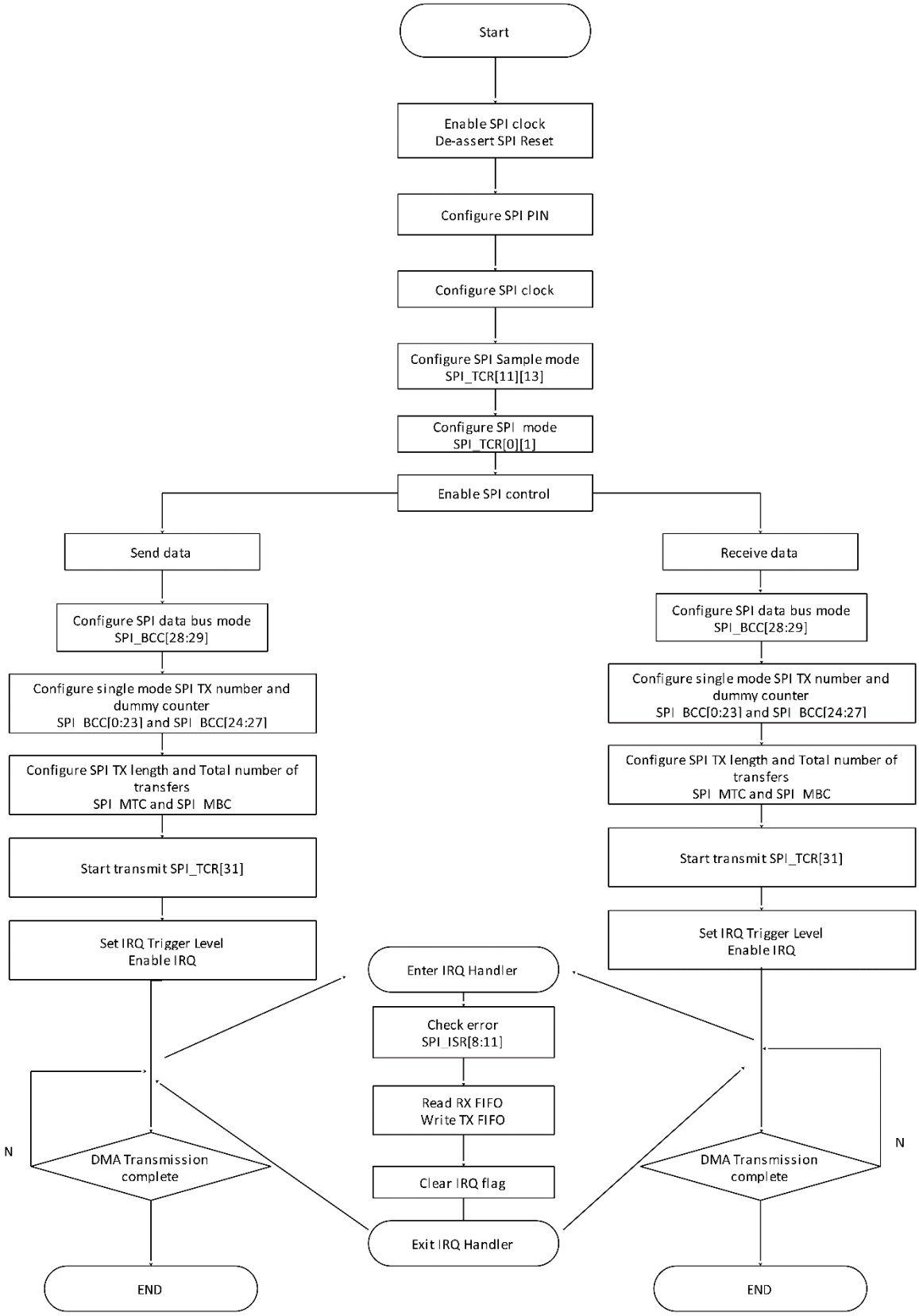
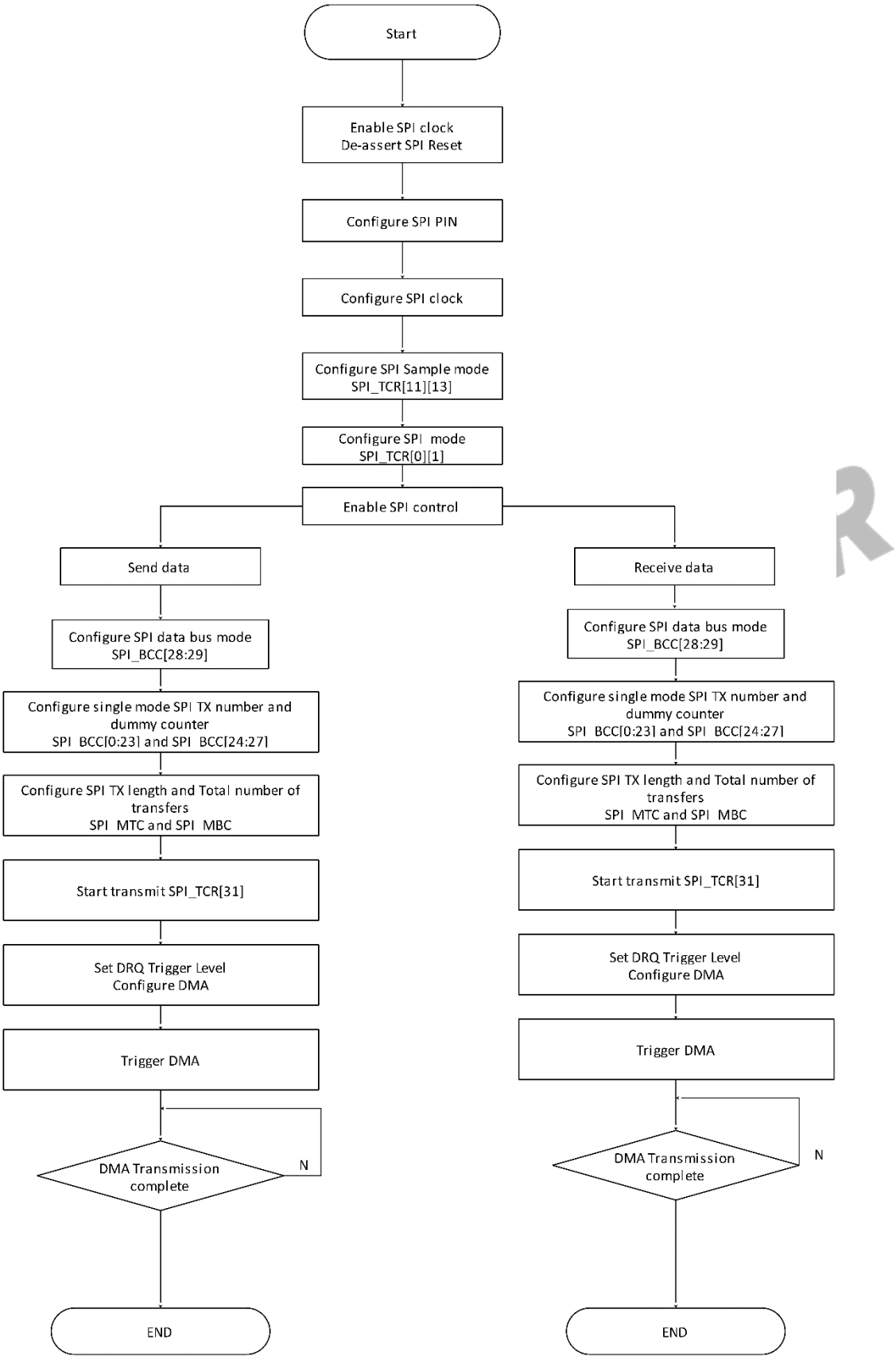


Figure 9-56 SPI Write/Read Data in DMA Mode



9.5.4.2 Calibrating Delay Chain in SPI Mode

There are one delay chains in SPI, 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:

Step 1 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.

Step 2 Configure a proper clock for SPI. Calibration delay chain is based on the clock for SPI from Clock Control Management Unit (CCMU)

Step 3 Set proper initial delay value. Write 0xA0 to the SPI_SAMP_DL to set initial delay value 0x20 to delay chain. Then write 0x0 to the SPI_SAMP_DL to clear this value.

Step 4 Write 0x8000 to the SPI_SAMP_DL to start calibrating delay chain.

Step 5 Wait until the flag (SPI_SAMP_DL [14]) of calibration done is set. The number of delay cells is shown at SPI_SAMP_DL [13:8]. 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.

Step 6 Calculate the delay time of one delay cell according to the cycle of SPI clock and the result of calibration.

9.5.4.3 Transmitting Write Command in DBI Mode

Step 1 Set the SPI_GCR[3] to 1 to select DBI mode.

Step 2 Set the DBI_CTL_1[30:29] to 0 to select the trigger mode of DBI.

Step 3 Configure the DBI_CTL_0.

- Set DBI_CTL_0[31] to 0 to configure the writing command.
- Set DBI_CTL_0[30:20] to configure the number of dummy cycles between commands.
- Set DBI_CTL_0[19] to select the MSB or LSB.
- Set DBI_CTL_0[15] to 0 to select the command path.
- Set DBI_CTL_0[14:12] to 0 to transmit the command.
- Set DBI_CTL_0[10:8] to select the DBI interface type.
- The remaining values of the DBI_CTL_0 remain the default value.

Step 4 Set DBI_CTL_1[22] to 0 to send the command.

Step 5 **DMA Path**: Configure the SPI_FCR.

- Set SPI_FCR[24] to 1 to enable TXFIFO DMA.

- Set SPI_FCR[23:16] to 255. It indicates the controller requests data from DMA if the remaining space of TX FIFO is greater than 255.

CPU Path: Write the command to be sent to the 0x200 address.

Step 6 Set SPI_GCR[4] to 1 to start transmitting the command.

Step 7 Wait until the TX FIFO underrun interrupt (SPI_ISR[11]) is 1. It indicates that the command written to the TX FIFO is transmitted completely.

9.5.4.4 Transmitting Parameter in DBI Mode

Step 1 Set the SPI_GCR[3] to 1 to select DBI mode.

Step 2 Set the DBI_CTL_1[30:29] to 0 to select the trigger mode of DBI.

Step 3 Configure the DBI_CTL_0.

- Set DBI_CTL_0[31] to 0 to configure the writing command.
- Set DBI_CTL_0[30:20] to configure the number of dummy cycles between commands.
- Set DBI_CTL_0[19] to select the MSB or LSB.
- Set DBI_CTL_0[15] to 0 to select the command path.
- Set DBI_CTL_0[14:12] to 0 to transmit the command.
- Set DBI_CTL_0[10:8] to select the DBI interface type.
- The remaining values of the DBI_CTL_0 remain the default value.

Step 4 Set DBI_CTL_1[22] to 1 to send the parameter.

Step 5 **DMA Path**: Configure the SPI_FCR.

- Set SPI_FCR[24] to 1 to enable TXFIFO DMA.
- Set SPI_FCR[23:16] to 255. It indicates the controller requests data from DMA if the remaining space of TX FIFO is greater than 255.

CPU Path: Write the command to be sent to the 0x0200 address.

Step 6 Set the SPI_GCR[4] to 1 to start transmitting the command.

Step 7 Wait until the TX FIFO underrun interrupt (SPI_ISR[11]) is 1. It indicates that the command written to the TX FIFO is transmitted completely.

9.5.4.5 Transmitting Video in DBI Mode

Set the SPI_GCR[3]to 1 to select DBI mode.

If the data is from the CPU path, the controller writes the command to be sent to the 0x0200 address by the AHB bus.

If the data is from the DMA path, configure DBI_CTL_1[15] to 1 and DBI_CTL_1[14:8] to 64, which indicates the controller requests data from DMA if the remaining space of TX FIFO is greater than 64.

Software Trigger

The software enables DBI_en_trigger when the edge interrupt of TE is detected.

After transmitting each frame image, the controller automatically clears the line_cnt, pixel_cnt and stops transmitting data.

Wait for the edge interrupt of TE, the software needs to enable DBI_en_trigger, in circulation.

The operation process is as follows.

Step 1 Set the SPI_GCR[3] to 1 to select DBI mode.

Step 2 Set the DBI_CTL_1[30:29] to 1 to select the software trigger mode.

Step 3 Configure the DBI_CTL_0.

- Set DBI_CTL_0[31] to 0 to set the writing command.
- Set DBI_CTL_0[30:20] to configure the number of dummy cycles between commands.
- Set DBI_CTL_0[19] to select the MSB or LSB.
- Set DBI_CTL_0[15] to 1 to select the image path.
- Set DBI_CTL_0[14:12] to select RGB111//444/565/666/888.
- Set DBI_CTL_0[10:8] to select the DBI interface type.
- The remaining values of the DBI_CTL_0 remain the default value.

Step 4 Set the DBI_CTL_1[22] to 0 to send the image data.

Step 5 Configure the DBI_Video_Size according to the sent image size.

Step 6 Configure the DBI_CTL_2 to set the TE-related parameter.

Step 7 Detect the TE interrupt of the DBI_INT.

Step 8 Configure the DBI_CTL_1[31] to 1.

Timer Trigger

The software configures timer_en to enable timer counting, and when the counter reaches the specified value, the DBI_EN automatically can be enabled to start transmitting data.

After transmitting each frame image, the controller clears automatically the line_cnt, pixel_cnt, and stops transmitting data.

The timer starts counting again. When the counter reaches the specified value, the controller automatically enables DBI_EN, and in circulation until the software turns off the timer_en.

The operation process is as follows.

Step 1 Set the SPI_GCR[3] to 1 to select DBI mode.

Step 2 Set the DBI_CTL_1[30:29] to 2 to select the timer trigger mode.

Step 3 Configure the DBI_CTL_0.

- Set DBI_CTL_0[31] to 0 to set the writing command.
- Set DBI_CTL_0[30:20] to configure the number of dummy cycles between commands.
- Set DBI_CTL_0[19] to select the MSB or LSB.
- Set DBI_CTL_0[15] to 1 to select the image path.
- Set DBI_CTL_0[14:12] to select RGB111/444/565/666/888.
- Set DBI_CTL_0[10:8] to select the DBI interface type.
- The remaining values of the DBI_CTL_0 remain the default value.

Step 4 Set the DBI_CTL_1[22] to 0 to send the image data.

Step 5 Configure the DBI_Video_Size to transmit the image size.

Step 6 Configure the related parameter of the DBI_Timer.

TE Trigger

When the edge changes of the TE are detected (The rising and falling edges are optional), the DBI_EN automatically can be enabled to start transmitting data.

After transmitting each frame image, the controller clears automatically the line_cnt, pixel_cnt, and stops transmitting data.

When the edge changes of the TE are detected (The rising and falling edges are optional), the DBI_EN automatically can be enabled to start transmitting data until the software shuts down TE_EN or the screen no longer sends TE signals.

The operation process is as follows.

Step 1 Set the SPI_GCR[3] to 1 to select DBI mode.

Step 2 Set the DBI_CTL_1[30:29] to 3 to select the TE trigger mode.

Step 3 Configure the DBI_CTL_0.

- Set DBI_CTL_0[31] to 0 to set the writing command.
- Set DBI_CTL_0[30:20] to configure the number of dummy cycles between commands.
- Set DBI_CTL_0[19] to select the MSB or LSB.
- Set DBI_CTL_0[15] to 1 to select the image path.
- Set DBI_CTL_0[14:12] to select RGB111/444/565/666/888.
- Set DBI_CTL_0[10:8] to select the DBI interface type.
- The remaining values of the DBI_CTL_0 remain the default value.

Step 4 Configure DBI_CTL_1[22] to 0 to send the image data.

Step 5 Configure DBI_Video_Size (0x0110) to transmit the image size.

Step 6 Configure DBI_CTL_2 (0x0108) to set the TE-related parameter.

9.5.4.6 Transmitting Read Command and Read Data in DBI Mode

Step 1 Set the SPI_GCR[3] to 1 to select DBI mode.

Step 2 Set the DBI_CTL_1[30:29] to 0.

Step 3 Configure the DBI_CTL_0.

- Set DBI_CTL_0[31] to 0 to set the reading command.
- Set DBI_CTL_0[19] to select the MSB or LSB.
- Set DBI_CTL_0[15] to 0 to select the command path.
- Set DBI_CTL_0[14:12] to 0.
- Set DBI_CTL_0[10:8] to select the DBI interface type.
- The remaining values of the DBI_CTL_0 remain the default value.

Step 4 Configure the DBI_CTL_1.

- Configure DBI_CTL_1[22] to 0 to send the command.
- Configure DBI_CTL_1[20] to select whether the first bit of the read data is the highest or lowest bit of data.
- Configure DBI_CTL_1[7: 0] to set the byte number to be read.
- Configure DBI_CTL_1[15:8] to set the dummy cycle between the read command and the read data, when the dummy cycle is complete, the data starts to be sampled.

Step 5 DMA Path: Configure the SPI_FCR.

- Set SPI_FCR[8] to 1 to enable RXFIFO DMA.
- Set SPI_FCR[7: 0] to 32, which indicates the controller requests receiving data from DMA if the data of the RX FIFO is greater than 64.

CPU Path: Read data in RX FIFO from the 0x0300 address.

Step 6 Set SPI_GCR[4] to 1 to start transmitting command.

Step 7 Wait until DBI_INT[11] is 1. It indicates that the data is read completely.

9.5.5 Register List

Module Name	Base Address
SPI1	0x4000F000

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

Register Name	Offset	Description
SPI_FCR	0x0018	SPI FIFO Control register
SPI_FSR	0x001C	SPI FIFO Status register
SPI_WCR	0x0020	SPI Wait Clock Counter 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 3Wire 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
DBI_CTL_0	0x0100	DBI Control Register 0
DBI_CTL_1	0x0104	DBI Control Register 1
DBI_CTL_2	0x0108	DBI Control Register 2
DBI_TIMER	0x010C	DBI Timer Control Register
DBI_VIDEO_SZIE	0x0110	DBI Video Size Configuration Register
DBI_INT	0x0120	DBI Interrupt Register
DBI_DEBUG_0	0x0124	DBI BEBUG 0 Register
DBI_DEBUG_1	0x0128	DBI BEBUG 1 Register
SPI_TXD	0x0200	SPI TX Data register
SPI_RXD	0x0300	SPI RX Data register

9.5.6 Register Description

9.5.6.1 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 Write '1' to this bit will clear the SPI controller, and auto clear to '0' when reset operation completes Write '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. 1: Stop transmit data when RXFIFO full 0: Normal operation, ignore RXFIFO status **Note**: *It cannot be written when XCH=1*
6:5	/	/	/

Offset: 0x0004			Register Name: SPI_GCR
Bit	**Read/Write**	**Default/Hex**	**Description**
4	R/W	0x0	DBI_EN DBI Module Enable Control 0: Disable 1: Enable
3	R/W	0x0	SPI_DBI_MODE_SEL 0: SPI MODE 1: DBI MODE
2	R/W	0x0	MODE_SELEC 0: Old mode of Sample Timing 1: New mode of Sample Timing **Note**: *It cannot be written when XCH=1*
1	R/W	0x0	MODE SPI Function Mode Select 0: Slave Mode 1: Master Mode **Note**: *It cannot be written when XCH=1*
0	R/W	0x0	EN SPI Module Enable Control 0: Disable 1: Enable **Note**: *After transforming from bit_mode to byte_mode, it must Enable the SPI Module again.*

9.5.6.2 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: Initiate exchange. Write “1” to this bit will start the SPI burst, and will auto clear after finishing the bursts transfer specified by BC. Write “1” to SRST will also clear this bit. Write ‘0’ to this bit has no effect. **Note**: *It cannot be written when XCH=1.*
30:16	/	/	/

Offset: 0x0008			Register Name: SPI_TCR
Bit	**Read/Write**	**Default/Hex**	**Description**
15	R/W	0x0	SDC1 Master Sample Data Control register1 Set this bit to '1' to make the internal read sample point with a delay of half cycle of SPI_CLK. It is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK propagating between master and slave. 0: Normal operation, do not delay internal read sample point 1: Delay internal read sample point **Note**: *It cannot be written when XCH=1.*
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 i/o mode for SPI mode 0.
13	R/W	0x0	SDM Master Sample Data Mode 1: Normal Sample Mode 0: Delay 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.
12	R/W	0x0	FBS First Transmit Bit Select 0: MSB first 1: LSB first **Note**: *It 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 **Note**: *It cannot be written when XCH=1.*

Offset: 0x0008			Register Name: SPI_TCR
Bit	**Read/Write**	**Default/Hex**	**Description**
10	R/W	0x0	RPSM Rapids mode select Select Rapids mode for high speed write. 0: Normal write mode 1: Rapids write mode **Note**: *It 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 **Note**: *It 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. **Note**: *It 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 **Note**: *It 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 **Note**: *It 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 **Note**: *It cannot be written when XCH=1.*

Offset: 0x0008			Register Name: SPI_TCR
Bit	**Read/Write**	**Default/Hex**	**Description**
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 **Note**: *It 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) **Note**: *It 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) **Note**: *It 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) **Note**: *It cannot be written when XCH=1.*

9.5.6.3 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 under run Interrupt Enable 0: Disable 1: Enable

Offset: 0x0010			Register Name: SPI_IER
Bit	**Read/Write**	**Default/Hex**	**Description**
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 under run Interrupt Enable 0: Disable 1: Enable
8	R/W	0x0	RF_OVF_INT_EN RX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
7	/	/	/
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

9.5.6.4 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 has been exchanged. In other condition, when set, this bit indicates that all the data in TXFIFO has been loaded in the Shift register, and the Shift register has shifted out all the bits. Writing 1 to this bit clears it. 0: Busy 1: Transfer Completed
11	R/W1C	0x0	TF_UDF TXFIFO under run 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 has overflowed.
7	/	/	/
6	R/W1C	0x0	TX_FULL TXFIFO Full This bit is set when the TXFIFO is full. Writing 1 to this bit clears it. 0: TXFIFO is not Full 1: TXFIFO is Full

Offset: 0x0014			Register Name: SPI_ISR
Bit	**Read/Write**	**Default/Hex**	**Description**
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. Where TX_WL is the water level of RXFIFO
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. Where RX_WL is the water level of RXFIFO.

9.5.6.5 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 Write ‘1’ to this bit will reset the control portion of the TX FIFO and auto clear to ‘0’ when completing reset operation, write to ‘0’ has no effect.

Offset: 0x0018			Register Name: SPI_FCR
Bit	**Read/Write**	**Default/Hex**	**Description**
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, write '1' to this bit will switch TX FIFO read and write function to AHB bus. This bit is used to test the TX FIFO, don't set in normal operation and don't set RF_TEST and TF_TEST at the same time.*
29:25	/	/	/
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 Write '1' to this bit will reset the control portion of the receiver FIFO, and auto clear to '0' when completing reset operation, write '0' to this bit has no effect.
14	R/W	0x0	RF_TEST RX Test Mode Enable 0: Disable 1: Enable **Note**: *In normal mode, RX FIFO can only be written by SPI controller, write '1' to this bit will switch RX FIFO read and write function to AHB bus. This bit is used to test the RX FIFO, don't set in normal operation and don't set RF_TEST and TF_TEST at the same time.*
13:9	/	/	/
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

9.5.6.6 0x001C SPI FIFO Status Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: SPI_FSR
Bit	**Read/Write**	**Default/Hex**	**Description**

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
27:26	/	/	/
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

9.5.6.7 0x0020 SPI Wait Clock Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: SPI_WCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/

Offset: 0x0020			Register Name: SPI_WCR
Bit	**Read/Write**	**Default/Hex**	**Description**
19:16	R/W	0x0	SWC Dual mode direction switch wait clock counter (for master mode only). 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 transmission in dual SPI mode. The SPI module counts SPI_SCLK by SWC for delaying next word data transmission. Cannot be written when XCH=1.*
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 transmission. The SPI module counts SPI_SCLK by WCC for delaying next word data transmission. 0: No wait states inserted N: N SPI_SCLK wait states inserted **Note**: *Cannot be written when XCH=1.*

9.5.6.8 0x0028 SPI Sample Delay Control Register (Default Value: 0x0000_2000)

Offset: 0x0028			Register Name: SPI_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	/	/	/

Offset: 0x0028			Register Name: SPI_SAMP_DL
Bit	**Read/Write**	**Default/Hex**	**Description**
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.

9.5.6.9 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 **Note**: *It cannot be written when XCH=1; Total transfer data, include the TXD, RXD and dummy burst.*

9.5.6.10 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 **Note**: *It cannot be written when XCH=1.*

9.5.6.11 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 **Note**: *It cannot be written when XCH=1. 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 **Note**: *It 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 is don't care by the device. 0: 0 burst 1: 1 burst … N: N bursts **Note**: *It 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 **Note**: *It cannot be written when XCH=1.*

9.5.6.12 0x0040 SPI Bit-Aligned Transfer Configure Register (Default Value: 0x0000_00A0)

Offset: 0x0040			Register Name: SPI_BATC
Bit	**Read/Write**	**Default/Hex**	**Description**

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 t1o transfer the serial bits frame, it is only valid when *Work Mode Select==0x10/0x11.* 0: Idle 1: Initiates transfer. Write "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. Write '0' to this bit has no effect.
30	R/W	0x0	MSMS Master Sample Standard 0 - Standard Sample Mode 1 - Delay 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 **Note**: *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 **Note**: *It is only valid when Work Mode Select==0x10/0x11.*
23:22	/	/	/
21:16	R/W	0x00	RX_FEM_LEN Configure the length of serial data frame(burst) of RX 000000: 0bit 000001: 1bit … 100000: 32bits Other values: Reserved **Note**: *It is only valid when Work Mode Select==0x10/0x11, and can't be written when TCE=1.*

Offset: 0x0040			Register Name: SPI_BATC
Bit	**Read/Write**	**Default/Hex**	**Description**
15:14	/	/	/
13:8	R/W	0x00	TX_FEM_LEN Configure the length of serial data frame(burst) of TX 000000: 0bit 000001: 1bit … 100000: 32bits Other values: Reserved **Note**: *It is only valid when Work Mode Select==0x10/0x11, and can't 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 **Note**: *It is only valid when Work Mode Select==0x10/0x11, and only work in Mode0, can't 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 **Note**: *It is only valid when Work Mode Select==0x10/0x11, and only work in Mode0, can't 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) **Note**: *It is only valid when Work Mode Select==0x10/0x11, and only work in Mode0, can't 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 **Note**: *It is only valid when Work Mode Select= =0x10/0x11, and only work in Mode0, can't be written when TCE=1.*

Offset: 0x0040			Register Name: SPI_BATC
Bit	Read/Write	Default/Hex	Description
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: Reserve 10: Data frame is bit aligned in 3-Wire SPI 11: Data frame is bit aligned in Standard SPI

9.5.6.13 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.

9.5.6.14 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. **Note**: *In the process of transmission, the LSB is transmitted first.*

NOTE

This register is only valid when Work Mode Select==0x10/0x11.

9.5.6.15 0x004C SPI RX Bit Register (Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: SPI_RBR
Bit	Read/Write	Default/Hex	Description

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. **Note**: *In the process of transmission, the LSB is transmitted first.*

NOTE

This register is only valid when Work Mode Select==0x10/0x11.

9.5.6.16 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	SPI_ACT_M SPI NDMA Active Mode 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	SPI_ACK_M SPI NDMA Acknowledge Mode 0: Active fall do not care ack 1: Active fall must after detect ack is high
4: 0	R/W	0x5	SPI_DMA_WAIT Delay Cycles The counts of hold cycles from DMA last signal high to dma_active high

9.5.6.17 0x0100 DBI Control Register 0 (Default Value: 0x0010_0000)

Offset: 0x0100			Register Name: DBI_CTL_0
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	CMDT Command Type 0: Write Command 1: Read Command

Offset: 0x0100			Register Name: DBI_CTL_0
Bit	**Read/Write**	**Default/Hex**	**Description**
30:20	R/W	0x1	WCDC Write Command Dummy Cycles Controls dummy cycles between two write commands Range 1~255 Default Condition: There is a dbi_clk cycle between each command or parameter.
19	R/W	0x0	DAT_SEQ Output Data Sequence 0: MSB First 1: LSB First
18:16	R/W	0x0	RGB_SEQ Output RGB Sequence 0: RGB 1: RBG 2: GRB 3: GBR 4: BRG 5: BGR 6~7: Reserve
15	R/W	0x0	TRAN_MOD Transmit Mode 0: Command / Parameter 1: Video
14:12	R/W	0x0	DAT_FMT Output Data Format 000: RGB111 001: RGB444 010: RGB565 011: RGB666 100: RGB888 (only for 2 Data Lane Interface) 101~111: Reserved
11	/	/	/
10:8	R/W	0x0	DBI Interface 000: 3 Line Interface I 001: 3 Line Interface II 010: 4 Line Interface I 011: 4 Line Interface II 100: 2 Data Lane Interface

Offset: 0x0100			Register Name: DBI_CTL_0
Bit	**Read/Write**	**Default/Hex**	**Description**
7:4	R/W	0x0	RGB_Source_Format When video_source_type is RGB32 (DBI_CTL_0[bit0] = 0) 0000: RGB 0001: RBG 0010: GRB 0011: GBR 0100: BRG 0101: BGR Others: Reserved When video_source_type is RGB16 (DBI_CTL_0[bit0] = 1) 0000: RGB 0001~0100: Reserved 0101: BGR 0110: GRBG_0 {G[5:3] R[4: 0] B[4: 0] G[2: 0]} 0111: GBRG_0 {G[5:3] B[4: 0] R[4: 0] G[2: 0]} 1000: GRBG_1 {G[2: 0] R[4: 0] B[4: 0] G[5:3]} 1001: GBRG_1 {G[2: 0] B[4: 0] R[4: 0] G[5:3]} Others: Reserved
3	R/W	0x0	DUM_VAL Dummy Cycle Value Output Value During Dummy Cycle
2	R/W	0x0	RGB_BO RGB Bit Order 0: Remain the sequence of RGB data 1: Swap the higher bit and the lower bit for each component of DRAM RGB
1	R/W	0x0	ELEMENT_A_POS Element A Position Only for RGB32 Data Format 0: A component is in the bit[31:24] of data source 1: A component is in the bit[7: 0] of data source
0	R/W	0x0	VI_SRC_TYPE Video Source Type 0: RGB32 1: RGB16

9.5.6.18 0x0104 DBI Control Register 1 (Default Value: 0x0000_0001)

Offset: 0x0104			Register Name: DBI_CTL_1
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0104			Register Name: DBI_CTL_1
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/WAC	0x0	DBI_SOFT_TRG DBI soft trigger It is only available for software trigger mode. Write '1' to this bit will start DBI TX module and auto clear to '0' when completing start operation, write to '0' has no effect.
30:29	R/W	0x0	DBI_EN_MODE_SEL DBI Enable Mode Select 00: Always on DBI mode 01: Software trigger mode 10: Timer trigger mode 11: TE trigger mode
28	/	/	/
27:26	R/W	0x0	RGB666_FMT 2 Data Lane RGB666 Format 00: Normal Format 01: Special Format for ILITEK 10: Special Format for New Vision
25	R/W	0x0	DBI_RXCLK_INV DBI rx clock inverse 0: Sample data by using the positive edge of the output clock 1: Sample data by using the negative edge of the output clock
24	R/W	0x0	DBI_CLKO_MOD DBI output clock mode 0: DBI clock always on (DCX Setup/hold equals one clock cycle) 1: DBI clock auto gating (DCX Setup/hold equals to a half clock cycle)
23	R/W	0x0	DBI_CLKO_INV DBI clock output inverse **When the bit24 (DBI output clock mode) is 0.** 0: The falling edge releases the CSX signal, and the falling edge releases data 1: The rising edge releases the CSX signal, and the rising edge releases data **When the bit24 (DBI output clock mode) is 1.** 0: The rising edge releases the CSX signal, and the falling edge releases data 1: The falling edge releases the CSX signal, and the rising edge releases data
22	R/W	0x0	DCX_DATA DCX Data Value 0: DCX Value equal to 0 1: DCX Value equal to 1

Offset: 0x0104			Register Name: DBI_CTL_1
Bit	**Read/Write**	**Default/Hex**	**Description**
21	R/W	0x0	RGB 16 Data Source Select 0: Pixel1 is stored in the higher bit of address, and Pixel0 is stored in the lower bit of address 1: Pixel0 is stored in the higher bit of address, and Pixel1 is stored in the lower bit of address
20	R/W	0x0	Read_MSB_First 0: A reading data is the higher bit 1: A reading data is the lower bit
19:16	/	/	/
15:8	R/W	0x0	RCDC Read Command Dummy Cycles The dummy cycle between the read command and read data. Reading 1-byte (8 bits) data has not dummy cycle.
7: 0	R/W	0x1	RDBN Read Data Number of Bytes Sample Bytes data based on configuration.

9.5.6.19 0x0108 DBI Control Register 2 (Default Value: 0x0000_4000)

Offset: 0x0108			Register Name: DBI_CTL_2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15	R/W	0x0	DBI_FIFO_ DRQ_EN DBI FIFO DMA Request Enable 0: Disable 1: Enable
14:8	R/W	0x40	DBI_TRIG_LEVEL DBI FIFO Empty Request Trigger Level
7	/	/	/
6	R/W	0x0	DBI_SDI_OUT_SEL DBI SDI PIN Output Select The signal is used with the DBI SDI PIN Function Sel bit. 0: Output WRX (When DBI DCX PIN Function Sel = 0, the SDI pin outputs data) 1: Output DCX
5	R/W	0x0	DBI_DCX_SEL DBI DCX PIN Function Select 0: DBI DCX Function 1: WRX (2 Data Lane Interface)

Offset: 0x0108			Register Name: DBI_CTL_2
Bit	**Read/Write**	**Default/Hex**	**Description**
4:3	R/W	0x0	DBI_SDI_SEL DBI SDI PIN Function Select 00: DBI_SDI (Interface II) 01: DBI_TE 10: DBI_DCX 11: Reserved
2	R/W	0x0	TE_DBC_SEL TE debounce function select 0: debounce 1: No-debounce
1	R/W	0x0	TE_TRIG_SEL TE edge trigger select 0: TE rising edge 1: TE falling edge
0	R/W	0x0	TE_EN TE enable 0: TE Disable 1: TE Enable

9.5.6.20 0x010C DBI Timer Register (Default Value: 0x0000_0000)

Offset: 0x010C			Register Name: DBI_TIMER
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	DBI_TM_EN DBI Timer Enable 0: Enable 1: Disable
30: 0	R/W	0x0	DBI Timer Value It sets the time interval between sending data twice, which is frame blanking. It is used to set the time at which the interrupt of the DBI Timer is triggered. When the Timer_EN is 1, the timer starts to count (the clock of the counting is SCLK), and the counter reaches the target value to trigger the Timer_INT of DBI, the data will start to send in series. **Note**: *Do not count when sending the series data.*

9.5.6.21 0x0110 DBI Video Size Register (Default Value: 0x01E0_0140)

Offset: 0x0110			Register Name: DBI_VIDEO_SIZE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:27	/	/	/

Offset: 0x0110			Register Name: DBI_VIDEO_SIZE
Bit	**Read/Write**	**Default/Hex**	**Description**
26:16	R/W	0x1e0	V_SIZE It is used to generate the Frame int.
15:11	/	/	/
10: 0	R/W	0x140	H_SIZE It is used to generate the Line int.

9.5.6.22 0x0120 DBI Interrupt Register (Default Value: 0x0000_4000)

Offset: 0x0120			Register Name: DBI_INT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:15	/	/	/
14	W1C	0x1	DBI_FIFO_EMPTY_INT 0: DBI_FIFO is not empty 1: DBI_FIFO is empty
13	W1C	0x0	DBI_FIFO_FULL_INT 0: DBI_FIFO is not full 1: DBI_FIFO is full
12	W1C	0x0	TIMER_INT it indicates that timer has been count sclk cycles to the value of DBI_Timer Register[30: 0]. Writing 1 to this bit clears it. 0: Timer has not been achieve objective 1: Timer has been achieve objective
11	W1C	0x0	RD_DONE_INT it indicates that the number of byte setting in DBI_Control Register 1 [19:8] has been read. Writing 1 to this bit clears it. 0: All of data has been not read 1: All of data has been read
10	W1C	0x0	TE_INT it indicates that TE signal has been changed. Writing 1 to this bit clears it. 0: TE signal is no changed 1: TE signal has been changed
9	W1C	0x0	FRAM_DONE_INT it indicates that a frame video data has been send. Writing 1 to this bit clears it. 0: A frame video has not been send 1: A frame video has been send
8	W1C	0x0	LINE_DONE_INT it indicates that a line of video data has been send. Writing 1 to this bit clears it. 0: A line of video data has not been send 1: A line of video data has been send

Offset: 0x0120			Register Name: DBI_INT
Bit	**Read/Write**	**Default/Hex**	**Description**
7	/	/	/
6	R/W	0x0	DBI_FIFO_EMPTY_INT_EN 0: Disable 1: Enable
5	R/W	0x0	DBI_FIFO_FULL_INT_EN 0: Disable 1: Enable
4	R/W	0x0	TIMER_INT_EN Timer Interrupt Enable 0: Disable 1: Enable
3	R/W	0x0	RD_DONE_INT_EN Read Done Interrupt Enable 0: Disable 1: Enable
2	R/W	0x0	TE_INT_EN TE Interrupt Enable 0: Disable 1: Enable
1	R/W	0x0	FRAM_DONE_INT_EN Frame Done Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	LINE_DONE_INT_EN Line Done Interrupt Enable 0: Disable 1: Enable

9.5.6.23 0x0124 DBI Debug Register 0 (Default Value: 0x007F_0000)

Offset: 0x0124			Register Name: DBI_DEBUG_0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
22:16	R	0x7F	DBI_FIFO ROOM VALID 0~127 Words
15:13	/	/	/
12	R	0x0	TE input 0: TE not Trigger 1: TE Trigger
11:8	R	0x0	dbi_rxcs RX_BS0 ~ RX_BS6 , Gray - Code

Offset: 0x0124			Register Name: DBI_DEBUG_0
Bit	**Read/Write**	**Default/Hex**	**Description**
7:4	R	0x0	sh_cs 0~11 : SH0~SH11
3:2	/	/	dbi_txcs 00: IDLE 01: SHIF 10: DUMY 11: Read
1: 0	R	0x0	tx_mem_cs 00: IDLE_FRM 01: FRM_POS 10: FRM_RDY

9.5.6.24 0x0128 DBI Debug Register 1 (Default Value: 0x0000_0000)

Offset: 0x0128			Register Name: DBI_DEBUG_1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25:16	R	0x0	line_cnt The number of pixel lines that are currently sent
15:12	/	/	/
11: 0	R	0x0	component_cnt The number of RGB components that are currently sent The field is equal to pixel_cnt *3.

9.5.6.25 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 increase by 2. In word accessing method, four SPI burst data are written and the TXFIFO depth is increased by 4. **Note**: *This address is writing-only if TF_TEST is '0', and if TF_TEST is set to '1', this address is readable and writable to test the TX FIFO through the AHB bus.*

9.5.6.26 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 decrease by 2. In word accessing method, the four SPI bursts are returned and the RXFIFO depth is decreased by 4. **Note**: *This address is read-only if RF_TEST is '0', and if RF_TEST is set to '1', this address is readable and writable to test the RX FIFO through the AHB bus.*

9.6 USB2.0 DRD

9.6.1 Overview

The USB2.0 dual-role device (USB2.0 DRD) supports both device and host functions which can also be configured as a Host-only or Device-only controller. It complies with the USB2.0 Specification.

For saving CPU bandwidth, the DMA interface of the DRD module can also support the external DMA controller to do the data transmission between the memory and the DRD FIFO. The DRD core also supports USB power saving functions.

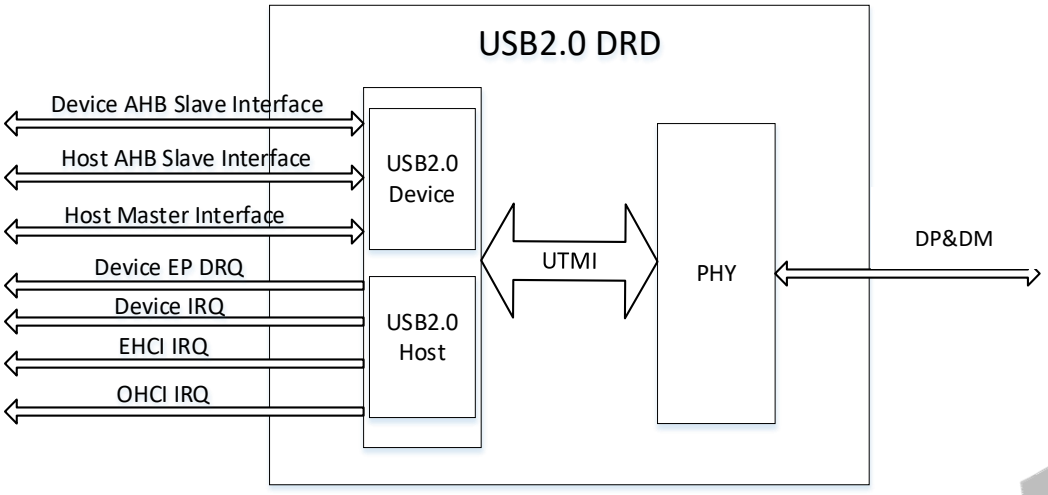
The USB2.0 DRD includes the following features:

- Complies with USB 2.0 Specification
- Supports High-Speed (HS, 480-Mbps), Full-Speed (FS, 12-Mbps), and Low-Speed (LS, 1.5-Mbps) in Host mode
- Supports High-Speed (HS, 480-Mbps), Full-Speed (FS, 12-Mbps) in Device mode
- Supports the UTMI+ Level 3 interface. The 8-bit bidirectional data buses are used
- Supports bi-directional endpoint0 for Control transfer
- Supports up to 10 User-Configurable Endpoints for Bulk, Isochronous and Interrupt bi-directional transfers (endpoint1_in/out, endpoint2_in/out, endpoint3_in/out, endpoint4_in/out, endpoint5_in/out)
- Supports up to (8 KB+64 Bytes) FIFO for 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

9.6.2 Block Diagram

The following figure shows the block diagram of USB2.0 DRD Controller.

Figure 9-57 USB2.0 DRD Controller Block Diagram



9.6.3 Functional Descriptions

9.6.3.1 External Signals

Table 9-18 USB2.0 OTG External Signals

Signal	Description	Type
VCC33-USB	Power Supply of USB	P
USB_DP	USB Bus Data	AI/O
USB_DM	USB Bus Data	AI/O

9.6.3.2 Clock and Reset

Figure 9-58 USB2.0_DRD Clock Description

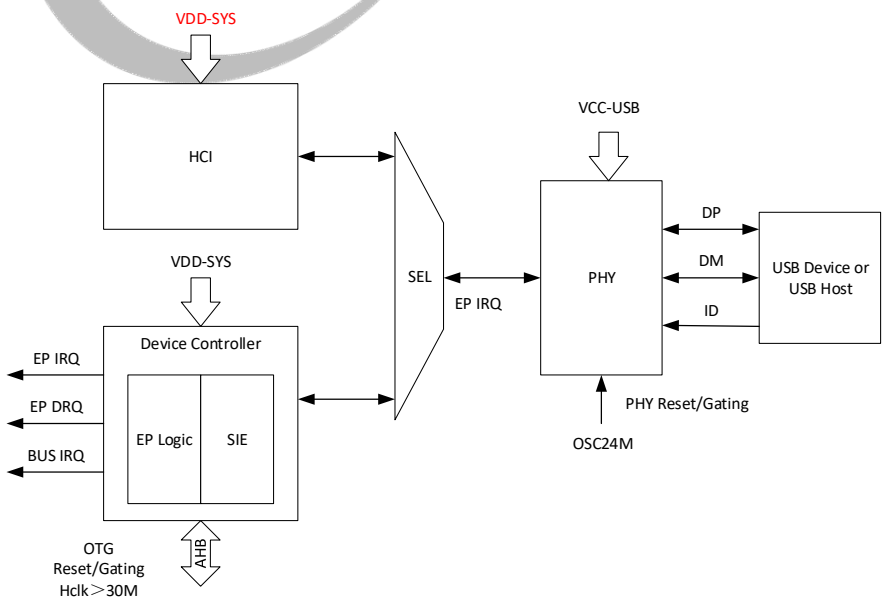


Table 9-19 USB_DRD Clock Sources

Clock Sources	Description
CCU_AON	High frequency crystal clock for system (optional frequency: 24M/24.576M/26M/32M/40M)
OSC40M	General-purpose high frequency crystal clock for system interfaces (frequency range: 0~392M)
32KCLK	The working clock of Device
CCU	The working closk of Host OHCI controller

9.6.4 USB_DRD_Device Register List

Module Name	Base Address
USB_DRD_Device	0x40B42000

Register Name	Offset	Description
USB_EPFIFOn	0x0000+N*0004	FIFO Entry for Endpoint N (N=0,1,2,3,4)
USB_GCS	0x0040	USB Global Control and Status Register
USB_EPINTF	0x0044	USB Endpoint Interrupt Flag Register
USB_EPINTF	0x0044	USB Endpoint Interrupt Flag Register
USB_EPINTE	0x0048	USB Endpoint Interrupt Enable Register
USB_BUSINTF	0x004C	USB Bus Interrupt Flag Register
USB_BUSINTE	0x0050	USB Bus Interrupt Enable Register
USB_FNUM	0x0054	USB Frame Number Register
USB_CSR0	0x0080	USB EP0 Control and Status Register
USB_TXCSR	0x0080	USB EP1~4 Tx Control and Status Register
USB_RXCSR	0x0084	USB EP1~4 Rx Control and Status Register
USB_COUNT0	0x0088	USB EP0 Rx Counter Register
USB_RXCOUNT	0x0088	USB EP1~4 Rx Counter Register
USB_ATTR0	0x008C	USB EP0 Attribute Register
USB_EPATTR	0x008C	USB EP1~4 Attribute Register
USB_TXFIFO	0x0090	USB EP1~4 Tx FIFO Setting Register
USB_RXFIFO	0x0094	USB EP1~4 Rx FIFO Setting Register
USB_FADDR	0x0098	USB Function Address Register
USB_ISCR	0x0400	USB Interface Status and Control Register
USB_PHYCTL	0x0410	USB PHY Control Register
USB_PHY_SEL	0x0420	USB PHY Select Register
PHY_STA	0x0424	USB PHY Status Register
USB_DMA_INTE	0x0500	USB DMA Interrupt Enable Register
USB_DMA_INTS	0x0504	USB DMA Interrupt Status Register
USB_DMA_CHAN_CFG	0x0540+N*0x10 (N=0~7)	USB DMA Channel Configuration Register
USB_DMA_SDRAM_AD	0x0544+N*0x10	USB DMA SDRAM Start Address Register

Register Name	Offset	Description
D	(N=0~7)	
USB_DMA_BC	0x0548+N*0x10 (N=0~7)	USB DMA Byte Counter Register
USB_DMA_RESIDUAL_BC	0x054C+N*0x10 (N=0~7)	USB DMA RESIDUAL Byte Counter Register

9.6.5 USB_DRD_Host Register List

Module Name	Base Address
USB_DRD_Host	0x40B43000

Register Name	Offset	Description
EHCI Capability Register		
E_CAPLENGTH	0x0000	EHCI Capability register Length 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_PERIODICLISTBASE	0x0024	EHCI Frame List Base Address Register
E_ASYNCLISTADDR	0x0028	EHCI Next Asynchronous List Address Register
E_CONFIGFLAG	0x0050	EHCI Configured Flag Register
E_PORTSC	0x0054	EHCI Port Status/Control Register
OHCI Control and Status Partition 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 Register
O_HcPeriodCurrentED	0x041C	OHCI Period Current ED Register
O_HcControlHeadED	0x0420	OHCI Control Head ED Register
O_HcControlCurrentED	0x0424	OHCI Control Current ED Register
O_HcBulkHeadED	0x0428	OHCI Bulk Head ED Register
O_HcBulkCurrentED	0x042C	OHCI Bulk Current ED Register
O_HcDoneHead	0x0430	OHCI Done Head Register
OHCI Frame Counter Partition Register		
O_HcFmInterval	0x0434	OHCI Frame Interval Register
O_HcFmRemaining	0x0438	OHCI Frame Remaining Register

Register Name	Offset	Description
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_CTRL1	0x0800	HCI Control 1 Register
HCI_CTRL2	0x0808	HCI Control 2 Register
PHY Control	0x0810	PHY Control Register
HCI SIE Port Disable Control	0x0828	HCI SIE Port Disable Control Register

9.6.6 USB_DRD_Device Register Description

9.6.6.1 0x0000+N*0x0004(N=0~4) USB FIFO Entry for Endpoint N (Default Value: Undefined)

Offset: 0x0000+N*0x0004(N=0~4)			Register Name: USB_EPFIFOn
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	UDF	EPnFIFO FIFO Entry for Endpoint n

9.6.6.2 0x0040 USB Global Control and Status Register (for Peripheral Mode) (Default Value: 0x0000_0020)

Offset: 0x0040			Register Name: USB_GCS_P
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	TX_EDMA 0: DMA_REQ signal for all IN Endpoints will be de-asserted when MAXP bytes have been written to and endpoint. This is late mode. 1: DMA_REQ signal for all IN Endpoints will be de-asserted when MAXP-8 bytes have been written to an endpoint. This is early mode.
30	R/W	0x0	RX_EDMA 0: DMA_REQ signal for all OUT Endpoints will be de-asserted when MAXP bytes have been read to an endpoint. This is late mode. 1: DMA_REQ signal for all OUT Endpoints will be de-asserted when MAXP-8 bytes have been read to and endpoint. This is early mode.
29	/	/	/

<table>
<tr><th colspan="3">Offset: 0x0040</th><th>Register Name: USB_GCS_P</th></tr>
<tr><th>Bit</th><th>Read/Write</th><th>Default/Hex</th><th>Description</th></tr>
<tr><td>28:25</td><td>R/W</td><td>0x0</td><td>BUS_DRQ_SEL
USB DMA Request Signal Source Select
0000: Select TX Endpoint 1 DRQ
0001: Select RX Endpoint 1 DRQ
0010: Select TX Endpoint 2 DRQ
0011: Select RX Endpoint 2 DRQ
0100: Select TX Endpoint 3 DRQ
0101: Select RX Endpoint 3 DRQ
0110: Select TX Endpoint 4 DRQ
0111: Select RX Endpoint 4 DRQ</td></tr>
<tr><td>24</td><td>R/W</td><td>0x0</td><td>FIFO_BUS_SEL
0: CPU bus for FIFO Access,
1: DMA bus for FIFO operation.</td></tr>
<tr><td>23:20</td><td>/</td><td>/</td><td>/</td></tr>
<tr><td>19:16</td><td>R/W</td><td>0x0</td><td>EPIND
Endpoint Index
Index is a 4-bit register that determines which endpoint control/status registers are accessed. Before accessing an endpoint’s control/status registers at 0x0080~0x00BF, the endpoint number should be written to the Index register to ensure that correct control/status registers in the memory map.
Note: The valid value for Index register is 0-5.</td></tr>
<tr><td>15</td><td>R</td><td>0x0</td><td>BDev
B-Device
0 => ‘A’ device;
1 => ‘B’ device;
Only valid while a session is in progress.
Note: If the core is in Force_Host mode (i.e. a session has been started with USB_TMCTL.7=1), this bit will indicate the state of the HOSTDISCON input signal from the PHY.</td></tr>
<tr><td>14:13</td><td>/</td><td>/</td><td>/</td></tr>
<tr><td>12:11</td><td>R</td><td>0x0</td><td>VBus
These bits encode the current VBus level as follows:
<table>
<tr><td>D4</td><td>D3</td><td>Meaning</td></tr>
<tr><td>0</td><td>0</td><td>Below SessionEnd</td></tr>
<tr><td>0</td><td>1</td><td>Above SessionEnd, below AValid</td></tr>
<tr><td>1</td><td>0</td><td>Above AValid, below VBusValid</td></tr>
<tr><td>1</td><td>1</td><td>Above VBusValid</td></tr>
</table></td></tr>
<tr><td>10</td><td>R</td><td>0x0</td><td>HostMode
Host Mode
This bit is set when the USB/DRD is acting as a Host.</td></tr>
</table>

Offset: 0x0040			Register Name: USB_GCS_P
Bit	**Read/Write**	**Default/Hex**	**Description**
9	/	/	/
8	R/W	0x0	Session *When operating as an 'A' device*, this bit is set or cleared by the CPU to start or end a session. *When operating as an 'B' device*, this bit is set/cleared by the USB/DRD when a session starts/ends. It is also set by the CPU to initiate the Session Request Protocol. When the USB/DRD is in Suspend mode, the bit may be cleared by the CPU to perform a software disconnect. **Note**: *Clearing this bit when the core is not suspending will result in undefined behavior.*
7	R/W	0x0	IsoUpdateEn Isochronous Update Enable When set by the CPU, the USB/DRD will wait for an SOF token from the Tx packet ready before sending the packet. If an IN token is received before an SOF token, then a zero length data packet will be send. **Note**: *This bit only affects endpoints performing Isochronous transfer.*
6	R/W	0x0	Soft Connect The USB D+/D- line is enabled when this bit is set by CPU and tri-stated when this bit is cleared by CPU. **Note**: *Only valid in Peripheral Mode (but not means 'B' Device).*
5	R/W	0x1	HSEN High-speed Mode Enable When set by CPU, the USB/DRD will negotiate for High-speed mode when the device is reset by host. If not set, the device will only operate in Full-speed mode.
4	R	0x0	HSFLAG High-speed Mode Flag When set, this read-only bit indicates High-speed mode successfully negotiated during USB reset. And this bit becomes valid when USB Reset completes (as indicated by USB reset interrupt).
3	R	0x0	Reset This bit is set when Reset Signaling is present on the bus.
2	R/W	0x0	Resume Set by the CPU to generate Resume signaling when the function is in Suspend mode. The CPU should clear this bit after 10 ms (a maximum of 15 ms) to end Resume signaling.
1	R	0x0	SuspendM Suspend Mode This bit is set on entry into Suspend mode.

