



# R328-S3 User Manual

**Revision 1.4**

**Jan,10, 2022**

## DECLARATION

---

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

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

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

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

## Revision History

Revision	Date	Description
1.4	Jan.10, 2022	<ol style="list-style-type: none"><li>1. Add Trustzone description in section 2.2.1.</li><li>2. Update block diagram in section 2.3.</li></ol>
1.3	Dec.30, 2019	<ol style="list-style-type: none"><li>1. Add the calculation formula and example of spread spectrum in section 3.3.3.5.</li><li>2. Modify the register description about AC_DAC_FIFOC[FIFO_MODE] and AC_ADC_FIFOC[RX_FIFO_MODE] in section 5.4.6.2, section 5.4.6.7.</li><li>3. Modify the register description about LRADC_CTRL[LEVELB_VOL] in section 6.8.6.1.</li></ol>
1.2	Apr.18, 2019	<ol style="list-style-type: none"><li>1. Update LRADC feature in section 2.2.6.7 and section 6.8.1.</li><li>2. Delete the description for SPI referenced mass production rate in section 6.4.1.</li></ol>
1.1	Mar.04, 2019	Update bus clock tree in Figure 3-3.
1.0	Feb.20, 2019	Initial release version

## Contents

---

Chapter 1 About This Document .....	5
1.1. Purpose .....	5
1.2. Intended Audience.....	5
1.3. Symbol Conventions.....	5
1.4. Notes .....	6
1.4.1. Register Attributes .....	6
1.4.2. Reset Value Conventions.....	6
1.4.3. Numerical System.....	6
1.5. Acronyms and Abbreviations .....	7

# Chapter 1 About This Document

## 1.1. Purpose

This document describes the features, logical structures, functions, operating modes, and related registers of each module about R328-S3. For details about the interface timings and related parameters, the pins, pin usages, performance parameters, and package dimension, please refer to the *Allwinner R328-S3 Datasheet*.

## 1.2. Intended Audience

The document is intended for:

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

## 1.3. Symbol Conventions

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

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

## 1.4. Notes

### 1.4.1. Register Attributes

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

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

### 1.4.2. Reset Value Conventions

In the register definition tables:

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

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

### 1.4.3. Numerical System

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

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

The expressions of addresses and data are described as follows.

Symbol	Example	Description
0x	0x0200,0x79	Address or data in hexadecimal
0b	0b010,0b00 000 111	Data or sequence in binary(register description is excluded.)

X	00X,XX1	In data expression,X indicates 0 or 1.For example, 00X indicates 000 or 001, XX1 indicates 001,011,101 or 111.
---	---------	--

## 1.5. Acronyms and Abbreviations

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

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

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

ROM	Read Only Memory
RSA	Rivest-Shamir-Adleman
RTC	Real Time Clock
SAR	Successive Approximation Register
SD	Secure Digital
SDIO	Secure Digital Input Output
SDK	Software Development Kit
SDRAM	Synchronous Dynamic Random Access Memory
SDXC	Secure Digital Extended Capacity
SLC	Single-Level Cell
SoC	System on Chip
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
TDES	Triple Data Encryption Standard
TWI	Two Wire Interface
UART	Universal Asynchronous Receiver Transmitter
UDF	Undefined
USB OTG	Universal Serial Bus On The Go
UTMI	USB2.0 Transceiver Macrocell Interface

## Contents

---

Chapter 2 Product Description .....	12
2.1. Overview .....	12
2.2. Features.....	13
2.2.1. CPU Architecture.....	13
2.2.2. Memory SubSystem .....	13
2.2.3. System Peripheral.....	13
2.2.4. Audio Interfaces .....	16
2.2.5. Security Engine.....	17
2.2.6. External Peripherals .....	18
2.2.7. Package .....	21
2.3. Block Diagram .....	22

## Figures

---

Figure 2- 1. R328-S3 System Block Diagram .....	22
Figure 2- 2. R328-S3 Intelligent Speaker Solution.....	22

## Chapter 2 Product Description

---

### 2.1. Overview

Allwinner's R328-S3 is a highly integrated dual-core SoC targeted for audio application markets. The R328-S3 integrates a dual-core ARM Cortex™-A7 operating up to 1.2GHz. An extensive set of audio interfaces such as audio codec, I2S/PCM, DMIC, one wire audio(OWA) are included for microphone voice wake-up/recognition/record/playback applications on connected audio products. In addition, voice activity detector(VAD) supports low power consumption wake-up function to reduce standby power consumption.

To reduce the BOM cost, a 1Gbit DDR3 die is embedded for the R328-S3. The R328-S3 comes with extensive connectivity and interfaces, such as USB, SDIO, SPI, UART, TWI, etc. Also the R328-S3 has ability to connect with other different peripherals like WIFI and BT via SDIO and UART.

Security functions are enabled and accelerated by hardware crypto engine, secure boot and secure efuse, etc. The small footprint with low-power consumption greatly reduces the PCB layout resource.

## 2.2. Features

### 2.2.1. CPU Architecture

- Dual-core ARM Cortex™-A7 Processor
- ARMv7 ISA standard ARM instruction set
- Thumb-2 Technology
- Trustzone technology supported
- Jazeller RCT
- NEON Advanced SIMD
- VFPv4 floating point
- Large Physical Address Extensions(LPAE)
- 32KB L1 Instruction cache and 32KB L1 Data cache for per CPU
- 256KB L2 cache shared

### 2.2.2. Memory SubSystem

#### 2.2.2.1. Boot ROM

- On-chip memory
- Supports system boot from the following devices:
  - SPI Nor Flash
  - SPI Nand Flash
- Supports secure boot and normal boot
- Supports system code download through USB OTG
- Secure brom supports load only certified firmware
- Secure brom ensures that the secure boot is a trusted environment

#### 2.2.2.2. SDRAM

- Embedded with 1Gbit DDR3
- Supports clock frequency up to 792MHz for DDR3

### 2.2.3. System Peripheral

#### 2.2.3.1. VAD

- Supports 3 I2S interfaces, 1 DMIC PDM audio transfer interface
- Supported sample rate: 16 kHz, 48 kHz

- Supports voice detection module based on energy recognition
- Supports 4 working status: idle, wait, run, normal
- Supports voice activity: idle->energy wake-up
- Supports 128KB SRAM used to store audio data
- Supports hardware interpolation for audio data channel numbers, including 5 modes: 2chs-4chs, 2chs-6chs, 2chs-8chs, 4chs-6chs, 4chs-8chs
- Supports secure protection for register configuration
- Supports secure protection of VAD\_SRAM
- Supports VAD bypass function, after bypass, VAD\_SRAM can be used as on-chip SRAM to work individual

#### 2.2.3.2. Timer

- The timer module implements the timing and counting functions, which includes Timer0, Timer1 and Watchdog
- Timer0 and Timer1 for system scheduler counting
  - Configurable 8 prescale factor
  - Programmable 32-bit down timer
  - Supports two working modes: continue mode and single count mode
  - Generates an interrupt when the count is decreased to 0
- 1 Watchdog for transmitting a reset signal to reset the entire system after an exception occurs in the system
  - Supports 12 initial values to configure
  - Generation of timeout interrupts
  - Generation of reset signal
  - Watchdog restart the timing

#### 2.2.3.3. High Speed Timer

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

#### 2.2.3.4. GIC

- Supports 16 Software Generated Interrupts(SGIs), 16 Private Peripheral Interrupts(PPIs) and 137 Shared Peripheral Interrupts(SPIs)
- Enabling, disabling, and generating processor interrupts from hardware interrupt
- Interrupt masking and prioritization
- Uniprocessor and multiprocessor environments
- ARM architecture security extensions
- ARM architecture virtualization extensions
- Wakeup events in power-management environments

### 2.2.3.5. DMA

- Up to 10-channel DMA
- Interrupt generated for each DMA channel
- Flexible data width of 8/16/32/64-bit
- Supports linear and IO address modes
- Supports data transfer types with memory-to-memory, memory-to-peripheral, peripheral-to-memory, peripheral-to-peripheral
- Supports transfer with linked list
- DRQ response includes wait mode and handshake mode
- DMA channel supports pause function

### 2.2.3.6. CCU

- 6 PLLs
- One on-chip RC oscillator
- Supports one external DCXO and one external 32.768 kHz oscillator
- Supports clock configuration and clock generated for corresponding modules
- Supports software-controlled clock gating and software-controlled reset for corresponding modules

### 2.2.3.7. Thermal Sensor Controller

- Temperature accuracy :  $\pm 3^{\circ}\text{C}$  from  $0^{\circ}\text{C}$  to  $+100^{\circ}\text{C}$ ,  $\pm 5^{\circ}\text{C}$  from  $-25^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$
- Supports over-temperature protection interrupt and over-temperature alarm interrupt
- Averaging filter for thermal sensor reading
- One thermal sensor located in CPU

### 2.2.3.8. CPU Configuration

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

### 2.2.3.9. LDO Power

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

- Input voltage is 2.4V~3.6V

#### 2.2.3.10. Reset

- Integrated internal reset, which can select reset source by RST-BYP-N pin
- Internal reset R328-S3, also reset other IC

### 2.2.4. Audio Interfaces

#### 2.2.4.1. Audio Codec

- One audio digital-to-analog(DAC) channel
  - Supports 16-bit and 20-bit sample resolution
  - 8 kHz to 192 kHz DAC sample rate
  - $95\pm2$ dB SNR@A-weight,  $-80\pm3$ dB THD+N
- One audio output:
  - One differential LINEOUTP/N or single-ended LINEOUTL output
  - Full-scale output level is 1.1Vrms@0dBFS differential LINEOUTP/N
  - Full-scale output level is 0.55Vrms@0dBFS single-ended LINEOUTL
- Three audio analog-to-digital(ADC) channels
  - Supports 16-bit and 20-bit sample resolution
  - 8 kHz to 48 kHz ADC sample rate
  - $95\pm2$ dB SNR@A-weight,  $-80\pm3$ dB THD+N
- Three audio inputs:
  - Three differential microphone inputs: MICIN1P/1N, MICIN2P/2N, MICIN3P/3N
  - Full-scale input level is 1.8Vpp and maximum undistorted input level is 1.7Vpp
- One low-noise analog microphone bias output
- Supports Dynamic Range Controller adjusting the DAC playback and ADC capture
- One 128x24-bits FIFO for DAC data transmit, one 128x24-bits FIFO for ADC data receive
- Programmable FIFO thresholds
- DMA and Interrupt support

#### 2.2.4.2. I2S/PCM

- Up to three I2S/PCM interfaces
- Compliant with standard Philips Inter-IC sound (I2S) bus specification
- Compliant with Left-justified, Right-justified, PCM mode, and Time Division Multiplexing(TDM) format
- TDM mode supports maximum 16 channels output and 16 channels input
- Supports full-duplex synchronous work mode
- Supports master/slave mode
- Supports clock up to 24.576 MHz
- Supports adjustable audio sample resolution from 8-bit to 32-bit

- Sample rate from 8 kHz to 384 kHz(channels =2)
- Supports 8-bit u-law and 8-bit A-law companded sample
- One 128 depth x 32-bit width TXFIFO for data transmit, one 64 depth x 32-bit width RXFIFO for data receive
- Supports programmable PCM frame width: 1 BCLK width (short frame) and 2 BCLKs width (long frame)
- Programmable FIFO thresholds

#### 2.2.4.3. DMIC

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

#### 2.2.4.4. One Wire Audio(OWA)

- One OWA TX and one OWA RX
- IEC-60958 transmitter and receiver functionality
- Supports channel status capture on the receiver
- Supports channel status insertion for the transmitter
- Hardware parity checking on the receiver
- Hardware parity generation on the transmitter
- One 128×24bits TXFIFO and one 64×24bits RXFIFO for audio data transfer
- Programmable FIFO thresholds
- Interrupt and DMA support
- Supports 16-bit,20-bit and 24-bit data formats

### 2.2.5. Security Engine

#### 2.2.5.1. Crypto Engine(CE)

- Supports Symmetrical algorithm: AES,DES,TDES
  - Supports ECB,CBC,CTS,CTR,CFB,OFB mode for AES
  - Supports 128/192/256-bit key for AES
  - Supports ECB,CBC,CTR mode for DES/TDES
- Supports Hash algorithm: MD5,SHA,HMAC
  - Supports SHA1,SHA224,SHA256,SHA384,SHA512 for SHA
  - Supports HMAC-SHA1,HMAC-SHA256 for HMAC
  - MD5,SHA,HMAC are padded using hardware
- Supports Asymmetrical algorithm: RSA
  - RSA supports 512/1024/2048-bit width
- Supports 160-bit hardware PRNG with 175-bit seed
- Supports 256-bit hardware TRNG
- Internal embedded DMA to do data transfer
- Supports secure and non-secure interfaces respectively

- Supports task chain mode for each request. Task or task chain are executed at request order
- 8 scatter group(sg) are supported for both input and output data
- DMA has multiple channels, each corresponding to one suit of algorithm

#### 2.2.5.2. Security ID

- Supports 1Kbit EFUSE for chip ID and security application
- EFUSE has secure zone and non-secure zone

#### 2.2.5.3. Secure Memory Control(SMC)

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

#### 2.2.5.4. Secure Peripherals Control(SPC)

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

### 2.2.6. External Peripherals

#### 2.2.6.1. SMHC

- One SD/MMC host controller(SMHC) interface
- SMHC1 controls the devices that comply with the Secure Digital Input/Output(SDIO3.0)
  - 4-bit bus width
  - SDR mode 50MHz@3.3V IO pad
  - SDR mode 150MHz@1.8V IO pad
  - DDR mode 50MHz@1.8V IO pad
- Supports hardware CRC generation and error detection
- Supports block size of 1 to 65535 bytes

#### 2.2.6.2. USB

- One USB 2.0 OTG, with integrated USB 2.0 analog PHY
- Compatible with USB2.0 Specification
- Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s) and Low-Speed (LS, 1.5 Mbit/s) in host mode

- Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s) in device mode
- Compatible with Enhanced Host Controller Interface(EHCI)Specification, Version 1.0, and the Open Host Controller Interface(OHCI) Specification, Version 1.0a for host mode
- Up to 8 User-Configurable Endpoints(EPs) for Bulk, Isochronous and Interrupt bi-directional transfers
- Supports (4KB+64Bytes) FIFO for all EPs(including EP0)
- Supports point-to-point and point-to-multipoint transfer in both host and peripheral mode

#### 2.2.6.3. UART

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

#### 2.2.6.4. Two Wire Interface(TWI)

- Up to 2 TWI controllers(TWI0,TWI1)
- Software-programmable for slave or master
- Supports repeated START signal
- Multi-master system supported
- Allows 10-bit addressing transactions
- Performs arbitration and clock synchronization
- Own address and General Call address detection
- Interrupt on address detection
- Supports Standard mode(up to 100 kbit/s) and Fast mode(up to 400 kbit/s)
- Allows operation from a wide range of input clock frequency
- TWI driver supports packet transmission and DMA when TWI works in Master mode

#### 2.2.6.5. SPI

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

- Interrupt or DMA support
- Supports 3-wire/4-wire SPI
- Supports programmable serial data frame length: 1-bit to 32-bit
- Supports Standard SPI, Dual-Output/Dual-Input SPI, Dual IO SPI, Quad-Output/Quad-Input SPI

#### 2.2.6.6. PWM

- 8 PWM channels(4 PWM pairs)
- Supports pulse, cycle and complementary pair output
- Supports capture input
- Programming deadzone output
- Build-in the programmable dead-time generator, controllable dead-time
- Three kinds of output waveform: continuous waveform, pulse waveform and complementary pair
- Output frequency range: 0~24MHz/100MHz
- Various duty-cycle: 0%~100%
- Minimum resolution: 1/65536
- Interrupt generation of PWM output and capture input

#### 2.2.6.7. LRADC

- One LRADC input channel
- 6-bit resolution
- Sample rate up to 2kHz
- Supports hold Key and general Key
- Supports normal, continue and single work mode
- Power supply voltage: AVCC, power reference voltage: 0.75\*AVCC, analog input and detected voltage range: 0~LEVELB(the maximum value is 1.266V)

#### 2.2.6.8. GPADC

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

#### 2.2.6.9. LEDC

- LEDC is used to control the external intelligent control LED lamp
- Configurable LED output high/low level width

- Configurable LED reset time
- LEDC data supports DMA configuration mode and CPU configuration mode
- Maximum 1024 LEDs serial connect
- LED data transfer rate up to 800 kbit/s

### 2.2.7. Package

- LFBGA144 balls, 0.65mm ball pitch, 0.35mm ball size, 9.15mm x 11.1mm body

## 2.3. Block Diagram

Figure 2-1 shows the system block diagram of the R328-S3.

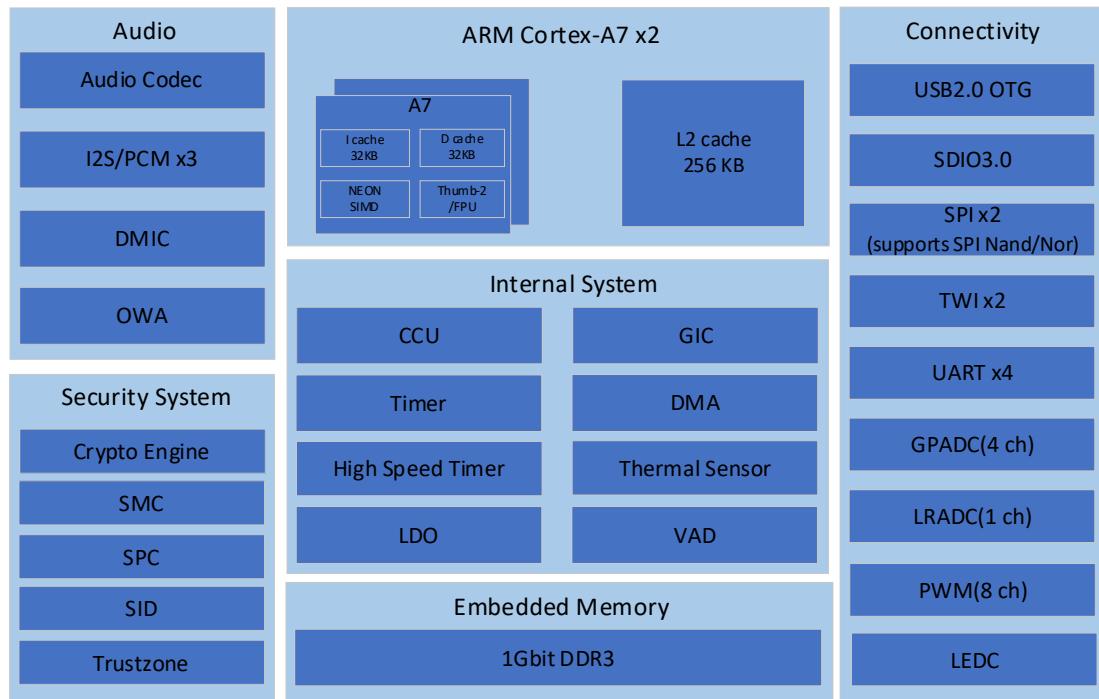


Figure 2- 1. R328-S3 System Block Diagram

Figure 2-2 shows the intelligent speaker solution of the R328-S3.

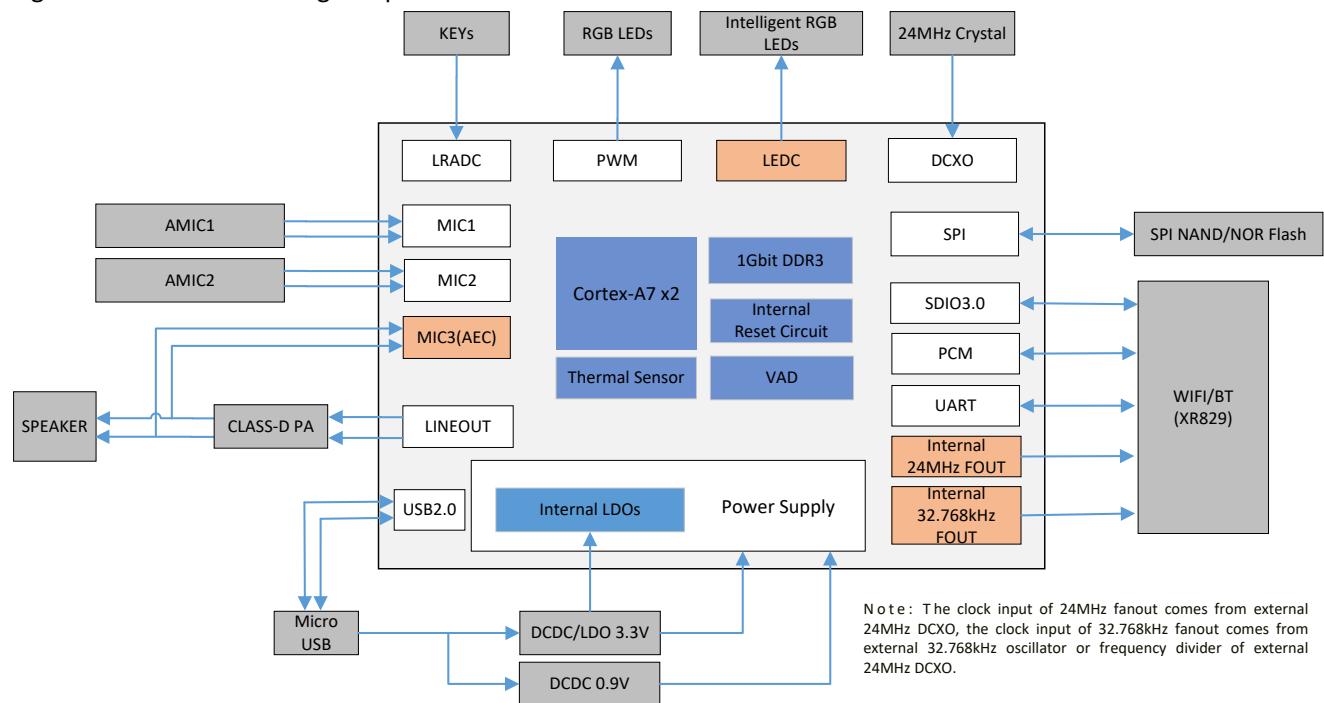


Figure 2- 2. R328-S3 Intelligent Speaker Solution

# Contents

Chapter 3 System .....	27
3.1. Memory Mapping .....	27
3.2. CPUX Configuration .....	29
3.2.1. Overview .....	29
3.2.2. Operations and Functional Descriptions .....	29
3.2.3. Programming Guidelines.....	31
3.2.4. Cluster 0 Configuration Register List .....	31
3.2.5. Cluster 0 Configuration Register Description .....	31
3.2.6. CPU Subsystem Control Register List.....	36
3.2.7. CPU Subsystem Control Register Description.....	36
3.3. CCU.....	39
3.3.1. Overview .....	39
3.3.2. Operations and Functional Descriptions .....	39
3.3.3. Programming Guidelines.....	44
3.3.4. Register List .....	46
3.3.5. Register Description .....	49
3.4. BROM System .....	98
3.4.1. Overview .....	98
3.4.2. Operations and Functional Descriptions .....	98
3.5. System Configuration .....	107
3.5.1. Overview .....	107
3.5.2. Register List .....	107
3.5.3. Register Description .....	107
3.6. Timer .....	109
3.6.1. Overview .....	109
3.6.2. Block Diagram .....	109
3.7. High Speed Timer.....	120
3.7.1. Overview .....	120
3.7.2. Block Diagram .....	120
3.7.3. Operations and Functional Description.....	120
3.7.4. Programming Guidelines.....	122
3.7.5. Register List .....	123
3.7.6. Register Description .....	123
3.8. GIC.....	129
3.8.1. Interrupt Source .....	129
3.9. DMA .....	132
3.9.1. Overview .....	132
3.9.2. Block Diagram .....	132
3.9.3. Operations and Functional Description.....	133
3.9.4. Programming Guidelines.....	141

3.9.5. Register List .....	142
3.9.6. Register Description .....	142
3.10. Thermal Sensor Controller .....	156
3.10.1. Overview .....	156
3.10.2. Block Diagram .....	156
3.10.3. Operations and Functional Descriptions .....	156
3.10.4. Programming Guidelines.....	158
3.10.5. Register List .....	159
3.10.6. Register Description .....	160
3.11. PSI.....	164
3.11.1. Overview .....	164
3.11.2. Block Diagram .....	164
3.12. VAD.....	165
3.12.1. Overview .....	165
3.12.2. Block Diagram .....	165
3.12.3. Operations and Functional Descriptions .....	166
3.12.4. Programming Guidelines.....	168
3.12.5. Register List .....	169
3.12.6. Register Description .....	170