Offset: 0x0040			Register Name: USB_GCS_P
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	SuspendMEn Enable SuspendM Set by the CPU to enable the SUSPENDM output of UTMI+ bus.

9.6.6.3 0x0044 USB Endpoint Interrupt Flag Register (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: USB_EPINTF
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20	R	0x0	EP4Rx Rx Endpoint 4 interrupt flag
19	R	0x0	EP3Rx Rx Endpoint 3 interrupt flag
18	R	0x0	EP2Rx Rx Endpoint 2 interrupt flag
17	R	0x0	EP1Rx Rx Endpoint 1 interrupt flag
16:5	/	/	/
4	R	0x0	EP4Tx Tx Endpoint 4 interrupt flag
3	R	0x0	EP3Tx Tx Endpoint 3 interrupt flag
2	R	0x0	EP2Tx Tx Endpoint 2 interrupt flag
1	R	0x0	EP1Tx Tx Endpoint 1 interrupt flag
0	R	0x0	EP0 Endpoint 0 interrupt flag

9.6.6.4 0x0048 USB Endpoint Interrupt Enable Register (Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: USB_EPINTE
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20	R/W	0x0	EP4Rx Rx Endpoint 4 interrupt enable
19	R/W	0x0	EP3Rx Rx Endpoint 3 interrupt enable
18	R/W	0x0	EP2Rx Rx Endpoint 2 interrupt enable
17	R/W	0x0	EP1Rx Rx Endpoint 1 interrupt enable

Offset: 0x0048			Register Name: USB_EPINTE
Bit	**Read/Write**	**Default/Hex**	**Description**
16:5	/	/	/
4	R/W	0x0	EP4Tx Tx Endpoint 4 interrupt enable
3	R/W	0x0	EP3Tx Tx Endpoint 3 interrupt enable
2	R/W	0x0	EP2Tx Tx Endpoint 2 interrupt enable
1	R/W	0x0	EP1Tx Tx Endpoint 1 interrupt enable
0	R/W	0x0	EP0 Endpoint 0 interrupt enable

9.6.6.5 0x004C USB Bus Interrupt Flag Register (Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: USB_BUSINTF
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R	0x0	VBusError Set when VBus drops below the VBus Valid threshold during a session. **Note**: *Only valid when USB/DRD is 'A' device.*
6	R	0x0	SessionRequest Set when Session Request signaling has been detected. **Note**: *Only valid when USB/DRD is 'A' device.*
5	R	0x0	Disconnect Set in Host mode when a device disconnect is detected. Set in Peripheral mode when a session ends. **Note**: *Valid at all transaction speeds.*
4	R	0x0	Connect Set in host mode when a device connection is detected. **Note**: *Only valid in Host mode. Valid at all transaction speeds.*
3	R	0x0	SOF Set when a new frame starts.
2	R	0x0	ResetBabble Reset Set in Peripheral mode when Reset signaling is detected on the bus. Babble Set in Host mode when babble is detected. **Note**: *Only active after first SOF has been sent.*

Offset: 0x004C			Register Name: USB_BUSINTF
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R	0x0	Resume Set when Resume signaling is detected on the bus while the USB/DRD is in Suspend mode.
0	R	0x0	Suspend Set when Suspend signaling is detected on the bus. **Note**: *Only valid in Peripheral mode.*

9.6.6.6 0x0050 USB Bus Interrupt Enable Register (Default Value: 0x0000_0000)

Offset: 0x0050			Register Name: USB_BUSINTE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	VBusError VBusError interrupt enable
6	R/W	0x0	Session Request Session Request interrupt enable
5	R/W	0x0	Disconnect Disconnect interrupt enable
4	R/W	0x0	Connect Connect interrupt enable
3	R/W	0x0	SOF SOF interrupt enable
2	R/W	0x0	ResetBabble Reset Reset interrupt enable Babble Babble interrupt enable
1	R/W	0x0	Resume Resume interrupt enable
0	R/W	0x0	Suspend Suspend interrupt enable

9.6.6.7 0x0054 USB Frame Number Register (Default Value: 0x0000_0000)

Offset: 0x0054			Register Name: USB_FRNUM
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10: 0	R	0x0	FRNUM Frame Number Hold the last received frame number.

9.6.6.8 0x0080 USB EP0 Control and Status Register (for EP0 in Peripheral Mode) (Default Value: 0x0000_0000)

Offset: 0x0080			Register Name: USB_CSR0_P
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	W	0x0	FlushFIFO The CPU writes a '1' to this bit to flush the next packet to be transmitted/read from the Endpoint 0 FIFO. The FIFO pointer is reset, and the TxPktRdy/RxPktRdy bit (below) is cleared. **Note***: Writing '0' to this bit is ignored. Flush FIFO should only be used when TxPktRdy/RxPktRdy is set, at other times, it may cause data to be corrupted.*
23	W	0x0	ServicedSetupEnd The CPU writes a '1' to this bit to clear the SetupEnd bit. It is cleared automatically.
22	W	0x0	ServicedRxPktRdy The CPU writes a 1 to this bit to clear the RxPktRdy bit. It is cleared automatically.
21	W	0x0	SendStall The CPU writes a '1' to this bit to terminate the current transaction. The STALL handshake will be transmitted and then this bit will be cleared automatically. **Note**: The *FIFO should be flushed before SendStall is set.*
20	R	0x0	SetupEnd This bit will be set when a control transaction ends before the DataEnd bit has been set. An interrupt will be generated and the FIFO flushed at this time. The bit is cleared by the CPU writing a '1' to the ServicedSetupEnd bit.
19	W	0x0	DataEnd The CPU sets this bit: When setting TxPktRdy for the last data packet. When clearing RxPktRdy after unloading the last data packet. When setting TxPktRdy for a zero length data packet. It is cleared automatically.
18	R/W	0x0	SentStall This bit is set when a STALL handshake is transmitted. The CPU should clear this bit.
17	R/W	0x0	TxPktRdy The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is also generated at this point (if enabled).

Offset: 0x0080			Register Name: USB_CSR0_P
Bit	**Read/Write**	**Default/Hex**	**Description**
16	R	0x0	RxPktRdy This bit is set when a data packet has been received. An interrupt is generated (if enabled) when this bit is set. The CPU clears this bit by setting the ServicedRxPktRdy bit.
15: 0	/	/	/

9.6.6.9 0x0080 USB EP1~4 Tx Control and Status Register (for Peripheral Mode) (Default Value: 0x0000_0000)

Offset: 0x0080			Register Name: USB_TXCSR_P
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	AutoSet If CPU sets this bit, TxPktRdy will be automatically set when data of maximum packet size (value in the USB_TXMAXP) is loaded into the Tx FIFO. If a packet of less than the maximum packet size is loaded, then TxPktRdy will have to be set manually. **Note**: *Should not be set for high-bandwidth Isochronous/Interrupt endpoints.*
30	R/W	0x0	ISO The CPU sets this bit to enable the Tx endpoint for Isochronous transfers, and clears it to enable the Tx endpoint for Bulk or Interrupt transfers. **Note**: *This is only effective in Peripheral mode. In Host mode, it always returns zero.*
29	R/W	0x0	Mode The CPU sets this bit to enable the endpoint direction as Tx, and clears the bit to enable it as Rx. **Note**: *This bit only has any affect where the same endpoint FIFO is used for Tx and Rx transactions.*
28	R/W	0x0	DMAReqEnab The CPU sets this bit to enable the DMA request for the Tx endpoint.
27	R/W	0x0	FrcDataTog The CPU sets this bit to force the endpoint data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an ACK was received. This can be used by Interrupt Tx endpoints that are used to communicate rate feedback for Isochronous endpoints.
26	R/W	0x0	DMAReqMode The CPU sets this bit to select DMA Request Mode 1 and clears it to select DMA Request Mode 0. **Note**: *This bit must not be cleared either before or in the same cycle as the above DMAReqEnab bit is cleared.*

Offset: 0x0080			Register Name: USB_TXCSR_P
Bit	**Read/Write**	**Default/Hex**	**Description**
25:24	/	/	/
23	R/W	0x0	IncompTx When the endpoint is being used for high-bandwidth Isochronous/Interrupt transfers, this bit is set to indicate where a large packet has been split into 2 or 3 packets for transmission but insufficient IN tokens have been received to send all the parts. **Note**: *In anything other than a high-bandwidth transfer, this bit will always return 0. And writing '1' to this bit is ignored.*
22	W	0x0	ClrDataTog The CPU writes a 1 to this bit to reset the endpoint data toggle to 0. It is cleared automatically. **Note**: *Writing '0' to this bit is ignored.*
21	R/W	0x0	SentStall This bit is set when a STALL handshake is transmitted. The FIFO is flushed and the TxPktRdy bit is cleared. The CPU should clear this bit. **Note**: *Writing '1' to this bit is ignored.*
20	R/W	0x0	SendStall The CPU writes a 1 to this to issue a STALL handshake to an IN token. The CPU clears this bit to terminate the stall condition. **Note***: The FIFO should be flushed before SendStall is set. This bit has no effect where the endpoint is being used for Isochronous transfers.*
19	W	0x0	FlushFIFO The CPU writes a 1 to this bit to flush the latest packet from the endpoint Tx FIFO. The FIFO pointer is reset, the TxPktRdy bit (below) is cleared and an interrupt is generated. May be set simultaneously with TxPktRdy to abort the packet that is currently being loaded into the FIFO. **Note***: Writing '0' to this bit is ignored. Flush FIFO should only be used when TxPktRdy is set, at other times, it may cause data to be corrupted. Also note that, if the FIFO is double-buffered, FlushFIFO may need to be set twice to completely clear the FIFO.*
18	R/W	0x0	UnderRun The USB sets this bit if an IN token is received when the TxPktRdy bit not set. The CPU should clear this bit. **Note**: *Writing '1' to this bit is ignored.*
17	R/W	0x0	FIFONotEmpty The USB sets this bit when there is at least 1 packet in the Tx FIFO. **Note**: *Writing '1' to this bit is ignored.*

Offset: 0x0080			Register Name: USB_TXCSR_P
Bit	**Read/Write**	**Default/Hex**	**Description**
16	R/W	0x0	TxPktRdy The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is also generated at this point (if enabled). TxPktRdy is also automatically cleared prior to loading a second packet to a double buffered FIFO. **Note**: *Writing '0' to this bit is ignored.*
15:11	R/W	0x0	PacketCount In the case of Bulk endpoints with the packet splitting option enabled, the Packet Count can be up to 32 and defines the maximum number of USB packets of specified payload into which a single data packet placed in the FIFO should be split, prior to transfer. If the packet splitting option is not enabled, Packet Count is not implemented. For Isochronous/Interrupt endpoints operating in High-speed mode and with the High-bandwidth option enabled, Packet Count may only either 2 or 3 (corresponding to bit 11 or bit 12 set, respectively) and it specifies the maximum number of such transactions that can be take place in a single microframe. If either bit 11 or bit 12 is non-zero, the USB/DRD will automatically split any data packet written to the FIFO into up to 2 or 3 USB packet, each containing the specified payload (or less). For Isochronous/Interrupt transfers in Full-speed mode or if High-bandwidth is not enabled, bit 11 and 12 are ignored. **Note**: *Value for this bits is (Packet Count - 1), but not Packet Count.*
10: 0	R/W	0x0	MaximumPayload These bits define the maximum payload (in bytes) transmitted in a single transaction. The value set can be up to 1024 bytes but is subject to the constraints placed by the USB Specification on packet sizes for Bulk, Interrupt and Isochronous transfers in Full-speed and High-speed operations. **Note***:* *The value written to Maximum Payload (multiplied by Packet Count in the case of high-bandwidth Isochronous/Interrupt transfer) must match the value given in the wMaxPacketSize field of the Standard Endpoint Descriptor for the associated endpoint, a mismatch could cause unexpected result.* *The total amount of data represented by the value written to this register (Maximum payload × Packet Count) must not exceed the FIFO size for Tx endpoint, and should not exceed half the FIFO size if double-buffering is required.*

9.6.6.10 0x0084 USB EP1~4 Rx Control and Status Register (for Peripheral Mode) (Default Value: 0x0000_0000)

Offset: 0x0084			Register Name: USB_RXCSR_P
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	AutoClear If CPU sets this bit then the RxPktRdy will be automatically cleared when data of maximum packet size (value in the USB_TXMAXP) is unloaded from the Rx FIFO. If a packet of less than the maximum packet size is unloaded, then RxPktRdy will have to be cleared manually. **Note**: *Should not be set for high-bandwidth Isochronous* endpoints.
30	R/W	0x0	ISO The CPU sets this bit to enable the Rx endpoint for Isochronous transfers, and clears it to enable the Rx endpoint for Bulk or Interrupt transfers.
29	R/W	0x0	DMAReqEnab The CPU sets this bit to enable the DMA request for the Rx endpoint.
28	R/W	0x0	DisNyet_PIDError DisNyet *Bulk/Interrupt Transactions: The* CPU sets this bit to disable the sending of NYET handshakes. When set, all successfully received Rx packets are ACK'd including at the point at which the FIFO becomes full. **Note**: *This bit only has any affect in High-speed mode, in which mode it should be set for all Interrupt endpoints.* PIDError ISO Transactions: The core sets this bit to indicate a PID error in the received packet.
27	R/W	0x0	DMAReqMode The CPU sets this bit to select DMA Request Mode 1 and clears it to select DMA Request Mode 0. **Note**: *This bit must not be cleared in the same cycle as the above RxPktRdy(or DMAReqEnab) bit is cleared.*
26:25	/	/	/
24	R/W	0x0	IncompRx This bit will be set in a high-bandwidth Isochronous/Interrupt transfer if the packet received is incomplete. It will be cleared when RxPktRdy is cleared. **Note***:* *Writing '1' to this bit is forbidden. In anything other than a high-bandwidth transfer, this bit will always return 0.*

Offset: 0x0084			Register Name: USB_RXCSR_P
Bit	**Read/Write**	**Default/Hex**	**Description**
23	W	0x0	ClrDataTog The CPU writes a '1' to this bit to reset the endpoint data toggle to 0. It is cleared automatically. **Note**: *Writing '0' to this bit is ignored.*
22	R/W	0x0	SentStall This bit is set when a STALL handshake is transmitted. The CPU should clear this bit. **Note**: *Writing '1' to this bit is ignored.*
21	R/W	0x0	SendStall The CPU writes a '1' to this to issue a STALL handshake. The CPU clears this bit to terminate the stall condition. **Note***:* *The FIFO should be flushed before SendStall is set. This bit has no effect where the endpoint is being used for Isochronous transfers.*
20	W	0x0	FlushFIFO The CPU writes a '1' to this bit to flush the next packet to be read from the endpoint Rx FIFO. The FIFO pointer is reset and the RxPktRdy bit (below) is cleared. **Note***: Writing '0' to this bit is ignored. Flush FIFO should only be used when RxPktRdy is set, at other times, it may cause data to be corrupted. If the FIFO is double-buffered, FlushFIFO may need to be set twice to completely clear the FIFO.*
19	R	0x0	DataError This bit is set when RxPktRdy is set if the data packet has a CRC or bit-stuff error. It is cleared when RxPktRdy is cleared. **Note**: *This bit is only valid when the endpoint is operating in ISO mode. In bulk mode, it always returns zero.*
18	R/W	0x0	OverRun The USB sets this bit if an OUT token cannot be loaded into the Rx FIFO. The CPU should clear this bit. **Note**: *Writing '1' to this bit is ignored. This bit is only valid when the endpoint is operating in ISO mode. In bulk mode, it always returns zero.*
17	R	0x0	FIFOFull The USB sets this bit when no more packets can be loaded into the Rx FIFO.
16	R/W	0x0	RxPktRdy This bit is set when a data packet has been received. The CPU should clear this bit when the packet has been unloaded from the Rx FIFO. An interrupt is generated when the bit is set. **Note**: *Writing '1' to this bit is ignored.*

Offset: 0x0084			Register Name: USB_RXCSR_P
Bit	**Read/Write**	**Default/Hex**	**Description**
15:11	R/W	0x0	PacketCount In the case of Bulk endpoints with the packet combining option enabled, the Packet Count can be up to 32 and defines the maximum number of USB packets of specified payload which are to be combined into a single data packet within the FIFO. For Isochronous/Interrupt endpoints operating in High-speed mode and with the High-bandwidth option enabled, Packet Count may only either 2 or 3 (corresponding to bit 11 or bit 12 set, respectively) and it specifies the maximum number of such transactions that can be take place in a single microframe. If either bit 11 or bit 12 is non-zero, the USB/DRD will automatically combine the separate USB packets received in any microframe into a single packet within the Rx FIFO. For Isochronous/Interrupt transfers in Full-speed mode or if High-bandwidth is not enabled, bit 11 and 12 are ignored. **Note**: *Value for this bits is (Packet Count - 1), but not Packet Count.*
10: 0	R/W	0x0	MaximumPayload These bits define the maximum payload (in bytes) transmitted in a single transaction. The value set can be up to 1024 bytes but is subject to the constraints placed by the USB Specification on packet sizes for Bulk, Interrupt and Isochronous transfers in Full-speed and High-speed operations. **Note***:* *The value written to Maximum Payload (multiplied by Packet Count in the case of high-bandwidth Isochronous/Interrupt transfer) must match the value given in the wMaxPacketSize field of the Standard Endpoint Descriptor for the associated endpoint, a mismatch could cause unexpected result.* *The total amount of data represented by the value written to this register (Maximum payload × Packet Count) must not exceed the FIFO size for OUT endpoint, and should not exceed half the FIFO size if double-buffering is required.*

9.6.6.11 0x0088 USB EP0 Rx Counter Register (Default Value: 0x0000_0000)

Offset: 0x0088			Register Name: USB_COUNT0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/

Offset: 0x0088			Register Name: USB_COUNT0
Bit	**Read/Write**	**Default/Hex**	**Description**
6: 0	R	0x0	RxCount0 Endpoint 0 Rx Count These bits indicate the number of received data bytes in the Endpoint 0 FIFO. **Note**: The *value returned changes as the FIFO is unloaded and is only valid while RxPktRdy (of USB_CSR0) is set.*

9.6.6.12 0x0088 USB EP1~4 Rx Counter Register (Default Value: 0x0000_0000)

Offset: 0x0088			Register Name: USB_RXCOUNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	RqPktCount Sets the number of packets of size MaxP that are to be transferred in a block transfer. **Note**: *Only used in Host mode when AutoReq (of USB_RXCSR) is set. Has no effect in Peripheral mode or AutoReq is not set.*
15:13	/	/	/
12: 0	R	0x0	RxCount Endpoint Rx Count These bits hold the number of data bytes in the packet currently in line to be read from the Rx FIFO. If the packet was transmitted as multiple bulk packets, the number given will be for the combined packet. **Note**: The *value returned changes as the FIFO is unloaded and is only valid while RxPktRdy (of USB_RXCSR) is set.*

9.6.6.13 0x0090 USB EP1~4 TxFIFO Setting Register (Default Value: 0x0000_0000)

Offset: 0x0090			Register Name: USB_TXFIFO
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28:16	R/W	0x0	AD AD[12: 0] Start address of the endpoint FIFO is in units of 8 bytes, and it equals (AD[12: 0]*8).
15:5	/	/	/
4	R/W	0x0	DPB Define whether double-packet buffering supported. When '1', double-packet buffering is supported. When '0', only single-packet buffering is supported.

Offset: 0x0090			Register Name: USB_TXFIFO
Bit	**Read/Write**	**Default/Hex**	**Description**
3: 0	R/W	0x0	SZ SZ[3: 0] Maximum packet size to be allowed for (*before any splitting within the FIFO of Bulk/High-Bandwidth packets prior to transmission*) is $2^{(SZ[3:0]+3)}$ bytes, and the valid values for SZ[3: 0] are 0x0–0x09. If DPB=0, the FIFO will also this size; if DPB=1, the FIFO will be twice this size.

9.6.6.14 0x0094 USB EP1~4 RxFIFO Setting Register (Default Value: 0x0000_0000)

Offset: 0x0094			Register Name: USB_RXFIFO
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28:16	R/W	0x0	AD AD[12: 0] Start address of the endpoint FIFO is in units of 8 bytes, and it equals (AD[12: 0]*8).
15:5	/	/	/
4	R/W	0x0	DPB Defines whether double-packet buffering supported. When '1', double-packet buffering is supported. When '0', only single-packet buffering is supported.
3: 0	R/W	0x0	SZ SZ[3: 0] Maximum packet size to be allowed for (*before any splitting within the FIFO of Bulk/High-Bandwidth packets prior to transmission*) is $2^{(SZ[3:0]+3)}$ bytes, and the valid values for SZ[3: 0] are 0x0–0x09. If DPB=0, the FIFO will also this size; if DPB=1, the FIFO will be twice this size.

9.6.6.15 0x0098 USB Function Address Register (Peripheral Mode) (Default Value: 0x0000_0000)

Offset: 0x0098			Register Name: USB_FADDR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6: 0	R/W	0x0	FADDR The function address in peripheral mode. This field is reset to zero after a USB bus reset, and should be updated by software after Set_Address Command during USB enumeration.

9.6.6.16 0x0400 USB Interface Status and Control Register (Default Value: 0x0000_0000)

Offset: 0x0400			Register Name: USB_ISCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	0x0	/
30	R	UDF	VBUSLS USB VBUS Valid Status detected from Line State
29	R	UDF	VBUSEX USB VBUS Valid Status detected from external VBUS input
28	R	UDF	IDEX USB ID Status detected from external ID input
27:26	R	UDF	LS USB Line Status [27]: DM [26]: DP
25	R	UDF	VBUS USB VBUS Status merged from both internal and external
24	R	UDF	ID USB ID Status merged from both internal and external
23:18	/	0x0	/
17	R/W	0x0	IDPullupEn ID pull up enable 0: Disable 1: Enable ID pull up
16	R/W	0x0	DataPullupEn DP/DM pull up enable 0: Disable DP/DM pull up 1: Enable DP/DM pull up
15:14	R/W	0x0	ForceID Force ID 0x: Use external ID Status 10: Force ID to LOW 11: Force ID to HIGH
13:12	R/W	0x0	ForceVBUS Force VBUS Valid 0x: Use external VBUS Valid Status from VBUS Input or Line State 10: Force VBUS Valid to LOW 11: Force VBUS Valid to HIGH
11:10	R/W	0x0	VBUSSEL External VBUS Valid Source Select 0x: External VBUS Valid detected from VBUS Input 10: External VBUS Valid detected from DP/DM Input 11: External VBUS Valid detected from either VBUS or DP/DM input

Offset: 0x0400			Register Name: USB_ISCR
Bit	**Read/Write**	**Default/Hex**	**Description**
9:8	/	0x0	/
7	R/W	0x0	WakeupEn USB Wakeup Enable 0: Disable 1: Enable
6	R/W	0x0	VBUSCDS VBUS Input Change Detect Status This bit is set by hardware after VBUS input changed when VBUS change detect is enable. Writing '1' will clear this bit.
5	R/W	0x0	IDCDS ID Input Change Detect Status This bit is set by hardware after ID input changed when ID change detect is enable. Writing '1' will clear this bit.
4	R/W	0x0	DATACDS DP/DM Input Change Detect Status This bit is set by hardware after DP/DM input changed when DP/DM change detect is enable. Writing '1' will clear this bit.
3	R/W	0x0	WakeupIE USB Wakeup IRQ Enable 1: Enable USB Wakeup IRQ 0: Disable USB Wakeup IRQ If this bit is set to zero, an USB wakeup event (VBUS/ID/DP/DM change) will generate an USB wakeup request to wake up the system, but not generate an USB wakeup IRQ to CPU.
2	R/W	0x0	VBUSCDE VBUS Input Change Detect enable 0: Disable 1: Enable
1	R/W	0x0	IDCDE ID Input Change Detect enable 0: Disable 1: Enable
0	R/W	0x0	DATACDE DP/DM Input Change Detect enable 0: Disable 1: Enable

9.6.6.17 0x0410 USB PHY Control Register (Default Value: 0x0000_0002)

Offset: 0x0410			Register Name: PHY_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/

Offset: 0x0410			Register Name: PHY_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
16	R/W	0x0	bist_en_a
15:8	R/W	0x0	vc_addr
7	R/W	0x0	vc_di
6:2	/	/	/
1	R/W	0x1	SIDDQ 1: Write 1 to disable phy. 0: Write 0 to enable phy.
0	R/W	0x0	vc_clk

9.6.6.18 0x0420 USB PHY Select Register (Default Value: 0x0000_0000)

Offset: 0x0420			Register Name: PHY_SEL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	OTG_SEL 1: Phy is connected to OTG SIE. 0: Phy is connected to HCI SIE.

9.6.6.19 0x0424 USB PHY Status Register (Default Value: 0x0000_0000)

Offset: 0x0424			Register Name: PHY_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R	0x0	bist_error
16	R	0x0	bist_done
15:1	/	/	/
0	R	0x0	vc_do

9.6.6.20 0x0500 USB DMA Interrupt Enable Register (Default Value: 0x0000_0000)

Offset: 0x0500			Register Name: USB_DMA_INTE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	Reserved
7	R/W	0x0	USB_DMA7_PKG_INT_EN DMA7 Package End Transfer Interrupt Enable 0: Disable 1: Enable
6	R/W	0x0	USB_DMA6_PKG_INT_EN DMA6 Package End Transfer Interrupt Enable 0: Disable 1: Enable

Offset: 0x0500			Register Name: USB_DMA_INTE
Bit	**Read/Write**	**Default/Hex**	**Description**
5	R/W	0x0	USB_DMA5_PKG_INT_EN DMA5 Package End Transfer Interrupt Enable 0: Disable 1: Enable
4	R/W	0x0	USB_DMA4_PKG_INT_EN DMA4 Package End Transfer Interrupt Enable 0: Disable 1: Enable
3	R/W	0x0	USB_DMA3_PKG_INT_EN DMA3 Package End Transfer Interrupt Enable 0: Disable 1: Enable
2	R/W	0x0	USB_DMA2_PKG_INT_EN DMA2 Package End Transfer Interrupt Enable 0: Disable 1: Enable
1	R/W	0x0	USB_DMA1_PKG_INT_EN DMA1 Package End Transfer Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	USB_DMA0_PKG_INT_EN DMA0 Package End Transfer Interrupt Enable 0: Disable 1: Enable

9.6.6.21 0x0504 USB DMA Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0504			Register Name: USB_DMA_INTS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	Reserved
7	R/W	0x0	USB_DMA7_PKG_INT_STA DMA7 Package End Transfer Interrupt Status. Set 1 to the bit will clean it. 0: No effect, 1: Pending.
6	R/W	0x0	USB_DMA6_PKG_INT_STA DMA6 Package End Transfer Interrupt Status. Set 1 to the bit will clean it. 0: No effect 1: Pending

Offset: 0x0504			Register Name: USB_DMA_INTS
Bit	**Read/Write**	**Default/Hex**	**Description**
5	R/W	0x0	USB_DMA5_PKG_INT_STA DMA5 Package End Transfer Interrupt Status. Set 1 to the bit will clean it. 0: No effect 1: Pending
4	R/W	0x0	USB_DMA4_PKG_INT_STA DMA4 Package End Transfer Interrupt Status. Set 1 to the bit will clean it. 0: No effect 1: Pending
3	R/W	0x0	USB_DMA3_PKG_INT_STA DMA3 Package End Transfer Interrupt Status. Set 1 to the bit will clean it. 0: No effect 1: Pending
2	R/W	0x0	USB_DMA2_PKG_INT_STA DMA2 Package End Transfer Interrupt Status. Set 1 to the bit will clean it. 0: No effect, 1: Pending.
1	R/W	0x0	USB_DMA1_PKG_INT_STA DMA1 Package End Transfer Interrupt Status. Set 1 to the bit will clean it. 0: No effect 1: Pending
0	R/W	0x0	USB_DMA0_PKG_INT_STA DMA0 Package End Transfer Interrupt Status. Set 1 to the bit will clean it. 0: No effect 1: Pending

9.6.6.22 0x0540+N*0x10 (N=0~7) USB DMA Channel Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0540+N*0x10 (N=0~7)			Register Name: USB_DMA_CHAN_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	DMA_EN DMA Channel Enable 1: DMA will start the data transmission between the source and the destination. The bit will hold on until the DMA finished. It will be cleared automatically. 0: Stop the corresponding DMA channel and reset its state machine.
30:27	/	/	/

Offset: 0x0540+N*0x10 (N=0~7)			Register Name: USB_DMA_CHAN_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
26:16	R/W	0x0	DMA_BST_LEN DMA Burst Length The value setting on this field should be equated to the usb max packet length of the corresponding endpoint.
15:5	/	/	/
4	R/W	0x0	DMA_DIR DMA Transfer Direction 0: SDRAM to USB FIFO 1: USB FIFO to SDRAM
3: 0	R/W	0x0	DMA_FOR_EP DMA Channel for Endpoint The Endpoint number setting on this field selects the dma channel for the corresponding endpoint.

9.6.6.23 0x0544+N*0x10 (N=0~7) USB DMA SDRAM Start Address Register (Default Value: 0x0000_0000)

Offset: 0x0544+N*0x10 (N=0~7)			Register Name: USB_DMA_SDRAM_ADD
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	DMA_SDRAM_STR_ADDR DMA SDRAM Start Address The sdram start address for the dma channel transfer between the sdram and usb fifo.

9.6.6.24 0x0548+N*0x10 (N=0~7) USB DMA Byte Counter Register (Default Value: 0x0000_0000)

Offset: 0x0548+N*0x10 (N=0~7)			Register Name: USB_DMA_BC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17: 0	R/W	0x0	DMA_BC DMA Byte Counter

9.6.6.25 0x054C+N*0x10 (N=0~7) USB DMA RESIDUAL Byte Counter Register (Default Value: 0x0000_0000)

Offset: 0x054C+N*0x10 (N=0~7)			Register Name: USB_DMA_RESIDUAL_BC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17: 0	R/W	0x0	DMA_RESIDUAL_BC DMA Residual Byte Counter This field contains the residual byte count in current transfer.

9.6.7 USB_DRD_Host Register Description

9.6.7.1 EHCI Register Description

0x0000 EHCI Identification Register (Default Value: 0x0000_0010)

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.

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'.

<table>
<tr><th colspan="3">Offset: 0x0004</th><th>Register Name: HCSPARAMS</th></tr>
<tr><th>Bit</th><th>Read/Write</th><th>Default/Hex</th><th>Description</th></tr>
<tr><td>7</td><td>R</td><td>0x0</td><td>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:
<table><tr><td>Value</td><td>Meaning</td></tr><tr><td>0</td><td>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.</td></tr><tr><td>1</td><td>The port routing is explicitly enumerated by the first N_PORTS elements of the HCSP-PORTTOUTE array.</td></tr></table>
This field will always be ‘0’.</td></tr>
<tr><td>6:4</td><td>/</td><td>/</td><td>/</td></tr>
<tr><td>3: 0</td><td>R</td><td>0x1</td><td>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.</td></tr>
</table>

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	0xa0	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’.

Offset: 0x0008			Register Name: HCCPARAMS
Bit	**Read/Write**	**Default/Hex**	**Description**
7:4	R	0x2	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	/	/	/
2	R	0x1	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	0x1	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	/	/	/

0x000C EHCI Companion Port Route Description (Default Value: 0x0000_0000)

Offset: 0x000C			Register Name: HCSP-PORTROUTE
Bit	**Read/Write**	**Default/Hex**	**Description**

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 of the companion host controllers this port is routed. Only the first N_PORTS elements have valid information. A value of zero indicates that the port is routed to the lowest numbered function companion host controller. A value of one indicates that the port is routed to the next lowest numbered function companion host controller, and so on.

0x0010 EHCI USB Command Register (Default Value: 0x0008_0000)

<table>
<tr><td colspan="3">Offset: 0x0010</td><td>Register Name: USBCMD</td></tr>
<tr><td>Bit</td><td>Read/Write</td><td>Default/Hex</td><td>Description</td></tr>
<tr><td>31:24</td><td>/</td><td>/</td><td>/</td></tr>
<tr><td>23:16</td><td>R/W</td><td>0x8</td><td>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:
<table>
<tr><td>Value</td><td>Minimum Interrupt Interval</td></tr>
<tr><td>0x00</td><td>Reserved</td></tr>
<tr><td>0x01</td><td>1 micro-frame</td></tr>
<tr><td>0x02</td><td>2 micro-frame</td></tr>
<tr><td>0x04</td><td>4 micro-frame</td></tr>
<tr><td>0x08</td><td>8 micro-frame(default, equates to 1 ms)</td></tr>
<tr><td>0x10</td><td>16 micro-frame(2 ms)</td></tr>
<tr><td>0x20</td><td>32 micro-frame(4 ms)</td></tr>
<tr><td>0x40</td><td>64 micro-frame(8 ms)</td></tr>
</table>
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.</td></tr>
<tr><td>15:12</td><td>/</td><td>/</td><td>/</td></tr>
</table>

Offset: 0x0010			Register Name: USBCMD
Bit	**Read/Write**	**Default/Hex**	**Description**
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 is safe for software to re-initialize the host controller. A host software read of this bit as a one indicates the Light Host

<table>
<tr><th colspan="3">Offset: 0x0010</th><th>Register Name: USBCMD</th></tr>
<tr><th>Bit</th><th>Read/Write</th><th>Default/Hex</th><th>Description</th></tr>
<tr><td>6</td><td>R/W</td><td>0x0</td><td>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.</td></tr>
<tr><td>5</td><td>R/W</td><td>0x0</td><td>Asynchronous Schedule Enable
This bit controls whether the host controller skips processing the Asynchronous Schedule. Values mean:
<table><tr><td>Bit Value</td><td>Meaning</td></tr><tr><td>0</td><td>Do not process the Asynchronous Schedule.</td></tr><tr><td>1</td><td>Use the ASYNLISTADDR register to access the Asynchronous Schedule.</td></tr></table>The default value of this field is ‘0b’.</td></tr>
<tr><td>4</td><td>R/W</td><td>0x0</td><td>Periodic Schedule Enable
This bit controls whether the host controller skips processing the Periodic Schedule. Values mean:
<table><tr><td>Bit Value</td><td>Meaning</td></tr><tr><td>0</td><td>Do not process the Periodic Schedule.</td></tr><tr><td>1</td><td>Use the PERIODICLISTBASE register to access the Periodic Schedule.</td></tr></table>The default value of this field is ‘0b’.</td></tr>
</table>

<table>
<tr><th colspan="3">Offset: 0x0010</th><th>Register Name: USBCMD</th></tr>
<tr><th>Bit</th><th>Read/Write</th><th>Default/Hex</th><th>Description</th></tr>
<tr><td>3:2</td><td>R/W</td><td>0x0</td><td>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:
<table>
<tr><td>Bits</td><td>Meaning</td></tr>
<tr><td>00b</td><td>1024 elements(4096bytes)Default value</td></tr>
<tr><td>01b</td><td>512 elements(2048byts)</td></tr>
<tr><td>10b</td><td>256 elements(1024bytes)For resource-constrained condition</td></tr>
<tr><td>11b</td><td>reserved</td></tr>
</table>
The default value is ‘00b’.</td></tr>
<tr><td>1</td><td>R/W</td><td>0x0</td><td>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.</td></tr>
</table>

Offset: 0x0010			Register Name: USBCMD
Bit	**Read/Write**	**Default/Hex**	**Description**
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.

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. 0: Disable the status of the Asynchronous Schedule. 1: Enable the status of the Asynchronous Schedule. 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. 0: Disable the status of the Periodic Schedule. 1: Enable the status of the Periodic Schedule. 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.

Offset: 0x0014			Register Name: USBSTS
Bit	**Read/Write**	**Default/Hex**	**Description**
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 1 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 the completion of USB transaction results in an error condition (e.g. error counter underflow). If the TD on which the error interrupt occurred also have its IOC bit set, both. This bit and USBINT bit are set.

Offset: 0x0014			Register Name: USBSTS
Bit	**Read/Write**	**Default/Hex**	**Description**
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)

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.

Offset: 0x0018			Register Name: USBINTR
Bit	**Read/Write**	**Default/Hex**	**Description**
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

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	0x0	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.

0x0024 EHCI Periodic Frame List Base Address Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: PERIODICLISTBASE
Bit	**Read/Write**	**Default/Hex**	**Description**

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-K byte aligned. The contents of this register are combined with the Frame Index Register (FRINDEX) to enable the Host Controller to step through the Periodic Frame List in sequence.
11: 0	/	/	/

NOTE

Writes must be Dword Writes.

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

Writes must be Dword Writes.

0x0050 EHCI Configure Flag Register (Default Value: 0x0000_0000)

Offset: 0x0050			Register Name: CONFIGFLAG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/

Offset: 0x0050			Register Name: CONFIGFLAG
Bit	**Read/Write**	**Default/Hex**	**Description**
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: Meaning 0: Port routing control logic default-routs each port to an implementation dependent classic host controller. 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 use in the normal implementation.

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><th colspan="3">Offset: 0x0054</th><th>Register Name: PortSC</th></tr>
<tr><th>Bit</th><th>Read/Write</th><th>Default/Hex</th><th>Description</th></tr>
<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 is as follow:
<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 is:
<table>
<tr><td>Bit</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>
Note: 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>
</table>

Offset: 0x0054			Register Name: PortSC
Bit	**Read/Write**	**Default/Hex**	**Description**
8	R/W	0x0	Port Reset 1=Port is in Reset. 0=Port is not in Reset. Default value = 0. 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. (When software writes this bit to a one, it must also write a zero to the Port Enable bit.) **Note**: *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.*

<table>
<tr><th colspan="3">Offset: 0x0054</th><th>Register Name: PortSC</th></tr>
<tr><th>Bit</th><th>Read/Write</th><th>Default/Hex</th><th>Description</th></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</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:
Software sets the Force Port Resume bit to a zero (from a one).
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.</td></tr>
</table>

Offset: 0x0054			Register Name: PortSC
Bit	**Read/Write**	**Default/Hex**	**Description**
6	R/W	0x0	Force Port Resume 1: Resume detected/driven on port. 0: No resume (K-state) detected/driven on port. The default value in this field is '0'. This functionality defined for manipulating this bit depends on the value of the Suspend bit. For example, if the port is not suspended 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**: *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 This bit is set to 1 when there is a change to Over-current Active. Software clears this bit by writing a one to this bit position. The default value of this bit is '0'.
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'.

Offset: 0x0054			Register Name: PortSC
Bit	**Read/Write**	**Default/Hex**	**Description**
3	R/WC	0x0	Port Enable/Disable Change 1: Port enabled/disabled status has changed. 0: No change. The default value of this bit is '0'. 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 The default value of this field is '0'. 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. 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, The default value of this field is '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.

Offset: 0x0054			Register Name: PortSC
Bit	Read/Write	Default/Hex	Description
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.

9.6.7.2 OHCI Register Description

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 the interrupts 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.

<table>
<tr><td>7:6</td><td>R/W</td><td>R/W</td><td>0x0</td><td>HostControllerFunctionalState for USB
<table><tr><td>00b</td><td>USB Reset</td></tr><tr><td>01b</td><td>USB Resume</td></tr><tr><td>10b</td><td>USB Operational</td></tr><tr><td>11b</td><td>USB Suspend</td></tr></table>A transition to USB Operational 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.</td></tr>
<tr><td>5</td><td>R/W</td><td>R</td><td>0x0</td><td>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.</td></tr>
<tr><td>4</td><td>R/W</td><td>R</td><td>0x0</td><td>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.</td></tr>
<tr><td>3</td><td>R/W</td><td>R</td><td>0x0</td><td>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).</td></tr>
</table>

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: No. of Control EDs Over Bulk EDs Served 0: 1:1 1: 2:1 2: 3:1 3: 4:1 The default value is 0x0.

0x0408 HcCommandStatus Register (Default Value: 0x0000_0000)

Offset: 0x0408				Register Name: HcCommandStatus
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:18	/	/	/	/
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

Offset: 0x0408				Register Name: HcCommandStatus
				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.

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

Offset: 0x040C				Register Name: HcInterruptStatus
Bit	Read/Write		Default/Hex	Description
	HCD	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.

0x0410 HcInterruptEnable Register (Default Value: 0x0000_0000)

Offset: 0x0410				Register Name: HcInterruptEnable
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31	R/W	R	0x0	Master Interrupt Enable 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	Root Hub Status Change 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	Unrecoverable Error Interrupt Enable 0: Ignore 1: Enable interrupt generation due to Unrecoverable Error

Offset: 0x0410				Register Name: HcInterruptEnable
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
3	R/W	R	0x0	Resume Detected Interrupt Enable 0: Ignore 1: Enable interrupt generation due to Resume Detected
2	R/W	R	0x0	Start of Frame Interrupt Enable 0: Ignore 1: Enable interrupt generation due to Start of Flame
1	R/W	R	0x0	Writeback Done Head Interrupt Enable 0: Ignore 1: Enable interrupt generation due to Write back Done Head
0	R/W	R	0x0	Scheduling Overrun Interrupt Enable 0: Ignore 1: Enable interrupt generation due to Scheduling Overrun

0x0414 HcInterruptDisable Register (Default Value: 0x0000_0000)

Offset: 0x0414				Register Name: HcInterruptDisable
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

Offset: 0x0414				Register Name: HcInterruptDisable
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
				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

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.

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 Periodic 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.

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]

Offset: 0x0420				Register Name: HcControlHeadED[CHED]
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
				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.

0x0424 HcControlCurrentED Register (Default Value: 0x0000_0000)

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.

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

Offset: 0x0428				Register Name: HcBulkHeadED[BHED]
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
				in the PCED, through bit 0 to bit 3 must be zero in this field.

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.

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.

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

Offset: 0x0434				Register Name: HcFmInterval Register
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
				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.

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	/	/	/	Reserved
13: 0	R	R/W	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.

0x043C HcFmNumber Register (Default Value: 0x0000_0000)

Offset: 0x043C				Register Name: HcFmNumber
Bit	Read/Write		Default/Hex	Description
	HCD	HC		

Offset: 0x043C				Register Name: HcFmNumber
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:16	/	/	/	Reserved
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.*

0x0440 HcPeriodicStart Register (Default Value: 0x0000_0000)

Offset: 0x0440				Register Name: HcPeriodicStatus
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:14	/	/	/	Reserved
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.

0x0444 HcLSThreshold Register (Default Value: 0x0000_0628)

Offset: 0x0444				Register Name: HcLSThreshold
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:12	/	/	/	Reserved
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 is larger than or equal to this field. The value is calculated by HCD with the consideration of transmission and setup overhead.

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]

Offset: 0x0448				Register Name: HcRhDescriptorA
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
				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	0x1	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	0x0	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	0x1	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).

Offset: 0x0448				Register Name: HcRhDescriptorA
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
8	R/W	R	0x0	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	0x1	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.

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.

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. Write 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

Offset: 0x0450				Register Name: HcRhStatus Register
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
				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.