## Figures

Figure 3- 1. CPUX Power Domain Block Diagram.....	30
Figure 3- 2. System Bus Tree.....	40
Figure 3- 3. Bus Clock Tree.....	41
Figure 3- 4. Module Clock Tree .....	42
Figure 3- 5. Normal Mode Boot Process.....	100
Figure 3- 6. Secure BROM Process Diagram .....	101
Figure 3- 7. NON_CPU0 Boot Process .....	102
Figure 3- 8. Mandatory Upgrade Process .....	103
Figure 3- 9. USB FEL Process .....	104
Figure 3- 10. Try Media Boot Process .....	105
Figure 3- 11. eFuse Boot Select Process.....	106
Figure 3- 12. Timer Block Diagram.....	109
Figure 3- 13. Timer Application Diagram .....	110
Figure 3- 14. HSTimer Block Diagram.....	120
Figure 3- 15. HSTimer Application Diagram .....	121
Figure 3- 16. HSTimer Initialization Process.....	122
Figure 3- 17. DMA Block Diagram .....	132
Figure 3- 18. DMA Typical Application Diagram .....	133
Figure 3- 19. DMA Descriptor .....	135
Figure 3- 20. DMA Chain Transfer.....	136
Figure 3- 21. DMA Transfer Mode .....	138
Figure 3- 22. DMA Transfer Process.....	140
Figure 3- 23. Thermal Sensor Controller Block Diagram.....	156
Figure 3- 24. Thermal Sensor Time Requirement .....	157
Figure 3- 25. Thermal Sensor Controller Interrupt Source .....	157
Figure 3- 26. THS Initial Process .....	158
Figure 3- 27. PSI Block Diagram .....	164
Figure 3- 28. VAD System Architeture Diagram.....	166
Figure 3- 29. VAD Data Flow .....	167
Figure 3- 30. VAD Clock System .....	168

## Tables

---

Table 3- 1. Reset Signal Description .....	29
Table 3- 2. PLL Features .....	43
Table 3- 3. BOOT_MODE Setting.....	98
Table 3- 4. eFuse Boot Select Configure .....	99
Table 3- 5. eFuse Boot Select Setting.....	99
Table 3- 6. DMA DRQ Table.....	133

# Chapter 3 System

## 3.1. Memory Mapping

Module	Address(It is for Cluster CPU)	Size(Bytes)
N-BROM	0x0000 0000---0x0000 7FFF	32KB
S-BROM	0x0000 0000---0x0000 7FFF	32KB
SRAM A1	0x0002 0000---0x0003 FFFF	128K(Borrow VAD sram, supports Byte operation, clock source is AHB1)
<b>Accelerator</b>		
CE	0x0190 4000---0x0190 4FFF	4K
<b>System Resources</b>		
SYS_CFG	0x0300 0000---0x0300 0FFF	4K
CCU	0x0300 1000---0x0300 1FFF	4K
DMA	0x0300 2000---0x0300 2FFF	4K
HSTIMER	0x0300 5000---0x0300 5FFF	4K
SID	0x0300 6000---0x0300 6FFF	4K
SMC	0x0300 7000---0x0300 7FFF	4K
SPC	0x0300 8000---0x0300 83FF	1K
TIMER	0x0300 9000---0x0300 93FF	1K
PWM	0x0300 A000---0x0300 A3FF	1K
GPIO	0x0300 B000---0x0300 B3FF	1K
PSI	0x0300 C000---0x0300 C3FF	1K
DCU	0x0301 0000---0x0301 FFFF	64K
GIC	0x0302 0000---0x0302 FFFF	64K
DCXO	0x0700 0000---0x0700 03FF	1K
<b>Memory</b>		
SMHC1	0x0402 1000---0x0402 1FFF	4K
MSI_CTRL	0x0400 2000---0x0400 2FFF	4K
DRAM_CTRL&PHY	0x0400 3000---0x0400 5FFF	12K
<b>Interfaces</b>		
UART0	0x0500 0000---0x0500 03FF	1K
UART1	0x0500 0400---0x0500 07FF	1K
UART2	0x0500 0800---0x0500 0BFF	1K
UART3	0x0500 0C00---0x0500 0FFF	1K
TWI0	0x0500 2000---0x0500 23FF	1K

TWI1	0x0500 2400---0x0500 27FF	1K
SPI0	0x0501 0000---0x0501 0FFF	4K
SPI1	0x0501 1000---0x0501 1FFF	4K
GPADC	0x0507 0000---0x0507 03FF	1K
THS	0x0507 0400---0x0507 07FF	1K
LRADC	0x0507 0800---0x0507 0BFF	1K
I2S/PCM0	0x0509 0000---0x0509 0FFF	4K
I2S/PCM1	0x0509 1000---0x0509 1FFF	4K
I2S/PCM2	0x0509 2000---0x0509 2FFF	4K
OWA	0x0509 3000---0x0509 33FF	1K
DMIC	0x0509 5000---0x0509 53FF	1K
Audio Codec	0x0509 6000---0x0509 6FFF	4K
USB0(USB2.0_OTG)	0x0510 0000---0x051F FFFF	1M
VAD	0x0540 0000---0x0540 0FFF	4K
VAD_SRAM	0x0548 0000---0x0549 FFFF	128K
LEDC	0x0670 0000---0x0670 03FF	1K
<b>CPUX Related</b>		
CPU_SYS_CFG	0x0810 0000---0x0810 03FF	1K
TimeStamp_STA	0x0811 0000---0x0811 0FFF	4K
TimeStamp_CTRL	0x0812 0000---0x0812 0FFF	4K
IDC	0x0813 0000---0x0813 0FFF	3K
C0_CPUX_CFG	0x0901 0000---0x0901 03FF	1K
C0_CPUX_MBIST	0x0902 0000---0x0902 0FFF	4K
<b>DRAM</b>		
DRAM	0x4000 0000---0xFFFF FFFF	3G

## 3.2. CPUX Configuration

### 3.2.1. Overview

The C0\_CPUX\_CFG module is used for configuring cluster0(dual-core Cortex-A7, 32KB I-cache and 32KB D-cache per core, shared 256KB L2 cache), such as reset, control, cache, debug, CPU status.

The CPU\_SUBSYS\_CTRL module is used for the system resource control of CPU sub-system, such as GIC-400,JTAG.

The CPUX\_CFG includes the following features:

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

### 3.2.2. Operations and Functional Descriptions

#### 3.2.2.1. Signal Description

For the detail of CPUX signal, please refer to **ARM Cortex-A7 TRM**, such as **DDI0464F\_cortex\_a7\_mpcore\_r0p5\_trm.pdf**

#### 3.2.2.2. L2 Idle Mode

When the L2 cache of Cluster needs to enter WFI mode, firstly make sure the CPU[1:0] of Cluster enter WFI mode, which can be checked through the bit[17:16] of **Cluster CPU Status Register**, and then pull high the **ACINACTM** of Cluster by writing 1 to the bit0 of **Cluster Control Register1**, and then check whether L2 enters idle status by checking whether the **STANDBYWFL2** is high. Note that set the **ACINACTM** to low when exiting the L2 idle mode.

#### 3.2.2.3. CPUX Reset System

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

**Table 3- 1. Reset Signal Description**

Reset Signal	Description
CORE_RST	This is the primary reset signal which resets the corresponding core logic that includes NEON and VFP, Debug, ETM, breakpoint and watchpoint logic. This maps to a warm reset that covers reset of the processor logic.

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

### 3.2.2.4. CPUX Power Block Diagram

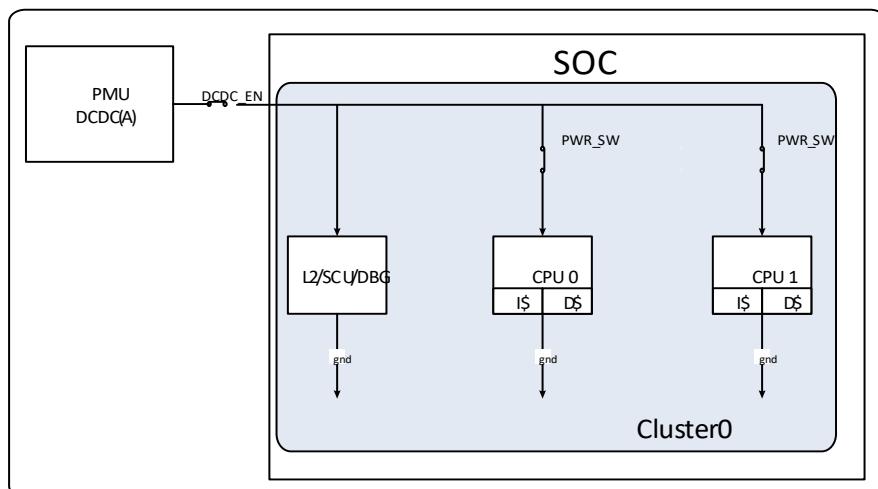


Figure 3-1. CPUX Power Domain Block Diagram

Figure 3-1 above lists the power domain of cluster in default. For cluster0, the power switch of all CPU core are power-on, the pwrn\_rst of all CPU cores is de-asserted, the core reset of CPU0 is de-asserted, the core reset of CPU1 is asserted.

C0\_CPUX\_CFG and cluster0 belong to the same power domain, within opening and closing cluster0 process, when cluster0 starts to power on again from power-off state, C0\_CPUX\_CFG holds in default state, at this time software need initial C0\_CPUX\_CFG after C0\_H\_RST is de-asserted.

CPU\_SUBSYS\_CTRL belongs to system power domain. The power domains of CPU related module are as follows.

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

### 3.2.2.5. Operation Principle

The CPU-related operations(such as open/close core, cluster switch, status query) need proper configuration of CO\_CPUX\_CFG module, as well as related system control resource including BUS, clock.

### 3.2.3. Programming Guidelines

For CPU core and cluster operation,please see the **R328-S3\_CPU\_AP\_Note**.

### 3.2.4. Cluster 0 Configuration Register List

Module Name	Base Address
CO_CPUX_CFG	0x09010000

Register Name	Offset	Description
C0_RST_CTRL	0x0000	Cluster 0 Reset Control Register
C0_CTRL_REG0	0x0010	Cluster 0 Control Register0
C0_CTRL_REG1	0x0014	Cluster 0 Control Register1
C0_CTRL_REG2	0x0018	Cluster 0 Control Register2
CACHE_CFG_REG	0x0024	Cache Configuration Register
C0_CPU_STATUS	0x0080	Cluster 0 CPU Status Register
L2_STATUS_REG	0x0084	Cluster 0 L2 Status Register
DBG_REG0	0x00C0	Cluster 0 Debug Control Register0
DBG_REG1	0x00C4	Cluster 0 Debug Control Register1

### 3.2.5. Cluster 0 Configuration Register Description

#### 3.2.5.1. Cluster 0 Reset Control Register(Default Value: 0x13FF\_0101)

Offset: 0x0000			Register Name: C0_RST_CTRL
Bit	Read/Write	Default/Hex	Description
31:29	/	/	/
28	R/W	0x1	DDR_RST AXI2MBUS logic circuit Reset 0: assert 1: de-assert
27:26	/	/	/
25	R/W	0x1	MBIST_RST

			CPUBIST Reset. The reset signal for test 0: assert 1: de-assert
24	R/W	0x1	SOC_DBG_RST Cluster SOC Debug Reset 0: assert 1: de-assert
23:20	R/W	0xF	ETM_RST Cluster ETM Reset Assert 0: assert 1: de-assert
19:16	R/W	0xF	DBG_RST Cluster Debug Reset Assert 0: assert 1: de-assert
15:9	/	/	/
8	R/W	0x1	L2_RST Cluster L2 Cache Reset 0: assert 1: de-assert
7:2	/	/	/
1:0	R/W	0x1	CORE_RESET Cluster CPU[1:0] Reset Assert 0: assert 1: de-assert

### 3.2.5.2. Cluster 0 Control Register0(Default Value: 0x8000\_0000)

Offset: 0x0010			Register Name: C0_CTRL_REG0
Bit	Read/Write	Default/Hex	Description
31	R/W	0x1	SYSBAR_DISABLE Disable broadcasting of barriers onto system bus: 0: Barriers are broadcast onto system bus, this requires an AMBA4 interconnect 1: Barriers are not broadcast onto the system bus. This is compatible with an AXI3 interconnect
30	R/W	0x0	BROADCAST_INNER Enable broadcasting of Inner Shareable transactions: 0: Inner shareable transactions are not broadcasted externally 1: Inner shareable transactions are broadcasted externally
29	R/W	0x0	BROADCAST_OUTER Enable broadcasting of outer shareable transactions 0: Outer Shareable transactions are not broadcasted externally 1: Outer Shareable transactions are broadcasted externally

28	R/W	0x0	BROADCAST_CACHE_MAINT Enable broadcasting of cache maintenance operations to downstream caches 0: Cache maintenance operations are not broadcasted to downstream caches 1: Cache maintenance operations are broadcasted to downstream caches
27:12	/	/	/
11:8	R/W	0x0	CP15S_DISABLE Disable write access to some secure CP15 register
7:5	/	/	/
4	R/W	0x0	L2_RST_DISABLE Disable automatic L2 cache invalidate at reset 0: L2 cache is reset by hardware 1: L2 cache is not reset by hardware
3:2	R	0x0	/
1:0	R/W	0x0	L1_RST_DISABLE Disable automatic Cluster CPU[1:0] L1 cache invalidate at reset 0: L1 cache is reset by hardware 1: L1 cache is not reset by hardware

### 3.2.5.3. Cluster 0 Control Register1(Default Value: 0x0000\_0000)

Offset: 0x0014			Register Name: C0_CTRL_REG1
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W	0x0	CRM auto select slow frequency enable 0 : disable auto select 1 : enable auto select
0	R/W	0x0	ACINACTM Snoop interface is inactive and no longer accepting requests 0: Snoop interface is active 1: Snoop interface is inactive

### 3.2.5.4. Cluster 0 Control Register2(Default Value: 0x0000\_0010)

Offset: 0x0018			Register Name: C0_CTRL_REG2
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	EVENTI Event input for processor wake-up from WFE state. This bit must remain high for at least one clock cycle to be visible by the cores.
23	/	/	/

22:20	R/W	0x0	EXM_CLR[1:0] Clear the status of interface.
19:0	/	/	/

### 3.2.5.5. Cache Configuration Register(Default Value: 0x001A\_001A)

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

### 3.2.5.6. Cluster0 CPU Status Register(Default Value: 0x000E\_0000)

Offset: 0x0080			Register Name: C0_CPU_STATUS
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
25:24	R	0x0	SMP_AMP CPU[1:0] is in Symmetric Multiprocessing mode or Asymmetric Multiprocessing mode. 0: AMP mode 1: SMP mode
23:18	/	/	/
17:16	R	0xE	STANDBYWFI. Indicates if Cluster CPU[1:0] is in WFI standby mode: 0: Processor not in WFI standby mode. 1: Processor in WFI standby mode
15:10	/	/	/
9:8	R	0x0	STANDBYWFE. Indicates if Cluster CPU[1:0] is in the WFE standby mode: 0: Processor not in WFE standby mode

			1: Processor in WFE standby mode
7:1	/	/	/
0	R	0x0	<p>STANDBYWFL2.</p> <p>Indicates if the Cluster L2 memory system is in WFI standby mode.</p> <p>0: Cluster L2 not in WFI standby mode</p> <p>1: Cluster L2 in WFI standby mode</p>

### 3.2.5.7. L2 Status Register(Default Value: 0x0000\_0000)

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

### 3.2.5.8. Cluster 0 Debug Control Register0(Default Value:0x0000\_000F)

Offset: 0x00C0			Register Name: DBG_REG0
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9:8	R/W	0x0	<p>DBGRESTART[1:0]</p> <p>External restart requests.</p>
7:0	/	/	/

### 3.2.5.9. Cluster 0 Debug Control Register1(Default Value: 0x0000\_0000)

Offset: 0x00C4			Register Name: DBG_REG1
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13:12	R/W	0x0	<p>DBGRESTARTED[1:0]</p> <p>Handshake for DBGRESTART.</p>
11:6	/	/	/
5:2	R	0x0	<p>C_DBGNOPWRDWN.</p> <p>No power-down request. Debugger has requested that processor is not powered down.</p> <p>Debug no power down[1:0].</p>
1:0	R	0x0	<p>C_DBGPWRUPREQ.</p> <p>Power up request.</p> <p>Debug power up request[1:0]</p>

			0: Do not request that the core is powered up 1: Request that the core is powered up
--	--	--	---

### 3.2.6. CPU Subsystem Control Register List

Module Name	Base Address
CPU_SUBSYS_CTRL	0x08100000

Register Name	Offset	Description
GENER_CTRL_REG0	0x0000	General Control Register0
GIC_JTAG_RST_CTRL	0x000C	GIC and Jtag reset control Register
C0_INT_EN	0x0010	Cluster0 interrupt enable control register
IRQ_FIQ_STATUS	0x0014	IRQ/FIQ status register
GENER_CTRL_REG2	0x0018	General control register2
DBG_STATE	0x001C	Debug state register

### 3.2.7. CPU Subsystem Control Register Description

#### 3.2.7.1. General Control Register0(Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: GENER_CTRL_REG0
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W	0x0	IDC clock enable 0: disable IDC clock 1: enable IDC clock
0	R/W	0x0	GIC_CFGSDISABLE Disables write access to some secure GIC registers.

#### 3.2.7.2. General Control Register1(Default Value: 0x0000\_000E)

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

### 3.2.7.3. GIC and Jtag Reset Control Register(Default Value: 0x0000\_0F07)

Offset: 0x000C			Register Name: GIC_JTAG_RST_CTRL
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17:16	R/W	0x0	EXM_CLR[1:0] Clear the status of interface, for debug
15:12	/	/	/
11	R/W	0x1	CS_RST CoreSight Reset. 0: assert 1: de-assert
10	R/W	0x1	DAP_RST DAP Reset. 0: assert 1: de-assert
9	R/W	0x1	PORTRST Jtag portrst. 0: assert 1: de-assert
8	R/W	0x1	TRST. Jtag trst. 0: assert 1: de-assert
7:2	/	/	/
1	R/W	0x1	IDC_RST Interrupt delay controller reset. 0: assert 1: de-assert
0	R/W	0x1	GIC_RST GIC_reset_cpu_reg 0: assert 1: de-assert

### 3.2.7.4. Cluster 0 Interrupt Enable Register(Default Value: 0x0000\_FFFF)

Offset: 0x0010			Register Name: C0_INT_EN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFFFF	C0_GIC_EN Interrupt enable control register. Mask irq_out/firq_out to system domain.

### 3.2.7.5. GIC IRQ/FIQ Status Register(Default Value: 0x0000\_0000)

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

### 3.2.7.6. General Control Register2(Default Value: 0x0000\_0000)

Offset: 0x0018			Register Name: GENER_CTRL_REG2
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	CDBG_RSTACK Debug Reset ACK
15:1	/	/	/
0	R/W	0x0	C0_TSCLKCHANGE Cluster 0 Time Stamp change bit

## 3.3. CCU

### 3.3.1. Overview

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

The CCU includes the following features:

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

### 3.3.2. Operations and Functional Descriptions

#### 3.3.2.1. System Bus Tree

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

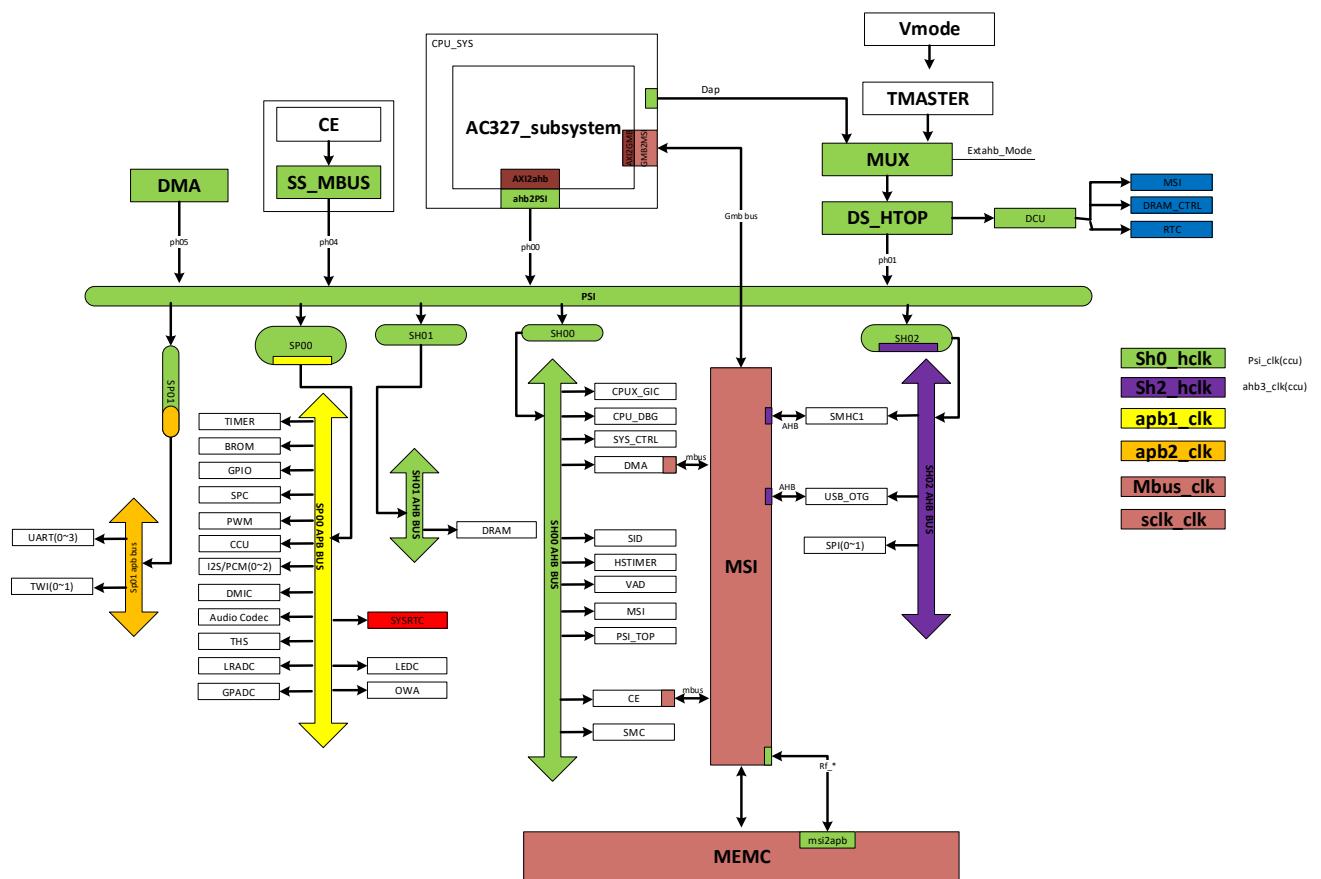


Figure 3- 2. System Bus Tree

The above system bus tree is used to introduce bus interface of every module. These modules can divide into two types: bus master, slave. For example, DMA, BIST\_MST(Built-in Self Test Master), CE, SMHC1, CPUS, CPUX(A7x2) and Debug System(DS) are as bus master that can access corresponding register of every slave through bus. Every slave hangs in corresponding bus.

For example, CPU accesses to RTC module, the process is as follows: CPU instruction firstly passes AXI bus, then goes to PSI bus through AXI2PSI bridge, then goes to operate RTC based on relevant bus protocol through PSI2APB bridge. The access time from CPU to RTC , is relevant with the CPU clock , AXI bus clock, PSI bus clock, and APB bus clock. Any lower bus clock will lead to access time very long.

In above module, the clocks of these modules(such as TWI and UART) to be hung on APB1 are from their respective bus clock , however the clocks of most other modules are from related CLK register. Each module clock requirement can refer to their module.