0x0454 HcRhPortStatus Register (Default Value: 0x0000_0100)

Offset: 0x0454				Register Name: HcRhPortStatus
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
31:21	/	/	0x0	Reserved
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 is 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,

Offset: 0x0454				Register Name: HcRhPortStatus
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
				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	/	/	/	**/**
9	R/W	R/W	UFD	(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. 0: Full speed device attached 1: Low speed device attached (write) ClearPortPower The HCD clears the PortPowerStatus bit by writing a '1' to this bit. Writing a '0' has no effect.
8	R/W	R/W	0x1	(read) PortPowerStatus 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. 0: Port power is off 1: Port power is on (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.*

Offset: 0x0454				Register Name: HcRhPortStatus
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
7:5	/	/	/	/
4	R/W	R/W	0x0	(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. 0: Port reset signal is not active 1: Port reset signal is active (write) SetPortReset The HCD sets the port reset signaling by writing a '1' to this bit. Writing a '0' has no effect. If CurrentConnectStatus is cleared, this write does not set PortResetStatus, but instead sets ConnectStatusChange. This informs the driver that it attempted to reset a disconnected port.
3	R/W	R/W	0x0	(read) PortOverCurrentIndicator 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. 0: No overcurrent condition. 1: overcurrent condition detected. (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.
2	R/W	R/W	0x0	(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. 0: Port is not suspended 1: Port is suspended (write) SetPortSuspend The HCD sets the PortSuspendStatus bit by writing a '1' to this bit. Writing a '0' has no effect. If CurrentConnectStatus is cleared, this write does not set PortSuspendStatus; instead it sets ConnectStatusChange. This informs the driver that it attempted to

Offset: 0x0454				Register Name: HcRhPortStatus
Bit	Read/Write		Default/Hex	Description
	HCD	HC		
				suspend a disconnected port.
1	R/W	R/W	0x0	(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. 0: Port is disabled 1: Port is enabled (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 (DeviceRemoveable[NumberDownstreamPort]).*

9.6.7.3 HCI Controller and PHY Interface Description

0x0800 HCI Control 1 Register (Default Value: 0x1000_0000)

Offset: 0x0800			Register Name: HCI_CTRL1
Bit	Read/Write	Default/Hex	Description
31:29	/	/	Reserved.
28	R	0x1	DMA Transfer Status Enable 0: Disable 1: Enable
27:26	/	/	/

Offset: 0x0800			Register Name: HCI_CTRL1
Bit	**Read/Write**	**Default/Hex**	**Description**
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.
19:13	/	/	/
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.
7:1	/	/	/
0	R/W	0x0	ULPI bypass enable 1: Enable UTMI interface, disable ULPI interface 0: Enable ULPI interface, disable UTMI interface

0x0808 HCI Control 2 Register (Default Value: 0x0000_0000)

Offset: 0x0808	Register Name: HCI_CTRL2

Bit	Read/Write	Default/Hex	Description
31:17	/	/	Reserved.
16	R/W1C	0x1	Linestate Change Detect 0: Linestate change is not detected. 1: Linestate change is detected. Write '1' to clear.
15:12	/	/	Reserved.
1	R/W	0x0	Linestate Change Interrupt Enable 1: Enable 0: Disable
0	R/W	0x0	Linestate Change Detect Enable 1: Enable 0: Disable

0x0810 PHY Control Register (Default Value: 0x0000_0002)

Offset: 0x0810			Register Name: PHY Control Register
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	bist_en_a
15:9	/	/	/
8	R/W	0x0	500k pullup enable
7:2	/	/	/
1	R/W	0x1	siddq write 1 to enable phy
0	R/W	0x0	vc_clk

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
15:2	/	/	/
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

9.7 GPIO

9.7.1 Overview

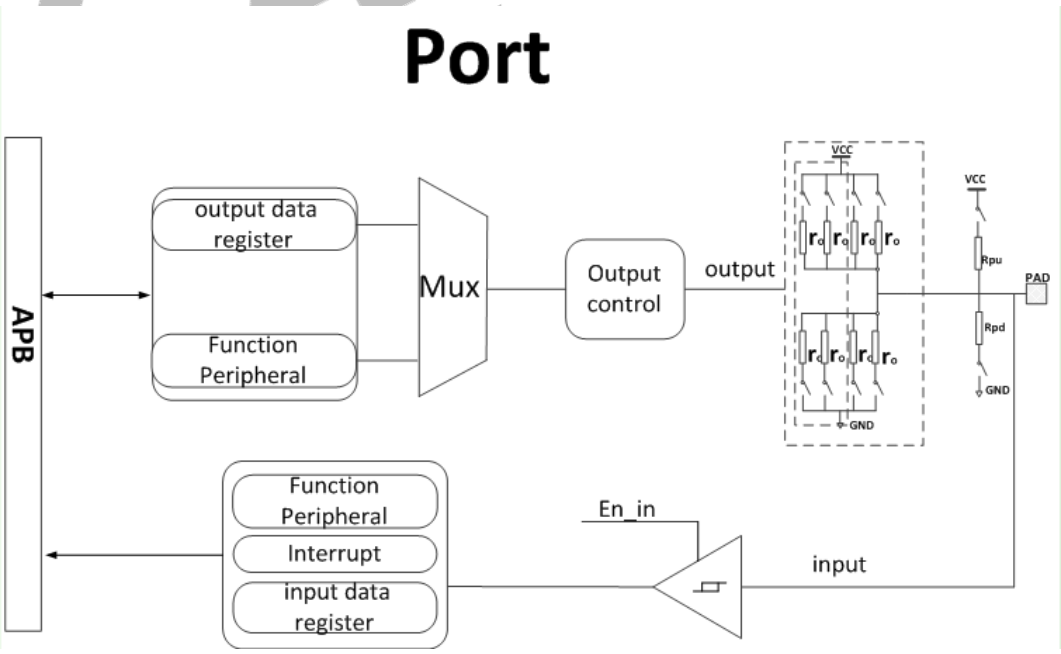
The general purpose input/output (GPIO) can be configured with multifunctional input/output pins. All these ports can be configured as GPIO only if multiplexed functions not used. The total 3 group external PIO interrupt sources are supported and interrupt mode can be configured by software.

GPIO has the following features:

- 2 ports (PA, PB)
- Software control for each signal pin
- GPIO peripheral can produce interrupt
- Pull-up/pull-down/no-pull register control
- Controls the direction of every signal
- 4 drive strengths in each operating mode
- Up to 37 interrupts (for R128-S1 and R128-S2) and 40 interrrupts (for R128-S3)
- Configurable interrupt edges

9.7.2 Block Diagram

The following figure shows the block diagram of GPIO.



The GPIO consists of the digital part (GPIO, external interface) and IO analog part (output buffer, dual pull down, pad). The digital part can select the output interface by using the MUX switch; the analog part can configure pull up/down and buffer strength.

When executing GPIO read state, the GPIO reads the current level of the pin into the internal register bus. When not executing GPIO read state, the external pin and the internal register bus are off-status, which is high-impedance.

9.7.3 Functional Description

9.7.3.1 Multi-function Port

R128-S1/S2 includes 37 multi-functional input/output port pins. R128-S3 includes 40 multi-functional input/output port pins. There are 2 ports as listed below.

Table 9-20 R128-S1/S2 Multi-function Port

Port Name	Number of Pins	Input Driver	Output Driver	Multiplex Pins	Typical Power Supply
PA	30	Schmitt	CMOS	SPI/TWI/LCD/UART/PWM/IR /I2S/DMIC/JTAG/CSI/	3.3V/1.8V
PB	7	Schmitt	CMOS	PWM/FLASH/UART/SD/LCD/ CSI/ADC	3.3V/1.8V

Table 9-21 R128-S3 Multi-function Port

Port Name	Number of Pins	Input Driver	Output Driver	Multiplex Pins	Typical Power Supply
PA	30	Schmitt	CMOS	SPI/TWI/LCD/UART/PWM/IR /I2S/DMIC/JTAG/CSI/	3.3V/1.8V
PB	10	Schmitt	CMOS	PWM/FLASH/UART/SD/LCD/ CSI/ADC	3.3V/1.8V

9.7.3.2 GPIO Multiplex Function

The following table shows the multiplex function pins of R128.

NOTE

For each GPIO, Function 0 is input function, and Function 1 is output function. Functions 9-13 are reserved.

Table 9-22 GPIO Multiplex Function

Pin Name	GPIO Group	IO Type	Function2	Function3	Function4	Function5	Function6	Function7	Function8	Function14
PA0	GPIOA	I/O	IR_RX		PWM7	TWI0_SCL	TWI1_SCL	LCD_VSYNC	LCD_D2	PA_EINT0
PA1/FEL0		I/O	IR_TX	FEM_CTRL1	IR_RX	TWI0_SDA	TWI1_SDA		LCD_D3	PA_EINT1
PA2/FEL1		I/O	SPI1_CS<DBI_CSX>	DMIC_DATA0	JTAG_RV_TMS	SDC_DATA1	I2S_LRCLK	JTAG_DSP_TMS	LCD_D4	PA_EINT2
PA3		I/O	SPI1_CLK<DBI_SCLK>	DMIC_DATA1	JTAG_RV_TDI	SDC_DATA0	I2S_BCLK	JTAG_DSP_TDI	LCD_D5	PA_EINT3
PA4		I/O	SPI1_MOSI<DBI_SDO>	DMIC_DATA2	UART0_TX	SDC_CLK	I2S_DIN	PWM1	LCD_D6	PA_EINT4
PA5		I/O	SPI1_MISO<DBI_SDI/DBI_TE/DBI_DCX>	DMIC_DATA3	JTAG_RV_TDO	SDC_CMD	I2S_DOUT	JTAG_DSP_TDO	LCD_D7	PA_EINT5
PA6		I/O	SPI1_HOLD<DBI_DCX/DBI_WRX>	DMIC_CLK	UART0_RX	SDC_DATA3	I2S_MCLK	LCD_CLK	LCD_D14	PA_EINT6
PA7		I/O	SPI1_WP<DBI_TE>	OWA_IN	JTAG_RV_TCK	SDC_DATA2	JTAG_DSP_TCK	LCD_HSYNC	LCD_D13	PA_EINT7
PA8		I/O	UART0_RX	OWA_OUT	PWM0	OWA_IN	TWI1_SCL	FEM_CTRL2	LCD_D12	PA_EINT8
PA9		I/O	UART0_TX		PWM1	LEDC	TWI1_SDA	LCD_DE	LCD_D15	PA_EINT9
PA10		I/O	UART2_RTS	IR_RX	PWM2	TWI1_SCL			LCD_D11	PA_EINT10
PA11/WUPIO0		I/O	UART2_CTS	IR_TX	PWM3	TWI1_SDA	32KOSCO	FEM_CTRL1	LCD_D10	PA_EINT11
PA12/WUPIO1/LXTAL-IN		I/O	UART2_TX	TWI0_SCL	UART2_RTS	IR_RX	SPI1_CS/DBI_CSX	LCD_VSYNC	LCD_D18	PA_EINT12
PA13/WUPIO2/LXTAL-OUT		I/O	UART2_RX	TWI0_SDA	UART2_CTS	IR_TX	SPI1_CLK/DBI_SCLK	LEDC	LCD_D19	PA_EINT13
PA14/WUPIO3		I/O	PWM0	SPI1_MOSI<DBI_SDO>	UART2_RX	SIM_DATA	UART1_RTS	TWI1_SCL	LCD_D20	PA_EINT14
PA15		I/O	PWM1	SPI1_HOLD<DBI_DCX/DBI_WRX>	UART2_TX	SIM_CLK	UART1_CTS	TWI1_SDA	LCD_D21	PA_EINT15
PA16		I/O	TWI0_SCL	OWA_IN	TWI1_SCL	UART0_TX	IR_RX	UART2_TX	SWD_TMS	PA_EINT16
PA17		I/O	TWI0_SDA	OWA_OUT	TWI1_SDA	UART0_RX	IR_TX	UART2_RX	SWD_TCK	PA_EINT17
PA18/WUPIO4		I/O	I2S_MCLK	IR_RX	IR_TX		SPI1_MOSI<DBI_SDO>	NCSI_HSYNC	LCD_VSYNC	PA_EINT18
PA19/WUPIO5		I/O	I2S_LRCLK	UART1_RTS	PWM4	DMIC_DATA0	SPI1_HOLD<DBI_DCX/DBI_WRX>	NCSI_VSYNC	LCD_HSYNC	PA_EINT19
PA20/WUPIO6		I/O	I2S_BCLK	UART1_CTS	PWM5	DMIC_DATA1	SPI1_WP<DBI_TE>	NCSI_PCLK	LCD_CLK	PA_EINT20
PA21/WUPIO7		I/O	I2S_DIN	UART1_RX	PWM6	DMIC_DATA2	SPI1_MISO<DBI_SDI/DBI_TE/DBI_DCX>	NCSI_MCLK	LCD_DE	PA_EINT21
PA22/WUPIO8		I/O	I2S_DOUT	UART1_TX	PWM7	DMIC_DATA3		LEDC	NCSI_D0	PA_EINT22
PA23/WUPIO9		I/O	I2S_MCLK	DCXO_PUP_OUT	SWD_SWO	DMIC_CLK	TWI0_SCL	PWM0	NCSI_D1	PA_EINT23
PA24		I/O	SDC_DATA3	SPI0_MISO	PWM4	UART2_RX	TWI0_SDA	SIM_DATA	NCSI_D6	PA_EINT24
PA25		I/O	SDC_CMD	SPI0_WP	PWM5	JTAG_M33_TDO	JTAG_RV_TDO	SIM_CLK	NCSI_D5	PA_EINT25

PA26		I/O	SDC_DATA0	SPI0_CLK	PWM6	JTAG_M33_TDI	JTAG_RV_TDI	LEDC	NCSI_D3	PA_EINT26
PA27		I/O	SDC_DATA1	SPI0_HOLD	PWM7	JTAG_M33_TMS	JTAG_RV_TMS	SIM_DET	NCSI_D2	PA_EINT27
PA28		I/O	SDC_DATA2	SPI0_CS	FEM_CTRL1	JTAG_M33_TCK	JTAG_RV_TCK	SIM_RST	NCSI_D7	PA_EINT28
PA29		I/O	SDC_CLK	SPI0_MOSI	FEM_CTRL2	UART2_TX	PWM1	LEDC	NCSI_D4	PA_EINT29
NOTE: LXTAL is the low-frequency clock exclusively for the RTC domain.										
For R128-S1/S2										
PB0/ADC0	GPIOB	I/O	UART0_TX	TWI1_SCL	IR_RX	UART2_RTS	PWM2		NCSI_HSYNC	PB_EINT0
PB1/ADC1		I/O	UART0_RX	TWI1_SDA	IR_TX	UART2_CTS	PWM3		NCSI_VSYNC	PB_EINT1
PB2/ADC2		I/O	PWM2	SPI1_MISO<DBI_SDI/DBI_TE/DBI_DCX>	TWI1_SCL	SIM_RST	UART1_RX	UART2_RTS	LCD_D23	PB_EINT2
PB3/ADC3		I/O	PWM3	SPI1_WP<DBI_TE>	TWI1_SDA	SIM_DET	UART1_TX	UART2_CTS	LCD_D22	PB_EINT3
PB4/ADC4		I/O					PWM4		LCD_DE	PB_EINT4
PB14/ADC6		I/O	UART1_TX						NCSI_PCLK	PB_EINT14
PB15/ADC7		I/O	UART1_RX				PWM0		NCSI_MCLK	PB_EINT15
For R128-S3										
PB0/ADC0	GPIOB	I/O	UART0_TX	TWI1_SCL	IR_RX	UART2_RTS	PWM2		NCSI_HSYNC	PB_EINT0
PB1/ADC1		I/O	UART0_RX	TWI1_SDA	IR_TX	UART2_CTS	PWM3		NCSI_VSYNC	PB_EINT1
PB2/ADC2		I/O	PWM2	SPI1_MISO<DBI_SDI/DBI_TE/DBI_DCX>	TWI1_SCL	SIM_RST	UART1_RX	UART2_RTS	LCD_D23	PB_EINT2
PB3/ADC3		I/O	PWM3	SPI1_WP<DBI_TE>	TWI1_SDA	SIM_DET	UART1_TX	UART2_CTS	LCD_D22	PB_EINT3
PB4/ADC4		I/O	UART1_RTS	SDC_CLK	SPI0_CS		PWM4		LCD_DE	PB_EINT4
PB5/ADC5		I/O	UART1_CTS	SDC_DATA1	SPI0_MOSI		PWM5			PB_EINT5
PB6		I/O	UART1_TX	SDC_DATA0	SPI0_CLK		PWM6			PB_EINT6
PB7		I/O	UART1_RX	SDC_DATA3	SPI0_HOLD		PWM7			PB_EINT7
PB14/ADC6		I/O	UART1_TX	SDC_DATA2	SPI0_WP				NCSI_PCLK	PB_EINT14
PB15/ADC7		I/O	UART1_RX	SDC_CMD	SPI0_MISO		PWM0		NCSI_MCLK	PB_EINT15

9.7.3.3 Port Function

The Port Controller supports 3 GPIOs. every GPIO can be configured as Input, Output, Function Peripheral, IO disable or Interrupt function. The configuration instruction of every function is as follows.

Type	Function	Buffer Strength	Pull Up	Pull Down
Input	GPIO/Multiplexing Input	/	X	X
Output	GPIO/Multiplexing Output	Y	X	X
Disable	Pull Up	/	Y	N
	Pull Down	/	N	Y
Interrupt	Various Triggering Ways	/	X	X

/: Configuration is unnecessary and invalid.

Y: Configuration is necessary.

X: Select configuration according to the actual situation

N: Configuration is forbidden.

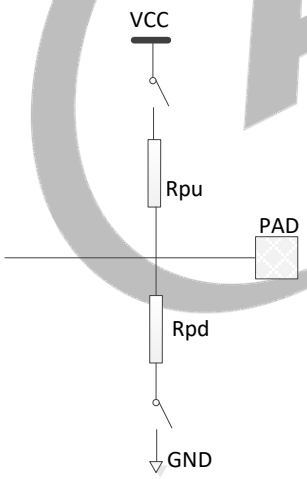
Pull Up/Down and High-Impedance Logic

Each IO pin can configure the internal pull-up/down function or high-impedance.

9.7.3.4 Pull Up/Down and High-Impedance Logic

Each IO pin can configure the internal pull-up/down function or high-impedance.

Figure 9-59 Pull up/down Logic



High impedance indicates that the output is at float state, and that all buffer is off. Its level is decided by external high/low level. At high-impedance state, Rpu and Rpd are switched off, and the multiplexing function of IO is set as IO disable or input through software configuration.

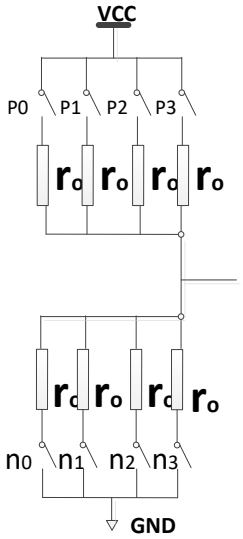
When pulling up, the switch on Rpu is conducted by software configuration, the IO is pulled up to VCC by Rpu. When pulling down, the switch on Rpd is conducted by software configuration, the IO is pulled down to GND by Rpd. The pull-up/down of each IO is weak pull-up/down.

The setting of pull-down, pull-up, high-impedance is decided by the external circuit.

9.7.3.5 Buffer Strength

Each IO can be set as different buffer strength. The IO buffer diagram is as follows.

Figure 9-60 IO Buffer Strength Diagram



In output high level, the n0, n1, n2, n3 of NMOS is off, and the p0, p1, p2, p3 of PMOS is on. When the buffer strength is set to 0 (buffer strength is weakest), only the p0 is on. The output impedance is maximum, and the impedance value is r0. When the buffer strength is set to 1, only the p0 and p1 is on. The output impedance is equivalent to two r0 in parallel, and the impedance value is r0/2. When the buffer strength is 2, only the p0, p1, and p2 is on. The output impedance is equivalent to three r0 in parallel, and the impedance value is r0/3. When buffer strength is 3, the p0, p1, p2, and p3 is on. The output impedance is equivalent to four r0 in parallel, and the impedance value is r0/4.

In output low level, the p0, p1, p2, p3 of PMOS is off, and the n0, n1, n2, n3 of NMOS is on. When the buffer strength is set to 0 (buffer strength is weakest), only the n0 is on. The output impedance is maximum, and the impedance value is r0. When the buffer strength is set to 1, only the n0 and n1 is on. The output impedance is equivalent to two r0 in parallel, and the impedance value is r0/2. When the buffer strength is 2, only the n0, n1, and n2 is on. The output impedance is equivalent to three r0 in parallel, and the impedance value is r0/3. When the buffer strength is 3, the n0, n1, n2, and n3 is on. The output impedance is equivalent to four r0 in parallel, and the impedance value is r0/4.

When GPIO is set to input or interrupt, output driver circuit is unconnected to the port. Therefore, the driver configuration is invalid.

9.7.3.6 Interrupts

Each group of IO has an independent interrupt number, and the IO within each group shares the same interrupt number. When one IO generates interrupt, the GPIO pins will send interrupt request to GIC. External Interrupt Status Register can be used to query which IO generates interrupt.

The interrupt trigger of GPIO supports the following triggering types.

Positive Edge: When a low level changes to a high level, the interrupt will generate. No matter how long a high level keeps, the interrupt generates only once.

Negative Edge: When a high level changes to a low level, the interrupt will generate. No matter how long a low level keeps, the interrupt generates only once.

High Level: Just keep a high level and the interrupt will always generate.

Low Level: Just keep a low level and the interrupt will always generate.

Double Edge: Positive and negative edge.

External Interrupt Configure Register: Used to configure the trigger type.

GPIO interrupt supports hardware debounce function by setting External Interrupt Debounce Register. Sample trigger signal using a lower sample clock, to reach the debounce effect because the dither frequency of the signal is higher than the sample frequency.

Set the sample clock source by PIO_INT_CLK_SELECT and the prescale factor by DEB_CLK_PRE_SCALE.

9.7.4 Register List

Module Name	Base Address
GPIO	0x4004A400

Register Name	Offset Address	Description
PA_CFG0	0*0x0024 + 0x0000	Port A Configure Register 0
PA_CFG1	0*0x0024 + 0x0004	Port A Configure Register 1
PA_CFG2	0*0x0024 + 0x0008	Port A Configure Register 2
PA_CFG3	0*0x0024 + 0x000C	Port A Configure Register 3
PB_CFG0	1*0x0024 + 0x0000	Port B Configure Register 0
PB_CFG1	1*0x0024 + 0x0004	Port B Configure Register 1
PA_DATA	0*0x0024 + 0x0010	Port A Data Register
PB_DATA	1*0x0024 + 0x0010	Port B Data Register
PA_DRV0	0*0x0024 + 0x0014	Port A Multi-Driving Register 0
PA_DRV1	0*0x0024 + 0x0018	Port A Multi-Driving Register 1
PB_DRV0	1*0x0024 + 0x0014	Port B Multi-Driving Register 0
PA_PULL0	0*0x0024 + 0x001C	Port A Pull Register 0
PA_PULL1	0*0x0024 + 0x0020	Port A Pull Register 1
PB_PULL0	1*0x0024 + 0x001C	Port B Pull Register 0
PA_INT_CFG0	0x0200 + 0*0x0020 + 0x0000	PIOA Interrupt Configure Register 0
PA_INT_CFG1	0x0200 + 0*0x0020 + 0x0004	PIOA Interrupt Configure Register 1
PA_INT_CFG2	0x0200 + 0*0x0020 + 0x0008	PIOA Interrupt Configure Register 2
PA_INT_CFG3	0x0200 + 0*0x0020 + 0x000C	PIOA Interrupt Configure Register 3
PB_INT_CFG0	0x0200 + 1*0x0020 + 0x0000	PIOB Interrupt Configure Register 0
PB_INT_CFG1	0x0200 + 1*0x0020 + 0x0004	PIOB Interrupt Configure Register 1
PA_INT_CTL	0x0200 + 0*0x0020 + 0x0010	PIOA Interrupt Control Register
PB_INT_CTL	0x0200 + 1*0x0020 + 0x0010	PIOB Interrupt Control Register

Register Name	Offset Address	Description
PA_INT_STA	0x0200 + 0*0x0020 + 0x0014	PIOA Interrupt Status Register
PB_INT_STA	0x0200 + 1*0x0020 + 0x0014	PIOB Interrupt Status Register
PA_INT_DBC	0x0200 + 0*0x0020 + 0x0018	PIOA Interrupt Debounce Register
PB_INT_DBC	0x0200 + 1*0x0020 + 0x0018	PIOB Interrupt Debounce Register

9.7.5 Register Description

9.7.5.1 0x0000 PA Configure Register 0 (Default Value: 0x0000_FFFF)

Offset: 0x0000			Register Name: PA_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0xf	PA7 function select 0x0: input 0x1: output 0x2: SPI1_WP/DBI_TE 0x3: OWA_IN 0x4: JTAG_RV_TCK 0x5: SD_DATA2 0x6: JTAG_DSP_TCK 0x7: LCD_HSYNC 0x8: LCD_D13 0xe: PA_EINT7 Other: No effect
27:24	R/W	0xf	PA6 function select 0x0: input 0x1: output 0x2: SPI1_HOLD/DBI_DCX/DBI_WRX 0x3: DMIC_CLK 0x4: UART0_RX 0x5: SD_DATA3 0x6: I2S_MCLK 0x7: LCD_DCLK 0x8: LCD_D14 0xe: PA_EINT6 Other: No effect
23:29	R/W	0xf	PA5 function select 0x0: input 0x1: output 0x2: SPI1_MISO/DBI_SDI/DBI_TE/DBI_DCX 0x3: DMIC_DATA3 0x4: JTAG_RV_TDO 0x5: SDC_CMD 0x6: I2S_DOUT 0x7: JTAG_DSP_TDO 0x8: LCD_D7 0xe: PA_EINT5 Other: No effect
19:16	R/W	0xf	PA4 function select 0x0: input 0x1: output 0x2: SPI1_MOSI/DBI_SDO 0x3: DMIC_DATA2 0x4: UART0_TX 0x5: SDC_CLK 0x6: I2S_DIN 0x7: PWM1 0x8: LCD_D6 0xe: PA_EINT4 Other: No effect

Offset: 0x0000			Register Name: PA_CFG0	
Bit	**Read/Write**	**Default/Hex**	**Description**	
15:12	R/W	0xf	PA3 function select 0x0: input 0x2: SPI1_CLK/DBI_SCLK 0x4: JTAG_RV_TDI 0x6: I2S_BCLK 0x8: LCD_D5 0xe: PA_EINT3	 0x1: output 0x3: DMIC_DATA1 0x5: SD_DATA0 0x7: JTAG_DSP_TDI Other: No effect
11:8	R/W	0xf	PA2 function select 0x0: input 0x2: SPI1_CS/DBI_CSX 0x4: JTAG_RV_TMS 0x6: I2S_LRCLK 0x8: LCD_D4 0xe: PA_EINT2	 0x1: output 0x3: DMIC_DATA0 0x5: SD_DATA1 0x7: JTAG_DSP_TMS Other: No effect
7:4	R/W	0xf	PA1 function select 0x0: input 0x2: IR_TX 0x4: IR_RX 0x6: TWI1_SDA 0x8: LCD_D3 0xe: PA_EINT1	 0x1: output 0x3: FEM_CTRL1 0x5: TWI0_SDA 0x7: Reserved Other: No effect
3: 0	R/W	0xf	PA0 function select 0x0: input 0x2: IR_RX 0x4: PWM7 0x6: TWI1_SCL 0x7: LCD_VSYNC 0x8: LCD_D2 0xe: PA_EINT0	 0x1: output 0x3: Reserved 0x5: TWI0_SCL Other: No effect

9.7.5.2 0x0004 PA Configure Register 1 (Default Value: 0x0000_FFFF)

Offset: 0x0004			Register Name: PA_CFG1	
Bit	**Read/Write**	**Default/Hex**	**Description**	
31:28	R/W	0xf	PA15 function select 0x0: input 0x2: PWM1 0x3: SPI1_HOLD 0x4: UART2_TX 0x6: UART1_CTS 0x8: LCD_D21 0xe: PA_EINT15	 0x1: output 0x5: SIM_CLK 0x7: TWI1_SDA Other: No effect

Offset: 0x0004			Register Name: PA_CFG1	
Bit	**Read/Write**	**Default/Hex**	**Description**	
27:24	R/W	0xf	PA14 function select 0x0: input 0x2: PWM0 0x4: UART2_RX 0x6: UART1_RTS 0x8: LCD_D20 0xe: PA_EINT14	 0x1: output 0x3: SPI1_MOSI/DBI_SDO 0x5: SIM_DATA 0x7: TWI1_SCL Other: No effect
23:20	R/W	0xf	PA13 function select 0x0: input 0x2: UART2_RX 0x4: UART2_CTS 0x6: SPI1_CLK/DBI_SCLK 0x8: LCD_D19 0xe: PA_EINT13	 0x1: output 0x3: TWI0_SDA 0x5: IR_TX 0x7: LEDC Other: No effect
19:16	R/W	0xf	PA12 function select 0x0: input 0x2: UART2_TX 0x4: UART2_RTS 0x6: SPI1_CS/DBI_CSX 0x7: LCD_VSYNC 0x8: LCD_D18 0xe: PA_EINT12	 0x1: output 0x3: TWI0_SCL 0x5: IR_RX Other: No effect
15:12	R/W	0xf	PA11 function select 0x0: input 0x2: UART2_CTS 0x4: PWM3 0x6: 32KOSCO 0x8: LCD_D10 0xe: PA_EINT11	 0x1: output 0x3: IR_TX 0x5: TWI1_SDA 0x7: FEM_CTRL1 Other: No effect
11:8	R/W	0xf	PA10 function select 0x0: input 0x2: UART2_RTS 0x4: PWM2 0x6: Reserved 0x8: LCD_D11 0xe: PA_EINT10	 0x1: output 0x3: IR_RX 0x5: TWI1_SCL 0x7: Reserved Other: No effect
7:4	R/W	0xf	PA9 function select 0x0: input 0x2: UART0_TX 0x4: PWM1 0x6: TWI1_SDA 0x8: LCD_D15 0xe: PA_EINT9	 0x1: output 0x3: Reserved 0x5: LEDC 0x7: LCD_DE Other: No effect

Offset: 0x0004			Register Name: PA_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
3: 0	R/W	0xf	PA8 function select 0x0: input 0x1: output 0x2: UART0_RX 0x3: OWA_OUT 0x4: PWM0 0x5: OWA_IN 0x6: TWI1_SCL 0x7: FEM_CTRL2 0x8: LCD_D12 0xe: PA_EINT8 Other: No effect

9.7.5.3 0x0008 PA Configure Register 2 (Default Value: 0x0000_FFFF)

Offset: 0x0008			Register Name: PA_CFG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0xf	PA23 function select 0x0: input 0x1: output 0x2: I2S_MCLK 0x3: DCXO_PUP_OUT 0x4: SWD_SWO 0x5: DMIC_CLK 0x6: TWI0_SCL 0x7: PWM0 0x8: CSI_D1 0xe: PA_EINT23 Other: No effect
27:24	R/W	0xf	PA22 function select 0x0: input 0x1: output 0x2: I2S_DOUT 0x3: UART1_TX 0x4: PWM7 0x5: DMIC_DATA3 0x6: Reserved 0x7: LEDC 0x8: CSI_D0 0xe: PA_EINT22 Other: No effect
23:20	R/W	0xf	PA21 function select 0x0: input 0x1: output 0x2: I2S_DIN 0x3: UART1_RX 0x4: PWM6 0x5: DMIC_DATA2 0x6: SPI1_MISO/DBI_SDI/DBI_TE/DBI_DCX 0x7: CSI_MCLK 0x8: LCD_DE 0xe: PA_EINT21 Other: No effect
19:16	R/W	0xf	PA20 function select 0x0: input 0x1: output 0x2: I2S_BCLK 0x3: UART1_CTS 0x4: PWM5 0x5: DMIC_DATA1 0x6: SPI1_WP/DBI_TE 0x7: CSI_PCLK 0x8: LCD_CLK 0xe: PA_EINT20 Other: No effect

Offset: 0x0008			Register Name: PA_CFG2
Bit	**Read/Write**	**Default/Hex**	**Description**
15:12	R/W	0xf	PA19 function select 0x0: input 0x1: output 0x2: I2S_LRCLK 0x3: UART1_RTS 0x4: PWM4 0x5: DMIC_DATA0 0x6: SPI1_HOLD/DBI_DCX/DBI_WRX 0x7: CSI_VSYNC 0x8: LCD_HSYNC 0xe: PA_EINT19 Other: No effect
11:8	R/W	0xf	PA18 function select 0x0: input 0x1: output 0x2: I2S_MCLK 0x3: IR_RX 0x4: IR_TX 0x5: Reserved 0x6: SPI1_MOSI/DBI_SDO 0x7: CSI_HSYNC 0x8: LCD_VSYNC 0xe: PA_EINT18 Other: No effect
7:4	R/W	0xf	PA17 function select 0x0: input 0x1: output 0x2: TWI0_SDA 0x3: OWA_OUT 0x4: TWI1_SDA 0x5: UART0_RX 0x6: IR_TX 0x7: UART2_RX 0x8: SWD_TCK 0xe: PA_EINT17 Other: No effect
3: 0	R/W	0xf	PA16 function select 0x0: input 0x1: output 0x2: TWI0_SCL 0x3: OWA_IN 0x4: TWI1_SCL 0x5: UART0_TX 0x6: IR_RX 0x7: UART2_TX 0x8: SWD_TMS 0xe: PA_EINT16 Other: No effect

9.7.5.4 0x000C PA Configure Register 3 (Default Value: 0x0000_FFFF)

Offset: 0x000C			Register Name: PA_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:20	R/W	0xf	PA29 function select 0x0: input 0x1: output 0x2: SDC_CLK 0x3: SPI0_MOSI 0x4: FEM_CTRL2 0x5: UART2_TX 0x6: PWM1 0x7: LEDC 0x8: NCSI_D4 0xe: PA_EINT29 Other: No effect

Offset: 0x000C			Register Name: PA_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
19:16	R/W	0xf	PA28 function select 0x0: input 0x1: output 0x2: SDC_DATA2 0x3: SPI0_CS 0x4: FEM_CTRL1 0x5: JTAG_M33_TCK 0x6: JTAG_RV_TCK 0x7: SIM_RST 0x8: NCSI_D7 0xe: PA_EINT28 Other: No effect
15:12	R/W	0xf	PA27 function select 0x0: input 0x1: output 0x2: SDC_DATA1 0x3: SPI0_HOLD 0x4: PWM7 0x5: JTAG_M33_TMS 0x6: JTAG_RV_TMS 0x7: SIM_DET 0x8: NCSI_D2 0xe: PA_EINT27 Other: No effect
11:8	R/W	0xf	PA26 function select 0x0: input 0x1: output 0x2: SDC_DATA0 0x3: SPI0_CLK 0x4: PWM6 0x5: JTAG_M33_TDI 0x6: JTAG_RV_TDI 0x7: LEDC 0x8: NCSI_D3 0xe: PA_EINT26 Other: No effect
7:4	R/W	0xf	PA25 function select 0x0: input 0x1: output 0x2: SDC_CMD 0x3: SPI0_WP 0x4: PWM5 0x5: JTAG_M33_TDO 0x6: JTAG_RV_TDO 0x7: SIM_CLK 0x8: NCSI_D5 0xe: PA_EINT25 Other: No effect
3: 0	R/W	0xf	PA24 function select 0x0: input 0x1: output 0x2: SDC_DATA3 0x3: SPI0_MISO 0x4: PWM4 0x5: UART2_RX 0x6: TWI0_SDA 0x7: SIM_DATA 0x8: NCSI_D6 0xe: PA_EINT24 Other: No effect

9.7.5.5 0x0024+0x0000 PB Configure Register 0 (Default Value: 0x0000_FFFF) (for R128-S1& R128-S2)

Offset: 0x0024+0x0000			Register Name: PB_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	Reserved

Offset: 0x0024+0x0000			Register Name: PB_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
19:16	R/W	0xf	PB4 function select 0x0: input 0x1: output 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Reserved 0x6: PWM4 0x7: Reserved 0x8: LCD_DE 0xe: PB_EINT4 Other: No effect
15:12	R/W	0xf	PB3 function select 0x0: input 0x1: output 0x2: PWM3/ECT3 0x3: SPI1_WP/DBI_TE 0x4: TWI1_SDA 0x5: SIM_DET 0x6: UART1_TX 0x7: UART2_CTS 0x8: LCD_D22 0xe: PB_EINT3 Other: No effect
11:8	R/W	0xf	PB2 function select 0x0: input 0x1: output 0x2: PWM2 0x3: SPI1_MISO/DBI_SDI/DBI_TE/DBI_DCX 0x4: TWI1_SCL 0x5: SIM_RST 0x6: UART1_RX 0x7: UART2_RTS 0x8: LCD_D23 0xe: PB_EINT2 Other: No effect
7:4	R/W	0xf	PB1 function select 0x0: input 0x1: output 0x2: UART0_RX 0x3: TWI1_SDA 0x4: IR_TX 0x5: UART2_CTS 0x6: PWM3 0x7: Reserved 0x8: CSI_VSYNC 0xe: PB_EINT1 Other: No effect
3: 0	R/W	0xf	PB0 function select 0x0: input 0x1: output 0x2: UART0_TX 0x3: TWI1_SCL 0x4: IR_RX 0x5: UART2_RTS 0x6: PWM2 0x7: Reserved 0x8: CSI_HSYNC 0xe: PB_EINT0 Other: No effect

9.7.5.6 0x0024+0x0000 PB Configure Register 0 (Default Value: 0x0000_FFFF) (for R128-S3)

Offset: 0x0024+0x0000			Register Name: PB_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/

Offset: 0x0024+0x0000			Register Name: PB_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
19:16	R/W	0xf	PB4 function select 0x0: input 0x1: output 0x2: UART1_RTS 0x3: SDC_CLK 0x4: SPI0_CS 0x5: Reserved 0x6: PWM4 0x7: Reserved 0x8: LCD_DE 0xe: PB_EINT4 Other: No effect
15:12	R/W	0xf	PB3 function select 0x0: input 0x1: output 0x2: PWM3 0x3: SPI1_WP/DBI_TE 0x4: TWI1_SDA 0x5: SIM_DET 0x6: UART1_TX 0x7: UART2_CTS 0x8: LCD_D22 0xe: PB_EINT3 Other: No effect
11:8	R/W	0xf	PB2 function select 0x0: input 0x1: output 0x2: PWM2 0x3: SPI1_MISO/DBI_SDI/DBI_TE/DBI_DCX 0x4: TWI1_SCL 0x5: SIM_RST 0x6: UART1_RX 0x7: UART2_RTS 0x8: LCD_D23 0xe: PB_EINT2 Other: No effect
7:4	R/W	0xf	PB1 function select 0x0: input 0x1: output 0x2: UART0_RX 0x3: TWI1_SDA 0x4: IR_TX 0x5: UART2_CTS 0x6: PWM3 0x7: Reserved 0x8: CSI_VSYNC 0xe: PB_EINT1 Other: No effect
3: 0	R/W	0xf	PB0 function select 0x0: input 0x1: output 0x2: UART0_TX 0x3: TWI1_SCL 0x4: IR_RX 0x5: UART2_RTS 0x6: PWM2 0x7: Reserved 0x8: CSI_HSYNC 0xe: PB_EINT0 Other: No effect

9.7.5.7 0x0024+0x0004 PB Configure Register 1 (Default Value: 0x0000_FFFF) (for R128-S1 & R128-S2)

Offset: 0x0024+0x0004			Register Name: PB_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0024+0x0004			Register Name: PB_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0xf	PB15 function select 0x0: input 0x1: output 0x2: UART1_RX 0x3: Reserved 0x4: Reserved 0x5: Reserved 0x6: PWM0 0x7: Reserved 0x8: CSI_MCLK 0xe: PB_EINT15 Other: No effect
27:24	R/W	0xf	PB14 function select 0x0: input 0x1: output 0x2: UART1_TX 0x3: Reserved 0x4: Reserved 0x5: Reserved 0x6: Reserved 0x7: Reserved 0x8: CSI_PCLK 0xe: PB_EINT14 Other: No effect
23: 0	/	/	/

9.7.5.8 0x0024+0x0004 PB Configure Register 1 (Default Value: 0x0000_FFFF) (for R128-S3)

Offset: 0x0024+0x0004			Register Name: PB_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0xf	PB15 function select 0x0: input 0x1: output 0x2: UART1_RX 0x3: SDC_CMD 0x4: SPI0_MISO 0x5: Reserved 0x6: PWM0 0x7: Reserved 0x8: NCSI_MCLK 0xe: PB_EINT15 Other: No effect
27:24	R/W	0xf	PB14 function select 0x0: input 0x1: output 0x2: UART1_TX 0x3: SDC_DATA2 0x4: SPI0_WP 0x5: Reserved 0x6: Reserved 0x7: Reserved 0x8: NCSI_PCLK 0xe: PB_EINT14 Other: No effect
23: 0	/	/	/

9.7.5.9 0x0010 PA Data Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: PA_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0010			Register Name: PA_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	DATA 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.

9.7.5.10 0x0024 + 0x0010 PB Data Register (Default Value: 0x0000_0000)

Offset: 0x0024 + 0x0010			Register Name: PB_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15: 0	R/W	0x0	DATA 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.

9.7.5.11 0x0014 PA Multi-Driving Register 0 (Default Value: 0x0000_5555)

Offset: 0x0014			Register Name: PA_DRV0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	R/W	0x1	PA15 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
29:28	R/W	0x1	PA14 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
27:26	R/W	0x1	PA13 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
25:24	R/W	0x1	PA12 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
23:22	R/W	0x1	PA11 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
21:20	R/W	0x1	PA10 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3

Offset: 0x0014			Register Name: PA_DRV0
Bit	**Read/Write**	**Default/Hex**	**Description**
19:18	R/W	0x1	PA9 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
17:16	R/W	0x1	PA8 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
15:14	R/W	0x1	PA7 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
13:12	R/W	0x1	PA6 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
11:10	R/W	0x1	PA5 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
9:8	R/W	0x1	PA4 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
7:6	R/W	0x1	PA3 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
5:4	R/W	0x1	PA2 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
3:2	R/W	0x1	PA1 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
1: 0	R/W	0x1	PA0 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3

9.7.5.12 0x0018 PA Multi-Driving Register 1 (Default Value: 0x0000_5555)

Offset: 0x0018			Register Name: PA_DRV1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	R/W	0x1	PA31 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
29:28	R/W	0x1	PA30 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3

Offset: 0x0018			Register Name: PA_DRV1
Bit	**Read/Write**	**Default/Hex**	**Description**
27:26	R/W	0x1	PA29 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
25:24	R/W	0x1	PA28 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
23:22	R/W	0x1	PA27 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
21:20	R/W	0x1	PA26 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
19:18	R/W	0x1	PA25 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
17:16	R/W	0x1	PA24 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
15:14	R/W	0x1	PA23 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
13:12	R/W	0x1	PA22 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
11:10	R/W	0x1	PA21 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
9:8	R/W	0x1	PA20 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
7:6	R/W	0x1	PA19 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
5:4	R/W	0x1	PA18 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
3:2	R/W	0x1	PA17 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
1: 0	R/W	0x1	PA16 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3

9.7.5.13 0x0024+ 0x0018 PB Multi-Driving Register 0 (Default Value: 0x0000_5555)

Offset: 0x0024+ 0x0018			Register Name: PB_DRV0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	R/W	0x1	PB15 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
29:28	R/W	0x1	PB14 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
27:26	R/W	0x1	PB13 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
25:24	R/W	0x1	PB12 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
23:22	R/W	0x1	PB11 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
21:20	R/W	0x1	PB10 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
19:18	R/W	0x1	PB9 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
17:16	R/W	0x1	PB8 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
15:14	R/W	0x1	PB7 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
13:12	R/W	0x1	PB6 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
11:10	R/W	0x1	PB5 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
9:8	R/W	0x1	PB4 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
7:6	R/W	0x1	PB3 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3

Offset: 0x0024+ 0x0018			Register Name: PB_DRV0
Bit	**Read/Write**	**Default/Hex**	**Description**
5:4	R/W	0x1	PB2 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
3:2	R/W	0x1	PB1 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3
1: 0	R/W	0x1	PB0 DRV_STRENGTH 00: Level 0 01: Level 1 10: Level 2 11: Level 3

9.7.5.14 0x001C PA Pull Register 0 (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: PA_PULL0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	R/W	0x0	PA15 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
29:28	R/W	0x0	PA14 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
27:26	R/W	0x0	PA13 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
25:24	R/W	0x0	PA12 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
23:22	R/W	0x0	PA11 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
21:20	R/W	0x0	PA10 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
19:18	R/W	0x0	PA9 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
17:16	R/W	0x0	PA8 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
15:14	R/W	0x0	PA7 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /

Offset: 0x001C			Register Name: PA_PULL0
Bit	**Read/Write**	**Default/Hex**	**Description**
13:12	R/W	0x0	PA6 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
11:10	R/W	0x0	PA5 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
9:8	R/W	0x0	PA4 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
7:6	R/W	0x0	PA3 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
5:4	R/W	0x0	PA2 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
3:2	R/W	0x0	PA1 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
1: 0	R/W	0x0	PA0 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /

9.7.5.15 0x001C PA Pull Register 1 (Default Value: 0x0000_0000)

Offset: 0x1C			Register Name: PA_PULL1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	R/W	0x0	PA31 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
29:28	R/W	0x0	PA30 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
27:26	R/W	0x0	PA29 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
25:24	R/W	0x0	PA28 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
23:22	R/W	0x0	PA27 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /

Offset: 0x1C			Register Name: PA_PULL1
Bit	**Read/Write**	**Default/Hex**	**Description**
21:20	R/W	0x0	PA26 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
19:18	R/W	0x0	PA25 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
17:16	R/W	0x0	PA24 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
15:14	R/W	0x0	PA23 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
13:12	R/W	0x0	PA22 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
11:10	R/W	0x0	PA21 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
9:8	R/W	0x0	PA20 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
7:6	R/W	0x0	PA19 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
5:4	R/W	0x0	PA18 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
3:2	R/W	0x0	PA17 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
1: 0	R/W	0x0	PA16 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /

9.7.5.16 0x0024+ 0x001C PB Pull Register 0 (Default Value: 0x0000_0000)

Offset: 0X0024+ 0x001C			Register Name: PB_PULL0
Bit	**Read/Write**	**Default**	**Description**
31:30	R/W	0x0	PB15 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /

Offset: 0X0024+ 0x001C			Register Name: PB_PULL0
Bit	**Read/Write**	**Default**	**Description**
29:28	R/W	0x0	PB14 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
27:26	R/W	0x0	PB13 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
25:24	R/W	0x0	PB12 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
23:22	R/W	0x0	PB11 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
21:20	R/W	0x0	PB10 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
19:18	R/W	0x0	PB9 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
17:16	R/W	0x0	PB8 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
15:14	R/W	0x0	PB7 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
13:12	R/W	0x0	PB6 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
11:10	R/W	0x0	PB5 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
9:8	R/W	0x0	PB4 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
7:6	R/W	0x0	PB3 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
5:4	R/W	0x0	PB2 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /
3:2	R/W	0x0	PB1 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /

Offset: 0X0024+ 0x001C			Register Name: PB_PULL0
Bit	**Read/Write**	**Default**	**Description**
1: 0	R/W	0x0	PB0 PULL SELECT 00: No PULL 01: Pull-Up 10: Pull-Down 11: /

9.7.5.17 0x0200 + 0x0000 PA External Interrupt Configure 0 Register (Default Value: 0x0000_0000)

Offset: 0x0200 + 0x0000			Register Name: PA_EINT_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	PA7 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
27:24	R/W	0x0	PA6 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
23:20	R/W	0x0	PA5 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
19:16	R/W	0x0	PA4 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /

Offset: 0x0200 + 0x0000			Register Name: PA_EINT_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
15:12	R/W	0x0	PA3 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
11:8	R/W	0x0	PA2 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
7:4	R/W	0x0	PA1 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
3: 0	R/W	0x0	PA0 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /

9.7.5.18 0x0200 + 0x0004 PA External Interrupt Configure 1 Register (Default Value: 0x0000_0000)

Offset: 0x0200 + 0x0004			Register Name: PA_EINT_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	PA15 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /

Offset: 0x0200 + 0x0004			Register Name: PA_EINT_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
27:24	R/W	0x0	PA14 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
23:20	R/W	0x0	PA13 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
19:16	R/W	0x0	PA12 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
15:12	R/W	0x0	PA11 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
11:8	R/W	0x0	PA10 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
7:4	R/W	0x0	PA9 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /

Offset: 0x0200 + 0x0004			Register Name: PA_EINT_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
3: 0	R/W	0x0	PA8 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /

9.7.5.19 0x0200 + 0x0008 PA External Interrupt Configure 2 Register (Default Value: 0x0000_0000)

Offset: 0x0200 + 0x0008			Register Name: PA_EINT_CFG2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	PA23 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
27:24	R/W	0x0	PA22 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
23:20	R/W	0x0	PA21 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
19:16	R/W	0x0	PA20 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /

Offset: 0x0200 + 0x0008			Register Name: PA_EINT_CFG2
Bit	**Read/Write**	**Default/Hex**	**Description**
15:12	R/W	0x0	PA19 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
11:8	R/W	0x0	PA18 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
7:4	R/W	0x0	PA17 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
3: 0	R/W	0x0	PA16 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /

9.7.5.20 0x0200 + 0x000C PA External Interrupt Configure 3 Register (Default Value: 0x0000_0000)

Offset: 0x0200 + 0x000C			Register Name: PA_EINT_CFG3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	PA31 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /

Offset: 0x0200 + 0x000C			Register Name: PA_EINT_CFG3
Bit	**Read/Write**	**Default/Hex**	**Description**
27:24	R/W	0x0	PA30 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
23:20	R/W	0x0	PA29 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
19:16	R/W	0x0	PA28 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
15:12	R/W	0x0	PA27 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
11:8	R/W	0x0	PA26 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
7:4	R/W	0x0	PA25 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /

Offset: 0x0200 + 0x000C			Register Name: PA_EINT_CFG3
Bit	**Read/Write**	**Default/Hex**	**Description**
3: 0	R/W	0x0	PA24 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /

9.7.5.21 0x0200 + 0x0020 + 0x0000 PB External Interrupt Configure 0 Register (Default Value: 0x0000_0000)

Offset: 0x0200 +0x020 + 0x0000			Register Name: PB_EINT_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	PB7 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
27:24	R/W	0x0	PB6 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
23:20	R/W	0x0	PB5 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
19:16	R/W	0x0	PB4 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /

Offset: 0x0200 +0x020 + 0x0000			Register Name: PB_EINT_CFG0
Bit	**Read/Write**	**Default/Hex**	**Description**
15:12	R/W	0x0	PB3 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
11:8	R/W	0x0	PB2 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
7:4	R/W	0x0	PB1 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
3: 0	R/W	0x0	PB0 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /

9.7.5.22 0x0200 + 0x0020 + 0x0004 PB External Interrupt Configure 1 Register (Default Value: 0x0000_0000)

Offset: 0x0200 +0x0020 +0x0004			Register Name: PB_EINT_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	PB15 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /

Offset: 0x0200 +0x0020 +0x0004			Register Name: PB_EINT_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
27:24	R/W	0x0	PB14 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
23:20	R/W	0x0	PB13 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
19:16	R/W	0x0	PB12 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
15:12	R/W	0x0	PB11 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
11:8	R/W	0x0	PB10 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
7:4	R/W	0x0	PB9 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /

Offset: 0x0200 +0x0020 +0x0004			Register Name: PB_EINT_CFG1
Bit	**Read/Write**	**Default/Hex**	**Description**
3: 0	R/W	0x0	PB8 EINT_MODE SELECT 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /

9.7.5.23 0x0200 + 0x0010 PA External Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0200 + 0x0010			Register Name: PA_EINT_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PA31 EINT_IRQEN 0: Disable 1: Enable
30	R/W	0x0	PA30 EINT_IRQEN 0: Disable 1: Enable
29	R/W	0x0	PA29 EINT_IRQEN 0: Disable 1: Enable
28	R/W	0x0	PA28 EINT_IRQEN 0: Disable 1: Enable
27	R/W	0x0	PA27 EINT_IRQEN 0: Disable 1: Enable
26	R/W	0x0	PA26 EINT_IRQEN 0: Disable 1: Enable
25	R/W	0x0	PA25 EINT_IRQEN 0: Disable 1: Enable
24	R/W	0x0	PA24 EINT_IRQEN 0: Disable 1: Enable
23	R/W	0x0	PA23 EINT_IRQEN 0: Disable 1: Enable
22	R/W	0x0	PA22 EINT_IRQEN 0: Disable 1: Enable

Offset: 0x0200 + 0x0010			Register Name: PA_EINT_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
21	R/W	0x0	PA21 EINT_IRQEN 0: Disable 1: Enable
20	R/W	0x0	PA20 EINT_IRQEN 0: Disable 1: Enable
19	R/W	0x0	PA19 EINT_IRQEN 0: Disable 1: Enable
18	R/W	0x0	PA18 EINT_IRQEN 0: Disable 1: Enable
17	R/W	0x0	PA17 EINT_IRQEN 0: Disable 1: Enable
16	R/W	0x0	PA16 EINT_IRQEN 0: Disable 1: Enable
15	R/W	0x0	PA15 EINT_IRQEN 0: Disable 1: Enable
14	R/W	0x0	PA14 EINT_IRQEN 0: Disable 1: Enable
13	R/W	0x0	PA13 EINT_IRQEN 0: Disable 1: Enable
12	R/W	0x0	PA12 EINT_IRQEN 0: Disable 1: Enable
11	R/W	0x0	PA11 EINT_IRQEN 0: Disable 1: Enable
10	R/W	0x0	PA10 EINT_IRQEN 0: Disable 1: Enable
9	R/W	0x0	PA9 EINT_IRQEN 0: Disable 1: Enable
8	R/W	0x0	PA8 EINT_IRQEN 0: Disable 1: Enable

Offset: 0x0200 + 0x0010			Register Name: PA_EINT_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
7	R/W	0x0	PA7 EINT_IRQEN 0: Disable 1: Enable
6	R/W	0x0	PA6 EINT_IRQEN 0: Disable 1: Enable
5	R/W	0x0	PA5 EINT_IRQEN 0: Disable 1: Enable
4	R/W	0x0	PA4 EINT_IRQEN 0: Disable 1: Enable
3	R/W	0x0	PA3 EINT_IRQEN 0: Disable 1: Enable
2	R/W	0x0	PA2 EINT_IRQEN 0: Disable 1: Enable
1	R/W	0x0	PA1 EINT_IRQEN 0: Disable 1: Enable
0	R/W	0x0	PA0 EINT_IRQEN 0: Disable 1: Enable

9.7.5.24 0x0200 + 0X0020 + 0x0010 PB External Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0200 + 0X0020+0x0010			Register Name: PB_EINT_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15	R/W	0x0	PB15 EINT_IRQEN 0: Disable 1: Enable
14	R/W	0x0	PB14 EINT_IRQEN 0: Disable 1: Enable
13	R/W	0x0	PB13 EINT_IRQEN 0: Disable 1: Enable
12	R/W	0x0	PB12 EINT_IRQEN 0: Disable 1: Enable

Offset: 0x0200 + 0X0020+0x0010			Register Name: PB_EINT_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
11	R/W	0x0	PB11 EINT_IRQEN 0: Disable 1: Enable
10	R/W	0x0	PB10 EINT_IRQEN 0: Disable 1: Enable
9	R/W	0x0	PB9 EINT_IRQEN 0: Disable 1: Enable
8	R/W	0x0	PB8 EINT_IRQEN 0: Disable 1: Enable
7	R/W	0x0	PB7 EINT_IRQEN 0: Disable 1: Enable
6	R/W	0x0	PB6 EINT_IRQEN 0: Disable 1: Enable
5	R/W	0x0	PB5 EINT_IRQEN 0: Disable 1: Enable
4	R/W	0x0	PB4 EINT_IRQEN 0: Disable 1: Enable
3	R/W	0x0	PB3 EINT_IRQEN 0: Disable 1: Enable
2	R/W	0x0	PB2 EINT_IRQEN 0: Disable 1: Enable
1	R/W	0x0	PB1 EINT_IRQEN 0: Disable 1: Enable
0	R/W	0x0	PB0 EINT_IRQEN 0: Disable 1: Enable

9.7.5.25 0x0200 + 0x0014 PA External Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0200 + 0x0014			Register Name: PA_EINT_ST
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0200 + 0x0014			Register Name: PA_EINT_ST
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	PA31 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
30	R/W	0x0	PA30 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
29	R/W	0x0	PA29 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
28	R/W	0x0	PA28 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
27	R/W	0x0	PA27 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
26	R/W	0x0	PA26 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
25	R/W	0x0	PA25 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
24	R/W	0x0	PA24 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
23	R/W	0x0	PA23 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
22	R/W	0x0	PA22 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear

Offset: 0x0200 + 0x0014			Register Name: PA_EINT_ST
Bit	**Read/Write**	**Default/Hex**	**Description**
21	R/W	0x0	PA21 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
20	R/W	0x0	PA20 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
19	R/W	0x0	PA19 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
18	R/W	0x0	PA18 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
17	R/W	0x0	PA17 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
16	R/W	0x0	PA16 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
15	R/W	0x0	PA15 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
14	R/W	0x0	PA14 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
13	R/W	0x0	PA13 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
12	R/W	0x0	PA12 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear

Offset: 0x0200 + 0x0014			Register Name: PA_EINT_ST
Bit	**Read/Write**	**Default/Hex**	**Description**
11	R/W	0x0	PA11 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
10	R/W	0x0	PA10 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
9	R/W	0x0	PA9 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
8	R/W	0x0	PA8 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
7	R/W	0x0	PA7 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
6	R/W	0x0	PA6 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
5	R/W	0x0	PA5 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
4	R/W	0x0	PA4 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
3	R/W	0x0	PA3 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
2	R/W	0x0	PA2 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear

Offset: 0x0200 + 0x0014			Register Name: PA_EINT_ST
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x0	PA1 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
0	R/W	0x0	PA0 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear

9.7.5.26 0x0200 + 0X0020 + 0x0014 PB External Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0200 +0X0020+ 0x0014			Register Name: PB_EINT_ST
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15	R/W	0x0	PB15 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
14	R/W	0x0	PB14 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
13	R/W	0x0	PB13 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
12	R/W	0x0	PB12 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
11	R/W	0x0	PB11 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
10	R/W	0x0	PB10 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
9	R/W	0x0	PB9 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear

Offset: 0x0200 +0X0020+ 0x0014			Register Name: PB_EINT_ST
Bit	**Read/Write**	**Default/Hex**	**Description**
8	R/W	0x0	PB8 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
7	R/W	0x0	PB7 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
6	R/W	0x0	PB6 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
5	R/W	0x0	PB5 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
4	R/W	0x0	PB4 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
3	R/W	0x0	PB3 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
2	R/W	0x0	PB2 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
1	R/W	0x0	PB1 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear
0	R/W	0x0	PB0 EINT_IRQST 0: Nothing 1: Irq pending write 1 to clear

9.7.5.27 0x0200 + 0x0018 PA External Interrupt Debounce Register (Default Value: 0x0000_0000)

Offset: 0x0200 + 0x0018			Register Name: PA_DBC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/

Offset: 0x0200 + 0x0018			Register Name: PA_DBC
Bit	**Read/Write**	**Default/Hex**	**Description**
6:4	R/W	0x0	Debounce Clock Pre-Scale n The selected clock is prescaled by 2^n
3:1	/	/	/
0	R/W	0x0	PIO_INT_CLK_SEL 0: LFCLK 1: HFCLK

9.7.5.28 0x0200 + 0X0020 + 0x0018 PB External Interrupt Debounce Register (Default Value: 0x0000_0000)

Offset: 0x0200 + 0X0020+0x0018			Register Name: PB_DBC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/
6:4	R/W	0x0	Debounce Clock Pre-Scale n The selected clock is prescaled by 2^n
3:1	/	/	/
0	R/W	0x0	PIO_INT_CLK_SEL 0: LFCLK 1: HFCLK

9.8 GPADC

9.8.1 Overview

The General Purpose ADC (GPADC) can convert the external signal into a certain proportion of digital value, to realize the measurement of analog signal, which can be applied to power detection and key detection. This ADC is a type of successive approximation register (SAR) A/D converter.

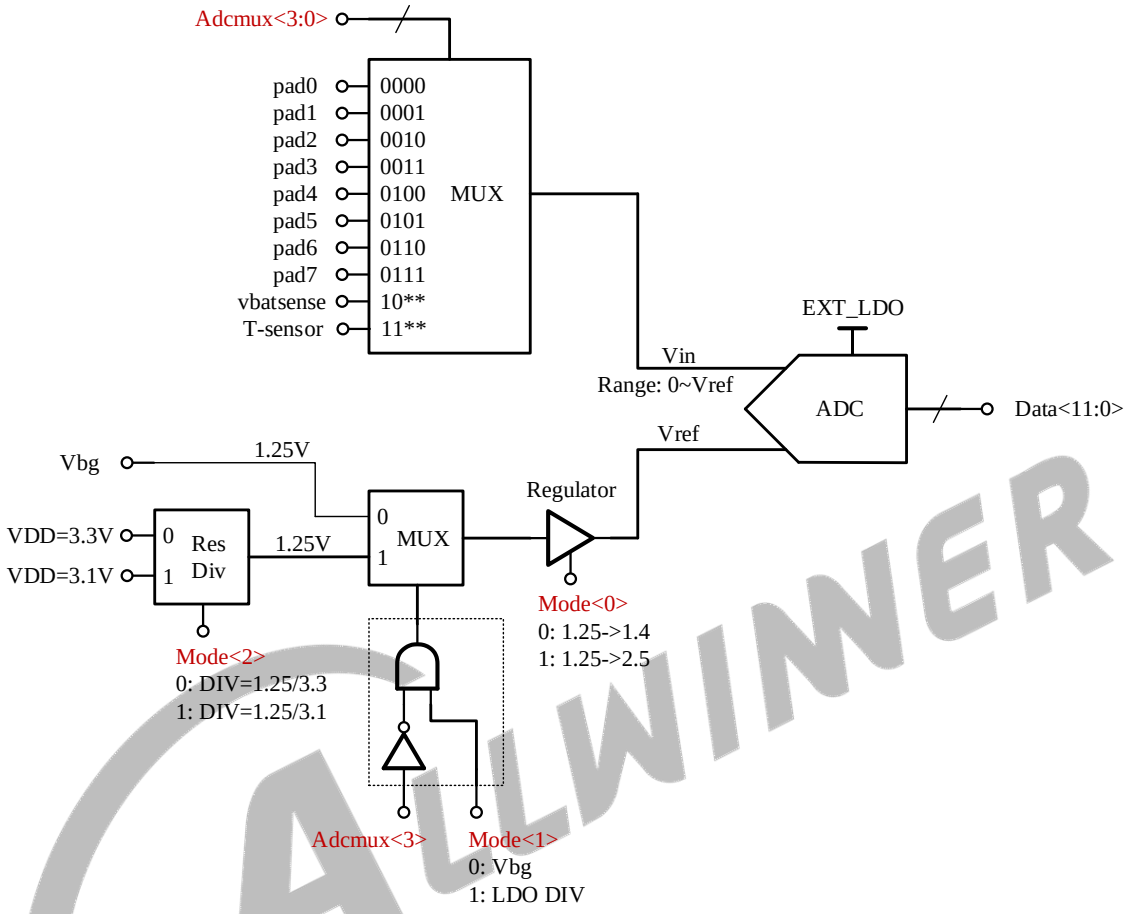
The GPADC has the following features:

- 12-bit Resolution and 7-bit effective SAR type A/D converter
- 9-channel multiplexer including 7 channels general purpose ADC (ADC0-ADC6) and 2 channels special ADC (ADC8, ADC12) for R128-S1 and R128-S2
- 10-channel multiplexer including 8 channels general purpose ADC (ADC0-ADC7) and 2 channels special ADC (ADC8, ADC12) for R128-S3
- The ADC8 is used for VBAT voltage detection and the ADC12 is used for temperature sensor
- 64 FIFO depth of data register
- Power Supply Voltage: 2.5V. Analog Input Range: 0 to 2V
- Maximum Sampling frequency: 1 MHz
- Support self-calibration
- Support data compare and interrupt
- Support four operation modes
  - Single conversion mode
  - Single-cycle conversion mode
  - Continuous conversion mode
  - Outbreak conversion mode

9.8.2 Block Diagram

The following figure shows the block diagram of GPADC.

Figure 9-61 GPADC Block Diagram



9.8.3 External Signals

The following table describes the external signals of the GPADC.

Table 9-23 External Signals

Signal	Description	Pin Number	Type
ADC_CH0	Analog Input	PB0	A
ADC_CH1	Analog Input	PB1	A
ADC_CH2	Analog Input	PB2	A
ADC_CH3	Analog Input	PB3	A
ADC_CH4	Analog Input	PB4	A
ADC_CH5	Analog Input	PB5	A
ADC_CH6	Analog Input	PB14	A
ADC_CH7	Analog Input	PB15	A
VIN_VBAT	VBAT Input	VBAT	P

9.8.4 Working Mode

- **Single conversion mode**

The GPADC completes one conversion in a specified channel, the converted data is updated at the data register of the corresponding channel.

- **Single-cycle conversion mode**

The GPADC completes one-cycle conversion in a specified channel, the converted data is updated at the data registers of the corresponding channel.

- **Continuous conversion mode**

The GPADC has continuous conversion in a specified channel until the software stops, the converted data is updated at the data registers of the corresponding channel.

- **Burst conversion mode**

The GPADC samples and converts in a specified channel, and sequentially stores the results in FIFO.

9.8.5 Register List

Module Name	Base Address
GPADC	0x4004A000

Register Name	Offset Address	Description
GP_SR_CON	0x0000	GPADC Sample Rate Configure Register
GP_CTRL	0x0004	GPADC Control Register
GP_CS_EN	0x0008	GPADC Compare and Select Enable Register
GP_FIFO_INTC	0x000C	GPADC FIFO Interrupt Control Register
GP_FIFO_INTS	0x0010	GPADC FIFO Interrupt Status Register
GP_FIFO_DATA	0x0014	GPADC FIFO Data Register
GP_CDATA	0x0018	GPADC Calibration Data Register
GP_DATAL_INTC	0x0020	GPADC Data Low Interrupt Configure Register
GP_DATAH_INTC	0x0024	GPADC Data High Interrupt Configure Register
GP_DATA_INTC	0x0028	GPADC Data Interrupt Configure Register
GP_DATAL_INTS	0x0030	GPADC Data Low Interrupt Status Register
GP_DATAH_INTS	0x0034	GPADC Data High Interrupt Status Register
GP_DATA_INTS	0x0038	GPADC Data Interrupt Status Register
GP_CHn_CMP_DATA	0x0040 + 4*n	GPADC CH n Compare Data Register
GP_CHn_DATA	0x0080 + 4 * n	GPADC CH n Data Register

9.8.6 Register Description

9.8.6.1 0x0000 GPADC Sample Rate Configure Register (Default Value: 0x0207_0034)

Offset: 0x0000	Register Name: GP_SR_CON

Bit	Read/Write	Default/Hex	Description
31: 16	R/W	0x207	FS_DIV ADC Sampling Clock Divider One sampling cycle= ADC acquire time + conversion time (+ idle time) The ADC conversion time is 17 HOSC cycles, idle time=FS_DIV-TACQ-17. Usually, FS_DIV should be greater than TACQ+17. Sampling rate=HOSC/(FS_DIV+1) Otherwise, sampling rate=HOSC/(TACQ+18) **Note**: *Set the ADC sampling rate to 50kHz, when HOSC=26MHz, the FS_DIV should be 0x207; when HOSC=40MHz, the FS_DIV should be 0x31F.*
15: 0	R/W	0x34	TACQ ADC acquire time= (TACQ+1)/HOSC **Note**: *Set the ADC acquire time to 2us, when HOSC=26MHz, the TACQ should be 0x33; when HOSC=40MHz, the TACQ should be 0x4F.*

9.8.6.2 0x0004 GPADC Control Register (Default Value: 0x00800000)

Offset: 0x0004			Register Name: GP_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/ W	0x0	ADC_FIRST_DLY. ADC First Convert Delay setting, ADC conversion of each channel is delayed by N samples
23:22	/	/	/
21:20	R/W	0x0	ADC_OP_BIAS. (Adjust the bandwidth of the ADC amplifier) ADC OP Bias
19:18	R/W	0x0	GPADC Work Mode 00: Single conversion mode 01: Single-cycle conversion mode 10: Continuous conversion mode 11: Burst conversion mode
17	R/W	0x0	ADC_CALI_EN. ADC Calibration 1: Start Calibration, it is clear to 0 after calibration
16	R/W	0x0	ADC_EN. ADC Function Enable 0: Disable 1: Enable **Note**: The *work mode and the channel number must be set before the ADC_EN bit being set.*
15:8	/	/	/

Offset: 0x0004			Register Name: GP_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
7	R/W	0x0	LP_TEMPSENS Temperature sensor lowpower mode enable, can reduce current to 100uA and ensure 100k sample rate
6	R/W	0x0	EN_ TEMPSENS Temperature sensor enable **Note**: The *12th channel is designed for temperature sensor and this bit should be set to 1 before enable the CH12*
5	R/W	0x0	VBAT_DET_EN 0: Disable 1: Enable **Note**: The *8th channel is designed for VBAT Voltage Detection and this bit should be set to 1 before enable the CH8*
4	R/W	0x0	LP_EN 0: Disable 1: Enable
3:1	R/W	0x0	VREF_MODE_SEL 000: Vref is 1.4V from BG, choose when vdd is 1.6V~3V 001: Vref is 2.5V from BG, choose when vdd is 3V~3.6V 010: Vref is 2.5V from divider of EXT_LDO, choose when vdd=3.3V 110: Vref is 2.5V from divider of EXT_LDO, choose when vdd=3.1V others: Reserved
0	R/W	0x0	ADC_LDO_EN 0: Disable 1: Enable **Note**: *Before enabling the ADC function, you must enable the LDO-ADC firstly.*

9.8.6.3 0x0008 GPADC Compare and Select Enable Register (Default Value: 0x00000000)

Offset: 0x0008			Register Name: GP _CS_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
others	/	/	/
[n+16] (n = 0-8, 12)	R/W	0x0	ADC_CHn_CMP_EN Channel n Compare Enable 0: Disable 1: Enable
others	/	/	/

Offset: 0x0008			Register Name: GP _CS_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
[n] (n = 0-8, 12)	R/W	0x0	ADC_CHn_SELECT. Analog input channel n Select 0: Disable 1: Enable

9.8.6.4 0x000C GPADC FIFO Interrupt Control Register (Default Value: 0x0000_0F00)

Offset: 0x000C			Register Name: GP_FIFO_INTC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:19	/	/	/
18	R/W	0x0	FIFO_DATA_DRQ_EN ADC FIFO Data DRQ Enable 0: Disable 1: Enable
17	R/W	0x0	FIFO_OVERRUN_IRQ_EN. ADC FIFO Over Run IRQ Enable 0: Disable 1: Enable
16	R/W	0x0	FIFO_DATA_IRQ_EN. ADC FIFO Data Available IRQ Enable 0: Disable 1: Enable
15:14	/	/	/
13:8	R/W	0x1F	FIFO_TRIG_LEVEL. Interrupt and DMA request trigger level for ADC Trigger Level = TXTL + 1
7:5	/	/	/
4	R/W1C	0x0	FIFO_FLUSH. ADC FIFO Flush Write ‘1’ to flush TX FIFO, self clear to ‘0’
3: 0	/	/	/

9.8.6.5 0x0010 GPADC FIFO Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: GP_FIFO_INTS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/

Offset: 0x0010			Register Name: GP_FIFO_INTS
Bit	**Read/Write**	**Default/Hex**	**Description**
17	R/W1C	0x0	FIFO_OVERRUN_PENDING. ADC FIFO Over Run IRQ pending 0: No Pending IRQ 1: FIFO Overrun Pending IRQ Write '1' to clear this interrupt or automatic clear if interrupt condition fails
16	R/W1C	0x0	FIFO_DATA_PENDING. ADC FIFO Data Available Pending Bit 0: No Pending IRQ 1: FIFO Available Pending IRQ Write '1' to clear this interrupt or automatic clear if interrupt condition fails
15:14	/	/	/
13:8	R	0x0	RXA_CNT. ADC FIFO available Sample Word Counter
7: 0	/	/	/

9.8.6.6 0x0014 GPADC FIFO Data Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: GP_FIFO_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11: 0	R/W	x	GP_FIFO_DATA GPADC Data in FIFO

9.8.6.7 0x0018 GPADC Calibration Data Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: GP_CDATA
Bit	**Read/Write**	**Default/Hex**	**Description**
11: 0	R/W	0x0	GP_CDATA GPADC Calibration Data

9.8.6.8 0x0020 GPADC Low Interrupt Configure Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: GP_DATAL_INTC
Bit	**Read/Write**	**Default/Hex**	**Description**
others	/	/	/
[n] (n = 0-8, 12)	R/W	0x0	CHn_ LOW_IRQ_EN 0: Disable 1: Enable

9.8.6.9 0x0024 GPADC HIGH Interrupt Configure Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: GP_DATAH_INTC
Bit	**Read/Write**	**Default/Hex**	**Description**
others	/	/	/
[n] (n = 0-8, 12)	R/W	0x0	CHn_ HIGH_IRQ_EN 0: Disable 1: Enable

9.8.6.10 0x0028 GPADC DATA Interrupt Configure Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: GP_DATA_INTC
Bit	**Read/Write**	**Default/Hex**	**Description**
others	/	/	/
[n] (n = 0-8, 12)	R/W	0x0	CHn_DATA_IRQ_EN 0: Disable 1: Enable

9.8.6.11 0x0030 GPADC Low Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: GP_DATAL_INTS
Bit	**Read/Write**	**Default/Hex**	**Description**
others	/	/	/
[n] (n = 0-8, 12)	R/W	0x0	CHn_ LOW_PENGDING 1: Channel n Voltage Low Available Pending IRQ Write '1' to clear this interrupt or automatic clear if interrupt condition fails

9.8.6.12 0x0034 GPADC High Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: GP_DATAH_INTS
Bit	**Read/Write**	**Default/Hex**	**Description**
others	/	/	/
[n] (n = 0-8, 12)	R/W	0x0	CHn_ HIGH_PENGDING 1: Channel n Voltage High Available Pending IRQ Write '1' to clear this interrupt or automatic clear if interrupt condition fails

9.8.6.13 0x0038 GPADC Data Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: GP_DATA_INTS
Bit	**Read/Write**	**Default/Hex**	**Description**
others	/	/	/

Offset: 0x0038			Register Name: GP_DATA_INTS
Bit	**Read/Write**	**Default/Hex**	**Description**
[n] (n = 0-8, 12)	R/W	0x0	CHn_DATA_ PENGDING 0: No Pending IRQ 1: Channel n Data Available Pending IRQ Write '1' to clear this interrupt or automatic clear if interrupt condition fails

9.8.6.14 0x0040 GPADC CHn Compare Data Register (Default Value: 0x0BFF_0400)

Offset: 0x0040 + 4*n (n = 0-8, 12)			Register Name: GP_CHn_CMP_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	/	/	/
27:16	R/W	0xBFF	CHn_CMP_HIG_DATA Channel n Voltage High Value
15:12	/	/	/
11: 0	R/W	0x400	CHn_CMP_LOW_DATA Channel n Voltage Low Value

9.8.6.15 0x0080 GPADC CHn Data Register (Default Value: 0x0000_0000)

Offset: 0x0080 + 4*n (n = 0-8, 12)			Register Name: GP_CHn_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11: 0	R	0x0	GP_CHn_DATA Channel n Data

9.9 PWM

9.9.1 Overview

The Pulse Width Modulation (PWM) module can output the configurable PWM waveforms and measure the external input waveforms.

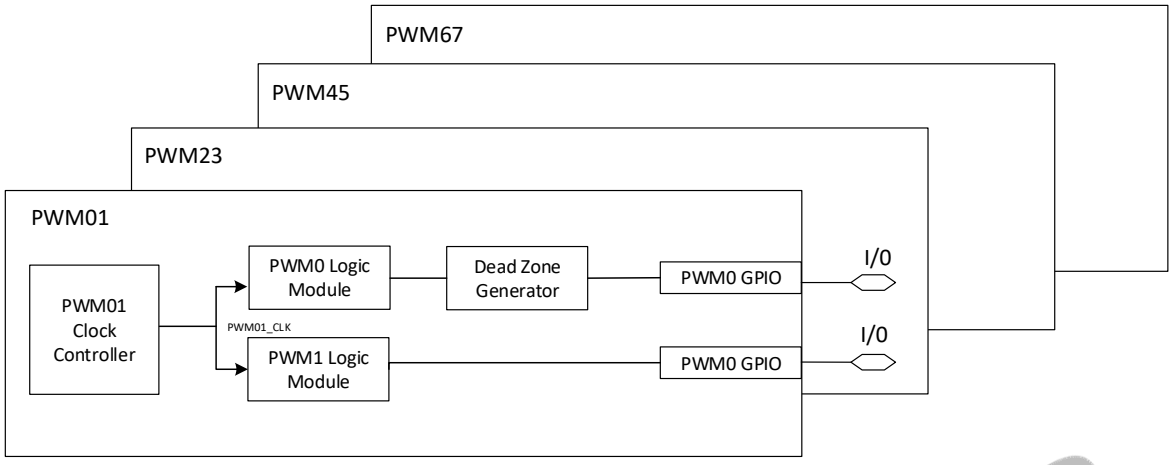
The PWM has the following features:

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

9.9.2 Block Diagram

The following figure shows the block diagram of PWM.

Figure 9-62 PWM Block Diagram



Each PWM pair consists of 1 clock module, 2 timer logic module, and 1 programmable dead-zone generator.

9.9.3 Functional Description

9.9.3.1 External Signals

The following table describes the external signals of the PWM.

Table 9-24 PWM External Signals

Signal	Description	Type
PWM_0	Pulse Width Module Channel0	I/O
PWM_1	Pulse Width Module Channel1	I/O
PWM_2	Pulse Width Module Channel2	I/O
PWM_3	Pulse Width Module Channel3	I/O
PWM_4	Pulse Width Module Channel4	I/O
PWM_5	Pulse Width Module Channel5	I/O
PWM_6	Pulse Width Module Channel6	I/O
PWM_7	Pulse Width Module Channel7	I/O

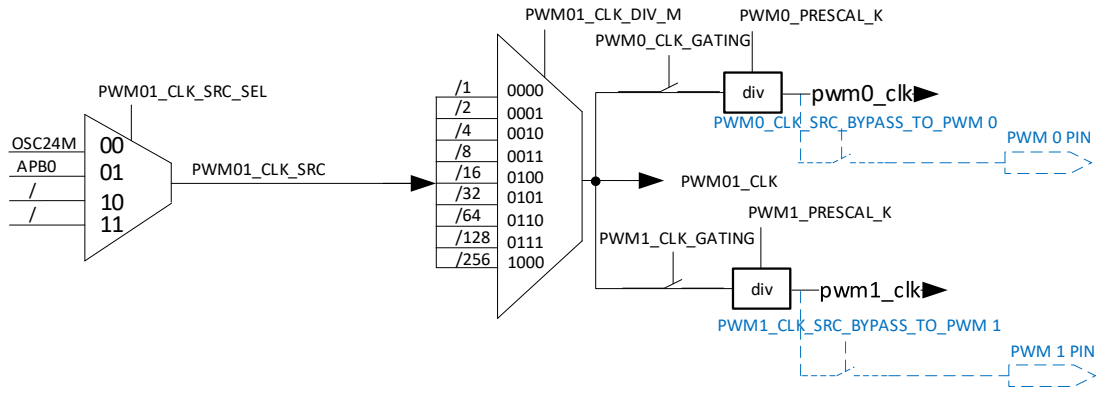
9.9.3.2 Typical Application

- Suitable for display device, such as LCD
- Suitable for electric motor control

9.9.3.3 Clock Controller

Using PWM01 as an example. The other PWM pairs are the same as PWM01.

Figure 9-63 PWM01 Clock Controller Diagram



The clock controller of each PWM pair includes clock source select (PWM01_CLK_SRC), 1~256 scaler (PWM01_CLK_DIV_M). Each PWM channel has the secondary frequency division (PWM_PRESCAL_K), clock source bypass (PWMx_CLK_BYPASS) and clock switch (PWMx_CLK_GATING).

The clock sources have HOSC and APB0. The HOSC comes from the external high-frequency oscillator; the APB0 is APB0 bus clock.

The bypass function of the clock source is that the clock source directly accesses PWM output, the PWM output waveform is the waveform of the clock controller output. The BYPASS gridlines in the above figure indicate the bypass function of the clock source, see Figure 10-85 for the details about implement. At last, the output clock of the clock controller is sent to the PWM logic module.

9.9.3.4 PWM Output

Taking PWM01 as an example, the following figure indicates the PWM01 output logic diagram. The logic diagrams of other PWM pairs are the same as PWM01.

The timer logic module of PWM consists of one 16-bit up-counter (PCNTR) and three 16-bit parameters (PWM_ENTIRE_CYCLE, PWM_ACT_CYCLE, PWM_COUNTER_START). The PWM_ENTIRE_CYCLE is used to control the PWM cycle, the PWM_ACT_CYCLE is used to control the duty-cycle, the PWM_COUNTER_START is used to control the output phase (multi-channel synchronization work requirements).

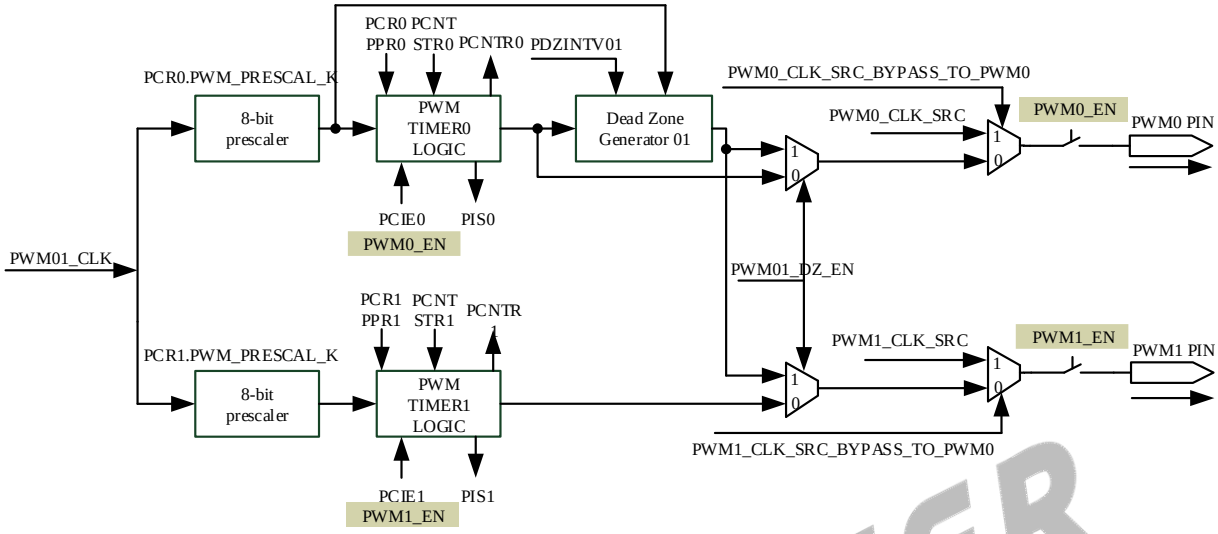
The PWM_ENTIRE_CYCLE and the PWM_ACT_CYCLE support the cache load, after PWM output is enabled, the register values of the PWM_ENTIRE_CYCLE and the PWM_ACT_CYCLE can be changed anytime, the changed value caches into the cache register. When the PCNTR counter outputs a period of PWM waveform, the value of the cache register can be updated for the PCNTR control. The purpose of the cache load is to avoid the unstable PWM output waveform with the burred feature when updating the values of the PWM_ENTIRE_CYCLE and PWM_ACT_CYCLE.

The PWM supports cycle and pulse waveform output.

Cycle mode: The PWM outputs the setting PWM waveform continually, that is, the output waveform is a continuous PWM square wave.

Pulse mode: After setting the PWM_PUL_NUM parameter, the PWM outputs (PWM_PULNUM+1) periods of PWM waveform, that is, the waveform with several pulses are output.

Figure 9-64 PWM01 Output Logic Module Diagram



9.9.3.5 Period, Duty-cycle and Phase of PWM Output Waveform

The period, duty-cycle, and phase of PWM output waveform are decided by the PCNTR, PWM_ENTIRE_CYCLE, PWM_ACT_CYCLE, and PWM_COUNTER_START. The rules are as follows.

- PCNTR= (PCNTR==PWM_ENTIRE_CYCLE)?0: PCNTR + 1
- PCNTR starts to count by PWM_COUNTER_START, the counter of a PWM period is (PWM_ENTIRE_CYCLE+1).
- PCNTR > (PWM_ENTIRE_CYCLE - PWM_ACT_CYCLE), output “active state”
- PCNTR <= (PWM_ENTIRE_CYCLE - PWM_ACT_CYCLE), output “~ (active state)”

Active state of PWM0 channel is high level (PCR0. PWM_ACT_STA = 1)

When PCNTR0 > (PPR0. PWM_ENTIRE_CYCLE - PPR0.PWM_ACT_CYCLE), then PWM0 outputs 1 (high level).

When PCNTR0 <= (PPR0. PWM_ENTIRE_CYCLE - PPR0.PWM_ACT_CYCLE), then PWM0 outputs 0 (low level).

The formula of the output period and the duty-cycle for PWM are as follows.

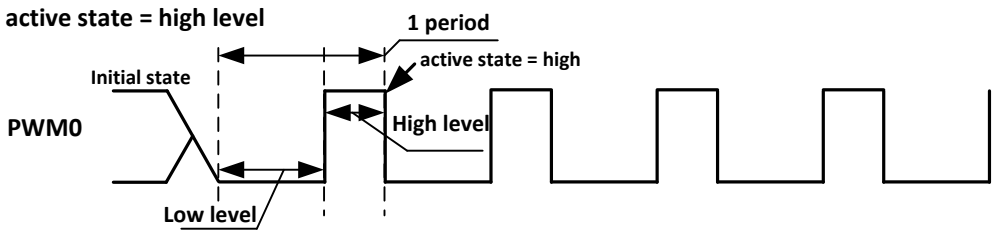
T_{period} = (PWM01_CLK / PWM0_PRESCALE_K)$^{-1}$ * (PPR0.PWM_ENTIRE_CYCLE + 1)

$T_{high\text{-}level}$ = (PWM01_CLK / PWM0_PRESCALE_K)$^{-1}$ * PPR0.PWM_ACT_CYCLE

$T_{low\text{-}level}$ = (PWM01_CLK / PWM0_PRESCALE_K)$^{-1}$ * (PPR0.PWM_ENTIRE_CYCLE + 1 - PPR0.PWM_ACT_CYCLE)

Duty-cycle = (high level time) / (1 period time) = $T_{high\text{-}level}$ / T_{period}

Figure 9-65 The Period and Duty-cycle of PWM0 Active State at High Level



Active state of PWM0 channel is low level (PCR0. PWM_ACT_STA = 0)

When PCNTR0 > (PPR0.PWM_ENTIRE_CYCLE - PPR0.PWM_ACT_CYCLE), then PWM0 outputs 0.

When PCNTR0 <= (PPR0.PWM_ENTIRE_CYCLE - PPR0.PWM_ACT_CYCLE), then PWM0 outputs 1.

The formula of the output period and the duty-cycle for PWM are as follows.

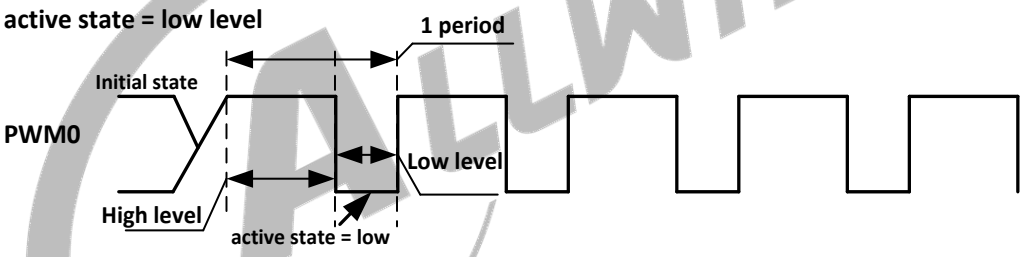
T_{period} = (PWM01_CLK / PWM0_PRESCALE_K)$^{-1}$ * (PPR0.PWM_ENTIRE_CYCLE + 1)

$T_{high\text{-}level}$ = (PWM01_CLK / PWM0_PRESCALE_K)$^{-1}$ * (PPR0.PWM_ENTIRE_CYCLE + 1 - PPR0.PWM_ACT_CYCLE)

$T_{low\text{-}level}$ = (PWM01_CLK / PWM0_PRESCALE_K)$^{-1}$ * PPR0.PWM_ACT_CYCLE

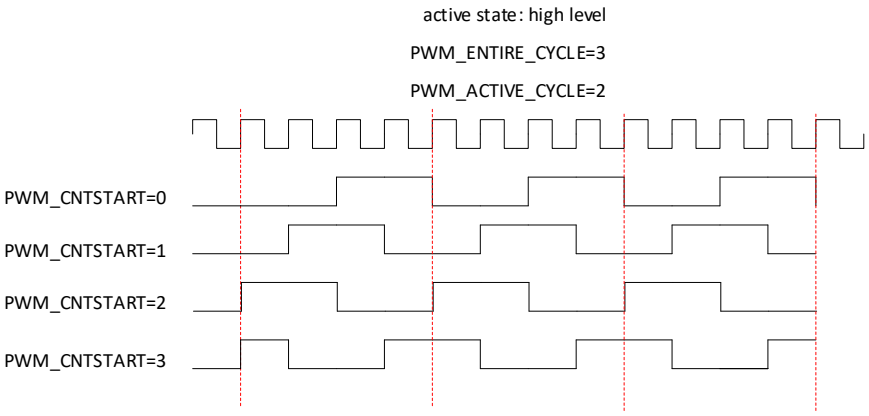
Duty-cycle = (low level time) / (1 period time) = $T_{low\text{-}level}$ / T_{period}

Figure 9-66 The Period and Duty-cycle of PWM0 Active State at Low Level



The counter of PCNTR starts from 0 by default, it can output the pulse control of the waveform by setting PWM_COUNTER_START. The figure is as follows.

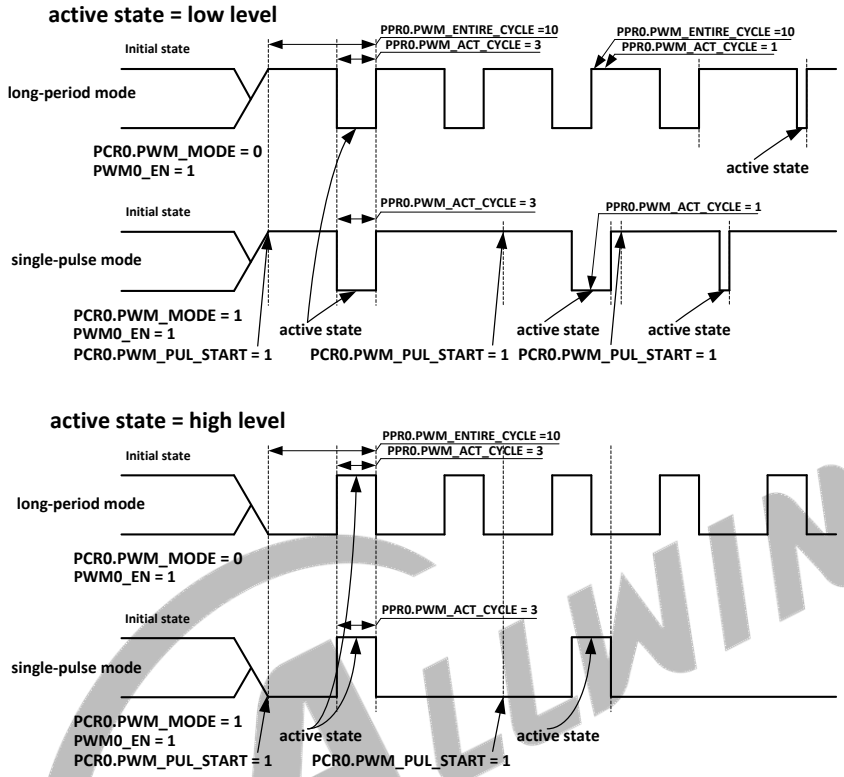
Figure 9-67 Phase of PWM0 Active State at High Level



9.9.3.6 Pulse Mode and Cycle Mode

The PWM output supports pulse mode and cycle mode. PWM in pulse mode outputs one pulse waveform, but PWM in cycle mode outputs continuous waveform. The following figure shows the PWM output waveform in pulse mode and cycle mode.

Figure 9-68 PWM0 Output Waveform in Pulse Mode and Cycle Mode



Each channel of the PWM module supports the PWM output of pulse mode and cycle mode, the active state of the PWM output waveform can be programmed to control.

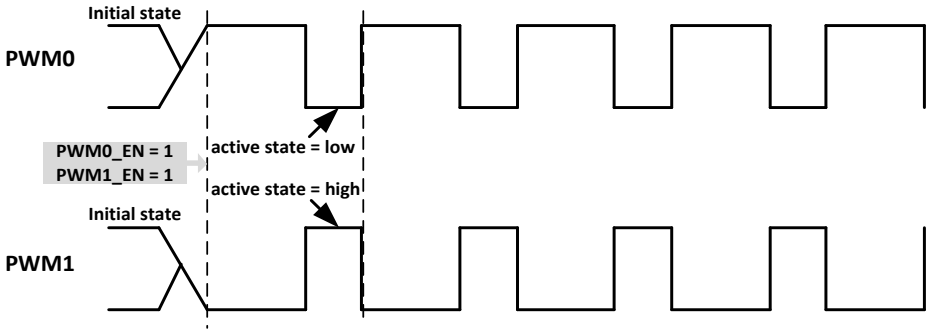
When PWM_MODE is 0, the PWM0 outputs in cycle mode. When PWM_MODE is 1, the PWM0 outputs in pulse mode.

Specifically, in pulse mode, after the PWM0 channel is enabled, PWM_PUL_START needs to be set to 1 when the PWM0 needs to output pulse waveform. After completing the output, PWM_PUL_START can be cleared to 0 by hardware. The next setting to 1 can be operated after PWM_PUL_START is cleared. In addition, the PPR needs to be configured. Only after the PWM_PERIOD_RDY is set to 0, PWM_PUL_START can be set to 1.

9.9.3.7 Complementary Pair Output

Every PWM pair supports complementary pair output and PWM pair with dead-time. The following figure shows the complementary pair output of PWM01.

Figure 9-69 PWM01 Complementary Pair Output



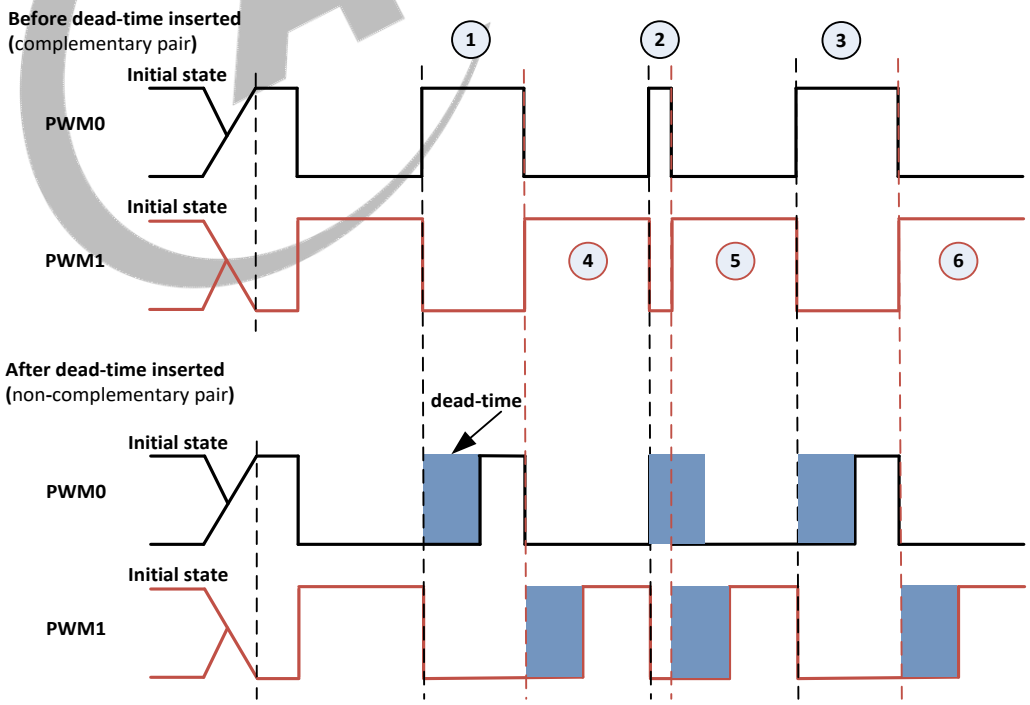
The complementary pair output needs to satisfy the following conditions:

- PWM0 and PWM1 have the same clock divider, frequency, duty-cycle, and phase
- PWM0 and PWM1 have an opposite active state
- Enable the clock gating of PWM0 and PWM1 at the same time
- Enable the waveform output of PWM0 and PWM1 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 the PWM pair is enabled, the PWM01 output waveform is decided by PWM timer logic and Dead Zone Generator. The following figure shows the output waveform.

Figure 9-70 Dead-time Output Waveform



The PWM waveform before the insertion of dead-time indicates a complementary waveform pair of non-inserted dead-time in Dead Zone Generator 01.

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 01. The PWM waveform pair at last outputs to PWM0 pin and PWM1 pin.

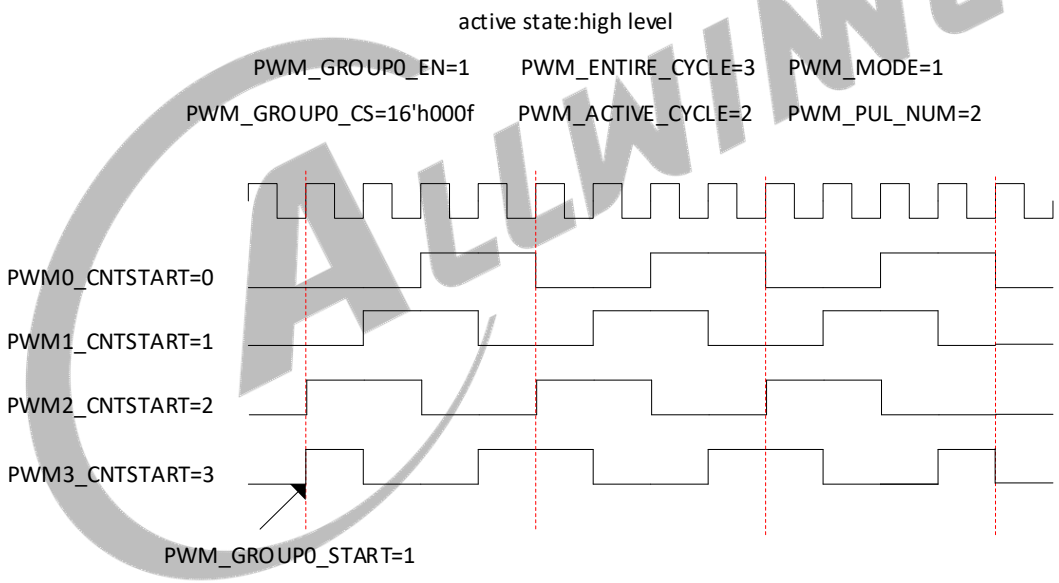
For the complementary pair of Dead Zone Generator 01, the principle of inserting dead-time is that to insert dead-time as soon as the rising edge came. If the high level time for mark② in the above figure is less than dead-time, then dead-time will override the high level. The setting of dead-time needs to consider the period and the duty-cycle of the output waveform. The dead-time formula is defined as follows:

Dead-time = (PWM01_CLK / PWM0_PRESCALE_K)$^{-1}$ * PDZINTV01

9.9.3.9 PWM Group Mode

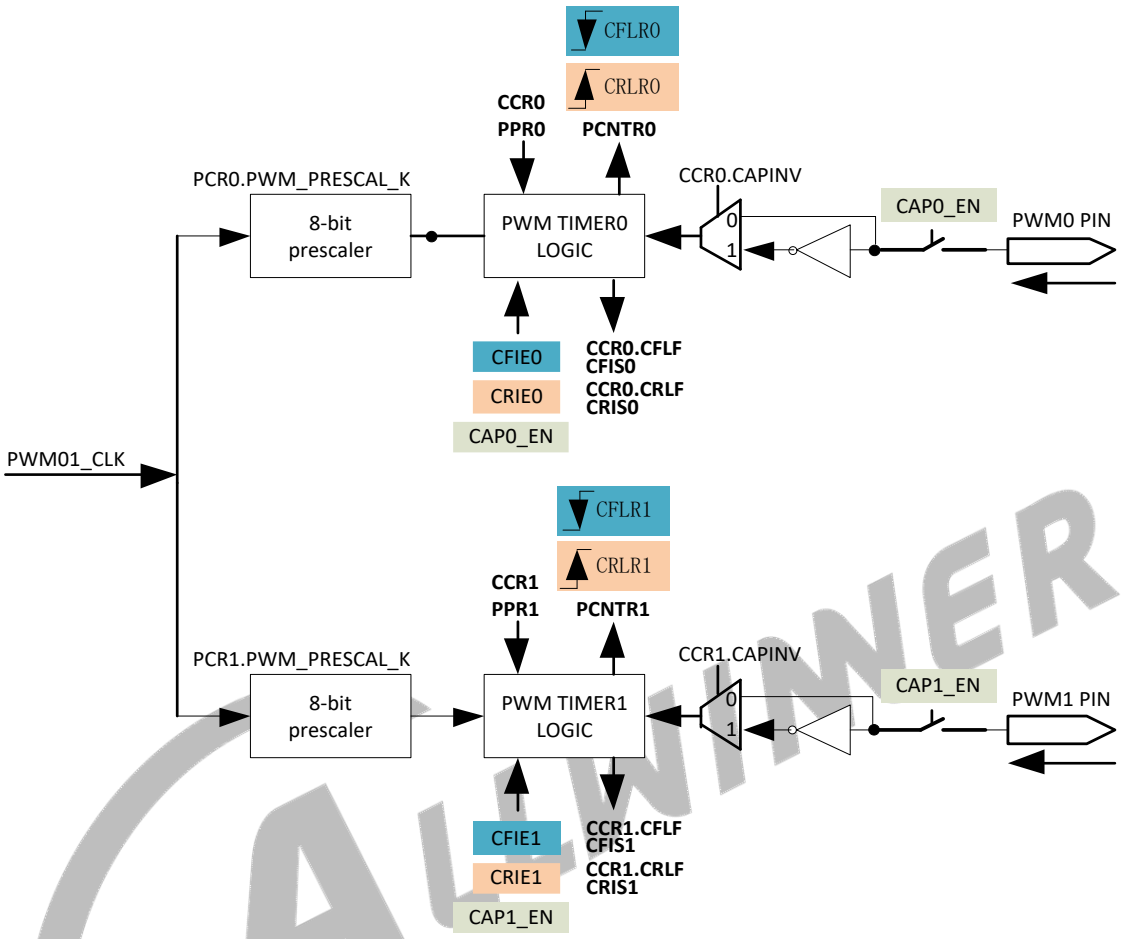
Taking PWM Group0 as an example. The same group of PWM channel is selected to work by PGR0.CS; the same PWM_ENTIRE_CYCLE, PWM_ACT_CYCLE are set by the same clock configuration; the different PWM_COUNTER_START can output PWM group signals with the same duty-cycle and the different phase.