### 3.3.2.2. Bus Clock Tree

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

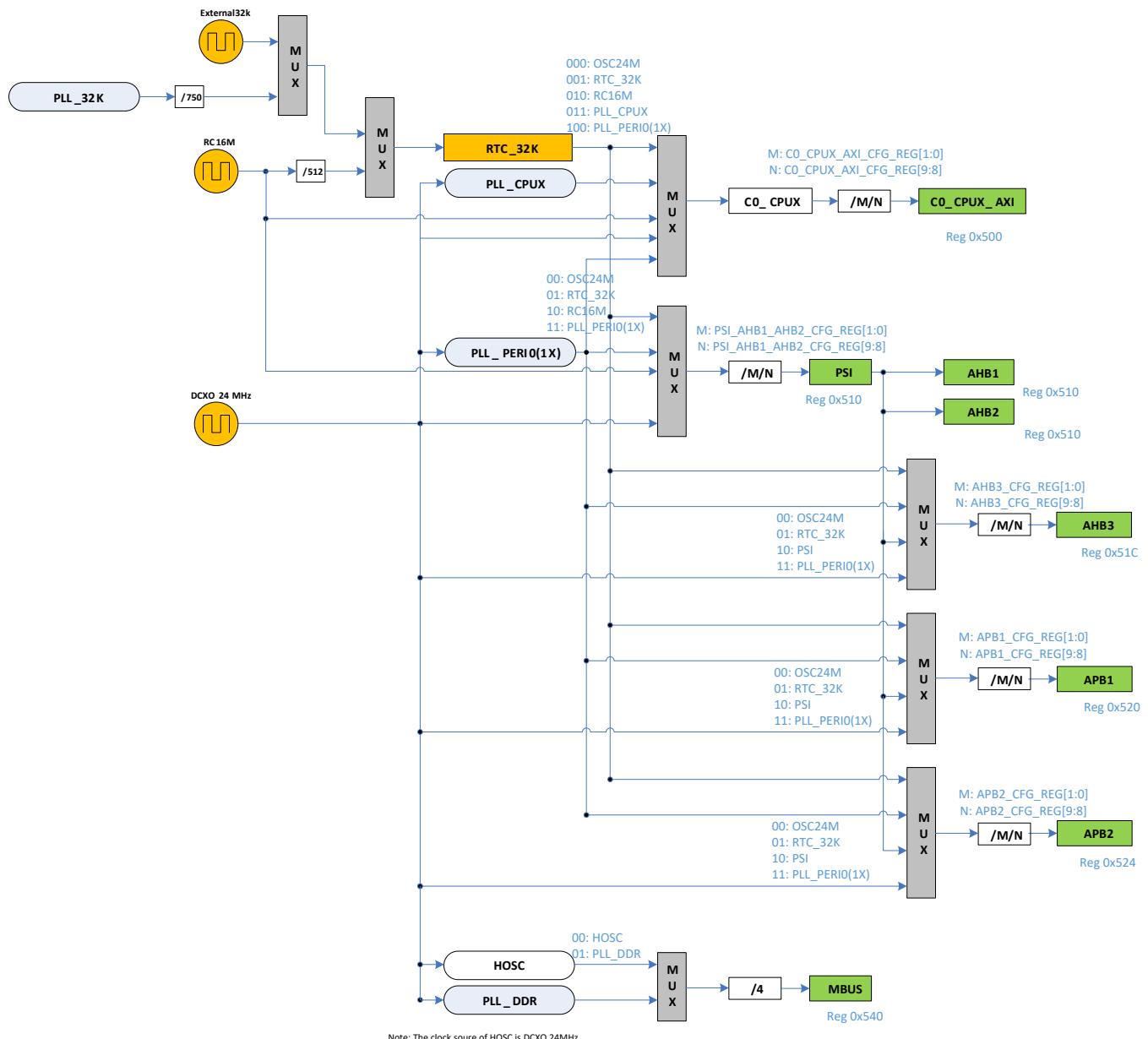


Figure 3- 3. Bus Clock Tree

The bus clock tree diagram shows the various bus clock source, relevant divider factor, etc.

For example, the clock source of CPU is from PLL\_CPU (48MHz ~ 2GHz), OSC\_24M, RC\_16M, RTC\_32K, etc. The internal AXI bus clock source is from the clock source of CPU, but on the base of CPU clock, two configurable divider factors(M,N) are added to adjust the AXI bus clock.

The clock source of RTC\_32K is from external 32 kHz, or internal RC\_16M by 512 fractional frequency.

The bus clocks commonly are used as follows.

AXI: 1/3 of CPU frequency.

PSI、AHB1、AHB2、AHB3: 200MHz.

PSI、AHB1、AHB2 is controlled by PSI\_AHB1\_AHB2\_CFG\_REG. AHB3 is controlled by AHB3\_CFG\_REG.

APB1、APB2: maximum of 100MHz, separately controlled by APB1\_CFG\_REG、APB2\_CFG\_REG .

### 3.3.2.3. Module Clock Tree

Figure 3-4 shows the block diagram of the Module Clock Tree.

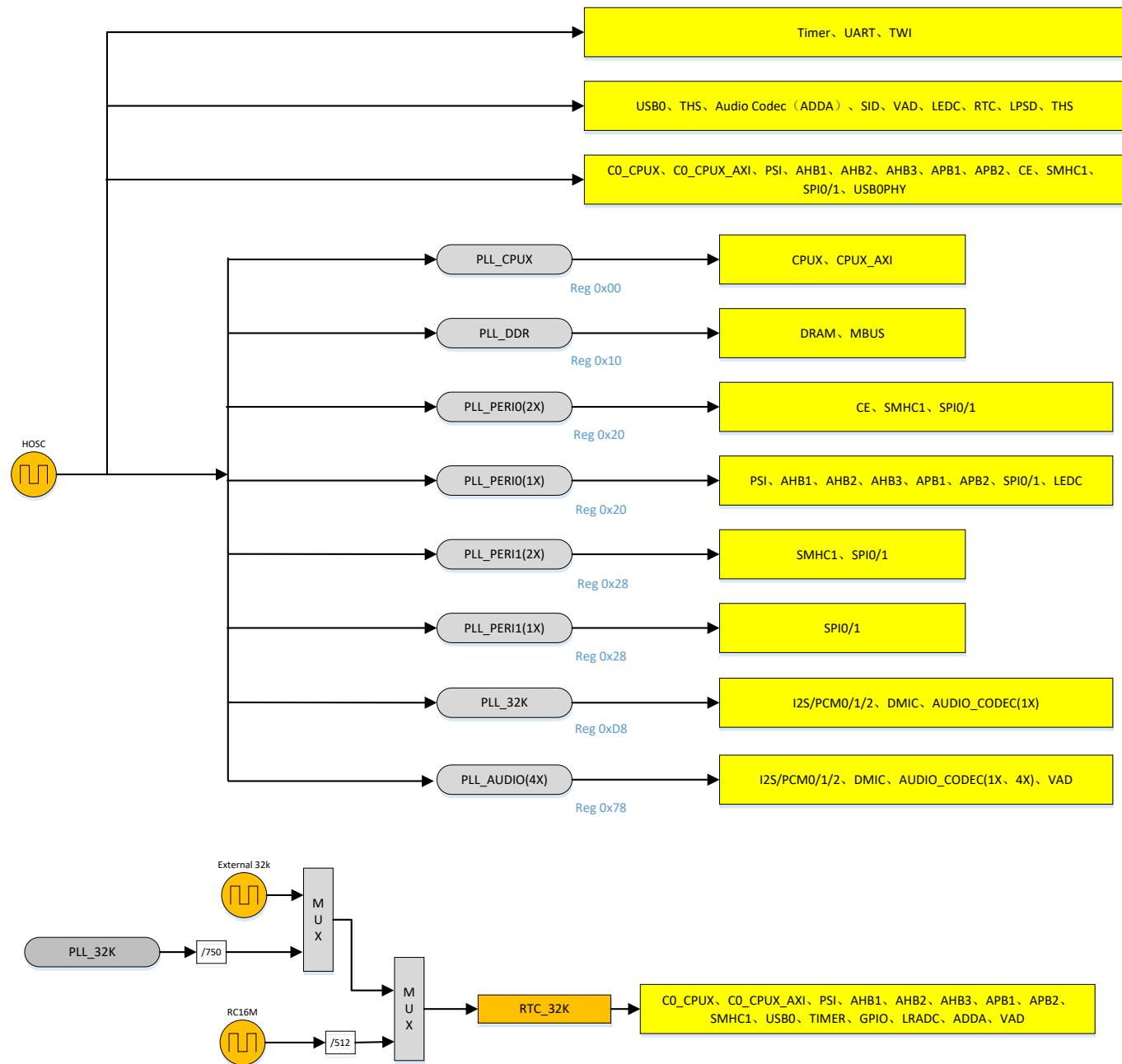


Figure 3- 4. Module Clock Tree

### 3.3.2.4. PLL Features

**Table 3- 2. PLL Features**

PLL	Stable Operating Frequency	Actual Operating Frequency	Spread Spectrum	Linear FM	Rate Control	Pk-Pk	Wait Duty	Lock Time
PLL_CPUX	288MHz~5.0GHz (24*N)	288MHz~1.8GHz	Yes	No	No	<200ps	Yes	2ms
PLL_AUDIO	24.576MHz, 22.5792MHz (24*N/div1/div2)	24.576MHz, 22.5792MHz, (24.576 * 8) MHz, (22.5792 * 8) MHz	Yes(fractional frequency division)	No	No	<200ps	When M0=2, meet wait-duty	500us
PLL_PERIO(2X)	180MHz~3.0GHz (24*N/div1/div2)	1.2GHz	No	No	No	<200ps	Yes	500us
PLL_PERI1(2X)	180MHz~3.0GHz (24*N/div1/div2)	1.2GHz	Yes	No	No	<200ps	Yes	500us
PLL_DDRO	180MHz~3GHz (24*N/div1/div2)	192MHz~2GHz	Yes	No	No	200MHz~800MHz(<200ps) 800MHz~1.3GHz(<140ps) 1.3GHz~2.0GHz(<100ps)	Yes	500us
PLL_32K	(24*N/div1/div2)	24.576MHz	Yes(fractional frequency division)	No	No	<200ps	When M0=2, meet wait-duty	500us

### 3.3.3. Programming Guidelines

#### 3.3.3.1. PLL

- (1) In practical application, other PLLs do not support dynamic frequency scaling except for PLL\_CPUX and PLL\_GPU.
- (2) The user guide of PLL Lock(using PLL\_CPUX as an example)
  - (a).PLL\_CPUX from close to open:
    - Write 0 to the bit29(Lock enable bit) of **PLL\_CPUX\_CTRL\_REG**.
    - Configure the parameters (**N,M,P**) of **PLL\_CPUX\_CTRL\_REG**.
    - Write 1 to the bit31(enable bit) of **PLL\_CPUX\_CTRL\_REG**.
    - Write 1 to the bit29(lock enable bit) of **PLL\_CPUX\_CTRL\_REG**.
    - Read the bit28 of **PLL\_CPUX\_CTRL\_REG**, when it is 1, then the CPUX PLL is locked.
    - Delay 20us.
  - (b).PLL\_CPUX frequency conversion:
    - Write 0 to the bit29(Lock enable bit) of **PLL\_CPUX\_CTRL\_REG**.
    - Configure the parameters (**N,M,P**) of **PLL\_CPUX\_CTRL\_REG**.
    - Write 1 to the bit29(Lock enable bit) of **PLL\_CPUX\_CTRL\_REG**.
    - Read the bit28 of **PLL\_CPUX\_CTRL\_REG**, when it is 1, then CPUX PLL is locked.
    - Delay 20us.
  - (c).PLL\_CPUX from open to close:
    - Write 0 to the bit31(PLL enable bit) of **PLL\_CPUX\_CTRL\_REG**.
    - Write 0 to the bit29(Lock enable bit) of **PLL\_CPUX\_CTRL\_REG**.



#### NOTE

**Parameter definition of PLL formula:** M、M0、M1、P is divider factor, N is multiplier factor, P changes exponentially, other factors change by adding 1.

#### 3.3.3.2. BUS

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

#### 3.3.3.3. Clock Switch

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

### 3.3.3.4. Gating and Reset

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

### 3.3.3.5. Spread Spectrum Function

#### 3.3.3.5.1. Configuration Process

The configuration of spread spectrum follows the following steps.

##### Step1: Configure PLL\_CTRL Register

- According to PLL frequency and PLL frequency formula, calculate factor N and decimal value X, and write M0, M1、N and PLL frequency to the PLL\_CTRL register.
- Configure the SDM\_Enable bit(bit24) of the PLL\_CTRL register to 1 to enable spread spectrum function.

##### Step 2: Configure PLL\_PAT Register

- According to decimal value X and spread spectrum frequency PREQ(the bit[18:17] of the PLL\_PAT register), calculate SDM\_BOT and WAVE\_STEP of PLL\_PAT register.
- Configure spread spectrum mode(SPR\_FREQ\_MODE) to 2 or 3.
- Configure the spread spectrum clock source select bit(SDM\_CLK\_SEL) to 0 by default. But if the PLL\_INPUT\_DIV\_M1 bit of the PLL\_CTRL register is 1, the bit should set to 1.
- Write SDM\_BOT、WAVE\_STEP、PREQ、SPR\_FREQ\_MODE and SDM\_CLK\_SEL to the PLL\_PAT register, and configure the SIG\_DELT\_PAT\_EN bit(bit31) of this register to 1.

##### Step 3: Delay 20us

#### 3.3.3.5.2. Configuration Instruction

- (1).The spread spectrum is an additional low frequency periodic frequency at the desired frequency point, which is not related with the configuration parameters(N, P, M1, M0) of the frequency point.
- (2).The parameter M0, P is not related with the configuration of spread spectrum, only used for frequency division configuration.

#### 3.3.3.5.3. Frequency Calculation Formula

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

Where,

P is the frequency division factor of module or PLL;  
M0 is the post-frequency division factor of PLL;  
M1 is the pre-frequency division factor of PLL;  
N is the frequency doubling factor of PLL;  
X is the amplitude coefficient of spread spectrum.



#### NOTE

Having different PLL calculate formula for different PLL, please refer to each **PLL\_CTRL** register.