Figure 9-71 Group 0~3 PWM Signal Output



9.9.3.10 Capture Input

Figure 9-72 PWM01 Capture Logic Module Diagram



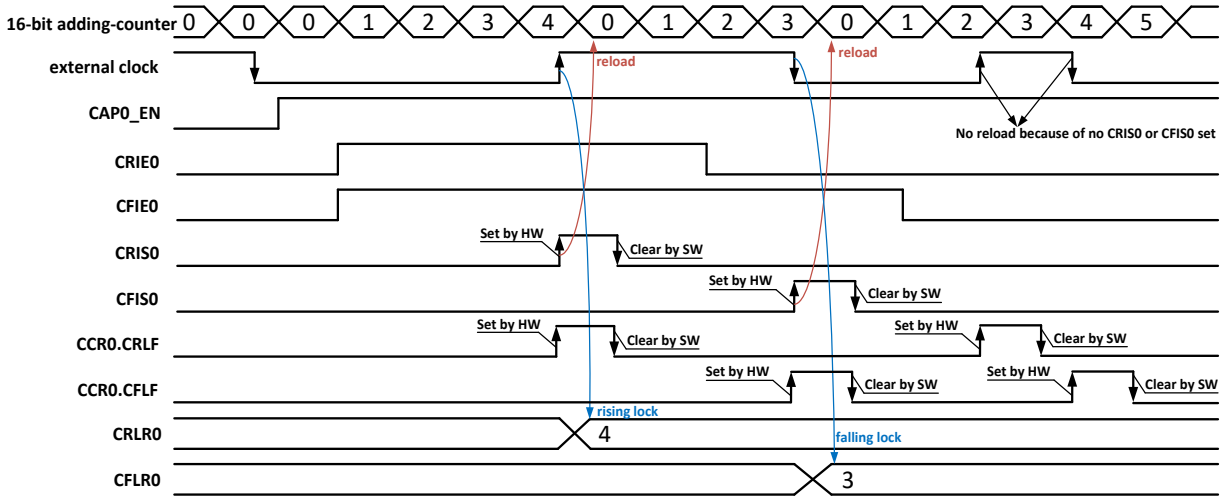
Besides the timer logic module of every PWM channel generates PWM output, it can be used to capture the rising edge and the falling edge of the external clock. Using the PWM0 channel as an example, the PWM0 channel has one CFLR0 and one CRLR0 for capturing up-counter value on the falling edge and rising edge, respectively. You can calculate the period of the external clock by CFLR0 and CRLR0.

$T_{high\text{-}level} = (PWM01\_CLK / PWM0\_PRESCALE\_K)^{-1} * CRLR0$

$T_{low\text{-}level} = (PWM01\_CLK / PWM0\_PRESCALE\_K)^{-1} * CFLR0$

$T_{period} = T_{high\text{-}level} + T_{low\text{-}level}$

Figure 9-73 PWM0 Channel Capture Timing



When the capture input function of the PWM0 channel is enabled, the PCNTR of the PWM0 channel starts to work.

When the timer logic module of PWM0 captures a rising edge, the current value of the up-counter is locked to CRLR0 and CCR0[CRLF] is set to 1. If CRIE0 is 1, then CRIS0 is set to 1, the PWM0 channel sends interrupt requests, and the up-counter is loaded to 0 and continues to count. If CRIE0 is 0, the timer logic module of PWM0 captures a rising edge, CRIS0 cannot be set to 1, the up-counter is not loaded to 0.

When the timer logic module of PWM0 captures one falling edge, the current value of PCNTR is locked to CFLR0 and CCR0[CFLF] is set to 1. If CFIE0 is 1, then CFIS0 is set to 1, the PWM0 channel sends interrupt requests, and the up-counter is loaded to 0 and continues to count. If CFIE0 is 0, the timer logic module of PWM0 captures a falling edge, CFIS0 cannot be set to 1, the up-counter is not loaded to 0.

9.9.3.11 Interrupt

The PWM supports an interrupt generation when configuring the PWM channel to PWM output or capturing input.

For PWM output function, when the controller outputs one period of PWM waveform in cycle mode, the PIS of the corresponding PWM channel is set to 1; when the controller outputs (PWM_PULNUM+1) periods of PWM waveform 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 capturing input function, when the timer logic module of the capture channel0 captures rising edge, and CRIE0 is 1, then CRIS0 is set to 1; when the timer logic module of the capture channel0 captures falling edge, and CFIE0 is 1, then CFIS0 is set to 1.

9.9.4 Programming Guidelines

The following working mode takes PWM01 as an example, other PWM pairs and PWM01 are consistent.

9.9.4.1 Configuring Clock

Step 1 PWM gating: When using PWM, write 1 to PCGR[PWMx_CLK_GATING].

Step 2 PWM clock source select: Set PCCR01[PWM01_CLK_SRC] to select HOSC or APB0 clock.

Step 3 PWM clock divider: Set PCCR01[PWM01_CLK_DIV_M] to select different frequency division coefficient (1/2/4/8/16/32/64/128/256).

Step 4 PWM clock bypass: Set PCGR[PWMx_CLK_BYPASS] to 1, output the PWM clock after the secondary frequency division to the corresponding PWM output pin.

Step 5 PWM internal clock configuration: Set PCR[PWM_PRESCAL_K] to select any frequency division coefficient from 1 to 256.

NOTE

For the channel of complementary output and group mode, firstly, set the same clock configurations (clock source selects APB0, clock division configures the same division factor); secondly, open clock gating at the same time; thirdly, configure PWM parameters; finally, enable PWM output at the same time to ensure each channel sync.

It is suggested that the two channels of the same PWM pair cannot subject to two groups because they have the same first level clock division and gating. If must allocate based on this way, the first level of clock division of the channel used by all groups needs to be set to the same coefficient and open gating at the same time. And the total module needs to be reset when the group mode regroups.

9.9.4.2 Configuring PWM

Step 1 PWM mode: Set PCR[PWM_MODE] to select cycle mode or pulse mode, if pulse mode, PCR[PWM_PUL_NUM] needs to be configured.

Step 2 PWM active level: Set PCR[PWM_ACT_STA] to select a low level or high level.

Step 3 PWM duty-cycle: Configure PPR[PWM_ENTIRE_CYCLE] and PPR[PWM_ACT_CYCLE] after clock gating is opened.

Step 4 PWM starting/stopping phase: Configure PCNTR[PWM_COUNTER_START] after the clock gating is enabled and before the PWM is enabled. You can verify whether the configuration was successful by reading back PCNTR [PWM_COUNTER_STATUS].

Step 5 Enable PWM: Configure PER to select the corresponding PWM enable bit; when selecting pulse mode, PCR[PWM_PUL_START] needs to be enabled.

9.9.4.3 Configuring Dead Zone

Step 1 Set initial value: Set [PDZINTV01].

Step 2 Enable Dead Zone: Set [PWM01_DZ_CN].

9.9.4.4 Configuring Capture Input

Step 1 Enable capture: Configure CER to enable the corresponding channel.

Step 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	0x40045000

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
PCCR67	0x002C	PWM67 Clock Configuration Register
PCGR	0x0040	PWM Clock Gating Register
PDZCR01	0x0060	PWM01 Dead Zone Control Register
PDZCR23	0x0064	PWM23 Dead Zone Control Register
PDZCR45	0x0068	PWM45 Dead Zone Control Register
PDZCR67	0x006C	PWM67 Dead Zone Control Register
PER	0x0080	PWM Enable Register
PGR0	0x0090	PWM Group0 Register
PGR1	0x0094	PWM Group1 Register
PGR2	0x0098	PWM Group2 Register
PGR3	0x009C	PWM Group3 Register
CER	0x00C0	Capture Enable Register
PCR	0x0100+0x0000+N*0x0020 (N= 0~7)	PWM Control Register
PPR	0x0100+0x0004+N*0x0020 (N= 0~7)	PWM Period Register
PCNTR	0x0100+0x0008+N*0x0020 (N= 0~7)	PWM Count Register
PPCNTR	0x0100+0x000C+N*0x0020	PWM Pulse Count Register

Register Name	Offset	Description
	(N= 0~7)	
CCR	0x0100+0x0010+N*0x0020 (N= 0~7)	Capture Control Register
CRLR	0x0100+0x0014+N*0x0020 (N= 0~7)	Capture Rise Lock Register
CFLR	0x0100+0x0018+N*0x0020 (N= 0~7)	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**
31:20	/	/	/
19	R/W	0x0	PGIE3 PWM Group 3 Interrupt Enable 0: Disable 1: Enable
18	R/W	0x0	PGIE2 PWM Group 2 Interrupt Enable 0: Disable 1: Enable
17	R/W	0x0	PGIE1 PWM Group 1 Interrupt Enable 0: Disable 1: Enable
16	R/W	0x0	PGIE0 PWM Group 0 Interrupt Enable 0: Disable 1: Enable
15:8	/	/	/
7	R/W	0x0	PCIE7 PWM Channel 7 Interrupt Enable 0: PWM Channel 7 Interrupt Disable 1: PWM Channel 7 Interrupt Enable
6	R/W	0x0	PCIE6 PWM Channel 6 Interrupt Enable 0: PWM Channel 6 Interrupt Disable 1: PWM Channel 6 Interrupt Enable
5	R/W	0x0	PCIE5 PWM Channel 5 Interrupt Enable 0: PWM Channel 5 Interrupt Disable 1: PWM Channel 5 Interrupt Enable

Offset: 0x0000			Register Name: PIER
Bit	**Read/Write**	**Default/Hex**	**Description**
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	R/W	0x0	PCIE0 PWM Channel 0 Interrupt Enable 0: PWM Channel 0 Interrupt Disable 1: PWM Channel 0 Interrupt Enable

9.9.6.2 0x0004 PWM IRQ Status Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: PISR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/
19	R/W1C	0x0	PGIS3 PWM Group 3 Interrupt Status
18	R/W1C	0x0	PGIS2 PWM Group 2 Interrupt Status
17	R/W1C	0x0	PGIS1 PWM Group 1 Interrupt Status
16	R/W1C	0x0	PGIS0 PWM Group 0 Interrupt Status
15:8	/	/	/

Offset: 0x0004			Register Name: PISR
Bit	**Read/Write**	**Default/Hex**	**Description**
7	R/W1C	0x0	PIS7 PWM Channel 7 Interrupt Status When the PWM channel 7 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Read 0: PWM channel 7 interrupt is not pending. Read 1: PWM channel 7 interrupt is pending. Write 0: No effect. Write 1: Clear PWM channel 7 interrupt status.
6	R/W1C	0x0	PIS6 PWM Channel 6 Interrupt Status When the PWM channel 6 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Read 0: PWM channel 6 interrupt is not pending. Read 1: PWM channel 6 interrupt is pending. Write 0: No effect. Write 1: Clear PWM channel 6 interrupt status.
5	R/W1C	0x0	PIS5 PWM Channel 5 Interrupt Status When the PWM channel 5 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Read 0: PWM channel 5 interrupt is not pending. Read 1: PWM channel 5 interrupt is pending. Write 0: No effect. Write 1: Clear PWM channel 5 interrupt status.
4	R/W1C	0x0	PIS4 PWM Channel 4 Interrupt Status When the PWM channel 4 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Read 0: PWM channel 4 interrupt is not pending. Read 1: PWM channel 4 interrupt is pending. Write 0: No effect. Write 1: Clear PWM channel 4 interrupt status.
3	R/W1C	0x0	PIS3 PWM Channel 3 Interrupt Status When the PWM channel 3 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Read 0: PWM channel 3 interrupt is not pending. Read 1: PWM channel 3 interrupt is pending. Write 0: No effect. Write 1: Clear PWM channel 3 interrupt status.

Offset: 0x0004			Register Name: PISR
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W1C	0x0	PIS2 PWM Channel 2 Interrupt Status When the PWM channel 2 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Read 0: PWM channel 2 interrupt is not pending. Read 1: PWM channel 2 interrupt is pending. Write 0: No effect. Write 1: Clear PWM channel 2 interrupt status.
1	R/W1C	0x0	PIS1 PWM Channel 1 Interrupt Status When the PWM channel 1 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Read 0: PWM channel 1 interrupt is not pending. Read 1: PWM channel 1 interrupt is pending. Write 0: No effect. Write 1: Clear PWM channel 1 interrupt status.
0	R/W1C	0x0	PIS0 PWM Channel 0 Interrupt Status When the PWM channel 0 counter reaches the Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Read 0: PWM channel 0 interrupt is not pending. Read 1: PWM channel 0 interrupt is pending. Write 0: No effect. Write 1: Clear PWM channel 0 interrupt status.

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**
31:16	/	/	/
15	R/W	0x0	CFIE7 If the enable bit is set to 1, when the capture channel 7 captures falling edge, it generates a capture channel 7 pending. 0: Capture channel 7 fall lock interrupt disable 1: Capture channel 7 fall lock interrupt enable
14	R/W	0x0	CRIE7 If the enable bit is set to 1, when the capture channel 7 captures rising edge, it generates a capture channel 7 pending. 0: Capture channel 7 rise lock interrupt disable 1: Capture channel 7 rise lock interrupt enable

Offset: 0x0010			Register Name: CIER
Bit	**Read/Write**	**Default/Hex**	**Description**
13	R/W	0x0	CFIE6 If the enable bit is set to 1, when the capture channel 6 captures falling edge, it generates a capture channel 6 pending. 0: Capture channel 6 fall lock interrupt disable 1: Capture channel 6 fall lock interrupt enable
12	R/W	0x0	CRIE6 If the enable bit is set to 1, when the capture channel 6 captures rising edge, it generates a capture channel 6 pending. 0: Capture channel 6 rise lock interrupt disable 1: Capture channel 6 rise lock interrupt enable
11	R/W	0x0	CFIE5 If the enable bit is set to 1, when the capture channel 5 captures falling edge, it generates a capture channel 5 pending. 0: Capture channel 5 fall lock interrupt disable 1: Capture channel 5 fall lock interrupt enable
10	R/W	0x0	CRIE5 If the enable bit is set to 1, when the capture channel 5 captures rising edge, it generates a capture channel 5 pending. 0: Capture channel 5 rise lock interrupt disable 1: Capture channel 5 rise lock interrupt enable
9	R/W	0x0	CFIE4 If the enable bit is set to 1, when the capture channel 4 captures falling edge, it generates a capture 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 enable bit is set to 1, when the capture channel 4 captures rising edge, it generates a capture channel 4 pending. 0: Capture channel 4 rise lock interrupt disable 1: Capture channel 4 rise lock interrupt enable
7	R/W	0x0	CFIE3 If the enable bit is set to 1, when the capture channel 3 captures falling edge, it generates a capture channel 3 pending. 0: Capture channel 3 fall lock interrupt disable 1: Capture channel 3 fall lock interrupt enable
6	R/W	0x0	CRIE3 If the enable bit is set to 1, when the capture channel 3 captures rising edge, it generates a capture channel 3 pending. 0: Capture channel 3 rise lock interrupt disable 1: Capture channel 3 rise lock interrupt enable

Offset: 0x0010			Register Name: CIER
Bit	**Read/Write**	**Default/Hex**	**Description**
5	R/W	0x0	CFIE2 If the enable bit is set to 1, when the capture channel 2 captures falling edge, it generates a capture channel 2 pending. 0: Capture channel 2 fall lock interrupt disable 1: Capture channel 2 fall lock interrupt enable
4	R/W	0x0	CRIE2 If the enable bit is set to 1, when the capture channel 2 captures rising edge, it generates a capture channel 2 pending. 0: Capture channel 2 rise lock interrupt disable 1: Capture channel 2 rise lock interrupt enable
3	R/W	0x0	CFIE1 If the enable bit is set to 1, when the capture channel 1 captures falling edge, it generates a capture channel 1 pending. 0: Capture channel 1 fall lock interrupt disable 1: Capture channel 1 fall lock interrupt enable
2	R/W	0x0	CRIE1 If the enable bit is set to 1, when the capture channel 1 captures rising edge, it generates a capture channel 1 pending. 0: Capture channel 1 rise lock interrupt disable 1: Capture channel 1 rise lock interrupt enable
1	R/W	0x0	CFIE0 If the enable bit is set to 1, when the capture channel 0 captures falling edge, it generates a capture channel 0 pending. 0: Capture channel 0 fall lock interrupt disable 1: Capture channel 0 fall lock interrupt enable
0	R/W	0x0	CRIE0 If the enable bit is set to 1, when the capture channel 0 captures rising edge, it generates a capture channel 0 pending. 0: Capture channel 0 rise lock interrupt disable 1: Capture channel 0 rise lock interrupt enable

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**
31:18	/	/	/

Offset: 0x0014			Register Name: CISR
Bit	**Read/Write**	**Default/Hex**	**Description**
15	R/W1C	0x0	CFIS7 Status of the capture channel 7 falling lock interrupt When the capture channel 7 captures falling edge, if the fall lock interrupt (CFIE7) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit. Read 0: The capture channel 7 interrupt is not pending. Read 1: The capture channel 7 interrupt is pending. Write 0: No effect. Write 1: Clear the status of the capture channel 7 interrupt.
14	R/W1C	0x0	CRIS7 Status of the capture channel 7 rising lock interrupt When the capture channel 7 captures rising edge, if the rise lock interrupt (CRIE7) is enabled, this bit is set to 1 by hardware. Write 1 to clear this bit. Read 0: The capture channel 7 interrupt is not pending. Read 1: The capture channel 7 interrupt is pending. Write 0: No effect. Write 1: Clear the status of the capture channel 7 interrupt.
13	R/W1C	0x0	CFIS6 Status of the capture channel 6 falling lock interrupt When the capture channel 6 captures falling edge, if the fall lock interrupt (CFIE6) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit. Read 0: The capture channel 6 interrupt is not pending. Read 1: The capture channel 6 interrupt is pending. Write 0: No effect. Write 1: Clear the status of the capture channel 6 interrupt.
12	R/W1C	0x0	CRIS6 Status of the capture channel 6 rising lock interrupt. When the capture channel 6 captures rising edge, if the rise lock interrupt (CRIE6) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit. Read 0: The capture channel 6 interrupt is not pending. Read 1: The capture channel 6 interrupt is pending. Write 0: No effect. Write 1: Clear the status of the capture channel 6 interrupt.

Offset: 0x0014			Register Name: CISR
Bit	**Read/Write**	**Default/Hex**	**Description**
11	R/W1C	0x0	CFIS5 Status of the capture channel 5 falling lock interrupt When the capturing channel 5 captures falling edge, if the fall lock interrupt (CFIE5) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit. Read 0: The capture channel 5 interrupt is not pending. Read 1: The capture channel 5 interrupt is pending. Write 0: No effect. Read 1: Clear the status of the capture channel 5 interrupt.
10	R/W1C	0x0	CRIS5 Status of the capture channel 5 rising lock interrupt When the capture channel 5 captures rising edge, if the rise lock interrupt (CRIE5) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit. Read 0: The capture channel 5 interrupt is not pending. Read 1: The capture channel 5 interrupt is pending. Write 0: No effect. Write 1: Clear the status of the capture channel 5 interrupt.
9	R/W1C	0x0	CFIS4 Status of the capture channel 4 falling lock interrupt When the capture channel 4 captures falling edge, if the fall lock interrupt (CFIE4) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit. Read 0: The capture channel 4 interrupt is not pending. Read 1: The capture channel 4 interrupt is pending. Write 0: No effect. Write 1: Clear the status of the capture channel 4 interrupt.
8	R/W1C	0x0	CRIS4 Status of the capture channel 4 rising lock interrupt. When the capture channel 4 captures rising edge, if the rise lock interrupt (CRIE4) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit. Read 0: The capture channel 4 interrupt is not pending. Read 1: The capture channel 4 interrupt is pending. Write 0: No effect. Write 1: Clear the status of the capture channel 4 interrupt status.

Offset: 0x0014			Register Name: CISR
Bit	**Read/Write**	**Default/Hex**	**Description**
7	R/W1C	0x0	CFIS3 Status of the capture channel 3 falling lock interrupt. When the capture channel 3 captures falling edge, if the fall lock interrupt (CFIE3) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit. Read 0: The capture channel 3 interrupt is not pending. Read 1: The capture channel 3 interrupt is pending. Write 0: No effect. Write 1: Clear the status of the capture channel 3 interrupt.
6	R/W1C	0x0	CRIS3 Status of the capture channel 3 rising lock interrupt When the capture channel 3 captures rising edge, if the rise lock interrupt (CRIE3) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit. Read 0: The capture channel 3 interrupt is not pending. Read 1: The capture channel 3 interrupt is pending. Write 0: No effect. Write 1: Clear the status of the capture channel 3 interrupt.
5	R/W1C	0x0	CFIS2 Status of the capture channel 2 falling lock interrupt When the capture channel 2 captures falling edge, if the fall lock interrupt (CFIE2) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit. Read 0: The capture channel 2 interrupt is not pending. Read 1: The capture channel 2 interrupt is pending. Write 0: No effect. Write 1: Clear the status of the capture channel 2 interrupt.
4	R/W1C	0x0	CRIS2 Status of the capture channel 2 rising lock interrupt. When the capture channel 2 captures rising edge, if the rise lock interrupt (CRIE2) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit. Read 0: The capture channel 2 interrupt is not pending. Read 1: The capture channel 2 interrupt is pending. Write 0: No effect. Write 1: Clear the status of the capture channel 2 interrupt.

Offset: 0x0014			Register Name: CISR
Bit	**Read/Write**	**Default/Hex**	**Description**
3	R/W1C	0x0	CFIS1 Status of the capture channel 1 falling lock interrupt When the capture channel 1 captures falling edge, if the fall lock interrupt (CFIE1) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit. Read 0: The capture channel 1 interrupt is not pending. Read 1: The capture channel 1 interrupt is pending. Write 0: No effect. Write 1: Clear the status of the capture channel 1 interrupt.
2	R/W1C	0x0	CRIS1 Status of the capture channel 1 rising lock interrupt. When the capture channel 1 captures rising edge, if the rise lock interrupt (CRIE1) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit. Read 0: The capture channel 1 interrupt is not pending. Read 1: The capture channel 1 interrupt is pending. Write 0: No effect. Write 1: Clear the status of the capture channel 1 interrupt.
1	R/W1C	0x0	CFIS0 Status of the capture channel 0 falling lock interrupt When the capture channel 0 captures falling edge, if the fall lock interrupt (CFIE0) is enabled, this bit is set to 1 by hardware. Writing 1 to clear this bit. Read 0: The capture channel 0 interrupt is not pending. Read 1: The capture channel 0 interrupt is pending. Write 0: No effect. Write 1: Clear the status of the capture channel 0 interrupt.
0	R/W1C	0x0	CRIS0 Status of the capture channel 0 rising lock interrupt When the capture channel 0 captures rising edge, if the rise lock interrupt (CRIE0) is enabled, this bit is set 1 by hardware. Writing 1 to clear this bit. Read 0: The capture channel 0 interrupt is not pending. Read 1: The capture channel 0 interrupt is pending. Write 0: No effect. Write 1: Clear the status of the capture channel 0 interrupt.

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	/	/	/

Offset: 0x0020			Register Name: PCCR01
Bit	**Read/Write**	**Default/Hex**	**Description**
8:7	R/W	0x0	PWM01_CLK_SRC Select PWM01 Clock Source 00: HOSC 01: APB0 Others: Reserved
6:4	/	/	/
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: HOSC 01: APB0 Others: Reserved
6:4	/	/	/
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: Reserved

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: HOSC 01: APB0 Others: Reserved
6:4	/	/	/
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: Reserved

9.9.6.8 0x002C PWM67 Clock Configuration Register (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: PCCR67
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8:7	R/W	0x0	PWM67_CLK_SRC_SEL Select PWM67 Clock Source 00: HOSC 01: APB0 Others: Reserved
6:4	/	/	/

Offset: 0x002C			Register Name: PCCR67
Bit	**Read/Write**	**Default/Hex**	**Description**
3: 0	R/W	0x0	PWM67_CLK_DIV_M PWM67 Clock Divide M 0000: /1 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 Others: Reserved

9.9.6.9 0x0040 PWM Clock Gating Register (Default Value: 0x0000_0000)

Offset: 0x0040			Register Name: PCGR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23	R/W	0x0	PWM7_CLK_BYPASS Bypass clock source (after pre-scale) to PWM7 output 0: Not bypass 1: Bypass
22	R/W	0x0	PWM6_CLK_BYPASS Bypass clock source (after pre-scale) to PWM6 output 0: Not bypass 1: Bypass
21	R/W	0x0	PWM5_CLK_BYPASS Bypass clock source (after pre-scale) to PWM5 output 0: Not bypass 1: Bypass
20	R/W	0x0	PWM4_CLK_BYPASS Bypass clock source (after pre-scale) to PWM4 output 0: Not bypass 1: Bypass
19	R/W	0x0	PWM3_CLK_BYPASS Bypass clock source (after pre-scale) to PWM3 output 0: Not bypass 1: Bypass
18	R/W	0x0	PWM2_CLK_BYPASS Bypass clock source (after pre-scale) to PWM2 output 0: Not bypass 1: Bypass

Offset: 0x0040			Register Name: PCGR
Bit	**Read/Write**	**Default/Hex**	**Description**
17	R/W	0x0	PWM1_CLK_BYPASS Bypass clock source (after pre-scale) to PWM1 output 0: Not bypass 1: Bypass
16	R/W	0x0	PWM0_CLK_BYPASS Bypass clock source (after pre-scale) to PWM0 output 0: Not bypass 1: Bypass
15:8	/	/	/
7	R/W	0x0	PWM7_CLK_GATING Gating clock for PWM7 0: Mask 1: Pass
6	R/W	0x0	PWM6_CLK_GATING Gating clock for PWM6 0: Mask 1: Pass
5	R/W	0x0	PWM5_CLK_GATING Gating clock for PWM5 0: Mask 1: Pass
4	R/W	0x0	PWM4_CLK_GATING Gating clock for PWM4 0: Mask 1: Pass
3	R/W	0x0	PWM3_CLK_GATING Gating clock for PWM3 0: Mask 1: Pass
2	R/W	0x0	PWM2_CLK_GATING Gating clock for PWM2 0: Mask 1: Pass
1	R/W	0x0	PWM1_CLK_GATING Gating clock for PWM1 0: Mask 1: Pass
0	R/W	0x0	PWM0_CLK_GATING Gating clock for PWM0 0: Mask 1: Pass

9.9.6.10 0x0060 PWM01 Dead Zone Control Register (Default Value: 0x0000_0000)

Offset: 0x0060			Register Name: PDZCR01
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:8	R/W	0x0	PWM01_DZ_INTV 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

9.9.6.11 0x0064 PWM23 Dead Zone Control Register (Default Value: 0x0000_0000)

Offset: 0x0064			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

9.9.6.12 0x0068 PWM45 Dead Zone Control Register (Default Value: 0x0000_0000)

Offset: 0x0068			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

9.9.6.13 0x006C PWM67 Dead Zone Control Register (Default Value: 0x0000_0000)

Offset: 0x006C			Register Name: PDZCR67
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x006C			Register Name: PDZCR67
Bit	**Read/Write**	**Default/Hex**	**Description**
15:8	R/W	0x0	PWM67_DZ_INTV PWM67 Dead Zone Interval Value
7:1	/	/	/
0	R/W	0x0	PWM67_DZ_EN PWM67 Dead Zone Enable 0: Dead Zone disable 1: Dead Zone enable

9.9.6.14 0x0080 PWM Enable Register (Default Value: 0x0000_0000)

Offset: 0x0080			Register Name: PER
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	PWM7_EN When PWM is enabled, the 16-bit up-counter starts working and PWM channel7 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable
6	R/W	0x0	PWM6_EN When PWM is enabled, the 16-bit up-counter starts working and PWM channel6 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable
5	R/W	0x0	PWM5_EN When PWM is enabled, the 16-bit up-counter starts working and PWM channel5 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable
4	R/W	0x0	PWM4_EN When PWM is enabled, 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 PWM is enabled, the 16-bit up-counter starts working and PWM channel3 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable

Offset: 0x0080			Register Name: PER
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x0	PWM2_EN When PWM is enabled, 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 PWM is enabled, the 16-bit up-counter starts working and PWM channel1 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable
0	R/W	0x0	PWM0_EN When PWM is enabled, the 16-bit up-counter starts working and PWM channel0 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable

9.9.6.15 0x0090 PWM Group0 Register (Default Value: 0x0000_0000)

Offset: 0x0090			Register Name: PGR0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R/WAC	0x0	PWMG0_START The PWM channels selected in PWMG0_CS start to output PWM waveform at the same time.
16	R/W	0x0	PWMG0_EN PWM Group0 Enable.
15: 0	R/W	0x0	PWMG0_CS If bit[i] is set, the PWM i is selected as one channel of PWM Group0.

9.9.6.16 0x0094 PWM Group1 Register (Default Value: 0x0000_0000)

Offset: 0x0094			Register Name: PGR1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R/WAC	0x0	PWMG1_START The PWM channels selected in PWMG1_CS start to output PWM waveform at the same time.
16	R/W	0x0	PWMG1_EN PWM Group1 Enable.

Offset: 0x0094			Register Name: PGR1
Bit	Read/Write	Default/Hex	Description
15: 0	R/W	0x0	PWMG1_CS If bit[i] is set, the PWM i is selected as one channel of PWM Group1.

9.9.6.17 0x0098 PWM Group2 Register (Default Value: 0x0000_0000)

Offset: 0x0098			Register Name: PGR2
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R/WAC	0x0	PWMG2_START The PWM channels selected in PWMG2_CS start to output PWM waveform at the same time.
16	R/W	0x0	PWMG2_EN PWM Group2 Enable.
15: 0	R/W	0x0	PWMG2_CS If bit[i] is set, the PWM i is selected as one channel of PWM Group2.

9.9.6.18 0x009C PWM Group3 Register (Default Value: 0x0000_0000)

Offset: 0x009C			Register Name: PGR3
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R/WAC	0x0	PWMG3_START The PWM channels selected in PWMG3_CS start to output PWM waveform at the same time.
16	R/W	0x0	PWMG3_EN PWM Group3 Enable.
15: 0	R/W	0x0	PWMG3_CS If bit[i] is set, the PWM i is selected as one channel of PWM Group3.

9.9.6.19 0x00C0 Capture Enable Register (Default Value: 0x0000_0000)

Offset: 0x00C0			Register Name: CER
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/

Offset: 0x00C0			Register Name: CER
Bit	**Read/Write**	**Default/Hex**	**Description**
7	R/W	0x0	CAP7_EN When enabling the capture function, the 16-bit up-counter starts working, and the capture channel7 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable
6	R/W	0x0	CAP6_EN When enabling the capture function, the 16-bit up-counter starts working, and the capture channel6 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable
5	R/W	0x0	CAP5_EN When enabling the capture function, the 16-bit up-counter starts working, and the capture channel5 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable
4	R/W	0x0	CAP4_EN When enabling the capture function, the 16-bit up-counter starts working, and the 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 enabling the capture function, the 16-bit up-counter starts working, and the 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 enabling the capture function, the 16-bit up-counter starts working, and the 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 enabling the capture function, the 16-bit up-counter starts working, and the capture channel1 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable

Offset: 0x00C0			Register Name: CER
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	CAP0_EN When enabling the capture function, the 16-bit up-counter starts working, and the capture channel is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable

9.9.6.20 0x0100 + N*0x20 PWM Control Register (Default Value: 0x0000_0000)

Offset: 0x0100+0x0+N*0x20 (N=0~7)			Register Name: PCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	PWM_PUL_NUM In pulse mode, the PWM outputs pulse for PWM_CYCLE_NUM+1 times and then stops.
15:12	/	/	/
11	R	0x0	PWM_PERIOD_RDY PWM Period Register Ready 0: PWM period register is ready to write 1: PWM period register is busy Note: In pulse mode, after the period is configured by software, wait for this bit to be cleared to 0 before configuring PWM_PUL_START.
10	R/WAC	0x0	PWM_PUL_START PWM Pulse Output Start 0: No effect 1: Output pulse for PWM_CYCLE_NUM+1. After finishing configuration for the output pulse, set this bit once, 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

Offset: 0x0100+0x0+N*0x20 (N=0~7)			Register Name: PCR
Bit	**Read/Write**	**Default/Hex**	**Description**
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

9.9.6.21 0x0104 + N*0x20 PWM Period Register (Default Value: 0x0000_0000)

Offset: 0x0100+0x04+N*0x20 (N=0~7)			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 needs 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

9.9.6.22 0x0108 + N*0x20 PWM Counter Register (Default Value: 0x0000_0000)

Offset: 0x0100+0x08+N*0x20 (N=0~7)			Register Name: PCNTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	PWM_COUNTER_START PWM counter value is set for phase control.
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.

9.9.6.23 0x010C + N*0x20 PWM Pulse Counter Register (Default Value: 0x0000_0000)

Offset: 0x0100+0x0C+N*0x20 (N=0~7)			Register Name: PPCNTR
Bit	**Read/Write**	**Default/Hex**	**Description**

Offset: 0x0100+0x0C+N*0x20 (N=0~7)			Register Name: PPCNTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15: 0	R	0x0	PWM_PUL_COUNTER_STATUS On PWM output, reading this register could get the current value of the PWM pulse counter.

9.9.6.24 0x0110 + N*0x20 PWM Capture Control Register (Default Value: 0x0000_0000)

Offset: 0x0100+0x10+N*0x20 (N=0~7)			Register Name: CCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4	R/W1C	0x0	CRLF When the capture channel captures a rising edge, the current value of the 16-bit up-counter is latched to CRLR, and then this bit is set 1 by hardware. Write 1 to clear this bit.
3	R/W1C	0x0	CFLF When the capture channel captures a falling edge, the current value of the 16-bit up-counter is latched to CFLR, and then this bit is set 1 by hardware. Write 1 to clear this bit.
2	R/W	0x0	CRTE Rising edge capture trigger enable
1	R/W	0x0	CFTE Falling edge capture trigger enable
0	R/W	0x0	CAPINV Inverse the signal input from capture channel before 16-bit counter of capture channel. 0: Not inverse 1: Inverse

9.9.6.25 0x0114 + N*0x20 PWM Capture Rise Lock Register (Default Value: 0x0000_0000)

Offset: 0x0100+0x14+N*0x20 (N=0~7)			Register Name: CRLR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15: 0	R	0x0	CRLR When the capture channel captures a rising edge, the current value of the 16-bit up-counter is latched to the register.

9.9.6.26 0x0118 + N*0x20 PWM Capture Fall Lock Register (Default Value: 0x0000_0000)

Offset: 0x0100+0x18+N*0x20 (N=0~7)	Register Name: CFLR

Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15: 0	R	0x0	CFLR When the capture channel captures a falling edge, the current value of the 16-bit up-counter is latched to the register.

9.10 LEDC

9.10.1 Overview

The LED Controller (LEDC) is used to control the external LED lamp.

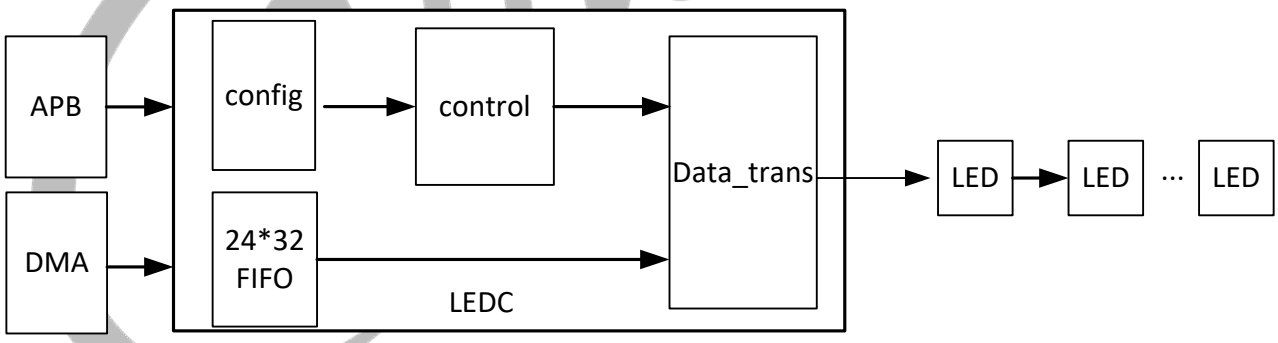
The LEDC has the following features:

- Configurable LED input high-/low-level width
- Configurable LED reset time
- LEDC data supports DMA configuration mode and CPU configuration mode
- Maximum 1024 LEDs serial connect
- LED data transmission rate up to 800 kbit/s
- Configurable RGB display mode

9.10.2 Block Diagram

The following figure shows a block diagram of the LEDC.

Figure 9-74 LEDC Block Diagram



LEDC contains the following sub-blocks:

Table 9-25 LEDC Sub-blocks

Sub-block	Description
config	register configuration
control	LEDC timing control and status control
FIFO	24-bit width x 32 depth
Data_trans	Convert input data to the 0 and 1 characters of LED

9.10.3 Functional Description

9.10.3.1 External Signals

The following table describes the external signals of the LEDC.

Table 9-26 LEDC External Signals

Signal	Description	Type
LEDC	Intelligent Control LED Signal Output	O

9.10.3.2 Clock Sources

The following table describes the clock sources for LEDC. For clock setting, configurations and gating information, refer to the section "CCU" and "CCU_AON".

Table 9-27 LEDC Clock Sources

Clock Sources	Description
HOSC	24 MHz
PLL_PERI(1X)	Peripheral Clock. The default value is 600 MHz

9.10.3.3 LEDC Timing

Figure 9-75 LEDC Package Output Timing Diagram

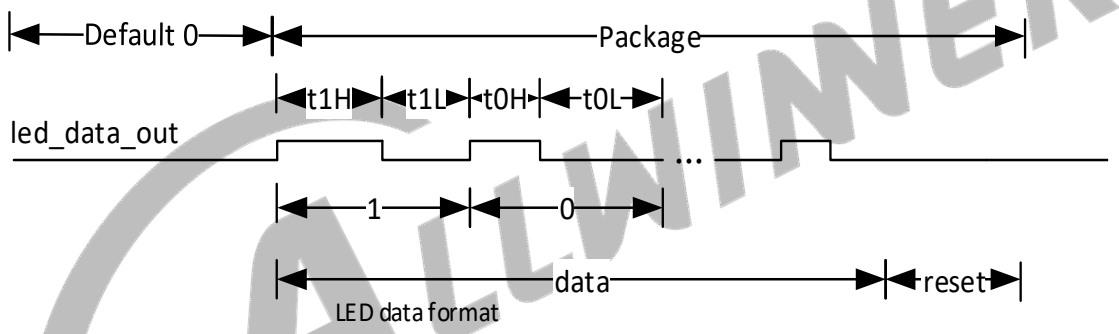


Figure 9-76 LEDC 1-frame Output Timing Diagram

9.10.3.4 LEDC Input Data Structure

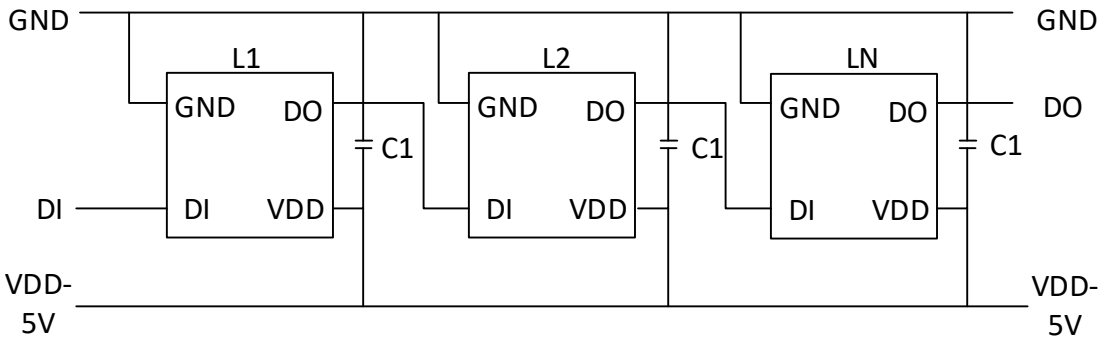
The RGB mode of LEDC data is configurable. By default, the data is sent in GRB order, and the higher bit is transmitted first.

Figure 9-77 LEDC Input Data Structure

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0

9.10.3.5 LEDC Typical Circuit

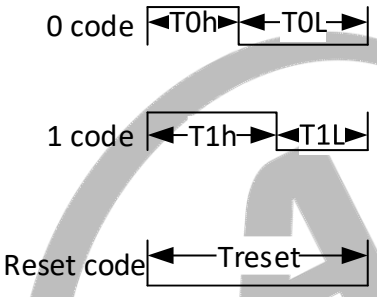
Figure 9-78 LEDC Typical Circuit



C1 is the bypass capacitor of LED light, and its value is usually 100 nF.

9.10.3.6 LEDC Data Input Code

Figure 9-79 LEDC Data Input Code



9.10.3.7 LEDC Data Transmission Time

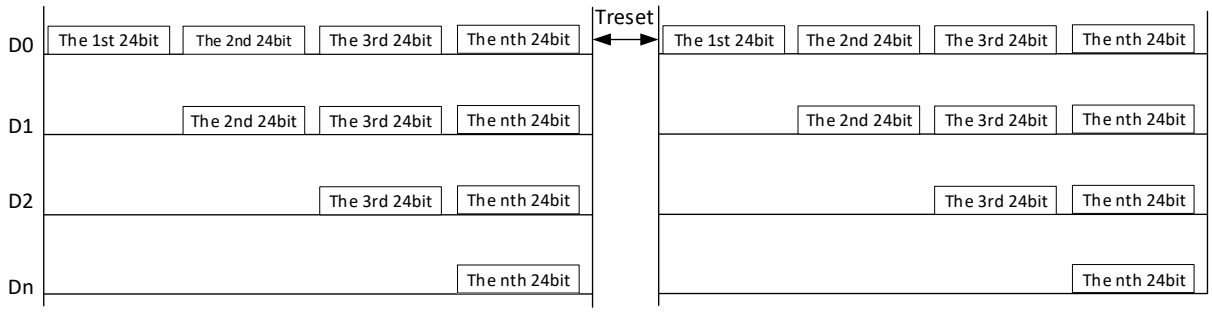
The time parameter of the typical LED specification shows as follows.

Table 9-28 Time Parameters of Typical LED Specification

T0H	0 code, high-level time	220 ns to 380 ns
T0L	0 code, low-level time	580 ns to 1.6 us
T1H	1 code, high-level time	580 ns to 1.6 us
T1L	1 code, low-level time	220 ns to 420 ns
RESET	Frame unit, low-level time	> 280 us

9.10.3.8 LEDC Data Transmission Mode

Table 9-29 LEDC Data Transmission Mode



9.10.3.9 LEDC Parameter

1.PAD rate > 800 kbit/s

2.LED number supported:

$T_{0\text{-}code}$: 800 ns to 1980 ns, T1-code: 800 ns to 2020 ns

When the LED refresh rate is 30 frame/s, LED number supported is (1 s/30-280 us)/((800 ns to 2020 ns)*24) =1024 ↓to 681.

When the LED refresh rate is 60 frame/s, LED number supported is (1 s/60-280 us)/((800 ns to 2020 ns)*24) =853 ↓to 338.

9.10.3.10 LEDC Data Transmission

The LEDC supports DMA data transmission mode or CPU data transmission mode. The DMA data transmission mode is set by LEDC_DMA_EN.

Data Transmission in DMA Mode

When the valid space of internal FIFO is greater than the setting FIFO free space threshold, the LEDC sends DMA_REQ to require DMA to transmission data from DRAM to LEDC. The maximum data transmission size in DMA mode is 16 words. (The internal FIFO level is 32.)

Data Transmission in CPU Mode

When the valid space of internal FIFO is greater than the setting FIFO free space threshold, the LEDC sends LEDC_CPUREQ_INT to require CPU to transfer data to LEDC. The transfer data size in CPU mode is controlled by software. The internal FIFO destination address is 0x06700014. The data width is 32-bit. (The lower 24-bit is valid.)

9.10.3.11 LEDC Interrupt

Module Name	Description
FIFO_OVERFLOW_INT	FIFO overflow interrupt.

Module Name	Description
	The data written by external is more than the maximum storage space of LED FIFO, the LEDC will be in data loss state. At this time, software needs to deal with the abnormal situation. The processing mode is as follows. The software can query LED_FIFO_DATA_REG to determine which data has been stored in the internal FIFO of LEDC. The LEDC performs soft_reset operation to refresh all data.
WAITDATA_TIMEOUT_INT	Wait for data timeout interrupt When internal FIFO of LEDC cannot get data because of some abnormal situation, the timeout interrupt is set after led_wait_data_time, now the LEDC is in WAIT_DATA state, and the LEDC outputs a level state configured by LED_POLARITY; in the course of wait_data, if the new data arrives, the LEDC will continue to send data, at this time the software needs to notice whether the waiting time of the LEDC exceeds the operation time of reset. If the waiting time of the LEDC exceeds the operation time of reset (this is equivalent to reset operation sent by LEDC), the LED may enter in refresh state, the data has not been sent.
FIFO_CPUREQ_INT	FIFO request CPU data interrupt When FIFO data is less than a threshold, the interrupt will be reported to the CPU.
LEDC_TRANS_FINISH_INT	Data transmission complete interrupt The value indicates that the data configured as total_data_length has been transferred completely.

LEDC interrupt usage scenario:

CPU mode

The software can enable GLOBAL_INT_EN, FIFO_CPUREQ_INT_EN, WAITDATA_TIMEOUT_INT_EN, FIFO_OVERFLOW_INT_EN, LEDC_TRANS_FINISH_INT_EN, and cooperate with LEDC_FIFO_TRIG_LEVEL to use. When FIFO_CPUREQ_INT is set to 1, the software can configure data of LEDC_FIFO_TRIG_LEVEL to LEDC.

DMA mode

The software can enable GLOBAL_INT_EN, WAITDATA_TIMEOUT_INT_EN, FIFO_OVERFLOW_INT_EN, LEDC_TRANS_FINISH_INT_EN, and cooperate with LEDC_FIFO_TRIG_LEVEL to use. When DMA receives LEDC DMA_REQ, DMA can transfer data of LEDC_FIFO_TRIG_LEVEL to LEDC.

9.10.4 Programming Guidelines

9.10.4.1 LEDC Normal Configuration Process

Step 1 Configure LEDC_CLK and bus pclk.

Step 2 Configure the written LEDC data.

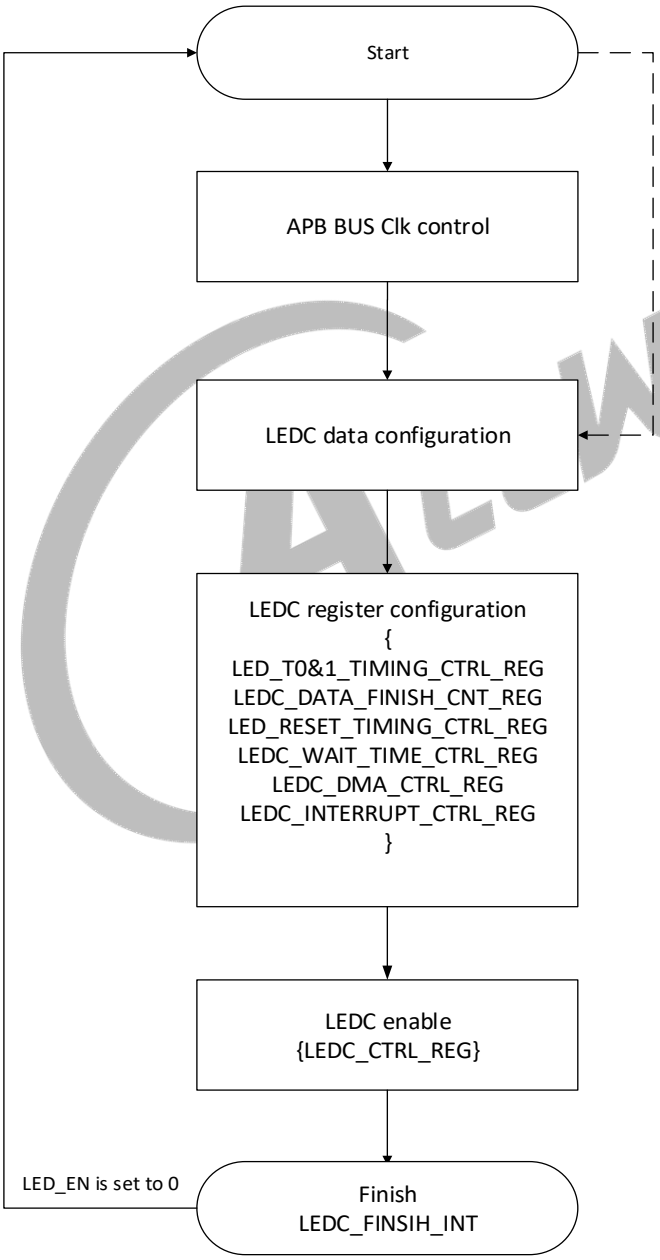
Step 3 Configure LED_T01_TIMING_CTRL_REG, LEDC_DATA_FINISH_CNT_REG, LED_RESET_TIMING_CTRL_REG, LEDC_WAIT_TIME0_CTRL_REG, LEDC_DMA_CTRL_REG, LEDC_INTERRUPT_CTRL_REG. Configure 0-code, 1-code, reset time, LEDC waiting time, and the number of external connected LEDC and the threshold of DMA transfer data.