If selecting M1 = 0, M0 = 0, P = 1(no frequency division)

Then the above formula can be simplified:

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

- $f_1 \sim f_2 = [N + 1 + (X_1 \sim X_2)] \cdot 24\text{MHz}$
- $\text{SDM\_BOT} = 2^{17} \cdot X_1$
- $\text{WAVE\_STEP} = 2^{17} \cdot (X_2 - X_1) / (24\text{M}/\text{PREQ}) * 2, 0 < X < 1$

PREQ is the frequency of spread spectrum.



#### NOTE

For decimal frequency division, the formula can be referenced by it, at this time X1 is equal to X2.

#### 3.3.3.5.4. Configuration Example

Configure 605.3 MHz ~ 609.7 MHz

M0 = 1,

$$N + 1 + (X_1 \sim X_2) = (605.3 \sim 609.7) / 24 = [600 + (5.3 \sim 9.7)] / 24 = 24 + 1 + (5.3/24 \sim 9.7/24)$$

Calculate to get the following values:

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

$$\text{SDM\_BOT} = 2^{17} * X_1 = 0x7111$$

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

If M0 = 1, P = 1, then total frequency division factor is  $(M0 + 1) * 1 = 2$ ,

So the actual output frequency of PLL is 1212.1 MHz ~ 1219.4 MHz.

Calculate to get the following values:

$$N = 49, X_1 = 12.1/24, X_2 = 19.4/24$$

Then calculate based on SDM\_BOT and WAVE\_STEP formula.

#### 3.3.4. Register List

##### CCU part:

Module Name	Base Address
CCU	0x03001000

Register Name	Offset	Description
PLL_CPUX_CTRL_REG	0x0000	PLL_CPUX Control Register
PLL_DDR_CTRL_REG	0x0010	PLL_DDR Control Register
PLL_PERIO_CTRL_REG	0x0020	PLL_PERIO Control Register
PLL_PERI1_CTRL_REG	0x0028	PLL_PERI1 Control Register
PLL_AUDIO_CTRL_REG	0x0078	PLL_AUDIO Control Register
PLL_32K_CTRL_REG	0x00D8	PLL_32K Control Register
PLL_DDR_PAT_CTRL_REG	0x0110	PLL_DDR Pattern Control Register
PLL_PERIO_PATO_CTRL_REG	0x0120	PLL_PERIO Pattern0 Control Register
PLL_PERIO_PAT1_CTRL_REG	0x0124	PLL_PERIO Pattern1 Control Register
PLL_PERI1_PATO_CTRL_REG	0x0128	PLL_PERI1 Pattern0 Control Register
PLL_PERI1_PAT1_CTRL_REG	0x012C	PLL_PERI1 Pattern1 Control Register
PLL_AUDIO_PATO_CTRL_REG	0x0178	PLL_AUDIO Pattern0 Control Register
PLL_AUDIO_PAT1_CTRL_REG	0x017C	PLL_AUDIO Pattern1 Control Register
PLL_32K_PATO_CTRL_REG	0x01D8	PLL_32K Pattern0 Control Register
PLL_32K_PAT1_CTRL_REG	0x01DC	PLL_32K Pattern1 Control Register
PLL_CPUX_BIAS_REG	0x0300	PLL_CPUX Bias Register
PLL_DDR_BIAS_REG	0x0310	PLL_DDR Bias Register
PLL_PERIO_BIAS_REG	0x0320	PLL_PERIO Bias Register
PLL_PERI1_BIAS_REG	0x0328	PLL_PERI1 Bias Register
PLL_AUDIO_BIAS_REG	0x0378	PLL_AUDIO Bias Register
PLL_32K_BIAS_REG	0x03D8	PLL_32K Bias Register
PLL_CPUX_TUN_REG	0x0400	PLL_CPUX Tuning Register
CO_CPUX_AXI_CFG_REG	0x0500	CPUX_AXI Configuration Register
PSI_AHB1_AHB2_CFG_REG	0x0510	PSI_AHB1_AHB2 Configuration Register
AHB3_CFG_REG	0x051C	AHB3 Configuration Register
APB1_CFG_REG	0x0520	APB1 Configuration Register
APB2_CFG_REG	0x0524	APB2 Configuration Register
CE_CLK_REG	0x0680	CE Clock Register
CE_BGR_REG	0x068C	CE Bus Gating Reset Register
DMA_BGR_REG	0x070C	DMA Bus Gating Reset Register
HSTIMER_BGR_REG	0x073C	HSTIMER Bus Gating Reset Register
AVS_CLK_REG	0x0740	AVS Clock Register
DBGSYS_BGR_REG	0x078C	DBGSYS Bus Gating Reset Register
PSI_BGR_REG	0x079C	PSI Bus Gating Reset Register
PWM_BGR_REG	0x07AC	PWM Bus Gating Reset Register
DRAM_CLK_REG	0x0800	DRAM Clock Register
MBUS_MAT_CLK_GATING_REG	0x0804	MBUS Master Clock Gating Register
DRAM_BGR_REG	0x080C	DRAM Bus Gating Reset Register

SMHC1_CLK_REG	0x0834	SMHC1 Clock Register
SMHC_BGR_REG	0x084C	SMHC Bus Gating Reset Register
UART_BGR_REG	0x090C	UART Bus Gating Reset Register
TWI_BGR_REG	0x091C	TWI Bus Gating Reset Register
SPI0_CLK_REG	0x0940	SPI0 Clock Register
SPI1_CLK_REG	0x0944	SPI1 Clock Register
SPI_BGR_REG	0x096C	SPI Bus Gating Reset Register
GPADC_BGR_REG	0x09EC	GPADC Bus Gating Reset Register
THS_BGR_REG	0x09FC	THS Bus Gating Reset Register
I2S/PCM0_CLK_REG	0x0A10	I2S/PCM0 Clock Register
I2S/PCM1_CLK_REG	0x0A14	I2S/PCM1 Clock Register
I2S/PCM2_CLK_REG	0x0A18	I2S/PCM2 Clock Register
I2S/PCM_BGR_REG	0x0A1C	I2S/PCM Bus Gating Reset Register
OWA_CLK_REG	0x0A20	OWA Clock Register
OWA_BGR_REG	0x0A2C	OWA Bus Gating Reset Register
DMIC_CLK_REG	0x0A40	DMIC Clock Register
DMIC_BGR_REG	0x0A4C	DMIC Bus Gating Reset Register
AUDIO_CODEC_1X_CLK_REG	0x0A50	AUDIO CODEC 1X Clock Register
AUDIO_CODEC_4X_CLK_REG	0x0A54	AUDIO CODEC 4X Clock Register
AUDIO_CODEC_BGR_REG	0x0A5C	AUDIO CODEC Bus Gating Reset Register
USBO_CLK_REG	0x0A70	USBO Clock Register
USB_BGR_REG	0x0A8C	USB Bus Gating Reset Register
VAD_BGR_REG	0x0ACC	VAD Bus Gating Reset Register
LPSD_CLK_REG	0x0AD0	LPSD Clock Register
LPSD_BGR_REG	0x0ADC	LPSD Bus Gating Reset Register
LEDC_CLK_REG	0x0BF0	LEDC Clock Register
LEDC_BGR_REG	0x0BFC	LEDC BUS GATING RESET Register
CCU_SEC_SWITCH_REG	0x0F00	CCU Security Switch Register
PLL_LOCK_DBG_CTRL_REG	0x0F04	PLL Lock Debug Control Register
PLL_CPUX_HW_FM_REG	0x0F20	PLL_CPUX Hardware FM Register
MOD_SPE_CLK_REG	0x0F30	Module Special Clock Register
HOSC_OUTPUT_CTRL_REG	0x0F40	HOSC Output Control Register

**DCXO part:**

Module Name	Base Address
DCXO	0x0700 0000

Register Name	Offset	Description
LOSC_CTRL_REG	0x0000	Low Oscillator Control Register
LOSC_AUTO_SWT_STA_REG	0x0004	LOSC Auto Switch Status Register
INTOSC_CLK_PRESCAL_REG	0x0008	Internal OSC Clock Prescalar Register

LOSC_OUT_GATING_REG	0x0060	LOSC output gating register
RTC_VER_REG	0x0070	RTC Version Register
GP_DATA_REG	0x0100+N*0x4	General Purpose Register (N=0~7)
XO_CTRL_REG	0x0160	XO Control Register
IC_CHARA_REG	0x01F0	IC Characteristic Register
CRY_CONFIG_REG	0x0210	Crypt Configuration Register
CRY_KEY_REG	0x0214	Crypt Key Register
CRY_EN_REG	0x0218	Crypt Enable Register
VDD_SYS_PWROFF_GATE_REG	0x0220	VDD_SYS Power Off Gating Register
PLL_CTRL_REG0	0x0224	PLL Control Register 0
PLL_CTRL_REG1	0x0228	PLL Control Register 1
RES_CAL_CTRL_REG	0x0230	Resistor Calibration Control Register
RES200_CTRL_REG	0x0234	200ohms Resistor Manual Control Register
RES240_CTRL_REG	0x0238	240ohms Resistor Manual Control Register
RES_CAL_STATUS_REG	0x023C	Resistor Calibration Status Register
NMI_CTRL_REG	0x0240	NMI Control Register
LDO_CTRL_REG	0x0254	LDOA and LDOB Control Register
CPU1_SOFT_ENT_REG	0x0258	Boot CPU1 Software Entry Register
EMPO_REG	0x0260	Empty0 Register
EMP1_REG	0x0264	Empty1 Register
EMP2_REG	0x0268	Empty2 Register

### 3.3.5. Register Description

CCU part:

#### 3.3.5.1. PLL\_CPUX Control Register (Default Value: 0xA00\_1000)

Offset: 0x0000			Register Name: PLL_CPUX_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable PLL_CPUX=InputFreq*N/P  <b>NOTE</b> <b>The PLL_CPUX output frequency must be in the range from 200MHz to 3GHz. When crystal oscillator is 24MHz, then the default value of PLL_CPUX is 408MHz.</b>
30	/	/	/
29	R/W	0x0	LOCK_ENABLE Lock Enable 0: Disable

			1: Enable
28	R	0x0	LOCK 0:Unlocked 1: Locked (It indicates that the PLL has been stable.)
27	R/W	0x1	PLL_OUTPUT_ENABLE 0:Disable 1:Enable The bit is used to control the output enable of PLL.
26:24	R/W	0x0	PLL_LOCK_TIME. PLL lock time The bit indicates the step amplitude from one frequency to another.
23:18	/	/	/
17:16	R/W	0x0	PLL_OUT_EXT_DIVP PLL Output External Divider P 00: 1 01: 2 10: 4 11: / When output clock is less than 288MHz, clock frequency is output by dividing P.
15:8	R/W	0x10	PLL_FACTOR_N PLL Factor N N= PLL_FACTOR_N +1 PLL_FACTOR_N is from 0 to 254. In application, PLL_FACTOR_N shall be more than or equal to 11.
7:2	/	/	/
1:0	R/W	0x0	PLL_FACTOR_M PLL Factor M M = PLL_FACTOR_M + 1 PLL_FACTOR_M is from 0 to 3.

### 3.3.5.2. PLL\_DDR Control Register (Default Value: 0x0800\_2301)

Offset: 0x0010			Register Name: PLL_DDR_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable PLL_DDR= InputFreq*N/M0/M1   <b>NOTE</b> <b>When crystal oscillator is 24MHz, and the bit23 is 0, then the default value of PLL_DDR is 432MHz.</b>
30	/	/	/

29	R/W	0x0	LOCK_ENABLE Lock Enable 0: Disable 1: Enable
28	R	0x0	LOCK 0:Unlocked 1: Locked (It indicates that the PLL has been stable.)
27	R/W	0x1	PLL_OUTPUT_ENABLE 0:Disable 1:Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_ENABLE 0:Disable 1:Enable
23:16	/	/	/
15:8	R/W	0x23	PLL_FACTOR_N PLL Factor N N= PLL_FACTOR_N +1 PLL_FACTOR_N is from 0 to 254. In application, PLL_FACTOR_N shall be more than or equal to 11.
7:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV_M1. PLL Input Div M1. M1=PLL_INPUT_DIV_M1 + 1 PLL_INPUT_DIV_M1 is from 0 to 1.
0	R/W	0x1	PLL_OUTPUT_DIV_M0. PLL Output Div M0. M0=PLL_OUTPUT_DIV_M0 + 1 PLL_OUTPUT_DIV_M0 is from 0 to 1.

### 3.3.5.3. PLL\_PERIO Control Register (Default Value: 0x0800\_3100)

Offset: 0x0020			Register Name: PLL_PERIO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable PLL_PERIO(2X) = InputFreq*N/M0/M1 PLL_PERIO(1X) = InputFreq*N/M0/M1/2  <span style="color: #ccc; font-size: 2em;">NOTE</span> <b>When crystal oscillator is 24MHz, and the bit23 is 0, then the default value of PLL_PERIO(2X) is 1.2GHz.The output clock of PLL_PERIO(2X) shall be</b>

			<b>fixed at 1.2GHz. The value is not recommended changing.</b>
30	/	/	/
29	R/W	0x0	LOCK_ENABLE Lock Enable 0: Disable 1: Enable
28	R	0x0	LOCK 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)
27	R/W	0x1	PLL_OUTPUT_ENABLE 0:Disable 1:Enable The bit is used to control the output enable of PLL.
26:24	/	/	/
23:16	/	/	/
15:8	R/W	0x1	PLL_FACTOR_N PLL Factor N N= PLL_FACTOR_N +1 PLL_FACTOR_N is from 0 to 254. In application, PLL_FACTOR_N shall be more than or equal to 11.
7:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV_M1. PLL Input Div M1. M1=PLL_INPUT_DIV_M1 + 1 PLL_INPUT_DIV_M1 is from 0 to 1.
0	R/W	0x0	PLL_OUTPUT_DIV_M0. PLL Output Div M0. M0=PLL_OUTPUT_DIV_M0 + 1 PLL_OUTPUT_DIV_M0 is from 0 to 1.

### 3.3.5.4. PLL\_PERI1 Control Register (Default Value: 0x0800\_3100)

Offset: 0x0028			Register Name: PLL_PERI1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	PLL_ENABLE 0: Disable 1: Enable PLL_PERI1(2X) = InputFreq*N/M0/M1 PLL_PERI1(1X) = InputFreq*N/M0/M1/2   <b>NOTE</b> <b>When crystal oscillator is 24MHz, and the bit23 is 0, then the default value of PLL_PERI1(2X) is 1.2GHz. The output clock of PLL_PERI1(2X) shall be fixed at 1.2GHz. The value is not recommended changing.</b>

30	/	/	/
29	R/W	0x0	<b>LOCK_ENABLE</b> Lock Enable 0: Disable 1: Enable
28	R	0x0	<b>LOCK</b> 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)
27	R/W	0x1	<b>PLL_OUTPUT_ENABLE</b> 0:Disable 1:Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	<b>PLL_SDM_ENABLE.</b> 0: Disable. 1: Enable.
23:16	/	/	/
15:8	R/W	0x1	<b>PLL_FACTOR_N</b> PLL Factor N $N = \text{PLL\_FACTOR\_N} + 1$ PLL_FACTOR_N is from 0 to 254. In application, PLL_FACTOR_N shall be more than or equal to 11.
7:2	/	/	/
1	R/W	0x0	<b>PLL_INPUT_DIV_M1.</b> PLL Input Div M1. $M1 = \text{PLL\_INPUT\_DIV\_M1} + 1$ PLL_INPUT_DIV_M1 is from 0 to 1.
0	R/W	0x0	<b>PLL_OUTPUT_DIV_M0.</b> PLL Output Div M0. $M0 = \text{PLL\_OUTPUT\_DIV\_M0} + 1$ PLL_OUTPUT_DIV_M0 is from 0 to 1.

### 3.3.5.5. PLL\_AUDIO Control Register (Default Value: 0x0814\_2A01)

Offset: 0x0078			Register Name: PLL_AUDIO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<b>PLL_ENABLE.</b> 0: Disable. 1: Enable. This PLL is for Audio. $\text{PLL\_AUDIO} = \text{InputFreq} * N / (\text{Input\_div} + 1) / (\text{Output\_div} + 1) / (P + 1)$ . $\text{PLL\_AUDIO}(2X) = \text{InputFreq} * N / (\text{Input\_div} + 1) / 4$ $\text{PLL\_AUDIO}(4X) = \text{InputFreq} * N / (\text{Input\_div} + 1) / 2$

			 <b>NOTE</b> When crystal oscillator is 24MHz, and the bit23 is 0, then the default value of PLL_AUDIO is 24.5714MHz.
30	/	/	/
29	R/W	0x0	LOCK_ENABLE Lock Enable 0: Disable 1: Enable
28	R	0x0	LOCK 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)
27	R/W	0x1	PLL_OUTPUT_ENABLE 0: Disable 1: Enable The bit is used to control the output enable of PLL.
26:25	/	/	/
24	R/W	0x0	PLL_SDM_ENABLE. 0: Disable. 1: Enable.
23:22	/	/	/
21:16	R/W	0x14	PLL_POST_DIV_P. PLL Post-div P. P= PLL_POST_DIV_P +1 PLL_POST_DIV_P is from 0 to 63.
15:8	R/W	0x2A	PLL_FACTOR_N PLL Factor N. N= PLL_FACTOR_N +1 PLL_FACTOR_N is from 0 to 254. In application, PLL_FACTOR_N shall be more than or equal to 11.
7:2	/	/	/
1	R/W	0x0	PLL_INPUT_DIV_M1. PLL Input Div M1. M1=PLL_INPUT_DIV_M1 + 1 PLL_INPUT_DIV_M1 is from 0 to 1.
0	R/W	0x1	PLL_OUTPUT_DIV_M0. PLL Output Div M0. M0=PLL_OUTPUT_DIV_M0 + 1 PLL_OUTPUT_DIV_M0 is from 0 to 1.

**3.3.5.6. PLL\_32K Control Register (Default Value: 0x0814\_2A01)**

Offset: 0x00D8	Register Name: PLL_32K_CTRL_REG
----------------	---------------------------------

Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>PLL_ENABLE 0: Disable 1: Enable This PLL is for Audio.</p> <p><math>\text{PLL\_32K}(24.576\text{MHz}) = \text{InputFreq} * N / (\text{Input\_div} + 1) / (\text{Output\_div} + 1) / (P + 1)</math>.</p> <p> <b>NOTE</b></p> <p><b>When crystal oscillator is 24MHz, and the bit23 is 0, then the default value of PLL_32K is 24.5714MHz.</b></p>
30	/	/	/
29	R/W	0x0	<p>LOCK_ENABLE Lock Enable 0: Disable 1: Enable</p>
28	R	0x0	<p>LOCK 0: Unlocked 1: Locked (It indicates that the PLL has been stable.)</p>
27	R/W	0x1	<p>PLL_OUTPUT_ENABLE 0: Disable 1: Enable The bit is used to control the output enable of PLL.</p>
26:25	/	/	/
24	R/W	0x0	<p>PLL_SDM_ENABLE. 0: Disable 1: Enable</p>
23:22	/	/	/
21:16	R/W	0x14	<p>PLL_POST_DIV_P. PLL Post-div P. <math>P = \text{PLL\_POST\_DIV\_P} + 1</math> PLL_POST_DIV_P is from 0 to 63.</p>
15:8	R/W	0x2A	<p>PLL_FACTOR_N PLL Factor N. <math>N = \text{PLL\_FACTOR\_N} + 1</math> PLL_FACTOR_N is from 0 to 254. In application, PLL_FACTOR_N shall be more than or equal to 11.</p>
7:2	/	/	/
1	R/W	0x0	<p>PLL_INPUT_DIV_M1. PLL Input Div M1. <math>M1 = \text{PLL\_INPUT\_DIV\_M1} + 1</math> PLL_INPUT_DIV_M1 is from 0 to 1.</p>
0	R/W	0x1	<p>PLL_OUTPUT_DIV_M0. PLL Output Div M0. <math>M0 = \text{PLL\_OUTPUT\_DIV\_M0} + 1</math> PLL_OUTPUT_DIV_M0 is from 0 to 1.</p>

### 3.3.5.7. PLL\_DDR Pattern Control Register (Default Value: 0x0000\_0000)

Offset: 0x0110			Register Name: PLL_DDR_PAT_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SIG_DELT_PAT_EN Sigma-Delta Pattern Enable
30:29	R/W	0x0	SPR_FREQ_MODE Spread Frequency Mode 00: DC=0 01: DC=1 10: Triangular(1bit) 11: Triangular(nbit)
28:20	R/W	0x0	WAVE_STEP Wave Step
19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz  <b>NOTE</b> When PLL_INPUT_DIV_M1 is 1, the bit is set to 1.
18:17	R/W	0x0	FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT Wave Bottom

### 3.3.5.8. PLL\_PERIO Pattern0 Control Register (Default Value: 0x0000\_0000)

Offset: 0x0120			Register Name: PLL_PERIO_PAT0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SIG_DELT_PAT_EN Sigma-Delta Pattern Enable
30:29	R/W	0x0	SPR_FREQ_MODE. Spread Frequency Mode. 00: DC=0 01: DC=1 10: Triangular(1bit) 11: Triangular(nbit)

28:20	R/W	0x0	WAVE_STEP. Wave Step.
19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz   <b>NOTE</b> When PLL_INPUT_DIV_M1 is 1, the bit is set to 1.
18:17	R/W	0x0	FREQ. Frequency. 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT. Wave Bottom.

### 3.3.5.9. PLL\_PERIO Pattern1 Control Register (Default Value: 0x0000\_0000)

Offset: 0x0124			Register Name: PLL_PERIO_PAT1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	DITHER_EN.
23:21	/	/	/
20	R/W	0x0	FRAC_EN.
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN.

### 3.3.5.10. PLL\_PERI1 Pattern0 Control Register (Default Value: 0x0000\_0000)

Offset: 0x0128			Register Name: PLL_PERI1_PAT0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SIG_DELT_PAT_EN Sigma-Delta Pattern Enable
30:29	R/W	0x0	SPR_FREQ_MODE Spread Frequency Mode 00: DC=0 01: DC=1 10: Triangular(1 bit) 11: Triangular(n bit)
28:20	R/W	0x0	WAVE_STEP

Wave Step			
19	R/W	0x0	<p>SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz</p> <p> <b>NOTE</b> <b>When PLL_INPUT_DIV_M1 is 1, the bit is set to 1.</b></p>
18:17	R/W	0x0	<p>FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz</p>
16:0	R/W	0x0	<p>WAVE_BOT Wave Bottom</p>

### 3.3.5.11. PLL\_PERI1 Pattern1 Control Register (Default Value: 0x0000\_0000)

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

### 3.3.5.12. PLL\_AUDIO Pattern0 Control Register (Default Value: 0x0000\_0000)

Offset: 0x0178			Register Name: PLL_AUDIO_PAT0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SIG_DELT_PAT_EN Sigma-Delta Pattern Enable
30:29	R/W	0x0	<p>SPR_FREQ_MODE Spread Frequency Mode 00: DC=0 01: DC=1 10: Triangular(1 bit) 11: Triangular(n bit)</p>
28:20	R/W	0x0	<p>WAVE_STEP Wave Step</p>

19	R/W	0x0	SDM_CLK_SEL SDM Clock Select 0: 24 MHz 1: 12 MHz  <b>NOTE</b> <b>When PLL_INPUT_DIV_M1 is 1, the bit is set to 1.</b>
18:17	R/W	0x0	FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz
16:0	R/W	0x0	WAVE_BOT Wave Bottom

### 3.3.5.13. PLL\_AUDIO Pattern1 Control Register (Default Value: 0x0000\_0000)

Offset: 0x017C			Register Name: PLL_AUDIO_PAT1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	DITHER_EN.
23:21	/	/	/
20	R/W	0x0	FRAC_EN.
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN.

### 3.3.5.14. PLL\_32K Pattern0 Control Register (Default Value: 0x0000\_0000)

Offset: 0x01D8			Register Name: PLL_32K_PAT0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SIG_DELT_PAT_EN Sigma-Delta Pattern Enable
30:29	R/W	0x0	SPR_FREQ_MODE Spread Frequency Mode 00: DC=0 01: DC=1 10: Triangular(1 bit) 11: Triangular(n bit)
28:20	R/W	0x0	WAVE_STEP Wave Step
19	R/W	0x0	SDM_CLK_SEL

			<p>SDM Clock Select 0: 24 MHz 1: 12 MHz</p> <p> <b>NOTE</b> When PLL_INPUT_DIV_M1 is 1, the bit is set to 1.</p>
18:17	R/W	0x0	<p>FREQ Frequency 00: 31.5 kHz 01: 32 kHz 10: 32.5 kHz 11: 33 kHz</p>
16:0	R/W	0x0	<p>WAVE_BOT Wave Bottom</p>

### 3.3.5.15. PLL\_32K Pattern1 Control Register (Default Value: 0x0000\_0000)

Offset: 0x01DC			Register Name: PLL_32K_PAT1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	DITHER_EN.
23:21	/	/	/
20	R/W	0x0	FRAC_EN.
19:17	/	/	/
16:0	R/W	0x0	FRAC_IN.