Step 4 Configure LEDC_CTRL_REG to enable LEDC_EN, the LEDC will start to output data.

Step 5 When the LEDC interrupt is pulled up, it indicates the configured data has transferred complete, at this time LED_EN will be set to 0, and the read/write point of LEDC FIFO is cleared to 0.

Step 6 Repeat step1, 2, 3, 4 to re-execute a new round of configuration, enable LEDC_EN, the LEDC will start new data transmission.

Figure 9-80 LEDC Normal Configuration Process



9.10.4.2 LEDC Abnormal Scene Processing Flow

WAITDATA_TIMEOUT Abnormal Status

Step 1 When WAITDATA_TIMEOUT_INT appears, it indicates the internal FIFO data request of LEDC cannot obtain a response, at this time if the default output level is low, then the external LED may think there was a reset operation and cause LED data to be flushed incorrectly.

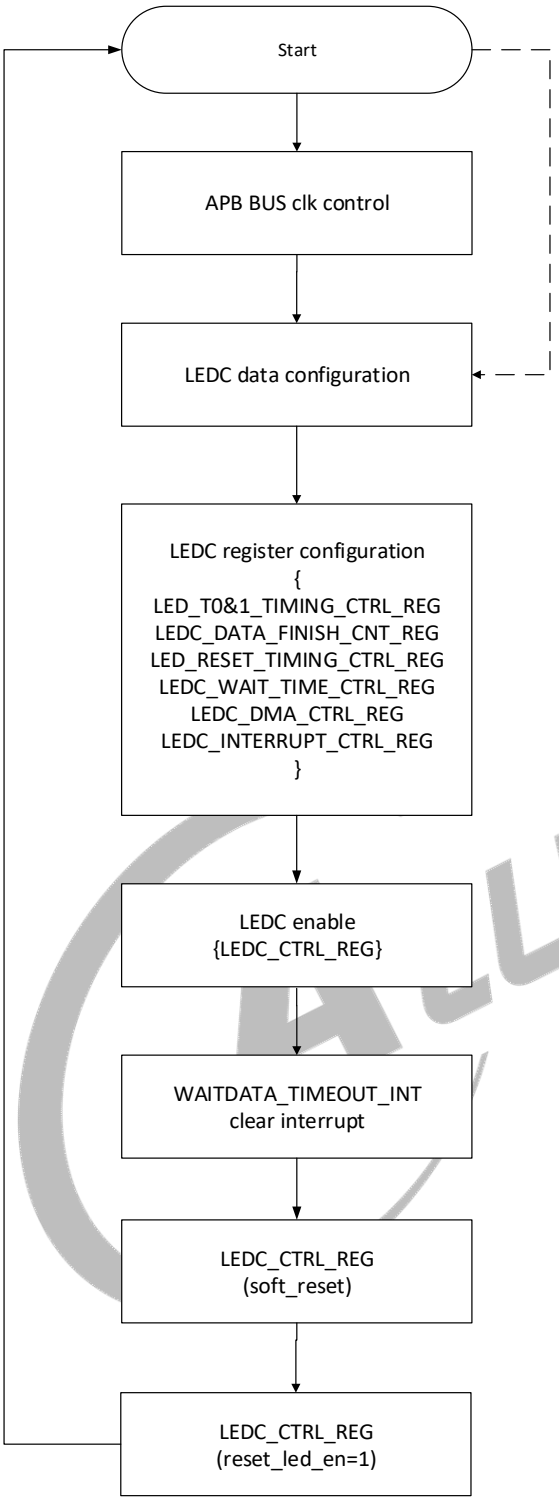
Step 2 The LEDC needs to be performed soft_reset operation (LEDC_SOFT_RESET=1); after soft_reset, the LEDC_EN will be pulled-down automatically, all internal status register and control state machine will return to the idle state, the LEDC FIFO read & write point is cleared to 0, the LEDC interrupt is cleared.

Step 3 Setting RESET_LED_EN to 1 indicates LEDC can actively send a reset operation to ensure the external LED lamp in the right state.

Step 4 The software Read the status of RESET_LED_EN, when the status value is 1, it indicates LEDC does not perform the transmission of LED reset operation; when the status value is 0, the LEDC completes the transmission of LED reset operation.

Step 5 When LEDC reset operation finishes, the LEDC data and register configuration need to be re-operated to start re-transmission data operation.

Figure 9-81 LEDC Timeout Abnormal Processing Flow



FIFO Overflow Abnormal Status

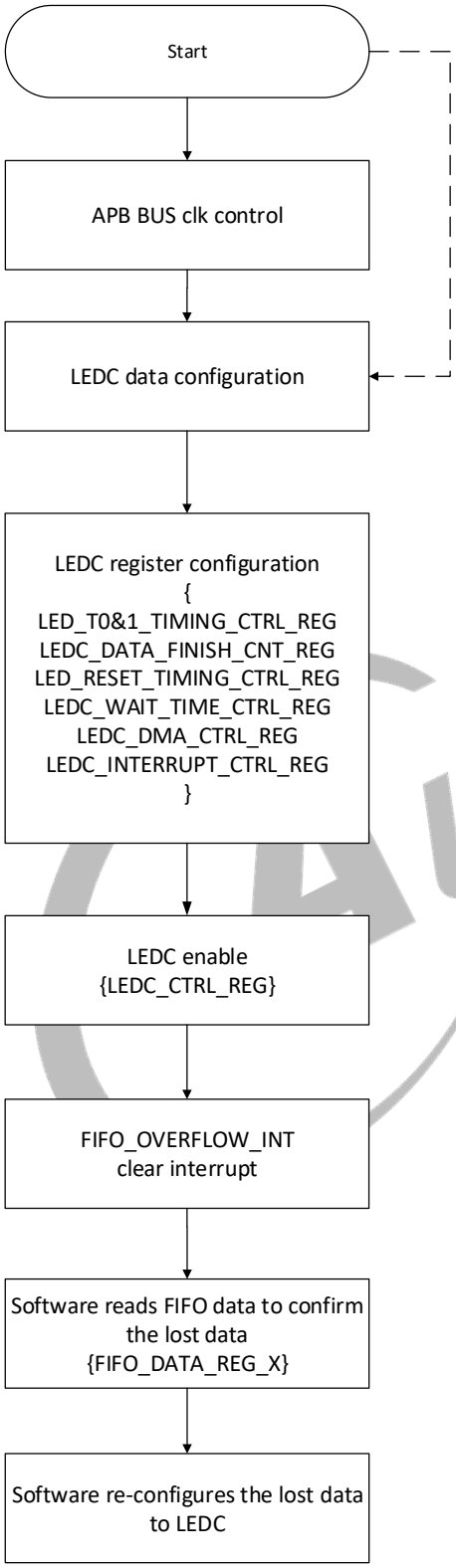
Step 1 When FIFO_OVERFLOW_INT appears, it indicates the data configured by software exceeds the LEDC FIFO space, at this time the redundant data will be lost.

Step 2 The software needs to read data in LEDC_FIFO_DATA_X to confirm the lost data.

Step 3 The software re-configures the lost data to the LEDC.

Step 4 If the software uses the soft_reset operation, the operation is the same with the timeout abnormal processing flow.

Figure 9-82 FIFO Overflow Abnormal Processing Flow



9.10.5 Register List

Module Name	Base Address

Module Name	Base Address
LEDC	0x40048000

Register Name	Offset	Description
LEDC_CTRL_REG	0x0000	LEDC Control Register
LED_T01_TIMING_CTRL_REG	0x0004	LEDC T0 & T1 Timing Control Register
LEDC_DATA_FINISH_CNT_REG	0x0008	LEDC Data Finish Counter Register
LED_RESET_TIMING_CTRL_REG	0x000C	LEDC Reset Timing Control Register
LEDC_WAIT_TIME0_CTRL_REG	0x0010	LEDC Wait Time0 Control Register
LEDC_DATA_REG	0x0014	LEDC Data
LEDC_DMA_CTRL_REG	0x0018	LEDC DMA Control Register
LEDC_INT_CTRL_REG	0x001C	LEDC Interrupt Control Register
LEDC_INT_STS_REG	0x0020	LEDC Interrupt Status Register
LEDC_WAIT_TIME1_CTRL_REG	0x0028	LEDC Wait Time1 Control Register
LEDC_FIFO_DATA_REG	0x0030+0x04*N	LEDC FIFO Data Register

9.10.6 Register Description

9.10.6.1 0x0000 LEDC Control Register (Default Value: 0x0000_003C)

Offset: 0x0000			Register Name: LEDC_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28:16	R/W	0x0	TOTAL_DATA_LENGTH Total length of transfer data (range: 0 to 8K, unit: 32-bit, only low 24-bit is valid) The field is recommended to be set to an integer multiple of (LED_NUM+1). If TOTAL_DATA_LENGTH is greater than (LED_NUM+1), but non-integer multiple, the last frame of data will transfer data less than (LED_NUM+1).
15:11	/	/	/

<table>
<tr><th colspan="3">Offset: 0x0000</th><th>Register Name: LEDC_CTRL_REG</th></tr>
<tr><th>Bit</th><th>Read/Write</th><th>Default/Hex</th><th>Description</th></tr>
<tr><td>10</td><td>R/W</td><td>0x0</td><td>RESET_LED_EN
Write operation:
The software writes 1 to the bit, the CPU triggers LEDC to transfer a reset to LED.
Only when LEDC is in IDLE status, the reset can be performed. After the reset finished, the control state machine returns to the IDLE status. To return LEDC to the IDLE status, it also needs to be used with SOFT_RESET.
When the software sets the bit, the software can read the bit to check if the reset is complete.
Read operation:
0: LEDC completes the transmission of the LED reset operation
1: LEDC does not complete the transmission of the LED reset operation</td></tr>
<tr><td>9</td><td>/</td><td>/</td><td>/</td></tr>
<tr><td>8:6</td><td>R/W</td><td>0x0</td><td>LED_RGB_MODE
000 GRB (bypass)
001 GBR
010 RGB
011 RBG
100 BGR
101 BRG
By default, the software configures data to LEDC according to GRB (MSB) mode, the LEDC internal combines data to output to the external LED.
Other modes configure as follows.
<table>
<tr><th>Software Input Mode</th><th>Configuration</th><th>LEDC Output Mode</th></tr>
<tr><td rowspan="6">GRB</td><td>000</td><td>GRB</td></tr>
<tr><td>001</td><td>GBR</td></tr>
<tr><td>010</td><td>RGB</td></tr>
<tr><td>011</td><td>RBG</td></tr>
<tr><td>100</td><td>BGR</td></tr>
<tr><td>101</td><td>BRG</td></tr>
<tr><td rowspan="6">GBR</td><td>000</td><td>GBR</td></tr>
<tr><td>001</td><td>GRB</td></tr>
<tr><td>010</td><td>BGR</td></tr>
<tr><td>011</td><td>BRG</td></tr>
<tr><td>100</td><td>RGB</td></tr>
<tr><td>101</td><td>RBG</td></tr>
<tr><td rowspan="2">RGB</td><td>000</td><td>RGB</td></tr>
<tr><td>001</td><td>RBG</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th colspan="3">Offset: 0x0000</th><th colspan="4">Register Name: LEDC_CTRL_REG</th></tr>
<tr><th>Bit</th><th>Read/Write</th><th>Default/Hex</th><th colspan="4">Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td>010</td><td>GRB</td></tr>
<tr><td></td><td></td><td></td><td></td><td>011</td><td>GBR</td></tr>
<tr><td></td><td></td><td></td><td></td><td>100</td><td>BRG</td></tr>
<tr><td></td><td></td><td></td><td></td><td>101</td><td>BGR</td></tr>
<tr><td></td><td></td><td></td><td>RBG</td><td>000</td><td>RBG</td></tr>
<tr><td></td><td></td><td></td><td></td><td>001</td><td>RGB</td></tr>
<tr><td></td><td></td><td></td><td></td><td>010</td><td>BRG</td></tr>
<tr><td></td><td></td><td></td><td></td><td>011</td><td>BGR</td></tr>
<tr><td></td><td></td><td></td><td></td><td>100</td><td>GRB</td></tr>
<tr><td></td><td></td><td></td><td></td><td>101</td><td>GBR</td></tr>
<tr><td></td><td></td><td></td><td>BGR</td><td>000</td><td>BGR</td></tr>
<tr><td></td><td></td><td></td><td></td><td>001</td><td>BRG</td></tr>
<tr><td></td><td></td><td></td><td></td><td>010</td><td>GBR</td></tr>
<tr><td></td><td></td><td></td><td></td><td>011</td><td>GRB</td></tr>
<tr><td></td><td></td><td></td><td></td><td>100</td><td>RBG</td></tr>
<tr><td></td><td></td><td></td><td></td><td>101</td><td>RGB</td></tr>
<tr><td></td><td></td><td></td><td>BRG</td><td>000</td><td>BRG</td></tr>
<tr><td></td><td></td><td></td><td></td><td>001</td><td>BGR</td></tr>
<tr><td></td><td></td><td></td><td></td><td>010</td><td>RBG</td></tr>
<tr><td></td><td></td><td></td><td></td><td>011</td><td>RGB</td></tr>
<tr><td></td><td></td><td></td><td></td><td>100</td><td>GBR</td></tr>
<tr><td></td><td></td><td></td><td></td><td>101</td><td>GRB</td></tr>
<tr><td>5</td><td>R/W</td><td>0x1</td><td colspan="4">LED_MSB_TOP
Adjust sequence of the combined GRB data
0: LSB
1: MSB</td></tr>
<tr><td>4</td><td>R/W</td><td>0x1</td><td colspan="4">LED_MSB_G
MSB control for Green data
0: LSB
1: MSB</td></tr>
<tr><td>3</td><td>R/W</td><td>0x1</td><td colspan="4">LED_MSB_R
MSB control for Red data
0: LSB
1: MSB</td></tr>
<tr><td>2</td><td>R/W</td><td>0x1</td><td colspan="4">LED_MSB_B
MSB control for Blue data
0: LSB
1: MSB</td></tr>
</table>

Offset: 0x0000			Register Name: LEDC_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W1C	0x0	LEDC_SOFT_RESET LEDC soft reset Write 1 to clear it automatically. The ranges of LEDC soft reset include the following points: All internal status registers, the control state machine returns to in idle status, the LEDC FIFO read & write point is cleared to 0, the LEDC interrupt is cleared; and the affected registers are followed. 1.LEDC_CTRL_REG (LEDC_EN is cleared to 0); 2. PLL_T0&1_TIMING_CTRL_REG remains unchanged; 3. LEDC_DATA_FINISH_CNT_REG (LEDC_DATA_FINISH_CNT is cleared to 0); 4.LED_RESET_TIMING_CTRL_REG remains unchanged; 5. LEDC_WAIT_TIME_CTRL_REG remains unchanged; 6. LEDC_DMA_CTRL_REG remains unchanged; 7. LEDC_INTERRUPT_CTRL_REG remains unchanged; 8.LEDC_INT_STS _REG is cleared to 0; 9. LEDC_CLK_GATING_REG remains unchanged; 10. LEDC_FIFO_DATA_REG remains unchanged;
0	R/W	0x0	LEDC_EN LEDC Enable 0: Disable 1: Enable That the bit is enabled indicates LEDC can be started when LEDC data finished transmission or LEDC_EN is cleared to 0 by hardware in LEDC_SOFT_RESET situation.

9.10.6.2 0x0004 LEDC T0 & T1 Timing Control Register (Default Value: 0x0286_01D3)

Offset: 0x0004			Register Name: LED_T01_TIMING_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:27	/	/	/
26:21	R/W	0x14	T1H_TIME LED T1H time Unit: cycle (24 MHz), T1H_TIME =42 ns*(N+1) The default value is 882 ns, and the range is 80 ns–2560 ns. N: 1–3F. When is 0, T1H_TIME = 3F
20:16	R/W	0x6	T1L_TIME LED T1L time Unit: cycle (24 MHz), T1L_TIME =42 ns*(N+1) The default value is 294 ns, and the range is 80 ns–1280 ns. N: 1–1F. When is 0, T1L_TIME = 1F

Offset: 0x0004			Register Name: LED_T01_TIMING_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
15:11	/	/	/
10:6	R/W	0x7	T0H_TIME LED T0h time Unit: cycle (24 MHz), T0H_TIME =42 ns*(N+1) The default value is 336 ns, and the range is 80 ns–1280 ns. N: 1–1F. When is 0, T0H_TIME = 1F
5: 0	R/W	0x13	T0L_TIME LED T0l time Unit: cycle (24 MHz), T0L_TIME =42 ns*(N+1) The default value is 840 ns, and the range is 80 ns–2560 ns. N: 1–3F. When is 0, T0L_TIME = 3F

9.10.6.3 0x0008 LEDC Data Finish Counter Register (Default Value: 0x1D4C_0000)

Offset: 0x0008			Register Name: LEDC_DATA_FINISH_CNT_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:16	R/W	0x1D4C	LED_WAIT_DATA_TIME The value is the time that internal FIFO in LEDC is waiting for data. When the time is exceeded, the LEDC will send the wait_data_timeout_int interrupt. (This is an abnormal situation, software needs to reset LEDC.) The value is about 300 us by default. The adjust range is 80 ns–655 us. led_wait_data_time=42ns*(N+1). N: 1–1FFF. When the field is 0, LEDC_WAIT_DATA_TIME=1FFF
15:13	/	/	/
12: 0	R	0x0	LED_DATA_FINISH_CNT The value is the total LED data that have been sent. (Range: 0–8k)

9.10.6.4 0x000C LEDC Reset Timing Control Register (Default Value: 0x1D4C_0000)

Offset: 0x000C			Register Name: LED_RESET_TIMING_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28:16	R/W	0x1D4C	TR_TIME Reset time control of LED lamp Unit: cycle (24 MHz), tr_time=42 ns*(N+1) The default value is 300 us. The adjust range is 80 ns–327 us. N: 1–1FFF

Offset: 0x000C			Register Name: LED_RESET_TIMING_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
15:10	/	/	/
9: 0	R/W	0x0	LED_NUM The value is the number of external LED lamp. Maximum up to 1024. The default value 0 indicates that 1 LED lamp is external connected. The range is from 0 to 1023.

9.10.6.5 0x0010 LEDC Wait Time 0 Control Register (Default Value: 0x0000_00FF)

Offset: 0x0010			Register Name: LEDC_WAIT_TIME0_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8	R/W	0x0	WAIT_TIM0_EN WAIT_TIME0 enable When it is 1, the controller automatically inserts waiting time between LED package data. 0: Disable 1: Enable
7: 0	R/W	0xFF	TOTAL_WAIT_TIME0 Waiting time between 2 LED data. The LEDC output is low level. The adjust range is 80 ns–10 us. wait_time0=42 ns*(N+1) Unit: cycle (24 MHz) N: 1–FF

9.10.6.6 0x0014 LEDC Data Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: LEDC_DATA_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	W	0x0	LEDC DATA LED display data (the lower 24-bit is valid)

9.10.6.7 0x0018 LEDC DMA Control Register (Default Value: 0x0000_002F)

Offset: 0x0018			Register Name: LEDC_DMA_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:6	/	/	/
5	R/W	0x1	LEDC_DMA_EN LEDC DMA request enable 0: Disable request of DMA transfer data 1: Enable request of DMA transfer data

Offset: 0x0018			Register Name: LEDC_DMA_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
4: 0	R/W	0xF	LEDC_FIFO_TRIG_LEVEL The remaining space of internal FIFO in LEDC The internal FIFO in LEDC is 24*32. When the remaining space of internal FIFO in LEDC is more than or equal to LEDFIFO_TRIG_LEVEL, the DMA or the CPU request will generate. The default value is 15. The adjusted value is from 1 to 31. The recommended configuration is 7 or 15. When the configuration value is 0, LEDFIFO_TRIG_LEVEL=F.

9.10.6.8 0x001C LEDC Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: LEDC_INTERRUPT_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:6	/	/	/
5	R/W	0x0	GLOBAL_INT_EN Global interrupt enable 0: Disable 1: Enable
4	R/W	0x0	FIFO_OVERFLOW_INT_EN FIFO overflow interrupt enable When the data written by the software is more than the internal FIFO level of LEDC, the LEDC is in the data loss state. 0: Disable 1: Enable
3	R/W	0x0	WAITDATA_TIMEOUT_INT_EN The internal FIFO in LEDC cannot get data because of some abnormal situation, after the time of led_wait_data_time, the interrupt will be enabled. 0: Disable 1: Enable
2	/	/	/
1	R/W	0x0	FIFO_CPUREQ_INT_EN FIFO request CPU data interrupt enable 0: Disable 1: Enable
0	R/W	0x0	LED_TRANS_FINISH_INT_EN Data transmission complete interrupt enable 0: Disable 1: Enable

9.10.6.9 0x0020 LEDC Interrupt Status Register (Default Value: 0x0002_0000)

Offset: 0x0020			Register Name: LEDC_INT_STS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
17	R	0x1	FIFO_EMPTY FIFO empty status flag
16	R	0x0	FIFO_FULL FIFO full status flag
15:10	R	0x0	FIFO_WLW FIFO internal valid data depth It indicates the space FIFO has been occupied.
9:5	/	/	/
4	R/W1C	0x0	FIFO_OVERFLOW_INT FIFO overflow interrupt The data written by external is more than the maximum storage space of LED FIFO, the LEDC will be in the data loss state. At this time, the software needs to deal with the abnormal situation. The processing mode is as follows. The software can query LED_FIFO_DATA_REG to determine which data has been stored in the internal FIFO of LEDC. The LEDC performs soft_reset operation to refresh all data. 0: FIFO not overflow 1: FIFO overflow
3	R/W1C	0x0	WAITDATA_TIMEOUT_INT When internal FIFO of LEDC cannot get data because of some abnormal situation, after led_wait_data_time, the timeout interrupt is set, the LEDC is in WAIT_DATA state, the LEDC outputs a level state configured by LED_POLARITY; in the course of wait_data, if the new data arrives, the LEDC will continue to send data, at this time software needs to notice whether the waiting time of LEDC exceeds the operation time of reset. If the waiting time of LEDC exceeds the operation time of reset (this is equivalent to reset operation sent by LEDC), the LED may enter in refresh state, the data has not been sent. 0: LEDC not timeout 1: LEDC timeout
2	/	/	/
1	R/W1C	0x0	FIFO_CPUREQ_INT FIFO request CPU data interrupt When FIFO data is less than the threshold, the interrupt will be reported to the CPU. 0: FIFO does not request that CPU transfers data 1: FIFO requests that CPU transfers data

Offset: 0x0020			Register Name: LEDC_INT_STS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W1C	0x0	LED_TRANS_FINISH_INT Data transmission complete interrupt The value indicates that the data configured as total_data_length is transferred completely. 0: Data is not transferred completely 1: Data is transferred completely

9.10.6.10 0x0028 LEDC Wait Time 1 Control Register (Default Value: 0x01FF_FFFF)

Offset: 0x0028			Register Name: LEDC_WAIT_TIME1_CTRL_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	WAIT_TIM1_EN 0: Disable 1: Enable WAIT_TIME1 enable When the bit is 1, the controller automatically inserts the waiting time between the LED frame data.
30: 0	R/W	0x01FFFFFF	TOTAL_WAIT_TIME1 Waiting time between 2 frame data. The LEDC output is low level. The adjust range is 80 ns– 85 s. wait_time1=42 ns*(N+1) Unit: cycle (24 MHz) N: 0x80–0x7FFFFFFF If the value is 0, TOTAL_WAIT_TIME1=0x7FFFFFFF

9.10.6.11 0x0030+N*0x04 LEDC FIFO Data Register X (Default Value: 0x0000_0000)

Offset: 0x0030+N*0x04 (N=0–31)			Register Name: LEDC_FIFO_DATA_X
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	LEDC_FIFO_DATA_X Internal FIFO data of LEDC The lower 24-bit is valid.

9.11 CIR Receiver

9.11.1 Overview

The Consumer Infrared (CIR) receiver captures pulse from the IR Receiver module and uses the Run-Length Code (RLC) to encode the pulse.

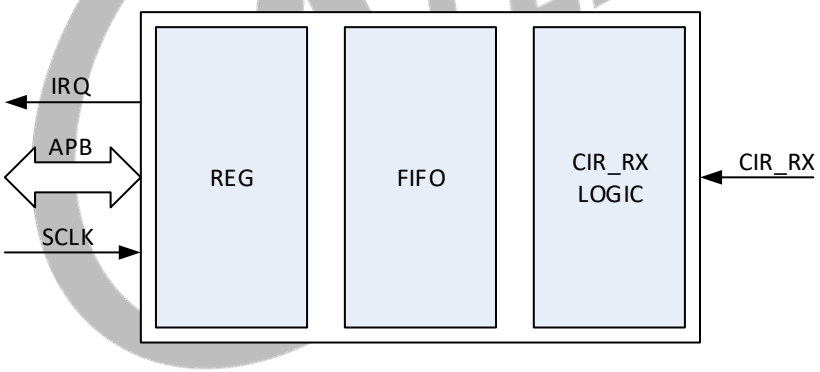
The CIR receiver has the following features:

- Supports CIR remote control receiver
- Supports NEC IR protocol
- 64x8 bits RX FIFO for data buffer
- Programmable RX FIFO thresholds
- Supports interrupt
- Sample clock up to 1 MHz

9.11.2 Block Diagram

The following figure shows the block diagram of CIR Receiver.

Figure 9-83 CIR Receiver Block Diagram



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 64 levels and 8-bit width RX FIFO; the MSB bit is used to record the polarity of the receiving CIR signal, the rest 7 bits are used for the length of RLC. The maximum length of the RLC is 128. If the duration of one level (high or low level) is more than 128, another byte is used.

9.11.3 Functional Description

9.11.3.1 External Signals

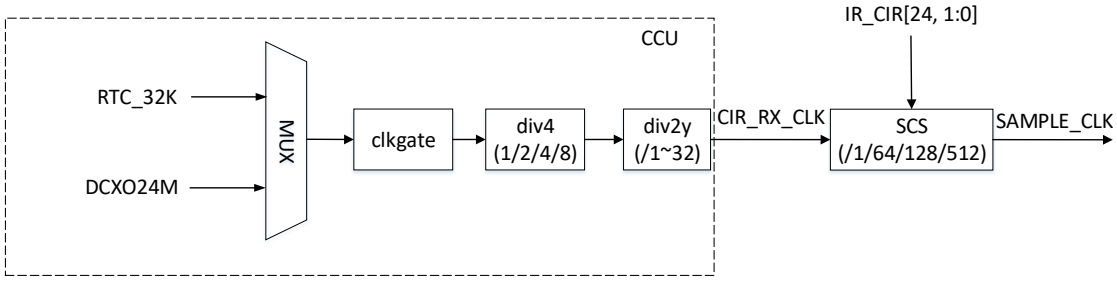
The following table describes the external signals of CIR Receiver.

Table 9-30 CIR Receiver External Signals

Signal	Description	Type
IR_RX	Consumer Infrared Receiver	I

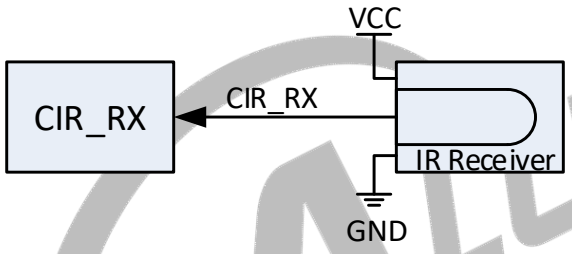
9.11.3.2 Clock Sources

Figure 9-84 CIR Receiver Clock



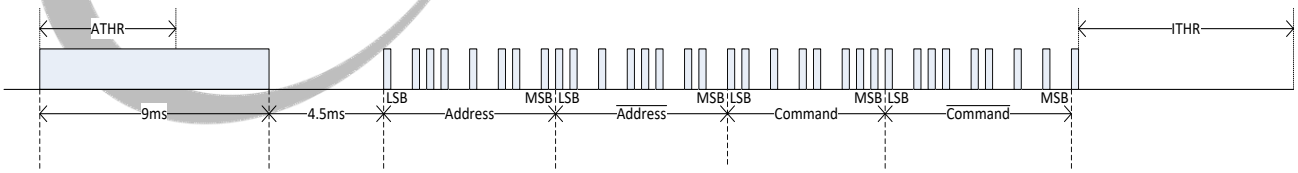
9.11.3.3 Typical Application

Figure 9-85 CIR Receiver Application Diagram



9.11.3.4 NEC Protocol Format

Figure 9-86 NEC Protocol



The CIR receiver module is a timer with a capture function.

When CIR_RX signals satisfy the Active Threshold (ATHR), the 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 the Receiver Packet End Flag is asserted.

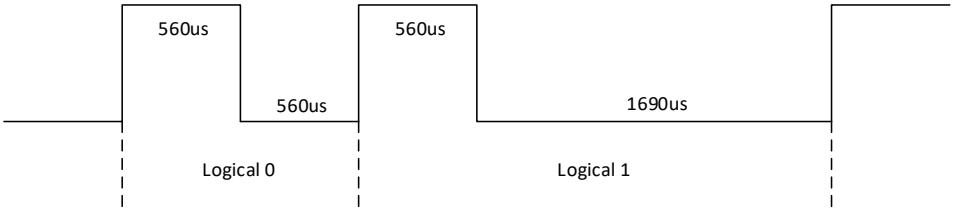
In a capture process, every effective pulse is buffered to FIFO in bytes according to the form of the Run-Length Code. The MSB bit of a byte is the polarity of pulse, and the rest 7 bits is pulse width by taking Sample Clock as a basic unit. This is the code form of the RLC-Byte. When the level changes or the pulse width counter

overflows, the RLC-Byte is buffered to FIFO. The CIR_RX module receives the infrared signals transmitted by the infrared remote control, the software decodes the signals.

9.11.3.5 Operating Mode

Sample Clock

Figure 9-87 Logical '0' and Logical '1' of NEC Protocol



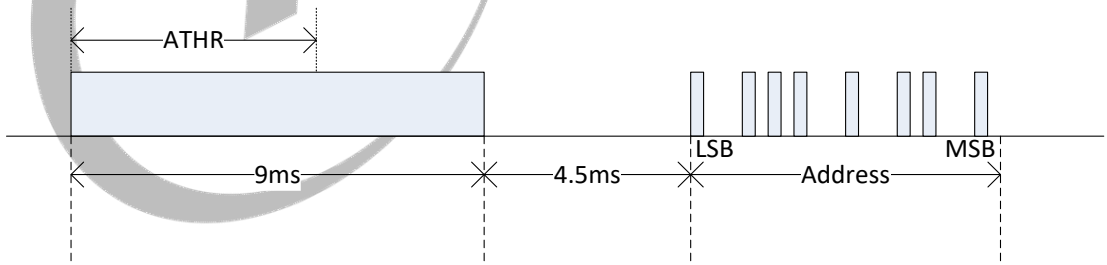
For NEC protocol, a logical "1" takes 2.25 ms (560 us+1680 us) to transmit, while a logical "0" is only half of that, being 1.12 ms (560 us+560 us).

For example, if the sample clock is 31.25 kHz, a sample cycle is 32 us, then 18 sample cycles are 560 us. So the RLC of 560 us low level is 0x12 (b'00010010), the RLC of 560 us high level is 0x92 (b'10010010). Then a logical "1" takes code 0x12 (b'00010010) and code 0xb5 (b'10110101) to transmit, a logical "0" takes code 0x12 and code 0x92 to transmit.

Active Threshold (ATHR)

When the CIR receiver is in Idle state, if the electrical level of the CIR_RX signal changes (positive jump or negative jump), and the duration reaches this threshold, then the CIR receiver takes the starting of the signal as a lead code, and the CIR receiver turns into an active state and starts to capture CIR_RX signals.

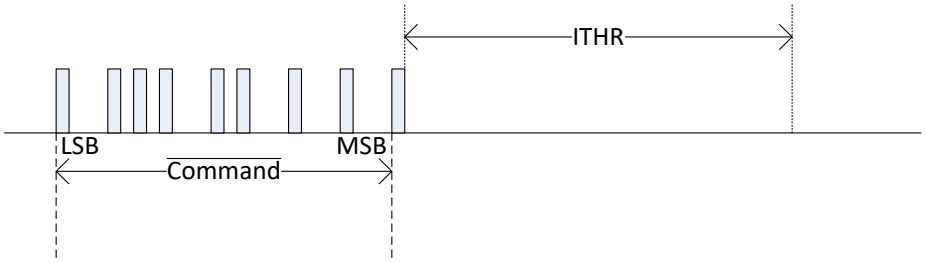
Figure 9-88 ATHR Definition



Idle Threshold (ITHR)

If the electrical level of CIR_RX signals has no change, and the duration reaches this threshold, then the CIR receiver enters into Idle state and ends this capture.

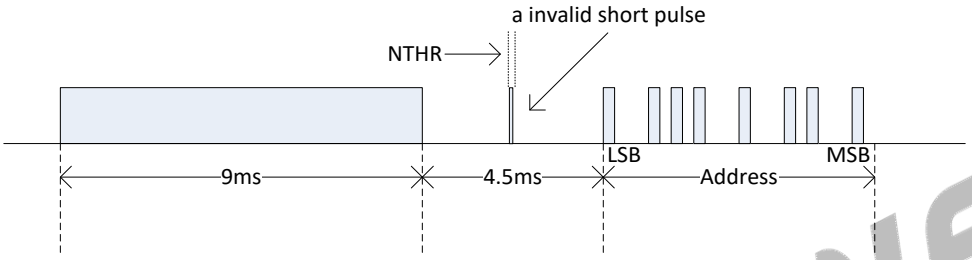
ITHR Definition



Noise Threshold (NTHR)

In the capture process, the pulse is ignored if the pulse width is less than the Noise Threshold.

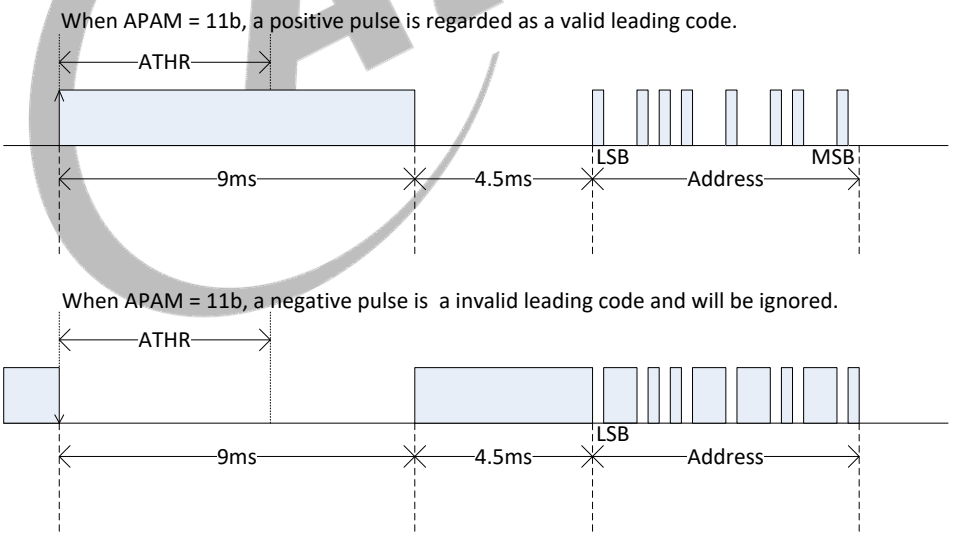
Figure 9-89 NTHR Definition



Active Pulse Accept Mode (APAM)

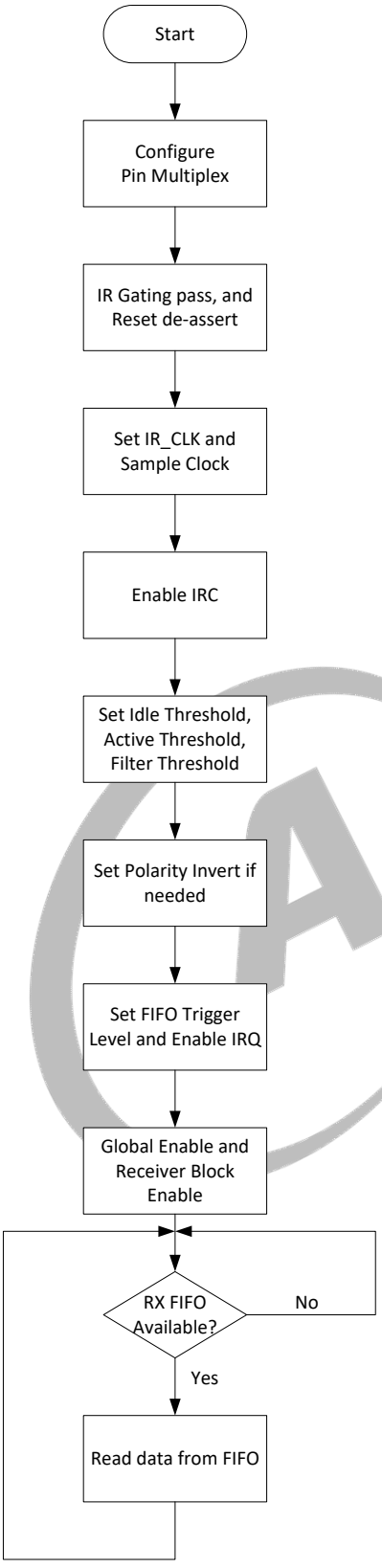
The APAM is used to fit the type of lead code. If a pulse does not fit the type of lead code, it is not regarded as a lead code even if the pulse width reaches ATHR.

Figure 9-90 APAM Definition



9.11.4 Programming Guidelines

Figure 9-91 CIR Receiver Process



9.11.5 Register List

Module Name	Base Address
IRRX	0x40046000

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

9.11.6 Register Description

9.11.6.1 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	APAM 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

9.11.6.2 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	/	/	/

9.11.6.3 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	RBF Receiver Byte FIFO

9.11.6.4 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 it is set to '1', the Receiver FIFO DRQ is asserted if reaching RAL. The DRQ is de-asserted when the condition fails.
4	R/W	0x0	RAI_EN RX FIFO Available Interrupt Enable 0: Disable 1: Enable When it is set to '1', the Receiver FIFO IRQ is asserted if reaching RAL. The IRQ is de-asserted when the condition fails.
3:2	/	/	/
1	R/W	0x0	RPEI_EN Receiver Packet End Interrupt Enable 0: Disable 1: Enable

Offset: 0x002C			Register Name: CIR_RXINT
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	ROI_EN Receiver FIFO Overrun Interrupt Enable 0: Disable 1: Enable

9.11.6.5 0x0030 CIR Receiver Status Register (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: CIR_RXSTA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:15	/	/	/
14:8	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 to its level 1: RX FIFO available according to its level Writing 1 clears this bit.
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. Writing 1 clears this bit.
0	R/W1C	0x0	ROI Receiver FIFO Overrun 0: Receiver FIFO not overrun 1: Receiver FIFO overrun Writing 1 clears this bit.

9.11.6.6 0x0034 CIR Receiver Configure Register (Default Value: 0x0000_1828)

Offset: 0x0034			Register Name: CIR_RXCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	SCS2 Bit2 of Sample Clock Select for CIR This bit is defined by SCS bits below.
23	R/W	0x0	ATHC Active Threshold Control for CIR 0: ATHR in a unit of (Sample Clock) 1: ATHR in a unit of (128*Sample Clocks)
22:16	R/W	0x0	ATHR Active Threshold for CIR These bits control the duration of CIR from the idle to the active state. The duration can be calculated by ((ATHR + 1)*(ATHC? Sample Clock: 128*Sample Clock)).
15:8	R/W	0x18	ITHR Idle Threshold for CIR The Receiver uses it to decide whether the CIR command is 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 receiver ends, 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 the CPU. When the duration of the signal keeps one status (high or low level) for the specified duration ((ITHR + 1)*128 sample_clk), this means that the previous CIR command is finished.
7:2	R/W	0xA	NTHR Noise Threshold for CIR When the duration of the 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.

<table>
<tr><th colspan="3">Offset: 0x0034</th><th>Register Name: CIR_RXCFG</th></tr>
<tr><th>Bit</th><th>Read/Write</th><th>Default/Hex</th><th>Description</th></tr>
<tr><td>1: 0</td><td>R/W</td><td>0x0</td><td>SCS
Sample Clock Select for CIR
<table>
<tr><th>SCS2</th><th>SCS[1]</th><th>SCS[0]</th><th>Sample Clock</th></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>
<tr><td>0</td><td>1</td><td>0</td><td>CIR_CLK/256</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>CIR_CLK/512</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>CIR_CLK</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Reserved</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Reserved</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Reserved</td></tr>
</table></td></tr>
</table>

9.12 CIR Transmitter

9.12.1 Overview

The CIR transmitter (CIR_TX) can transfer arbitrary waves which can be modulated with configurable carrier waves such as 38 kHz.

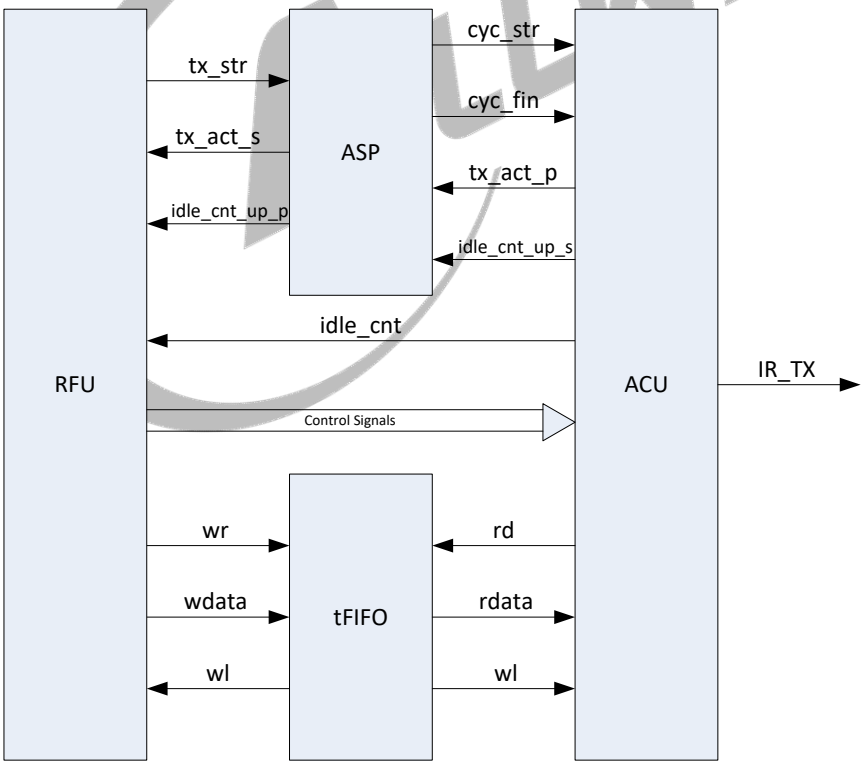
The CIR_TX has the following features:

- Supports CIR remote control transmitter
- 128 bytes FIFO for data buffer
- Configurable carrier frequency
- Supports Interrupt and DMA
- Supports handshake mode and waiting mode of DMA

9.12.2 Block Diagram

The following figure shows a block diagram of the CIR_TX.

Figure 9-92 CIR_TX Block Diagram



9.12.3 Functional Description

9.12.3.1 External Signals

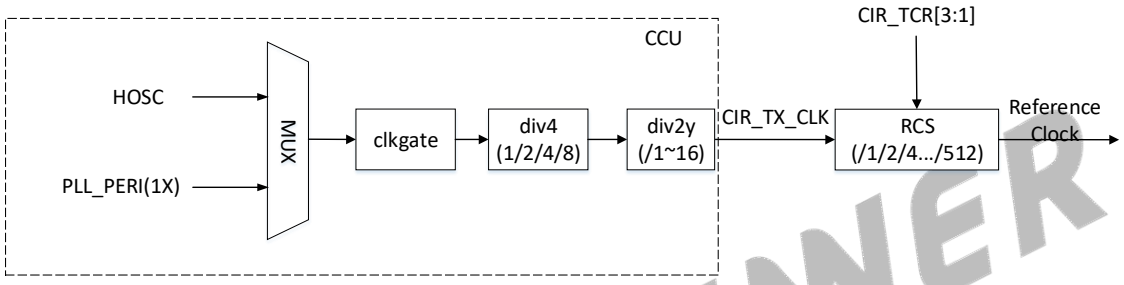
The following table describes the external signals of CIR_TX.

Figure 9-93 CIR_TX External Signals

Signal	Description	Type
CIR_TX	Consumer infrared transmitter	O

9.12.3.2 Clock Sources

Figure 9-94 CIR_TX Clock Description



9.12.3.3 Function Implementation

The CIR_TX is used to generate a waveform of arbitrary length, arbitrary shape, and no high-speed requirement, and it can change the data into the high-/low-level sequence of a certain length. Every transmitting data is in bytes, the Bit[7] of a byte means whether the level of a transmitting wave is high or low, the Bit[6: 0] is the length of this wave. If the current transmitting frequency-division is 1, 0x88 is a high level of 8 cycles, 0x08 is a low level of 8 cycles. If the current transmitting frequency-division is 4, 0x88 is a high level of 32 cycles, 0x08 is a low level of 32 cycles.

The CIR_TX has two transmission modes: Non-cycle transmission, and cycle transmission.

The non-cycle transmission is to transmit all the data in TX_FIFO until the FIFO is empty.

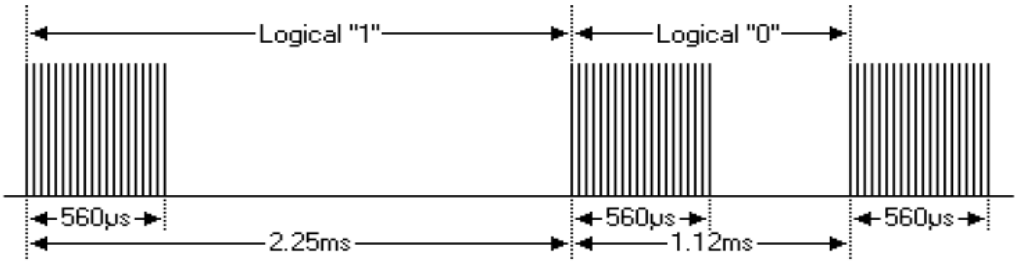
The cycle transmission is to transmit all the data in TX_FIFO, after the transmission completion, wait for a certain time to recover the data in TX_FIFO and then send it until a stop signal is detected. The data recovery in FIFO is implemented by clearing the read pointer.

9.12.3.4 Timing Diagram

The CIR remote control contains many protocols designed by different manufacturers. Here to NEC protocol as an example, the CIR-TX module uses a variable pulse-width modulation technique to encompass the various formats of infrared encoding for remote-control applications. A message is started by a 9 ms AGC burst, which is used to set the gain of the earlier CIR receivers. This AGC burst is then followed by a 4.5 ms space, which is then followed by the address and command.

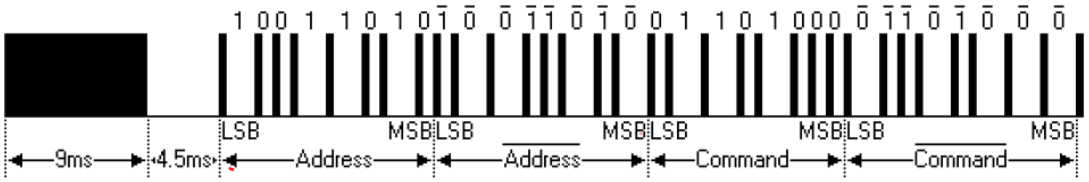
Bit definition: The logical “1” takes 2.25 ms to transmit, while a logical “0” is only 1.12 ms.

Figure 9-95 Definitions of Logical “1” and Logical “0”



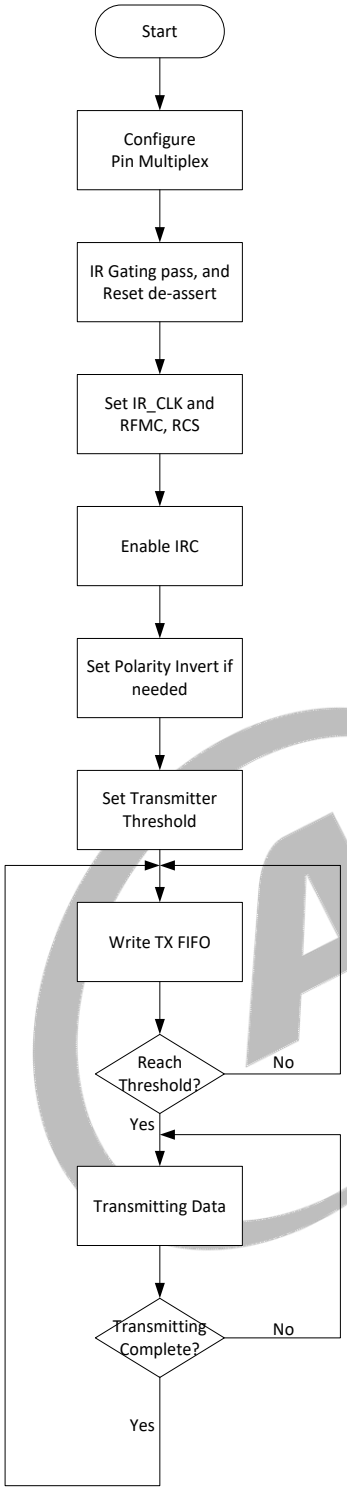
Timing for a message:

Figure 9-96 CIR Message Timing Diagram



9.12.4 Programming Guidelines

Figure 9-97 CIR Transmitter Process



9.12.5 Register List

Module Name	Base Address
IRTX	0x40046400

Register Name	Offset	Description
CIR_TGLR	0x0000	CIR Transmit Global Register
CIR_TMCR	0x0004	CIR Transmit Modulation Control Register
CIR_TCR	0x0008	CIR Transmit Control Register
CIR_IDC_H	0x000C	CIR Transmit Idle Duration Threshold High Bit Register
CIR_IDC_L	0x0010	CIR Transmit Idle Duration Threshold Low Bit Register
CIR_TICR_H	0x0014	CIR Transmit Idle Counter High Bit Register
CIR_TICR_L	0x0018	CIR Transmit Idle Counter Low Bit Register
CIR_TEL	0x0020	CIR TX FIFO Empty Level Register
CIR_TXINT	0x0024	CIR Transmit Interrupt Control Register
CIR_TAC	0x0028	CIR Transmit FIFO Available Counter Register
CIR_TXSTA	0x002C	CIR Transmit Status Register
CIR_TXT	0x0030	CIR Transmit Threshold Register
CIR_DMA	0x0034	CIR DMA Control Register
CIR_TXFIFO	0x0080	CIR Transmit FIFO Data Register

9.12.6 Register Description

9.12.6.1 0x0000 CIR Transmitter Global Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: CIR_TGLR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	IMS Internal Modulation Select 0: The transmitting signal is not modulated 1: The transmitting signal is modulated internally
6:5	R/W	0x0	DRMC Duty ratio of modulated carrier is high level/low level. 00: Low level is equal to high level 01: Low level is the double of high level 10: Low level is the triple of high level 11: Reserved
4:3	/	/	/
2	R/W	0x0	TPPI Transmit Pulse Polarity Invert 0: Not invert transmit pulse 1: Invert transmit pulse
1	R/W	0x0	TR Transmit Reset When this bit is set, the transmitting is reset. The FIFO will be flushed, the TIC filed and the CSS field will be cleared during Transmit Reset. This field will automatically be cleared when the Transmit Reset is finished, and the CIR transmitter will state Idle.

Offset: 0x0000			Register Name: CIR_TGLR
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	TXEN Transmit Block Enable 0: Disable the CIR Transmitter 1: Enable the CIR Transmitter

9.12.6.2 0x0004 CIR Transmitter Modulation Control Register (Default Value: 0x0000_009E)

Offset: 0x0004			Register Name: CIR_TMCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7: 0	R/W	0x9E	RFMC Reference Frequency of modulated carrier. Reference Frequency of modulated carrier based on a division of a fixed functional clock (FCLK). The range of the modulated carrier is usually 30 kHz to 60 kHz. Most consumer electronics is 38Hz. **Note**: The *default modulated carrier is 38 kHz when FCLK is 12 MHz. RFMC= FCLK/((N+1)*(DRMC+2)).*

9.12.6.3 0x0008 CIR Transmitter Control Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: CIR_TCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	CSS Cyclical Pulse Start/Stop Control 0: Stop when cleared to '0'. From start to stop, all data in FIFO must be transmitted. 1: Start. Start to transmit when it is set to '1'.
6:4	/	/	/
3:1	R/W	0x0	RCS Reference Clock Select for CIR Transmit The data in TX_FIFO is used to describe the pulse in Run-Length Code. The basic unit of pulse width is Reference Clock. 000: CIR Transmit reference clock is ir_clk 001: CIR Transmit reference clock is ir_clk/2 010: CIR Transmit reference clock is ir_clk/4 011: CIR Transmit reference clock is ir_clk/8 100: CIR Transmit reference clock is ir_clk/64 101: CIR Transmit reference clock is ir_clk/128 110: CIR Transmit reference clock is ir_clk/256 111: CIR Transmit reference clock is ir_clk/512

Offset: 0x0008			Register Name: CIR_TCR
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	TTS Type of the transmission signal 0: The transmitting wave is a single non-cyclical pulse. 1: The transmitting wave is a cyclical short-pulse.

9.12.6.4 0x000C CIR Transmitter Idle Duration Counter High Bit Register (Default Value: 0x0000_0000)

Offset: 0x000C			Register Name: CIR_IDC_H
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
3: 0	R/W	0x0	IDC_H Idle Duration Counter Threshold (High 4 bits) Idle Duration = 128*IDC*Ts (IDC = 0~4095) It is used in cyclical transmission mode. When all the data in FIFO is transmitted, the signals can be transmitted after a specific time.

9.12.6.5 0x0010 CIR Transmitter Idle Duration Counter Low Bit Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: CIR_IDC_L
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
3: 0	R/W	0x0	IDC_L Idle Duration Counter Threshold (Low 8 bits) Idle Duration = 128*IDC*Ts (IDC = 0~4095) It is used in cyclical transmission mode. When all the data in FIFO is transmitted, the signals can be transmitted after a specific time.

9.12.6.6 0x0014 CIR Transmitter Idle Counter High Bit Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: CIR_TICR_H
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7: 0	R	0x0	TIC_H Transmit Idle Counter_H (High 8 bits) It is used to count the idle duration of CIR transmitter by software. Count in 128*Ts (Sample Duration, 1/Fs) when the transmitter is idle, and it should be reset when the transmitter is active. When this counter reaches the maximum value (0xFFFF), it will stop automatically, and should not be cleared to zero.

9.12.6.7 0x0018 CIR Transmitter Idle Counter Low Bit Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: CIR_TICR_L
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7: 0	R	0x0	TIC_L Transmit Idle Counter_L (Low 8 bits) It is used to count the idle duration of CIR transmitter by software. Count in 128*Ts (Sample Duration, 1/Fs) when the transmitter is idle, and it should be reset when the transmitter is active. When this counter reaches the maximum value (0xFFFF), it will stop automatically, and should not be cleared to zero.

9.12.6.8 0x0020 CIR Transmitter FIFO Empty Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: CIR_TEL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7: 0	R/W	0x0	TEL TX FIFO empty Level for DRQ and IRQ. TRIGGER_LEVEL = TEL + 1

9.12.6.9 0x0024 CIR Transmitter Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: CIR_TXINT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x0	DRQ_EN TX FIFO DMA Enable 0: Disable 1: Enable When it is set to '1', the TX FIFO DRQ is asserted if the number of the transmitting data in the FIFO is less than the RAL. The DRQ is de-asserted when the condition fails.
1	R/W	0x0	TAI_EN TX FIFO Available Interrupt Enable 0: Disable 1: Enable

Offset: 0x0024			Register Name: CIR_TXINT
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	TPEI_EN Transmit Packet End Interrupt Enable for Cyclical Pulse 0: Disable 1: Enable TUI_EN Transmitter FIFO Underrun Interrupt Enable for Non-cyclical Pulse 0: Disable 1: Enable

9.12.6.10 0x0028 CIR Transmitter FIFO Available Counter Register (Default Value: 0x0000_0080)

Offset: 0x0028			Register Name: CIR_TAC
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7: 0	R	0x80	TAC TX FIFO Available Space Counter 0x00: No available space in TX FIFO 0x01: 1 Byte available space in TX FIFO 0x02: 2 Bytes available space in TX FIFO … 0x80: 128 bytes available space in TX FIFO

9.12.6.11 0x002C CIR Transmitter Status Register (Default Value: 0x0000_0002)

Offset: 0x002C			Register Name: CIR_TXSTA
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3	R	0x0	STCT Status of CIR Transmitter 0: Idle 1: Active This bit will automatically set when the controller begins to transmit the data in the FIFO. The “1” will last when the data in the FIFO. It will automatically be cleared to “0” when all data in the FIFO is transmitted. The bit is for debugging. The output Level of Idle state is determined by the level of the last data output.

Offset: 0x002C			Register Name: CIR_TXSTA
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R	0x0	DRQ DMA Request Flag When set to '1', the TX FIFO DRQ is asserted if the number of the transmission data in the FIFO is less than the RAL. The DRQ is de-asserted when the condition fails. This bit is for debugging.
1	R/W	0x1	TAI TX FIFO Available Interrupt Flag 0: TX FIFO not available by its level 1: TX FIFO available by its level Writing 1 clears this bit.
0	R/W	0x0	TPE Transmitter Packet End Flag for Cyclical Pulse 0: Transmissions of address, control and data fields not completed 1: Transmissions of address, control and data fields completed TUR Transmitter FIFO Underrun Flag for Non-cyclical Pulse 0: No transmitter FIFO underrun 1: Transmitter FIFO underrun Writing 1 clears this bit.

9.12.6.12 0x0030 CIR Transmitter Threshold Register (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: CIR_TXT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7: 0	R/W	0x0	NCTT Non-cyclical Pulse Transmit Threshold The controller will trigger transmitting the data in the FIFO when the data byte number has reached the Transmit Threshold set in this field.

9.12.6.13 0x0034 CIR Transmitter DMA Control Register (Default Value: 0x0000_00A5)

Offset: 0x0034			Register Name: CIR_DMA_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7: 0	R/W	0xA5	DMA Handshake Configuration 0xA5: DMA waiting cycle mode 0xEA: DMA handshake mode

9.12.6.14 0x0080 CIR Transmitter FIFO Data Register (Default Value: 0x0000_0000)

Offset: 0x0080			Register Name: CIR_TXFIFO
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7: 0	W	0x0	TBF Transmit Byte FIFO When the transmission is triggered, the data in the FIFO will be transmitted until the data number is transmitted completely.

9.13 Smart Card Reader (SCR)

9.13.1 Overview

The Smart Card Reader (SCR) is a communication controller that transmits data between the system and Smart Card. The controller can perform a complete smart card session, including card activation, card deactivation. Cold/warm reset, Answer to Reset (ATR) response reception, data transmission, etc.

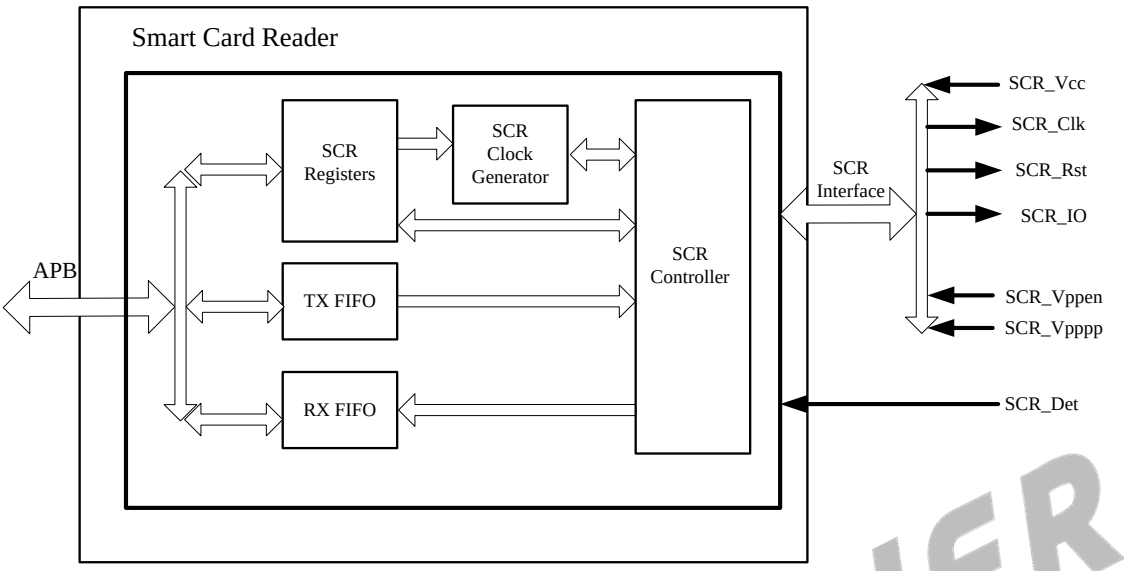
The SCR includes the following features:

- Supports the ISO/IEC 7816-3:1997(E) and EMV2000 (4.0) Specifications
- Performs functions needed for complete smart card sessions, including:
  - Card activation and deactivation
  - Cold/warm reset
  - Answer to Reset (ATR) response reception
  - Data transfers to and from the card
- Supports adjustable clock rate and bit rate
- Configurable automatic byte repetition
- Supports commonly used communication protocols:
  - T=0 for asynchronous half-duplex character transmission
  - T=1 for asynchronous half-duplex block transmission
- 128-bit FIFO for data transmit & receive.
- Supports FIFOs for receive and transmit buffers (up to 128 bits) with threshold
- Supports configurable timing functions:
  - Smart card activation time
  - Smart card reset time
  - Guard time
  - Timeout timers
- Supports synchronous and any other non-ISO 7816 and non-EMV cards

9.13.2 Block Diagram

The following figure shows the block diagram of the SCR.

Figure 9-98 SCR Block Diagram



9.13.3 Functional Description

9.13.3.1 External Signals

The following table describes the external signals of SCR.

Table 9-31 SCR External Signals

Signal	Description	Type
SIM_DATA	Data signal	I/O
SIM_CLK	Clock of SCR	O
SIM_DET	Card Detect Signal	I
SIM_RST	Reset Signal	O

9.13.3.2 Clock Sources

The following table describes the clock source of SCR. For clock setting, configurations and gating information, refer to the section "CCU" and "CCU_AON".

Table 9-32 SCR Clock Source

Clock Sources	Description
PCLK_SRC	apb bus clock, default value is 96MHz
CLK_HOSC	24MHz Crystal

9.13.3.3 Timing Diagram

Please refer ISO/IEC 7816 and EMV2000 Specification.

9.13.4 Programming Guidelines

9.13.4.1 Clock Generator

The Clock Generator generates the Smart Card Clock signal and the Baud Clock Impulse signal, used in timing the Smart Card Reader.

The Smart Card Clock signal is used as the main clock for the smart card. Its frequency can be adjusted using the Smart Card Clock Divisor (SCCDIV). This value is used to divide the system clock. The SCCLK frequency is given by the following equation:

$$f_{scclk} = \frac{f_{sysclk}}{2*(SCCDIV+1)}$$

f_{scclk} -- Smart Card Clock Frequency

f_{sysclk} -- System Clock (PCLK) Frequency

The Baud Clock Impulse signal is used to transmit and receive serial between the Smart Card Reader and the Smart Card. The baud rate can be modified using the Baud Clock Divisor (BAUDDIV). The value is used to divide the system clock. The BUAD rate is given by the following equation:

$$BAUD = \frac{f_{sysclk}}{2*(BAUDDIV+1)}$$

$BAUD$ -- Baud rate of the data stream between Smart Card and Reader

The duration of one bit, Elementary Time Unit (ETU), is defined in the ISO/IEC 7816-3 specification. During the first answer to reset response after the cold reset, the initial ETU must be equal to 372 Smart Card Clock Cycles.

$$\frac{1}{BAUD} = ETU = \frac{372}{f_{scclk}}$$

In this case, the BAUDDIV should be

$$BAUDDIV = \frac{372*f_{sysclk}}{2*f_{scclk}} - 1 = 372*(SCCDIV+1) - 1$$

After the ATR is completed, the ETU can be changed according to Smart Card abilities.

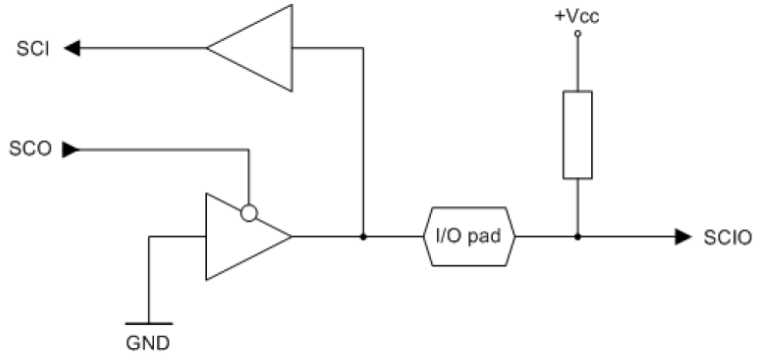
$$\frac{1}{BAUD} = ETU = \frac{F}{D} * \frac{1}{f_{scclk}}$$

Parameters F and D are defined in the ISO/IEC 7816-3 Specification. F is the clock rate conversion integer and D the baud rate adjustment integer.

9.13.4.2 SCR IO Pad Configuration

The following figure shows the SCR IO Pad Configuration.

Figure 9-99 SCR IO Pad Configuration



NOTE

Connect a pull-up resistor to the pin.

9.13.5 Register List

Module Name	Base Address
SMCARD	0x40045400

Register Name	Offset	Description
SCR_CSR	0x0000	Smart Card Reader Control and Status Register
SCR_INTEN	0x0004	Smart Card Reader Interrupt Enable Register 1
SCR_INTST	0x0008	Smart Card Reader Interrupt Status Register 1
SCR_FCSR	0x000C	Smart Card Reader FIFO Control and Status Register
SCR_FCNT	0x0010	Smart Card Reader RX and TX FIFO Counter Register
SCR_RPT	0x0014	Smart Card Reader RX and TX Repeat Register
SCR_DIV	0x0018	Smart Card Reader Clock and Baud Divisor Register
SCR_LTIM	0x001C	Smart Card Reader Line Time Register
SCR_CTIM	0x0020	Smart Card Reader Character Time Register
SCR_LCTLR	0x0030	Smart Card Reader Line Control Register
SCR_FSM	0x003C	Smart Card Reader FSM Register
SCR_DT	0x0040	Smart Card Reader Debounce Time Register
SCR_FIFO	0x0100	Smart Card Reader RX and TX FIFO Access Point

9.13.6 Register Description

9.13.6.1 0x0000 Smart Card Reader Control and Status Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: SCR_CSR
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x0	SCDET Smart Card Detected This bit is set to '1' when the scdetect input is active at least for a debounce time.
30:25	/	/	/
24	R/W	0x0	SCDETPOL Smart Card Detect Polarity This bit set polarity of scdetect signal. 0: Low Active 1: High Active
23:22	R/W	0x0	Protocol Selection (PTLSEL) 0x0: T=0. 0x1: T=1, no character repeating and no guard time is used when T=1 protocol is selected. 0x2: Reserved 0x3: Reserved
21	R/W	0x0	ATRSTFLUSH ATR Start Flush FIFO When enabled, both FIFOs are flushed before the ATR is started.
20	R/W	0x0	TSRXE TS Receive Enable When set to '1', the TS character (the first ATR character) will be store in RXFIFO during card session.
19	R/W	0x0	CLKSTPPOL Clock Stop Polarity The value of the scclk output during the clock stop state.
18	R/W	0x0	PECRXE Parity Error Character Receive Enable Enables storage of the characters received with wrong parity in RX FIFO.
17	R/W	0x0	MSBF MSB First When high, inverse bit ordering convention (msb to lsb) is used.
16	R/W	0x0	DATAPOL Data Polarity When high, inverse level convention is used (A='1', Z='0').
15:12	/	/	/

Offset: 0x0000			Register Name: SCR_CSR
Bit	**Read/Write**	**Default/Hex**	**Description**
11	R/WAC	0x0	DEACT Setting of this bit initializes the deactivation sequence. When the deactivation is finished, the DEACT bit is automatically cleared.
10	R/WAC	0x0	ACT Activation. Setting of this bit initializes the activation sequence. When the activation is finished, the ACT bit is automatically cleared.
9	W	0x0	WARMRST Warm Reset Command. Writing '1' to this bit initializes Warm Reset of the Smart Card. This bit is always read as '0'.
8	R/W	0x0	CLKSTOP Clock Stop. When this bit is asserted and the smart card I/O line is in 'Z' state, the SCR core stops driving of the smart card clock signal after the CLKSTOPDELAY time expires. The smart card clock is restarted immediately after the CLKSTOP signal is deasserted. New character transmission can be started after CLKSTARTDELAY time. The expiration of both times is signaled by the CLKSTOPRUN bit in the interrupt registers.
7:3	/	/	/
2	R/W	0x0	GINTEN Global Interrupt Enable. When high, IRQ output assertion is enabled.
1	R/W	0x0	RXEN Receiving Enable. When enabled the characters sent by the Smart Card are received by the UART and stored in RX FIFO. Receiving is internally disabled while a transmission is in progress.
0	R/W	0x0	TXEN Transmission Enable. When enabled the characters are read from TX FIFO and transmitted through UART to the Smart Card.

9.13.6.2 0x0004 Smart Card Reader Interrupt Enable Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: SCR_INTEN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23	R/W	0x0	SCDEA Smart Card Deactivation Interrupt Enable.
22	R/W	0x0	SCACT Smart Card Activation Interrupt Enable.

Offset: 0x0004			Register Name: SCR_INTEN
Bit	**Read/Write**	**Default/Hex**	**Description**
21	R/W	0x0	SCINS Smart Card Inserted Interrupt Enable.
20	R/W	0x0	SCREM Smart Card Removed Interrupt Enable.
19	R/W	0x0	ATRDONE ATR Done Interrupt Enable.
18	R/W	0x0	ATRFAIL ATR Fail Interrupt Enable.
17	R/W	0x0	C2CFULL Two Consecutive Characters Limit Interrupt Enable.
16	R/W	0x0	CLKSTOPRUN Smart Card Clock Stop/Run Interrupt Enable.
15:13	/	/	/
12	R/W	0x0	RXPERR RX Parity Error Interrupt Enable.
11	R/W	0x0	RXDONE RX Done Interrupt Enable.
10	R/W	0x0	RXFIFOTHD RX FIFO Threshold Interrupt Enable.
9	R/W	0x0	RXFIFOFULL RX FIFO Full Interrupt Enable.
8:5	/	/	/
4	R/W	0x0	TXPERR TX Parity Error Interrupt Enable.
3	R/W	0x0	TXDONE TX Done Interrupt Enable.
2	R/W	0x0	TXFIFOTHD TX FIFO Threshold Interrupt Enable.
1	R/W	0x0	TXFIFOEMPTY TX FIFO Empty Interrupt Enable.
0	R/W	0x0	TXFIFODONE TX FIFO Done Interrupt Enable.

9.13.6.3 0x0008 Smart Card Reader Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: SCR_INTST
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/

Offset: 0x0008			Register Name: SCR_INTST
Bit	**Read/Write**	**Default/Hex**	**Description**
23	R/W1C	0x0	SCDEA Smart Card Deactivation Interrupt. When enabled, this interrupt is asserted after the Smart Card deactivation sequence is complete. Write '1' to clear.
22	R/W1C	0x0	SCACT Smart Card Activation Interrupt. When enabled, this interrupt is asserted after the Smart Card activation sequence is complete. Write '1' to clear.
21	R/W1C	0x0	SCINS Smart Card Inserted Interrupt. When enabled, this interrupt is asserted after the smart card insertion. Write '1' to clear.
20	R/W1C	0x0	SCREM Smart Card Removed Interrupt. When enabled, this interrupt is asserted after the smart card removal. Write '1' to clear.
19	R/W1C	0x0	ATRDONE ATR Done Interrupt. When enabled, this interrupt is asserted after the ATR sequence is successfully completed. Write '1' to clear.
18	R/W1C	0x0	ATRFAIL ATR Fail Interrupt. When enabled, this interrupt is asserted if the ATR sequence fails. Write '1' to clear.
17	R/W1C	0x0	C2CFULL Two Consecutive Characters Limit Interrupt. When enabled, this interrupt is asserted if the time between two consecutive characters, transmitted between the Smart Card and the Reader in both directions, is equal the Two Characters Delay Limit described below. The C2CFULL interrupt is internally enabled from the ATR start to the deactivation or ATR restart initialization. It is recommended to use this counter to detect unresponsive Smart Cards. Write '1' to clear.
16	R/W1C	0x0	CLKSTOPRUN Smart Card Clock Stop/Run Interrupt. When enabled, this interrupt is asserted in two cases: When the smart card clock is stopped. When the new character can be started after the clock restart. To distinguish between the two interrupt cases, we recommend reading the CLKSTOP bit in SCR_CTRL1 register. Write '1' to clear.

Offset: 0x0008			Register Name: SCR_INTST
Bit	**Read/Write**	**Default/Hex**	**Description**
15:13	/	/	/
12	R/W1C	0x0	RXPERR RX Parity Error Interrupt. When enabled, this interrupt is asserted after the character with wrong parity was received when the number of repeated receptions exceeds RXREPEAT value or T=1 protocol is used. Write '1' to clear.
11	R/W1C	0x0	RXDONE RX Done Interrupt. When enabled, this interrupt is asserted after a character was received from the Smart Card. Write '1' to clear.
10	R/W1C	0x0	RXFIFOTHD RX FIFO Threshold Interrupt. When enabled, this interrupt is asserted if the number of bytes in RX FIFO is equal or exceeds the RX FIFO threshold. Write '1' to clear.
9	R/W1C	0x0	RXFIFOFULL RX FIFO Full Interrupt. When enabled, this interrupt is asserted if the RX FIFO is filled up. Write '1' to clear.
8:5	/	/	/
4	R/W1C	0x0	TXPERR TX Parity Error Interrupt. When enabled, this interrupt is asserted if the Smart Card signals wrong character parity during the guard time after the character transmission was repeated TXREPEAT times or T=1 protocol is used. Write '1' to clear.
3	R/W1C	0x0	TXDONE TX Done Interrupt. When enabled, this interrupt is asserted after one character was transmitted to the smart card. Write '1' to clear.
2	R/W1C	0x0	TXFIFOTHD TX FIFO Threshold Interrupt. When enabled, this interrupt is asserted if the number of bytes in TX FIFO is equal or less than the TX FIFO threshold. Write '1' to clear.
1	R/W1C	0x0	TXFIFOEMPTY TX FIFO Empty Interrupt. When enabled, this interrupt is asserted if the TX FIFO is emptied out. Write '1' to clear.

Offset: 0x0008			Register Name: SCR_INTST
Bit	Read/Write	Default/Hex	Description
0	R/W1C	0x0	TXFIFODONE TX FIFO Done Interrupt. When enabled, this interrupt is asserted after all bytes from TX FIFO ware transferred to the Smart Card. Write '1' to clear.

9.13.6.4 0x000C Smart Card Reader FIFO Control and Status Register (Default Value: 0x0000_0101)

Offset: 0x000c			Register Name: SCR_FCSR
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10	W	0x0	RXFIFOFLUSH Flush RX FIFO. RX FIFO is flushed, when '1' is written to this bit.
9	R	0x0	RXFIFOFULL RX FIFO Full.
8	R	0x1	RXFIFOEMPTY RX FIFO Empty.
7:3	/	/	/
2	W	0x0	TXFIFOFLUSH Flush TX FIFO. TX FIFO is flushed, when '1' is written to this bit.
1	R	0x0	TXFIFOFULL TX FIFO Full.
0	R	0x1	TXFIFOEMPTY TX FIFO Empty.

9.13.6.5 0x0010 Smart Card Reader FIFO Count Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: SCR_FIFOCNT
Bit	Read/Write	Default/Hex	Description
31:29	/	/	/
28:24	R/W	0x0	RXFTH RX FIFO Threshold These bits set the interrupt threshold of RX FIFO. The interrupt is asserted when the number of bytes it receives is equal to, or exceeds the threshold.
23:21	/	/	/
20:16	R/W	0x0	TXFTH TX FIFO Threshold These bits set the interrupt threshold of TX FIFO. The interrupt is asserted when the number of bytes in TX FIFO is equal to or less than the threshold.

Offset: 0x0010			Register Name: SCR_FIFOCNT
Bit	**Read/Write**	**Default/Hex**	**Description**
15:8	R	0x0	RXFCNT RX FIFO Counter These bits provide the number of bytes stored in the RXFIFO.
7: 0	R	0x0	TXFCNT TX FIFO Counter These bits provide the number of bytes stored in the TXFIFO.

9.13.6.6 0x0014 Smart Card Reader Repeat Control Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: SCR_REPEAT
Bit	**Read/Write**	**Default/Hex**	**Description**
15:8	/	/	/
7:4	R/W	0x0	RXRPT RX Repeat This is a 4-bit register that specifies the number of attempts to request character re-transmission after wrong parity was detected. The re-transmission of the character is requested using the error signal during the guard time.
3: 0	R/W	0x0	TXRPT TX Repeat This is a 4-bit register that specifies the number of attempts to re-transmit the character after the Smart Card signals the wrong parity during the guard time.

9.13.6.7 0x0018 Smart Card Reader Clock Divisor Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: SCR_CLKDIV
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	BAUDDIV Baud Clock Divisor. This 16-bit register defines the divisor value used to generate the Baud Clock impulses from the system clock. $BAUD = \frac{f_{sysclk}}{2*(BAUDDIV+1)}$

Offset: 0x0018			Register Name: SCR_CLKDIV
Bit	**Read/Write**	**Default/Hex**	**Description**
15: 0	R/W	0x0	SCCDIV Smart Card Clock Divisor. This 16-bit register defines the divisor value used to generate the Smart Card Clock from the system clock. $f_{scclk} = \frac{f_{sysclk}}{2*(SCCDIV+1)}$ f_{scclk} is the frequency of Smart Card Clock Signal. f_{sysclk} is the frequency of APB Clock.

9.13.6.8 0x001C Smart Card Reader Line Time Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: SCR_LTIM
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:16	R/W	0x0	ATR ATR Start Limit. This 8-bit register defines the maximum time between the rising edge of the *scrstn* signal and the start of ATR response. ATR Start Limit = 256* ATR* T_{scclk} .
15:8	R/W	0x0	RST Reset Duration. This 16-bit register sets the duration of the Smart Card reset sequence. This value is same for the cold and warm reset. Cold/Warm Reset Duration = 256* RST* T_{scclk} .
7: 0	R/W	0x0	ACT Activation/Deactivation Time. This 16-bit register sets the duration of each part of the activation and deactivation sequence. Activation/Deactivation Duration = 256* ACT * T_{scclk} . $T_{scclk} = \frac{1}{f_{scclk}}$ is the Smart Card Clock Cycle.

9.13.6.9 0x0020 Smart Card Reader Character Time Register (Default Value: 0x0000_0000)

Offset: 0x0020	Register Name: SCR_CTIM

Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x0	CHARLIMIT Character Limit. This 16-bit register sets the maximum time between the leading edges of two consecutive characters. The value is ETUs.
15:8	/	/	/
7: 0	R/W	0x0	GUARDTIME Character Guard time. This 8-bit register sets a delay at the end of each character transmitted from the Smart Card Reader to the Smart Card. The value is in ETUs. The parity error is besides signaled during the guard time.

9.13.6.10 0x0030 Smart Card Reader Line Control Register (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: SCR_PAD
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	DSCVPPPP Direct Smart Card Vpp Pause/Prog. It provides direct access to SCVPPPP output.
6	R/W	0x0	DSCVPPEN Direct Smart Card Vpp Enable. It provides direct access to SCVPPEN output.
5	R/W	0x0	AUTOADEAVPP Automatic Vpp Handling. When high, it enables automatic handling of DSVPPEN and DSVPPPP signals during activation and deactivation sequence.
4	R/W	0x0	DSCVCC Direct Smart Card VCC. When DIRACCPADS='1', the DSCVCC bit provides direct access to SCVCC pad.
3	R/W	0x0	DSCRST Direct Smart Card Clock. When DIRACCPADS='1', the DSCRST bit provides direct access to SCRST pad.
2	R/W	0x0	DSCCLK Direct Smart Card Clock. When DIRACCPADS='1', the DSCCLK bit provides direct access to SCCLK pad.
1	R/W	0x0	DSCIO Direct Smart Card Input/Output. When DIRACCPADS='1', the DSCIO bit provides direct access to SCIO pad.
0	R/W	0x0	DIRACCPADS Direct Access to Smart Card Pads. When high, it disables a serial interface functionality and enables direct control of the smart card pads using following 4 bits.

9.13.6.11 0x003C Smart Card Reader FSM Register (Default Value: 0x0000_0000)

Offset: 0x003C			Register Name: SCR_FSM
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R	0x0	ATR_Structure_FSM
23:16	R	0x0	ATR_FSM
15:8	R	0x0	ACT_FSM
7: 0	R	0x0	SCR_FSM

9.13.6.12 0x0040 Smart Card Reader Debounce Time Register (Default Value: 0x0000_03ff)

Offset: 0x0040			Register Name: SCR_DT
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x3ff	SCR_Debounce_Time Set the debounce time value. The time unit is the cycle of SCCLK.

9.13.6.13 0x0100 Smart Card Reader FIFO Data Register (Default Value: 0x0000_0000)

Offset: 0x0100			Register Name: SCR_FIFO
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7: 0	R/W	0x0	FIFO_DATA This 8-bit register provides access to the RX and TX FIFO buffers. The TX FIFO is accessed during the APB write transfer. The RX FIFO is accessed during the APB read transfer.

Contents

Figures

Tables

10 Security Subsystem

10.1 Crypto Engine

10.1.1 Overview

The Crypto Engine (CE) module is one encryption/decryption algorithms accelerator. It supports kinds of symmetric, asymmetric, HASH, and RNG algorithms. There are two software interfaces for secure and non-secure world. The software interface is simple for configuration, only by setting interrupt control, task description address and load tag. Algorithm control information is written in memory in task descriptor, then CE automatically reads it when executing request. It supports parallel requests from 4 channels each world, and has an internal DMA controller to transfer data between CE and memory.

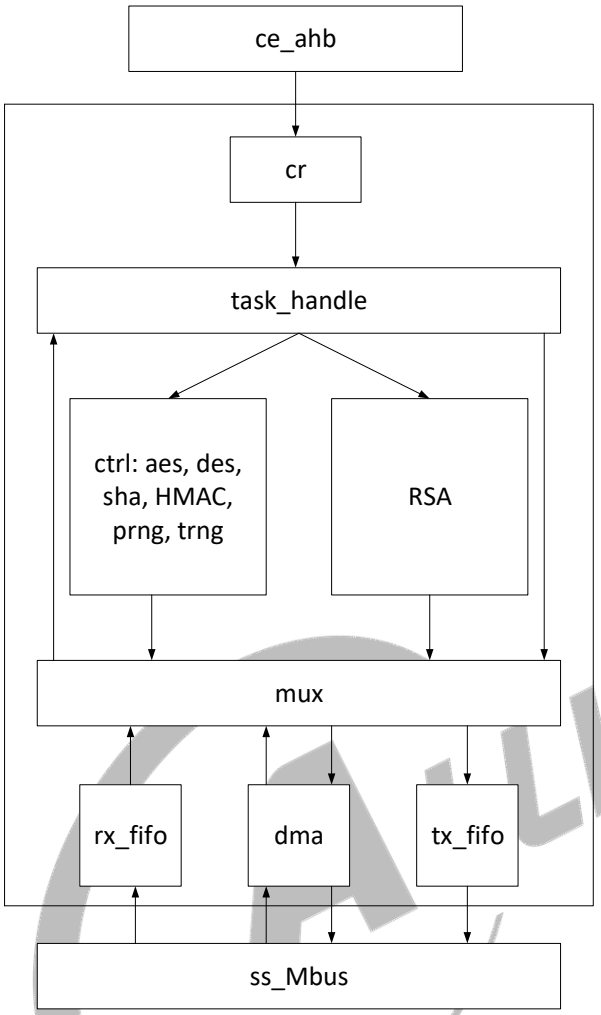
The CE includes the following features:

- Supports Symmetrical Algorithm: AES, DES, 3DES
  - Supports 128-bit, 192-bit and 256-bit key size for AES
  - Supports ECB, CBC, CTR, CTS, OFB, CFB modes for AES
  - Supports 1-bit, 8-bit, 64-bit, 128-bit width for AES-CFB
  - Supports 16-bit, 32-bit, 64-bit, 128-bit wide size for AES CTR
  - Supports ECB, CBC, CTR, CBC_MAC modes for DES/3DES
  - Supports 16-bit, 32-bit, 64-bit wide size for DES/3DES CTR
- Supports Hash Algorithms: MD5, SHA1, SHA224, SHA256, SHA384, SHA512, HMAC
  - Supports multi-package mode for MD5, SHA1, SHA224, SHA256, SHA384, SHA512
- Supports Asymmetrical Algorithm: RSA512/1024/2048bit
- Supports internal DMA Controller for data transferring with memory
- Supports secure and non-secure interfaces respectively
- Supports accessing secure and non-secure inferfaces by non-secure host when secure_mode is 0

10.1.2 Block Diagram

The following figure shows the block diagram of CE.

Figure 10-1 CE Block Diagram

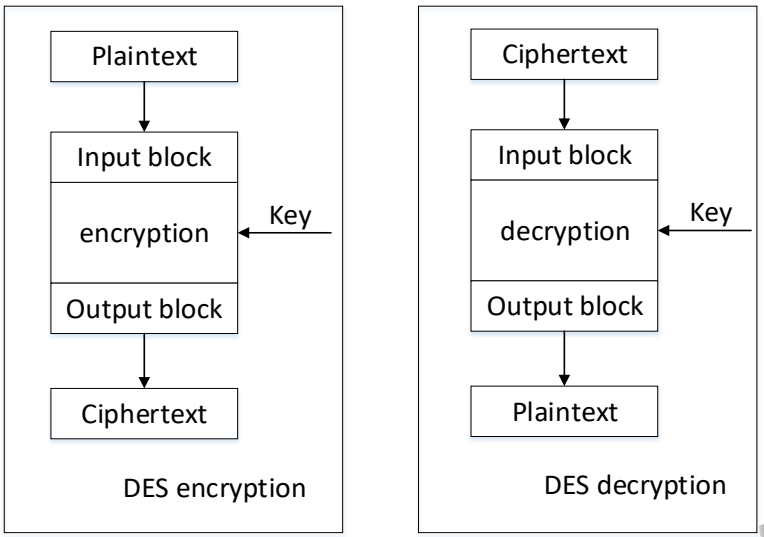


10.1.3 Functional Description

10.1.3.1 DES Algorithm

The following figure shows the DES encryption and decryption operation.

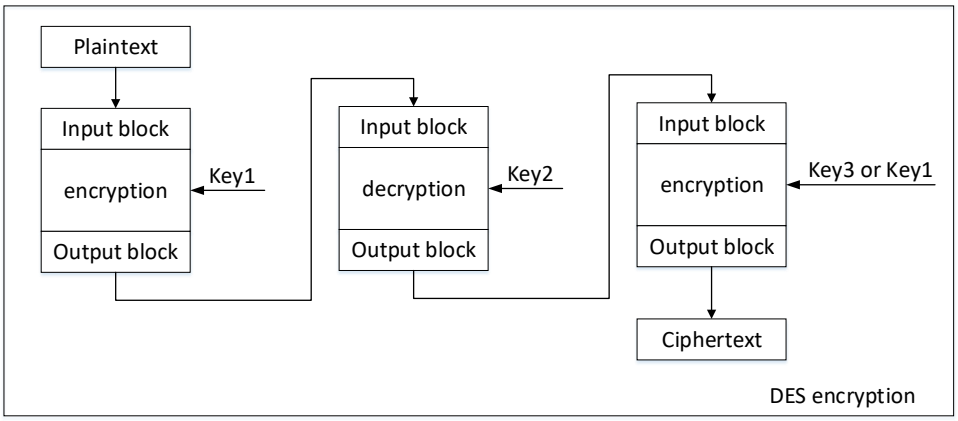
Figure 10-2 DES Encryption and Decryption

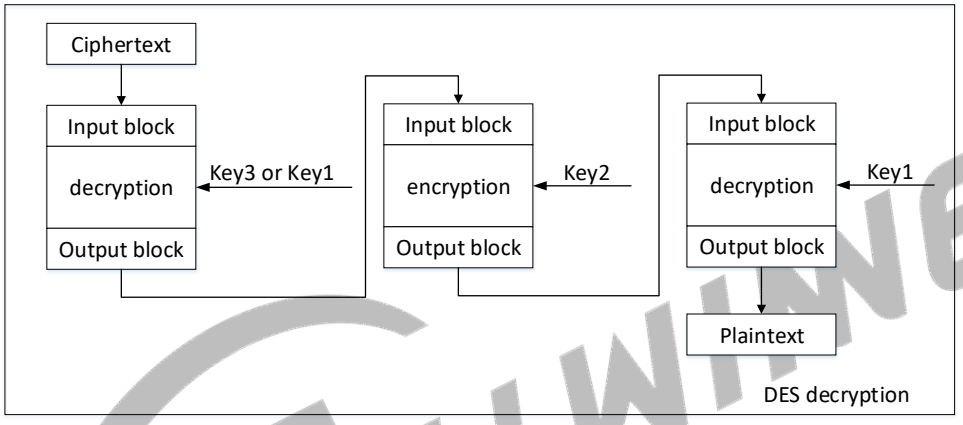


10.1.3.2 3DES Algorithm

The 3DES algorithm supports both 3-key and 2-key operations. A 2-key operation can be regarded as a simplified 3-key operation. To be specific, key 3 is represented by key 1 in a 2-key operation. The following figure shows the 3DES encryption and decryption operation of a 3-key operation and a 2-key operation.

Figure 10-3 3DES Encryption and Decryption of a 3-key Operation and a 2-key Operation



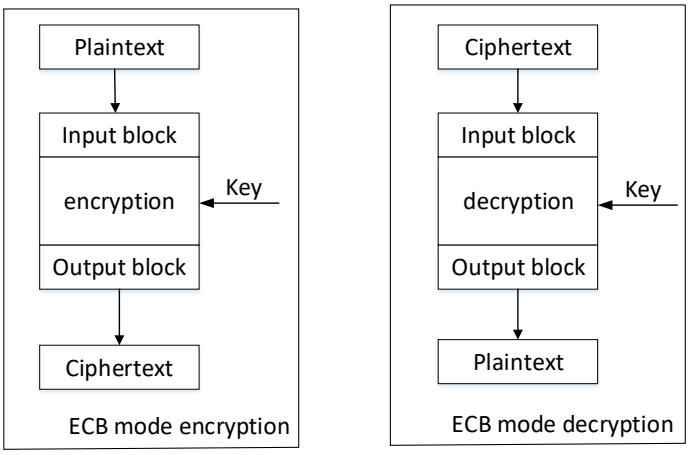


10.1.3.3 ECB Mode

The ECB mode is a confidentiality mode that features, for a given key, the assignment of a fixed ciphertext block to each plaintext block, analogous to the assignment of code words in a codebook.

In ECB mode, encryption and decryption algorithms are directly applied to the block data. The operation of each block is independent, so the plaintext encryption and ciphertext decryption can be performed concurrently.

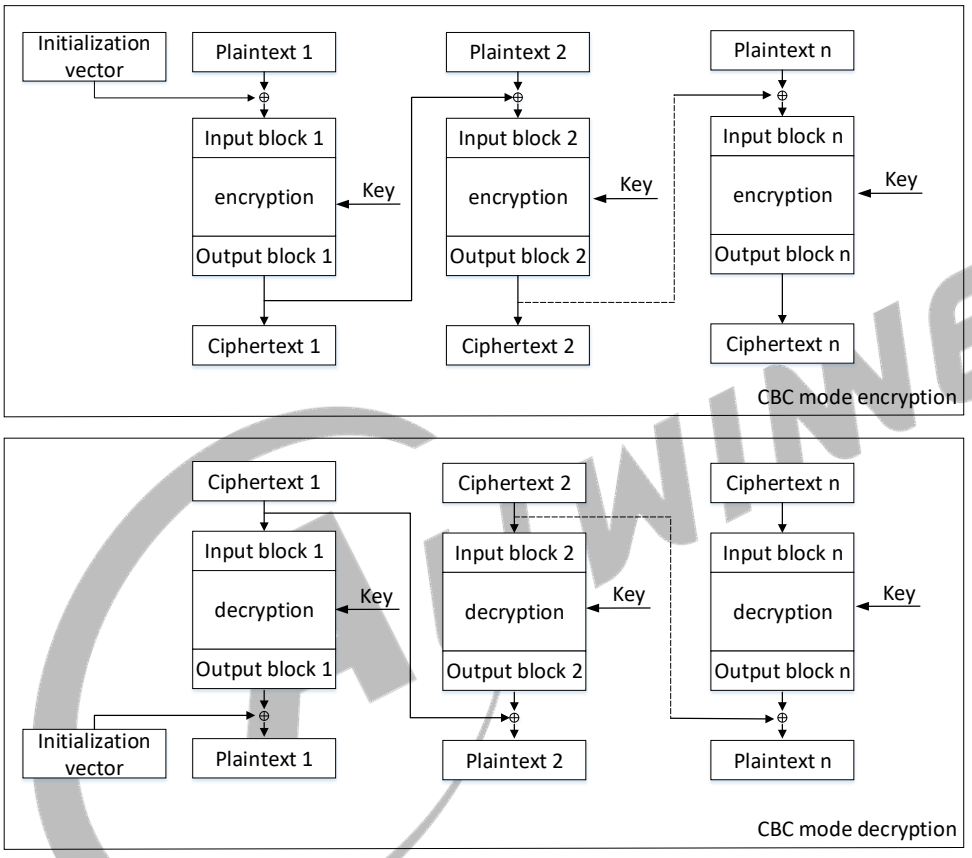
Figure 10-4 ECB Mode Encryption and Decryption



10.1.3.4 CBC Mode

The CBC mode is a confidentiality mode whose encryption process features the combining of the plaintext blocks with the previous ciphertext blocks. The CBC mode requires an initialization vector (IV) to combine with the first plaintext block. The encryption process of each plaintext block is related to the block processing result of the previous ciphertext blocks, so encryption operations cannot be concurrently performed in CBC mode. The decryption operation is independent of output plain text of the previous block, so decryption operations can be performed concurrently.

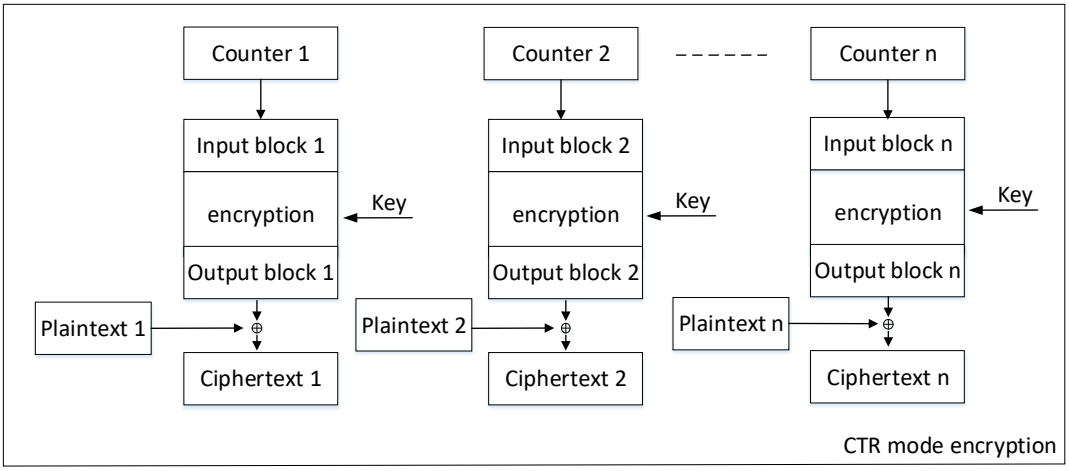
Figure 10-5 CBC Mode Encryption and Decryption

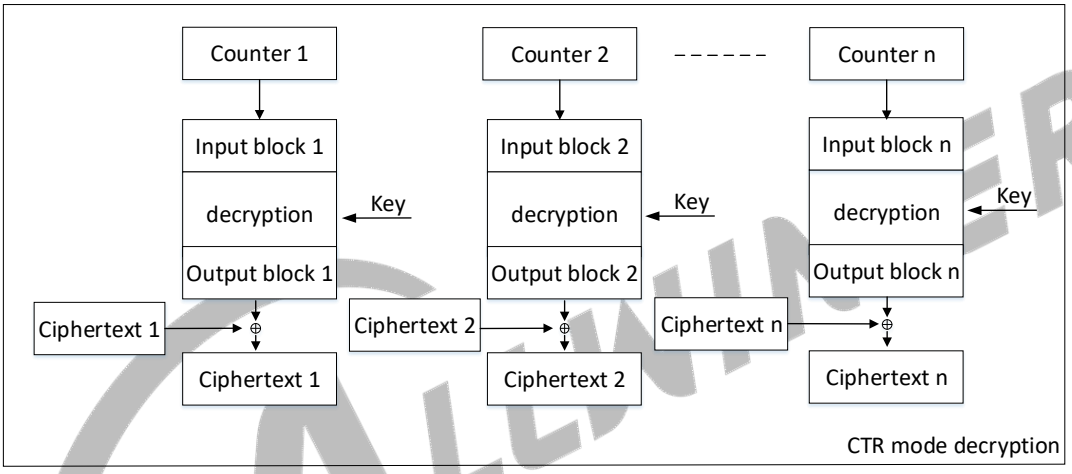


10.1.3.5 CTR Mode

The CTR mode is a confidentiality mode that features the application of the forward cipher to a set of input blocks, called counters, to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. All of the counters must be distinct.

Figure 10-6 CTR Mode Encryption and Decryption

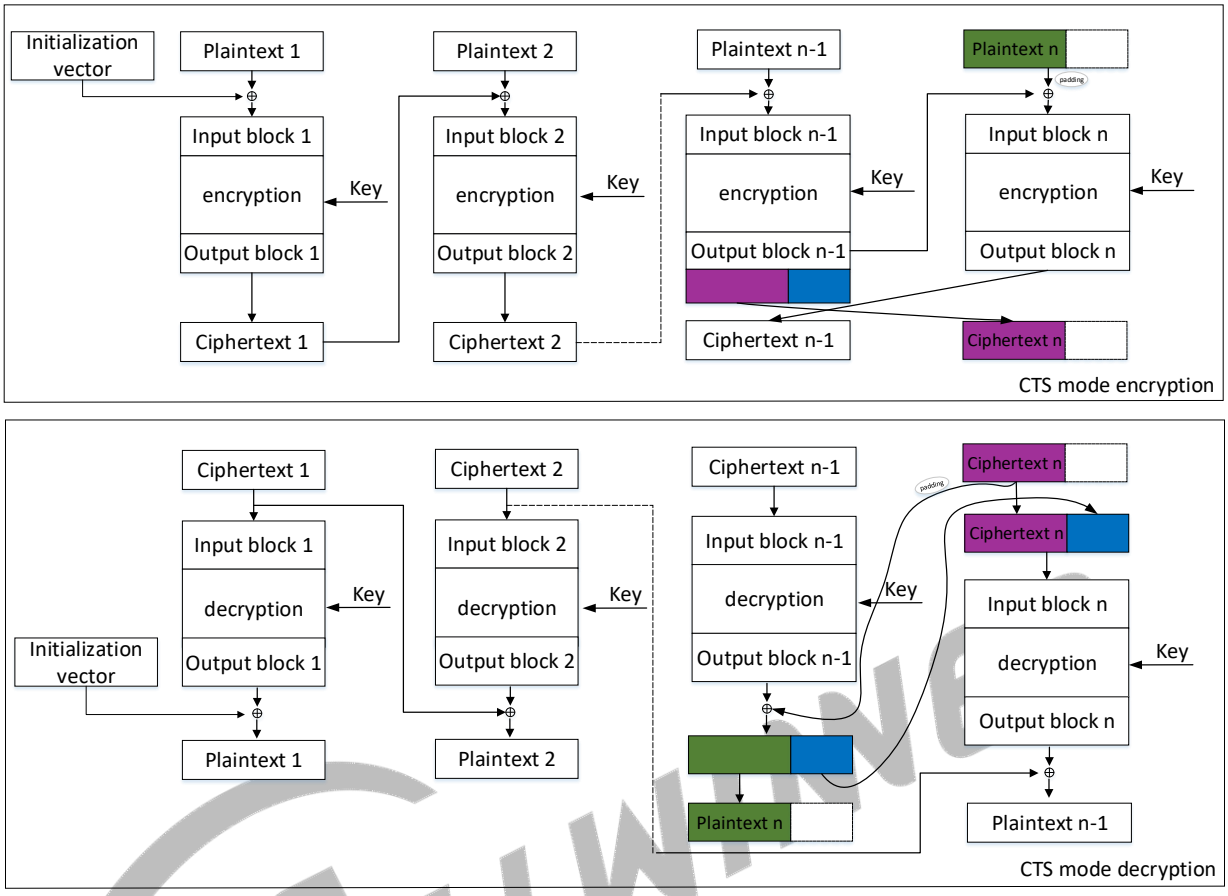




10.1.3.6 CTS Mode

The CTS mode is a confidentiality mode that accepts any plaintext input whose bit length is greater than or equal to the block size but not necessarily a multiple of the block size. Below are the diagrams for CTS encryption and decryption.

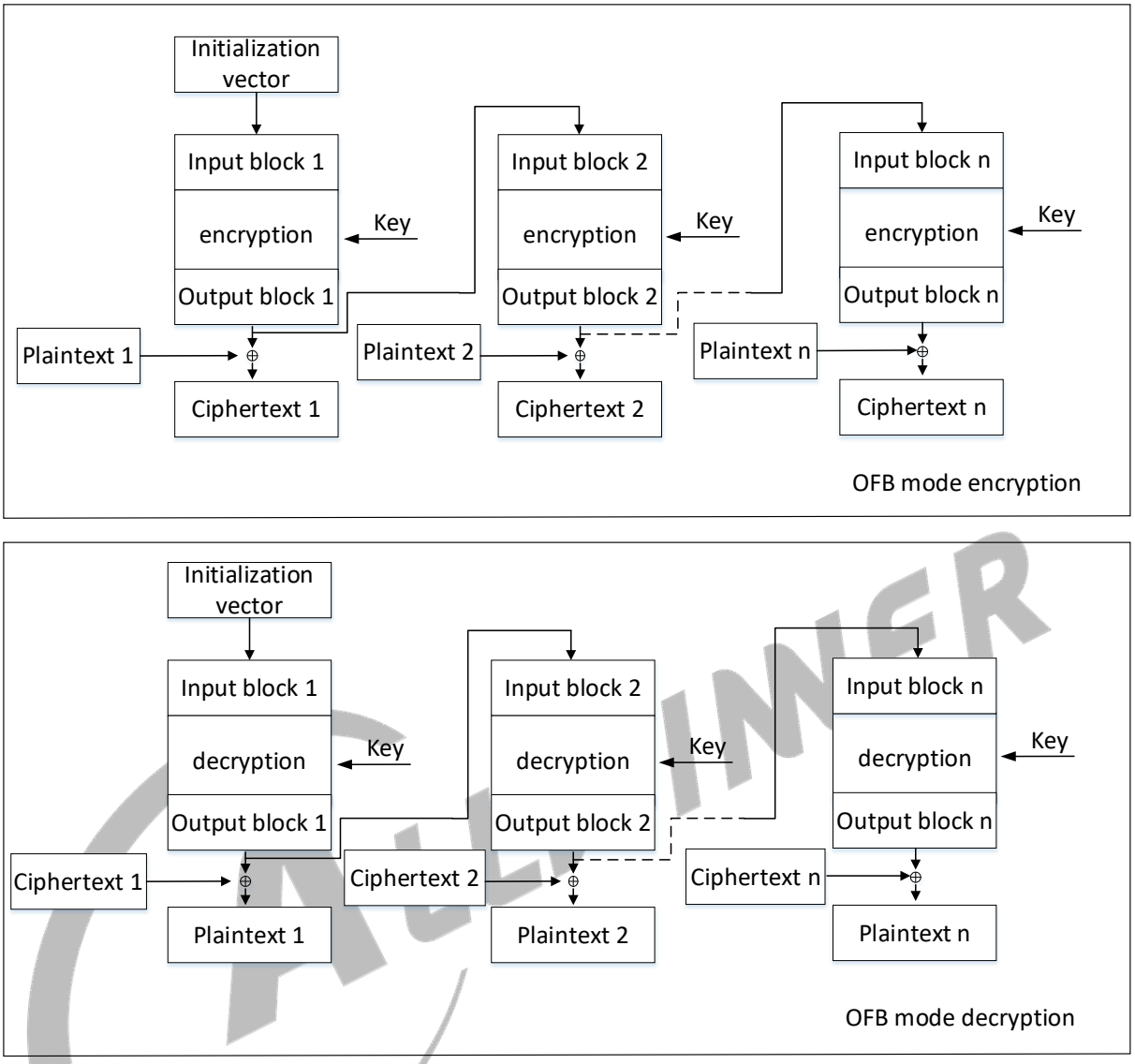
Figure 10-7 CTS Mode Encryption and Decryption



10.1.3.7 OFB Mode

The OFB mode is a confidentiality mode that features the iteration of the forward cipher on an IV to generate a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. If a same key is used, different IVs must be used to ensure operation security.

Figure 10-8 OFB Mode Encryption and Decryption

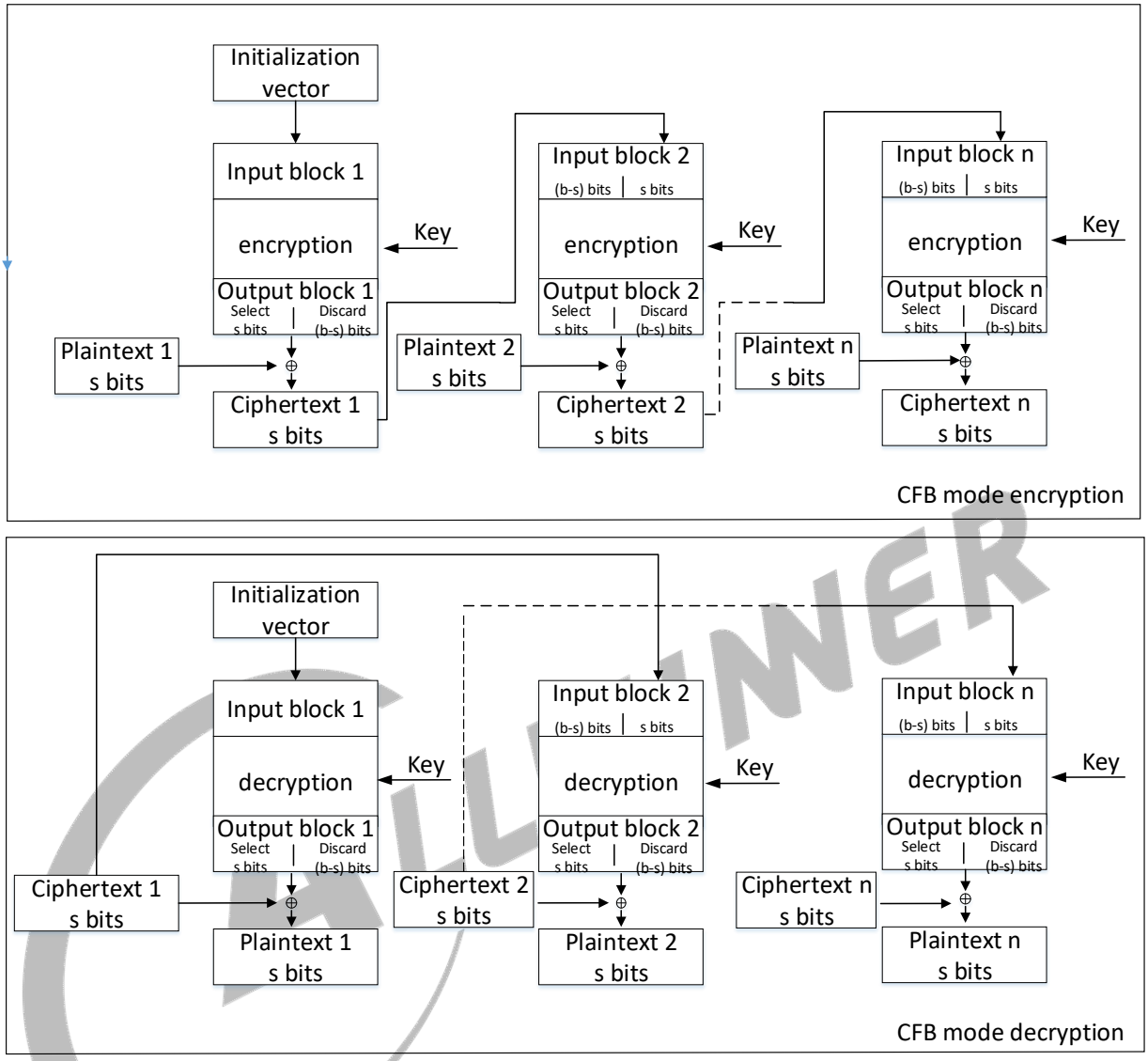


10.1.3.8 CFB Mode

The CFB mode is a confidentiality mode that features the feedback of successive ciphertext segments into the input blocks of the forward cipher to generate output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. The CFB mode requires an IV as the initial input block, and the forward cipher operation is applied to the IV to produce the first output block. The first ciphertext segment is produced by exclusive-ORing the first plaintext segment with the s most significant bits of the first output block. The value of s is 1 bit, 8 bits, 64 bits, or 128 bits.

The following figure shows the s-bit CFB mode of the AES algorithms.

Figure 10-9 CFB Mode Encryption and Decryption



10.1.3.9 HASH Algorithm

The hash algorithms support MD5, SHA1, SHA224, SHA256, SHA384, SHA512, HMAC-SHA1, and HMAC-SHA256. All algorithms are iterative, one-way hash functions that can process a message to produce a condensed representation called a message digest. When a message is received, the message digest can be used to verify whether the data has changed, that is, to verify its integrity.

The hash algorithm of R128 supports block-aligned total length of the input data (padded by software), that is, a multiple of 64 bytes. The message length after padding by software is used as the configured data length for the hash algorithm.

10.1.3.10 RSA Algorithm

The RSA is a public key encryption/decryption algorithm implemented through the modular exponentiation operation.

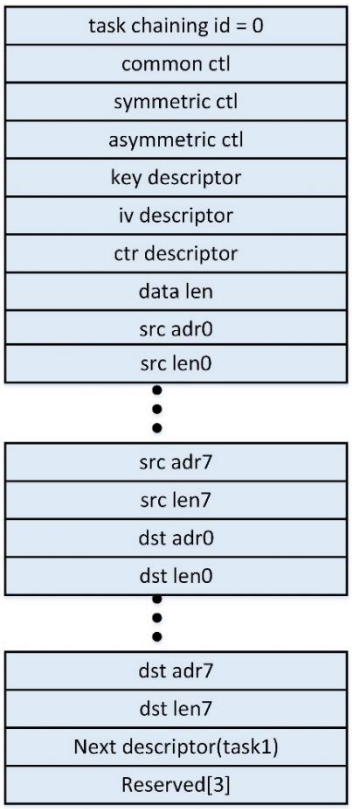
The ciphertext is obtained as follows: C = ME mod N. The plaintext is obtained as follows: M = CD mod N.

M indicates the plaintext, C indicates the ciphertext, (N, E) indicates the public key, and (N, D) indicates the private key.

10.1.3.11 Task Descriptor

The software makes request through task descriptor, including algorithm type, algorithm mode, key address, source/destination address and size, and so on. The structure of the task descriptor is as follows.

Figure 10-10 Structure of Task Descriptor Chaining



The bit definitions of the task descriptor are as follows.

Table 10-1 Task ID

Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3: 0	R/W	0x0	CHN Task channel ID Indicates which channel the task is running on. It supports 0 to 3.

Table 10-2 Common Control

Bit	Read/Write	Default/Hex	Description

Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	Interrupt enable (IE) for the current task 0: Disable interrupt 1: Enable interrupt Represents whether an interrupt signal is generated when the task chain ends at the end of this task. When the last task in a task chain ends, the operation of the task chain will end normally; if a task fails in the middle, the task chain will be aborted abnormally. And it is determined whether to generate an interrupt signal according to the IE configuration of the current task when the current task ends or aborts. Therefore, if you want to use interrupts, it is recommended that not only the IE of the last task of each task chain is configured to 1 to generate the end interrupt of the task chain, but also the IEs of other tasks in this task chain are also configured to 1. The purpose is to generate an interrupt signal once an abnormal error occurs in these tasks and the interrupt is aborted.
30:17	/	/	/
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 1: The last HMAC plaintext package
14:9	/	/	/
8	R/W	0x0	OP DIR Algorithm Operation Direction 0: Encryption 1: Decryption Configure according to the requirements of encryption or decryption.
7	/	/	/

Bit	Read/Write	Default/Hex	Description
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 0x20: RSA Others: Reserved

Table 10-3 Symmetric Control

Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
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} 0100-0111: Reserved 1000-1111: Select internal Key n (n from 0 to 7)
19:18	R/W	0x0	cfb_width AES-CFB width 00: CFB1 01: CFB8 10: CFB64 11: CFB128
17	/	/	/
16	R/W	0x0	CTS_LPKG AES CTS last package flag When set to '1', it means this is the last package for AES-CTS mode (the size of the last package is larger than 128 bits).
15:12	/	/	/

Bit	Read/Write	Default/Hex	Description
11:8	R/W	0x0	ALGORITHM_MODE CE algorithm mode 0000: Electronic Code Book (ECB) mode 0001: Cipher Block Chaining (CBC) mode 0010: Counter (CTR) mode 0011: Cipher Text Stealing (CTS) mode 0100: Output Feedback (OFB) mode 0101: Cipher Feedback (CFB) 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

Table 10-4 Asymmetric Control

Bit	Read/Write	Default/Hex	Description
31	/	/	/
30:28	R/W	0x0	RSA Pubic Modulus Width 000: 512-bit 001: 1024-bit 010: 2048-bit Other: Reserved
27:19	/	/	/
18:16	R/W	0x0	RSA MODE RSA algorithm mode. For modular computation: 000: modular exponent(RSA) 001: modular div 010: modular mul 011: modular inv others: Reserved
15: 0	/	/	/

Table 10-5 Key Descriptor

Bit	Read/Write	Default/Hex	Description

Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	Key Address The address of KEY that needs to be stored.

Table 10-6 IV Descriptor

Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	IV Address The address of IV that needs to be stored.

Table 10-7 Counter Descriptor

Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	CTR Data Output Address The address of CTR data output that needs to be stored.

Table 10-8 Data Length

Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	Data Length Configure the data length of the corresponding segment. The data length size needs to be consistent with dst_data_length (destination data length 0 +... + destination data length 7). The data length field in the task descriptor has different meanings for different algorithms. For AES-CTS, the data length field indicates byte numbers of source data, for others indicate word numbers of source data.

Table 10-9 Source Address 0~7

Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	Source Data Address The address of the source data that needs to be stored.

Table 10-10 Source Data Length 0~7

Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	Source Data Length The length of the source data. Unit: byte

Table 10-11 Destination Address 0~7

Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	Destination Data Address The address of the destination data that needs to be stored.

Table 10-12 Destination Data Length 0~7

Bit	Read/Write	Default/Hex	Description

Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	Destination Data Length The length of the destination data. Unit: byte

Table 10-13 Next Descriptor Address

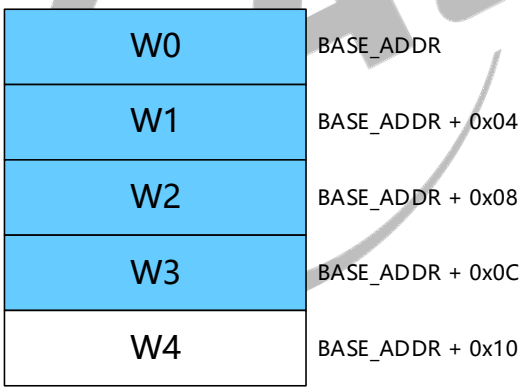
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	Next Task Address The address where the descriptor of the next task in a task-chain is saved. If there is the only task or the last task of a task-chain, the next task address must be 32'h0.

10.1.3.12 Storing Message

In the application, a message may not be stored contiguously in the memory, but divided into multiple segments. Or a piece of continuously stored messages can be artificially split into multiple pieces as needs. Then each segment corresponds to a set of the source address and source length in the descriptor. Multiple segments correspond to groups 0-7 source address/source length in sequence.

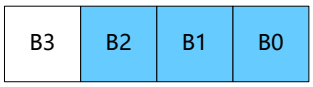
Each task supports up to 8 message segments, and the data volume of each message segment supports up to 4 GWord (AES-CTS is 1 GByte). The total amount of all segments in a task (that is a package) supports up to 4 GWord (AES-CTS is 1 GByte). If a message is divided into multiple packages, all others are required to be whole words; when the last package of AES-CTS is less than one word, 0 needs to be padded, and those less than one word are counted as one word. The following figure shows the address order structure.

Figure 10-11 Word Address of Message



Byte order: Low byte first, high byte last. When the data is less than one word, the low byte is filled first. The following figure shows the byte order structure (blue means it is filled by the message).

Figure 10-12 Byte Order



Bit order: high bit first, low bit last. When the data is less than one Byte, the high bit is filled first. The following figure shows the bit order structure.

Figure 10-13 Bit Order

10.1.3.13 Storing Key

The length of KEY must be an integer multiple of word, refer to the section "Algorithm Length Properties".

10.1.3.14 Storing IV

For different algorithms, the length of IV is different. But they are integer multiples of word. To keep the byte order of IV and HASH digest output consistent, the byte order of IV is different from that of the message. For the multi-packet operation, the first address of the digest output result of the previous HASH can be directly configured to the first address of the next IV, and the software does not need to do any processing on the digest.

The following figure shows the storage method of 32-bit IV value.

Figure 10-14 The Storage Method of 32-bit IV

IV0[31:0]	BASE_ADDR
IV1[31:0]	BASE_ADDR + 0x04
……	……
IV7[31:0]	BASE_ADDR + 0x1C

The following figure shows the storage method of 64-bit IV value.

Figure 10-15 The Storage Method of 64-bit IV

IV0[63:32]	BASE_ADDR
IV0[31:00]	BASE_ADDR + 0x04
IV1[63:32]	BASE_ADDR + 0x08
IV1[31:00]	BASE_ADDR + 0x0C
……	……
IV7[63:32]	BASE_ADDR
IV7[31:00]	BASE_ADDR + 0x3C

10.1.3.15 Algorithm Length Properties

The algorithm length has different requirements for different algorithms.

Table 10-14 Symmetric Algorithm Configuration Properties

Algorithm	Length Setting				Alignment	Software Padding
	Source Size	Destination Size	KEY	IV		
AES (except CTS)	< 4 GWord	< 4 GWord	AES-128: 4 Word AES-192: 6 Word AES 256: 8 Word	4 Word	Word-aligned	need
AES-CTS	< 1 GByte	< 1 GByte	AES-128: 4 word AES-192: 6 word AES 256: 8 word	4 Word	Word-aligned	need
DES	< 4 GWord	< 4 GWord	2 Word	2 Word	Word-aligned	need
TDES	< 4 GWord	< 4 GWord	6 Word	2 Word	Word-aligned	need

Table 10-15 Hash Algorithm Configuration Properties

Algorithm	Length Setting				Alignment	Software Padding
	Source Size	Destination Size	KEY	IV		
MD5	< 4 GWord	4 Word	Fixed to 0	4 Word	Word-aligned	need
SHA-1	< 4 GWord	5 Word	Fixed to 0	5 Word	Word-aligned	need
SHA-224	< 4 GWord	8 Word	Fixed to 0	8 Word	Word-aligned	need
SHA-256	< 4 GWord	8 Word	Fixed to 0	8 Word	Word-aligned	need
SHA-384	< 4 GWord	16 Word	Fixed to 0	16 Word	Word-aligned	need
SHA-512	< 4 GWord	16 Word	Fixed to 0	16 Word	Word-aligned	need
HMAC-SHA1	< 4 GWord	5 Word	16 Word	5 Word	Word-aligned	need
HMAC-SHA256	< 4 GWord	8 Word	16 Word	8 Word	Word-aligned	need

Table 10-16 Asymmetric Algorithm Configuration Properties

Algorithm	Length Setting				Alignment	Software Padding
	Source Size	Destination Size	KEY	IV		
RSA512	16 Word	16 Word	16 Word	Not use IV	Word-aligned	need
RSA1024	32 Word	32 Word	32 Word	Not use IV	Word-aligned	need
RSA2048	64 Word	64 Word	64 Word	Not use IV	Word-aligned	need

10.1.3.16 Security Operation

When the CPU issues request to the CE module, the CE module will save the secure mode of CPU. When executing this request, this state bit works as a access flag for the inner and system resources. For access to

SID module through the AHB bus, only the secure mode can succeed, or else these keys will be read to 0 or cannot write. When issuing MBUS read and write requests, the CE will use send this secure mode bit to BUS, so secure requests can access secure and non-secure space, but non-secure requests only can access non-secure space.

10.1.3.17 Error Detection

The CE module includes error detection for task configuration, data computing error, and authentication invalid. When the algorithm type in task descriptor is read into the CE module, the CE will check whether this type is supported through checking algorithm type field in common control. If the type value is out of scope, the CE will issue interrupt signal and set error state. Each type has certain input and output data size. After getting a task descriptor, the input size and output size configuration will be checked to avoid size error. If the size configuration is wrong, the CE will issue interrupt signal and set error state.

10.1.3.18 Clock Frequency Requirement

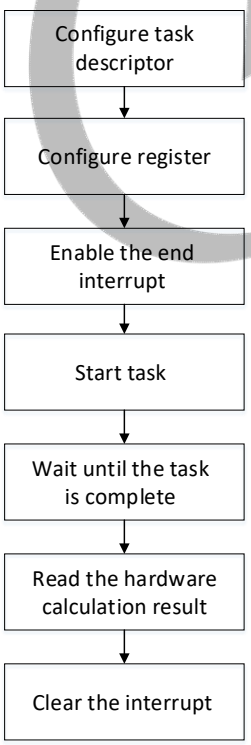
Clock Name	Description	Requirement
hfclk	AHB bus clock	24 MHz – 200 MHz
mclk	MBUS clock	24 MHz – 400 MHz
ce_clk	CE work clock	24 MHz – 200 MHz

10.1.4 Programming Guidelines

10.1.4.1 Symmetrical/Asymmetrical/Hash/RNG Algorithm Operation Process

The following figure shows the process of an algorithm operation.

Figure 10-16 Task Request Process



Step 1 The software should configure a task descriptor in memory, including the related fields in the descriptor. The channel id corresponds to one channel in CE. According to algorithm type, the 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. The source and destination address and size are set based on the upper application. If there is another task concatenating after this task, then set its descriptor address at the next descriptor field. For more details for task descriptor, see section 10.1.4.2, section 10.1.4.3 and section 10.1.4.4.

Step 2 The software should set registers. Configure the first address of the task descriptor structure to CE Task Descriptor Address Register. Configure the source/destination address to CE Current Source Address Register/CE Current Destination Address Register.

Step 3 Enable the end interrupt of the corresponding task channel by setting CE Interrupt Control Register.

Step 4 The software Read CE Task Load Register to ensure that the bit0 is 0. If the bit0 is not read out to be 0, wait until it is 0, then configure the bit0 to be 1 to start task.

Step 5 Wait for interrupt status by reading CE Interrupt Status Register.

Step 6 Read the result from the destination address.

Step 7 Clear the interrupt.

10.1.4.2 Configuring Task Descriptor of AES

- **Common control**: Configure Common Control[6: 0] to 0x0 to select AES algorithm type.
- **Symmetric control**: According to the corresponding algorithm requirements, configure Symmetric Control to select the key size, CTR width, CTS last package flag, CFB width, and AES algorithm mode, and so on.
- **Asymmetric control**: The symmetric algorithm does not need to be configured for this field.
- **Key descriptor**: Because the storage of the key requires word alignment, ensure that this descriptor is the first address of the KEY (word address).
- **IV descriptor**: In the task that requires the IV value, configure the first address of the storage space where the IV is stored here. Because the storage of the IV requires word alignment, ensure that this descriptor is the first address of the IV (word address).
- **Data length**: Configure the data length of the corresponding segment. The data length size needs to be consistent with dst_data_length (destination data length 0 +... + destination data length 7). When the algorithm is CTS mode, the higher 30-bit of the data length is the word numbers of data volume; when the data_length[1: 0] is 0, the data length is the higher 30-bit, otherwise it is increased by 1. For AES CTS, the data length indicates the byte numbers of the source data; for other algorithms, it indicates the word numbers.
- **Source address**: The first address of source data segments. Because the storage of the source data requires word alignment, ensure that this descriptor is the first address (word address).

- **Source data length**: The data volume of source data segments. The unit is word, and those less than one word are counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.
- **Destination address**: The first address of destination data segments. Because the storage of the destination data requires word alignment, ensure that this descriptor is the first address (word address).
- **Destination data length**: The data volume of destination data segments. The unit is word, and those less than one word are counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.
- **Next descriptor**: The first address of the next task descriptor. Because the storage of the descriptor requires word alignment, ensure that this descriptor is the first address (word address).
- **Reserved**: Configure to 0x0.

10.1.4.3 Configuring Task Descriptor of HASH

Common Control

- Algorithm type: Configure Common Control[6: 0] to select SHA or HMAC algorithm type.
- Last HMAC plaintext: If the algorithm type is HMAC, and the task is the last package of the message or if the message has only one package, then Common Control[15] needs to set to 1.
- IV mode: The Common Control[16] (IV MODE) bit is only set to 1 in the following two scenarios, except that the bit must be configured to 0. (1). When the message is split into multiple packages, the Common Control[16] bit of other packages needs to be set to 1, except that the bit of the first package needs to be cleared to 0. (2). In special applications, if you need to customize the IV value to form the initial value of a certain HASH algorithm, you need to set the Common Control[16] bit of the first (or only one) package to 1, and the first address of the storage space where the customized IV value is stored in IV address.

Key Descriptor: Because the storage of the key requires word alignment, ensure that this descriptor is the first address of the KEY (word address).

IV Descriptor: In the task that requires the IV value, configure the first address of the storage space where the IV is stored here. Because the storage of the IV requires word alignment, ensure that this descriptor is the first address of the IV (word address).

Data Length: Configure the data length of the corresponding segment. The data length size needs to be consistent with dst_data_length (destination data length 0 +... + destination data length 7).

Source Address: The first address of source data segments. Because the storage of the source data requires word alignment, ensure that this descriptor is the first address (word address).

Source Data Length: The data volume of source data segments. The unit is word, and those less than one word is counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.

Destination Address: The first address of destination data segments. Because the storage of the destination data requires word alignment, ensure that this descriptor is the first address (word address).

Destination Data Length: The data volume of destination data segments. The unit is word, and those less than one word is counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.

Next Descriptor: The first address of the next task descriptor. Because the storage of the descriptor requires word alignment, ensure that this descriptor is the first address (word address).

Reserved: Configure to 0x0.

10.1.4.4 Configuring Task Descriptor of RSA

Common Control: Configure Common Control[6: 0] to 0x20 to select RSA algorithm type.

Asymmetric Control: Configure Asymmetric Control[30:28] to select the RSA width.

Key Descriptor: Because the storage of the key requires word alignment, ensure that this descriptor is the first address of the KEY (word address).

Data Length: Configure the data length of the corresponding segment. The data length size needs to be consistent with dst_data_length (destination data length 0 +... + destination data length 7).

Source Address: The first address of source data segments. Because the storage of the source data requires word alignment, ensure that this descriptor is the first address (word address).

Source Data Length: The data volume of source data segments. The unit is word, and those less than one word is counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.

Destination Address: The first address of destination data segments. Because the storage of the destination data requires word alignment, ensure that this descriptor is the first address (word address).

Destination Data Length: The data volume of destination data segments. The unit is word, and those less than one word is counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.

Next Descriptor: The first address of the next task descriptor. Because the storage of the descriptor requires word alignment, ensure that this descriptor is the first address (word address).

Reserved: Configure to 0x0.

10.1.5 Register List

Module Name	Base Address
CE_NS	0x40004000
CE_S	0x40004800

Register Name	Offset	Description
CE_TDA	0x0000	CE Task Descriptor Address Register
CE_CTL	0x0004	CE Control Register
CE_ICR	0x0008	CE Interrupt Control Register

Register Name	Offset	Description
CE_ISR	0x000C	CE Interrupt Status Register
CE_TLR	0x0010	CE Task Load Register
CE_TSR	0x0014	CE Task Status Register
CE_ESR	0x0018	CE Error Status Register
CE_CSA	0x0024	CE Current Source Address Register
CE_CDA	0x0028	CE Current Destination Address Register
CE_TPR	0x002C	CE Throughput Register

10.1.6 Register Description

10.1.6.1 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_DES_ADDR Task Descriptor Address

10.1.6.2 0x0004 CE Control Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: CE_CTL
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3	R/W	0x0	RSA_CLK_GATE_EN RSA CLK Gating Enable (only for S world) 0: RSA clk gating enable 1: RSA clk gating disable
2: 0	/	/	/

10.1.6.3 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_CHAN_INT_EN Task channel0-3 interrupt enable 0: Interrupt disable 1: Interrupt enable

10.1.6.4 0x000C CE Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x000C			Register Name: CE_ISR
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/

Offset: 0x000C			Register Name: CE_ISR
Bit	**Read/Write**	**Default/Hex**	**Description**
3: 0	R/W	0x0	TC_EN_PENDING Task channel0-3 end pending 0: Not finished 1: Finished It indicates if task has been completed. **Note**: Write *'1' to clear it.*

10.1.6.5 0x0010 CE Task Load Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: CE_TLR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	TASK_LOAD When set, CE can load the descriptor of task if task FIFO is not full.

10.1.6.6 0x0014 CE Task Status Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: CE_TSR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1: 0	R	0x0	TASK_CHAN_STATE indicate which channel in run 0: Task channel0 1: Task channel1 2: Task channel2 3: Task channel3

10.1.6.7 0x0018 CE Error Status Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: CE_ESR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:12	R	0x0	CHAN3_ERR_STATE Task channel3 error type xxx1: Algorithm not support xx1x: Data length error x1xx: Keysram access error for AES. Write '1' to clear. 1xxx: Reserved

Offset: 0x0018			Register Name: CE_ESR
Bit	**Read/Write**	**Default/Hex**	**Description**
11:8	R	0x0	CHAN2_ERR_STATE Task channel2 error type xxx1: Algorithm not support xx1x: Data length error x1xx: Keysram access error for AES. Write '1' to clear. 1xxx: Reserved
7:4	R	0x0	CHAN1_ERR_STATE Task channel1 error type xxx1: Algorithm not support xx1x: Data length error x1xx: Keysram access error for AES. Write '1' to clear. 1xxx: Reserved
3: 0	R	0x0	CHAN0_ERR_STATE Task channel0 error type xxx1: Algorithm not support xx1x: Data length error x1xx: Keysram access error for AES. Write '1' to clear. 1xxx: Reserved

10.1.6.8 0x0024 CE Current Source Address Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: CE_CSA
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	CUR_SRC_ADDR Current source address DMA Read now

10.1.6.9 0x0028 CE Current Destination Address Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: CE_CDA
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	CUR_DST_ADDR Current destination address DMA writes now

10.1.6.10 0x002C CE Throughput Register (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: CE_TPR
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R/W	0x0	TP_NUM It indicates the throughput since last write to this register. **Note**: Write *to this register will clear it to 0.*

10.2 TRNG

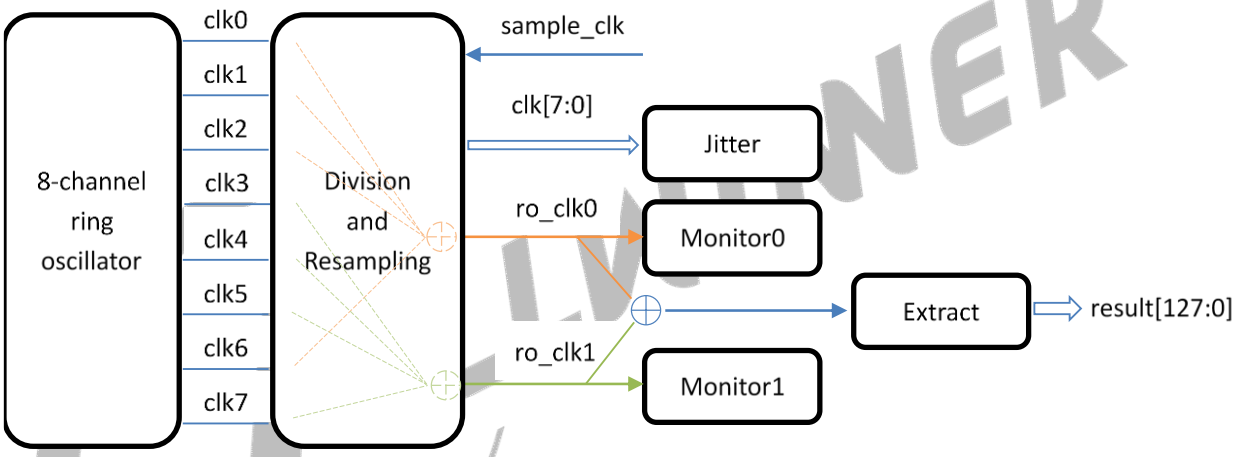
10.2.1 Overview

TRNG is the True Random Number Generator. The clock source of TRNG is the 8-channel independent ring oscillators. After the power-supply noise of analog devices generates frequency jitter, TRNG will resample the output clock of the ring oscillators through sample_clk and then perform elastic extraction and entropy extraction to generate 128-bit true random numbers.

10.2.2 Block Diagram

The follow figure shows the block diagram of TRNG.

Figure 10-17 TRNG Block Diagram



10.2.3 Functional Description

TRNG consists of jitter module, monitor module, and extract module.

The Jitter module is used for counting the number of rising edges in the 8-channel clocks sent from the ring oscillator within a period of time.

The monitor module is used for automatically detecting whether the clocks from analog meet the requirements. Monitor 0 and Monitor 1 respectively monitor one-channel clock, and eight-channel clocks are divided into two groups to be monitored. To pass the monitoring, at least two-channel clocks should start oscillation. This module has two indexes: RTC and ATP.

- RTC monitors the duration of consecutive 0 or 1. An error will be reported if the threshold (1032) is exceeded.
- ATP counts the number of cumulative 0 and 1 within the set data length. An error will be reported if the threshold (1009) is exceeded.

The extract module will perform elastic extraction and entropy extraction when the 8-channel clock is finally synthesized into one-channel random 0/1 sequence. A 128bit true random number will be output during each extraction.

10.2.4 Register List

Module Name	Base Address
TRNG	0x40048C00

Register Name	Offset Address	Description
TRNG_CTRL_CFG	0x0000	TRNG Control Register
TRNG_JITTER_CFG	0x0004	TRNG Jitter Control Register
TRNG_JITTER_CNT_TIMING	0x0008	TRNG Jitter Counter Timing Register
TRNG_MONITOR_RCT	0x000C	TRNG Monitor RTC Register
TRNG_MONITOR_APT	0x0010	TRNG Monitor APT Register
TRNG_EXTRACT_CFG	0x0014	TRNG Extract Control Register
TRNG_RAND_BIT_URN0	0x0018	TRNG Rand Data0 Register
TRNG_RAND_BIT_URN1	0x001C	TRNG Rand Data1 Register
TRNG_RAND_BIT_URN2	0x0020	TRNG Rand Data2 Register
TRNG_RAND_BIT_URN3	0x0024	TRNG Rand Data3 Register
TRNG_JITTER_CNT_RESULT0	0x0028	TRNG Jitter Counter Result0 Register
TRNG_JITTER_CNT_RESULT1	0x002C	TRNG Jitter Counter Result1 Register
TRNG_JITTER_CNT_RESULT2	0x0030	TRNG Jitter Counter Result2 Register
TRNG_JITTER_CNT_RESULT3	0x0034	TRNG Jitter Counter Result3 Register
TRNG_JITTER_CNT_RESULT4	0x0038	TRNG Jitter Counter Result4 Register
TRNG_JITTER_CNT_RESULT5	0x003C	TRNG Jitter Counter Result5 Register
TRNG_JITTER_CNT_RESULT6	0x0040	TRNG Jitter Counter Result6 Register
TRNG_JITTER_CNT_RESULT7	0x0044	TRNG Jitter Counter Result7 Register
TRNG_JITTER_COUNTER_READY	0x0048	TRNG Jitter Ready Register
TRNG_REG_ACCESS_CTR1	0x004C	TRNG Access Counter Low Register
TRNG_REG_ACCESS_CTR2	0x0050	TRNG Access Counter High Register

10.2.5 Register Description

10.2.5.1 0x0000 TRNG Control Register (Default Value: 0x0000_0FF0)

Offset: 0x0000			Register Name: TRNG_CTRL_CFG
Bit	Read/Write	Default/Hex	Description
31:15	/	/	/

Offset: 0x0000			Register Name: TRNG_CTRL_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
14:12	R	0x0	trng_ready Indicate the status of TRNG 4: Two groups of ring oscillators failed 3: The first group of ring oscillators failed 2: The second group of ring oscillators failed 1: Busy 0: IDLE
11:4	R/W	0xFF	trng_ro_ctrl The control bits of the open/closed states of 8-channel ring oscillators, which are valid in high level.
3	/	/	/
2:1	R/W	0x0	TURN_ro_detune Tune the oscillating frequency of ring oscillators. The larger value of TURN_RO_DETUNE contributes to the faster frequencies.
0	R/W	0x0	TRNG_EN 0: Disable 1: Enable

10.2.5.2 0x0004 TRNG Jitter Control Register (Default Value: 0x0000_0002)

Offset: 0x0004			Register Name: TRNG_JITTER_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x0	trng_jitter_counter_start Initiate this bit when the TRNG Jitter Counter is configured. It is valid in the rising edge.
1	R/W	0x1	trng_counter_2div The enable signal of 2DIV is optional, which is valid in the high level. (The level is high by default.)
0	R/W	0x0	trng_jitter_work_en Enable the measuring function of Jitter and control eight divider circuits. It is valid in high level.

10.2.5.3 0x0008 TRNG Jitter Counter Timing Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: TRNG_JITTER_CNT_TIMING
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	/	/	/
20: 0	R/W	0x0	trng_jitter_counter_timing Count the set time

10.2.5.4 0x000C TRNG Monitor RTC Register (Default Value: 0x0000_0408)

Offset: 0x000C			Register Name: TRNG_MONITOR_RCT
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11	R/W	0x0	trng_monitor_en Monitor the enable signal. It is valid in high level. (The level is low by default.)
10: 0	R/W	0x0408	trng_RCT_C The monitoring threshold of RCT (1032 by default). It continues monitoring the length of 0 or 1. An error will be reported when the threshold is exceeded.

10.2.5.5 0x0010 TRNG Monitor APT Register (Default Value: 0x0040_03F1)

Offset: 0x0010			Register Name: TRNG_MONITOR_APT
Bit	Read/Write	Default/Hex	Description
31:23	/	/	/
22:11	R/W	0x0400	trng_APT_W APT detects the data length (1024 by default). TRNG_APT_W counts the number of 0 and 1 of continuous TRNG_APT_W. An error will be reported if any counted number of 0 or 1 exceeds TRNG_APT_C.
10: 0	R/W	0x03F1	trng_APT_C APT monitoring threshold (1009 by default)

10.2.5.6 0x0014 TRNG Extract Control Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: TRNG_EXSTRACT_CFG
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
25:12	R/W	0x0	TRNG_RESILIENT_RATIO The extraction ratio of elastic extraction function
11:10	/	/	/
9	R/W1C	0x0	TRNG_EXTRACT_START 1: Initiate elastic extraction function to generate random numbers.
8	R/W	0x0	TRNG_RESILIENT_TYPE entropy extraction 0: CRC32 1: XOR

Offset: 0x0014			Register Name: TRNG_EXSTRACT_CFG
Bit	Read/Write	Default/Hex	Description
7:4	R/W	0x0	TRNG_ro_sampling_ration1 Obtain the post-divider of trng_clk, which is needed by trng_ro_sampling_clk. The effective division value can be set by the equation: division value=trng_ro_sampling_div1+1
3: 0	R/W	0x0	TRNG_ro_sampling_ration0 Obtain the pre-divider of trng_clk, which is needed by trng_ro_sampling_clk. The effective division value can be set by the equation: division value =trng_ro_sampling_div0+1.

10.2.5.7 0x0018 TRNG Rand Data0 Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: TRNG_RAND_BIT_URN0
Bit	Read/Write	Default/Hex	Description
31: 0	RC	0x0	The content of random blocks, which is read only and read to clear.

10.2.5.8 0x001C TRNG Rand Data1 Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: TRNG_RAND_BIT_URN1
Bit	Read/Write	Default/Hex	Description
31: 0	RC	0x0	The content of random blocks, which is read only and read to clear.

10.2.5.9 0x0020 TRNG Rand Data2 Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: TRNG_RAND_BIT_URN2
Bit	Read/Write	Default/Hex	Description
31: 0	RC	0x0	The content of random blocks, which is read only and read to clear.

10.2.5.10 0x0024 TRNG Rand Data3 Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: TRNG_RAND_BIT_URN3
Bit	Read/Write	Default/Hex	Description
31: 0	RC	0x0	The content of random blocks, which is read only and read to clear.

10.2.5.11 0x0028 TRNG Jitter Counter Result0 Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: TRNG_JITTER_CNT_RESULT0
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x0	TRNG_JITTER_COUNTER_RESULT0 The number of rising edges output by the 0-channel divider within the intervals of TRNG_JITTER_COUNTER_TIMING

10.2.5.12 0x002C TRNG Jitter Counter Result1 Register (Default Value: 0x0000_0000)

Offset: 0x002C	Register Name: TRNG_JITTER_CNT_RESULT1

Bit	Read/Write	Default/Hex	Description
31: 0	R	0x0	TRNG_JITTER_COUNTER_RESULT1 The number of rising edges output by the 1st channel divider within the intervals of TRNG_JITTER_COUNTER_TIMING

10.2.5.13 0x0030 TRNG Jitter Counter Result2 Register (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: TRNG_JITTER_CNT_RESULT2
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	TRNG_JITTER_COUNTER_RESULT2 The number of rising edges output by the 2nd channel divider within the intervals of TRNG_JITTER_COUNTER_TIMING

10.2.5.14 0x0034 TRNG Jitter Counter Result3 Register (Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: TRNG_JITTER_CNT_RESULT3
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	TRNG_JITTER_COUNTER_RESULT3 The number of rising edges output by the 3rd channel divider within the intervals of TRNG_JITTER_COUNTER_TIMING

10.2.5.15 0x0038 TRNG Jitter Counter Result4 Register (Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: TRNG_JITTER_CNT_RESULT4
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	TRNG_JITTER_COUNTER_RESULT4 The number of rising edges output by the 4th channel divider within the intervals of TRNG_JITTER_COUNTER_TIMING

10.2.5.16 0x003C TRNG Jitter Counter Result5 Register (Default Value: 0x0000_0000)

Offset: 0x003C			Register Name: TRNG_JITTER_CNT_RESULT5
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	TRNG_JITTER_COUNTER_RESULT5 The number of rising edges output by the 5th channel divider within the intervals of TRNG_JITTER_COUNTER_TIMING

10.2.5.17 0x0040 TRNG Jitter Counter Result6 Register (Default Value: 0x0000_0000)

Offset: 0x0040			Register Name: TRNG_JITTER_CNT_RESULT6
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 0	R	0x0	TRNG_JITTER_COUNTER_RESULT6 The number of rising edges output by the 6th channel divider within the intervals of TRNG_JITTER_COUNTER_TIMING

10.2.5.18 0x0044 TRNG Jitter Counter Result7 Register (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: TRNG_JITTER_CNT_RESULT7
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x0	TRNG_JITTER_COUNTER_RESULT7 The number of rising edges output by the 7th channel divider within the intervals of TRNG_JITTER_COUNTER_TIMING

10.2.5.19 0x0048 TRNG Jitter Ready Register (Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: TRNG_JITTER_COUNTER_READY
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x0	TRNG_JITTER_COUNTER_READY It indicates the status of TRNG JITTER COUNTER 1: Jitter Counter Result is ready for reading; 0: Jitter Counter Result is not ready for reading.

10.2.5.20 0x004C TRNG Access Counter Low Register (Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: TRNG_REG_ACCESS_CTR1
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x0	It counts the times of APB accessing TRNG (low 32 bits). Accessing this register won't increase the times and generate initial value. The value of this register will be retained after wakeup.

10.2.5.21 0x0050 TRNG Access Counter High Register (Default Value: 0x0000_0000)

Offset: 0x0050			Register Name: TRNG_REG_ACCESS_CTR2
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7: 0	R	0x0	It counts the times of APB accessing TRNG (high 8 bits). Accessing this register won't increase the times and generate initial value. The value of this register will be retained after wakeup.

11 Appendix: Glossary

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

Term	Meaning
A	
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
AGC	Automatic Gain Control
AHB	AMBA High-Speed Bus
ALC	Automatic Level Control
APB	Advanced Peripheral Bus
ARM	Advanced RISC Machine
B	
BROM	Boot ROM
C	
CIR	Consumer Infrared
CMOS	Complementary Metal-Oxide Semiconductor
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSI	Camera Serial Interface
D	
DDR	Double Data Rate
DES	Data Encryption Standard
DLL	Delay-Locked Loop
DMA	Direct Memory Access
DRC	Dynamic Range Compression
E	
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
F	
FEL	Fireware Exchange Launch
FIFO	First In First Out
G	
GPIO	General Purpose Input Output
I	
I2S	Inter IC Sound
ISP	Image Signal Processor
J	
JEDEC	Joint Electron Device Engineering Council

Term	Meaning
JPEG	Joint Photographic Experts Group
JTAG	Joint Test Action Group
L	
LCD	Liquid-Crystal Display
LSB	Least Significant Bit
M	
MAC	Media Access Control
MIC	Microphone
MMC	Multimedia Card
MSB	Most Significant Bit
N	
N/A	Not Application
NTSC	National Television Standards Committee
O	
OHCI	Open Host Controller Interface
OWA	One Wire Audio
P	
PAL	Phase Alternating Line
PCM	Pulse Code Modulation
PHY	Physical Layer Controller
PID	Packet Identifier
PLIC	Platform-level Interrupt Controller
PLL	Phase-Locked Loop
POR	Power-On Reset
PRCM	Power Reset Clock Management
PWM	Pulse Width Modulation
R	
R	Read only/non-Write
RGB	Read Green Blue
ROM	Read Only Memory
RSA	Rivest-Shamir-Adleman
RTC	Real Time Clock
S	
SAR	Successive Approximation Register
SD	Secure Digital
SDIO	Secure Digital Input Output
SDRAM	Synchronous Dynamic Random Access Memory
SoC	System on Chip
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
T	
TDES	Triple Data Encryption Standard
TWI	Two Wire Interface
U	

Term	Meaning
UART	Universal Asynchronous Receiver Transmitter Transmitter
UDF	Undefined
USB DRD	Universal Serial Bus Dual Role Device
UTMI	USB2.0 Transceiver Macrocell Interface