### 3.3.5.16. PLL\_CPUX Bias Register (Default Value: 0x8010\_0000)

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

### 3.3.5.17. PLL\_DDR Bias Register (Default Value: 0x0003\_0000)

Offset: 0x0310			Register Name: PLL_DDR_BIAS_REG
Bit	Read/Write	Default/Hex	Description

31:21	/	/	/
20:16	R/W	0x3	PLL_BIAS_CTRL. PLL bias control [4:0].
15:0	/	/	/

### 3.3.5.18. PLL\_PERIO Bias Register (Default Value: 0x0003\_0000)

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

### 3.3.5.19. PLL\_PERI1 Bias Register (Default Value: 0x0003\_0000)

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

### 3.3.5.20. PLL\_AUDIO Bias Register (Default Value: 0x0003\_0000)

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

### 3.3.5.21. PLL\_32K Bias Register (Default Value: 0x0003\_0000)

Offset: 0x03D8			Register Name: PLL_32K_BIAS_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20:16	R/W	0x3	PLL_BIAS_CTRL. PLL bias control [4:0].
15:0	/	/	/

### 3.3.5.22. PLL\_CPUX Tuning Register (Default Value: 0x4440\_4000)

Offset: 0x0400			Register Name: PLL_CPUX_TUN_REG
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30:28	R/W	0x4	VCO_RNG_CTRL. VCO range control [2:0].
27	/	/	/
26:24	R/W	0x4	KVCO_GAIN_CTRL. KVCO gain control [2:0].
23	/	/	/
22:16	R/W	0x40	CNT_INIT_CTRL. Counter initial control [6:0].
15	R/W	0x0	C_OD0. C-REG-OD0 for verify.
14:8	R/W	0x40	C_B_IN. C-B-IN [6:0] for verify.
7	R/W	0x0	C_OD1. C-REG-OD1 for verify.
6:0	RO	0x0	C_B_OUT. C-B-OUT [6:0] for verify.

### 3.3.5.23. CPUX\_AXI Configuration Register (Default Value: 0x0000\_0301)

Offset: 0x0500			Register Name: C0_CPUX_AXI_CFG_REG
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL. Clock Source Select 000: OSC24M 001: RTC_32K 010: RC16M 011: PLL_CPUX 100: PLL_PERIO(1X) 101: reserved 110: reserved 111: reserved
23:10	/	/	/
9:8	R/W	0x3	CPUX_APB_FACTOR_N. 0:/1 1:/2

			2:/4 3:/
7:2	/	/	/
1:0	R/W	0x1	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 3

### 3.3.5.24. PSI\_AHB1\_AHB2 Configuration Register (Default Value: 0x0000\_0000)

Offset: 0x0510			Register Name: PSI_AHB1_AHB2_CFG_REG
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL. Clock Source Select 00: OSC24M 01: RTC_32K 10: RC16M 11: PLL_PERIO(1X) PSI_AHB1_AHB2 CLK = Clock Source/M/N.
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N. Factor N 00: 1 01: 2 10: 4 11: 8
7:2	/	/	/
1:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 3.

### 3.3.5.25. AHB3 Configuration Register (Default Value: 0x0000\_0000)

Offset: 0x051C			Register Name: AHB3_CFG_REG
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL. Clock Source Select 00: OSC24M 01: RTC_32K 10: PSI 11: PLL_PERIO(1X)

			AHB3 CLK = Clock Source/M/N.
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N. Factor N 00: 1 01: 2 10: 4 11: 8
7:2	/	/	/
1:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 3.

### 3.3.5.26. APB1 Configuration Register (Default Value: 0x0000\_0000)

Offset: 0x520			Register Name: APB1_CFG_REG
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL. Clock Source Select 00: OSC24M 01: RTC_32K 10: PSI 11: PLL_PERIO(1X) APB1 CLK = Clock Source/M/N.
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N. Factor N 00: 1 01: 2 10: 4 11: 8
7:2	/	/	/
1:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 3.

### 3.3.5.27. APB2 Configuration Register (Default Value: 0x0000\_0000)

Offset: 0x524			Register Name: APB2_CFG_REG
Bit	Read/Write	Default/Hex	Description
31:26	/	/	/

25:24	R/W	0x0	CLK_SRC_SEL. Clock Source Select 00: OSC24M 01: RTC_32K 10: PSI 11: PLL_PERIO(1X) APB2 CLK = Clock Source/M/N.
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N. Factor N 00: 1 01: 2 10: 4 11: 8
7:2	/	/	/
1:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 3.

### 3.3.5.28. MBUS Configuration Register (Default Value: 0xC000\_0000)

Offset: 0x540			Register Name: MBUS_CFG_REG
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30	R/W	0x1	MBUS_RST MBUS Reset 0: Assert 1: De-assert
29:0	/	/	/

### 3.3.5.29. CE Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0680			Register Name: CE_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:25	/	/	/
24	R/W	0x0	CLK_SRC_SEL Clock Source Select

			0: OSC24M 1: PLL_PERIO(2X)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 15.

### 3.3.5.30. CE Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x068C			Register Name: CE_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	CE_RST CE Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	CE_GATING Gating Clock for CE 0: Mask 1: Pass

### 3.3.5.31. DMA Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x070C			Register Name: DMA_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	DMA_RST DMA Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	DMA_GATING Gating Clock for DMA 0: Mask

			1: Pass
--	--	--	---------

### 3.3.5.32. HSTIMER Bus Gating Reset Register (Default Value: 0x0000\_0000)

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

### 3.3.5.33. AVS Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0740			Register Name: AVS_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = OSC24M.
30:0	/	/	/

### 3.3.5.34. DBGSYS Bus Gating Reset Register (Default Value: 0x0000\_0000)

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

			1: Pass
--	--	--	---------

### 3.3.5.35. PSI Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x079C			Register Name: PSI_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	PSI_RST PSI Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	PSI_GATING Gating Clock for PSI 0: Mask 1: Pass

### 3.3.5.36. PWM Bus Gating Reset Register (Default Value: 0x0000\_0000)

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

### 3.3.5.37. DRAM Clock Register (Default Value: 0x0100\_0000)

Offset: 0x0800			Register Name: DRAM_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x1	Gating Special Clock 0: Clock is OFF 1: Clock is ON
30	R/W	0x0	MODULE_RST Module Reset

			0: Assert 1: De-assert SCLK = Clock Source/M.
29:28	/	/	/
27	R/WAC	0x0	SDRCLK_UPD SDRCLK Configuration 0 Update 0:Invalid 1:Valid  <b>NOTE</b> <b>Setting this bit will validate Configuration 0. It will be auto cleared after the Configuration 0 is valid.</b>
26	/	/	/
25:24	R/W	0x1	CLK_SRC_SEL Clock Source Select 00: HOSC 01: PLL_DDR Others: /
23:2	/	/	/
1:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 3.

### 3.3.5.38. MBUS Master Clock Gating Register (Default Value: 0x0000\_0000)

Offset: 0x0804			Register Name: MBUS_MAT_CLK_GATING_REG
Bit	Read/Write	Default/Hex	Description
31:6	/	/	/
5	R/W	0x0	Reserved
4:3	/	/	/
2	R/W	0x0	CE_MCLK_GATING. Gating MBUS Clock For CE 0: Mask 1: Pass
1	/	/	/
0	R/W	0x0	DMA_MCLK_GATING. Gating MBUS Clock For DMA 0: Mask 1: Pass

**3.3.5.39. DRAM Bus Gating Reset Register (Default Value: 0x0000\_0000)**

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

**3.3.5.40. SMHC1 Clock Register (Default Value: 0x0000\_0000)**

Offset: 0x0834			Register Name: SMHC1_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: OSC24M 01: PLL_PERIO(2X) 10: PLL_PERI1(2X) 11: /
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 15.

### 3.3.5.41. SMHC Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x084C			Register Name: SMHC_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/
18	R/W	0x0	Reserved
17	R/W	0x0	SMHC1_RST. SMHC1 Reset. 0: Assert 1: De-assert
16	R/W	0x0	Reserved
15:3	/	/	/
2	R/W	0x0	Reserved
1	R/W	0x0	SMHC1_GATING. Gating Clock For SMHC1 0: Mask 1: Pass
0	R/W	0x0	Reserved

### 3.3.5.42. UART Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x090C			Register Name: UART_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:20	/	/	/
19	R/W	0x0	UART3_RST UART3 Reset 0: Assert 1: De-assert
18	R/W	0x0	UART2_RST UART2 Reset 0: Assert 1: De-assert
17	R/W	0x0	UART1_RST UART1 Reset 0: Assert 1: De-assert
16	R/W	0x0	UART0_RST UART0 Reset 0: Assert 1: De-assert
15:4	/	/	/

3	R/W	0x0	UART3_GATING Gating Clock for UART3 0: Mask 1: Pass
2	R/W	0x0	UART2_GATING Gating Clock for UART2 0: Mask 1: Pass
1	R/W	0x0	UART1_GATING Gating Clock for UART1 0: Mask 1: Pass
0	R/W	0x0	UART0_GATING Gating Clock for UART0 0: Mask 1: Pass

### 3.3.5.43. TWI Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x091C			Register Name: TWI_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/
18	R/W	0x0	TWI2_RST TWI2 Reset 0: Assert 1: De-assert
17	R/W	0x0	TWI1_RST TWI1 Reset 0: Assert 1: De-assert
16	R/W	0x0	TWI0_RST TWI0 Reset 0: Assert 1: De-assert
15:3	/	/	/
2	R/W	0x0	TWI2_GATING Gating Clock for TWI2 0: Mask 1: Pass
1	R/W	0x0	TWI1_GATING Gating Clock for TWI1 0: Mask 1: Pass
0	R/W	0x0	TWI0_GATING

			Gating Clock for TWI0 0: Mask 1: Pass
--	--	--	---

### 3.3.5.44. SPI0 Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0940			Register Name: SPI0_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M/N.
30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: OSC24M 001: PLL_PERIO(1X) 010: PLL_PERI1(1X) 011: PLL_PERIO(2X) 100: PLL_PERI1(2X) 1XX: /
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 15.

### 3.3.5.45. SPI1 Clock Register (Default Value: 0x0000\_0000)

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

30:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 000: OSC24M 001: PLL_PERI0(1X) 010: PLL_PERI1(1X) 011: PLL_PERI0(2X) 100: PLL_PERI1(2X) 1XX: /
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 15.

### 3.3.5.46. SPI Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x096C			Register Name: SPI_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R/W	0x0	SPI1_RST SPI1 Reset 0: Assert 1: De-assert
16	R/W	0x0	SPI0_RST SPI0 Reset 0: Assert 1: De-assert
15:2	/	/	/
1	R/W	0x0	SPI1_GATING Gating Clock for SPI1 0: Mask 1: Pass
0	R/W	0x0	SPI0_GATING Gating Clock for SPI0 0: Mask 1: Pass

### 3.3.5.47. GPADC Bus Gating Reset Register (Default Value: 0x0000\_0000)

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

### 3.3.5.48. THS Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x09FC			Register Name: THS_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	THS_RST THS Reset 0: Assert 1: De-assert
15:1	/	/	/
0	R/W	0x0	THS_GATING Gating Clock for THS 0: Mask 1: Pass

### 3.3.5.49. I2S/PCM0 Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0A10			Register Name: I2S/PCM0_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/N
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select

			00: PLL_AUDIO(4X) 01: PLL_32K(24.576MHz) 10:/ 11:/
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:0	/	/	/

### 3.3.5.50. I2S/PCM1 Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0A14			Register Name: I2S/PCM1_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/N.
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO(4X) 01: PLL_32K(24.576MHz) 10:/ 11:/
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:0	/	/	/

### 3.3.5.51. I2S/PCM2 Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0A18			Register Name: I2S/PCM2_CLK_REG
Bit	Read/Write	Default/Hex	Description

31	R/W	0x0	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/N.
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO(4X) 01: PLL_32K(24.576MHz) 10:/ 11:/
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:0	/	/	/

### 3.3.5.52. I2S/PCM Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x0A1C			Register Name: I2S/PCM_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/
18	R/W	0x0	I2S/PCM2_RST I2S/PCM2 Reset 0: Assert 1: De-assert
17	R/W	0x0	I2S/PCM1_RST I2S/PCM1 Reset 0: Assert 1: De-assert
16	R/W	0x0	I2S/PCMO_RST I2S/PCMO Reset 0: Assert 1: De-assert
15:3	/	/	/
2	R/W	0x0	I2S/PCM2_GATING Gating Clock for I2S/PCM2 0: Mask 1: Pass

1	R/W	0x0	I2S/PCM1_GATING Gating Clock for I2S/PCM1 0: Mask 1: Pass
0	R/W	0x0	I2S/PCM0_GATING Gating Clock for I2S/PCM0 0: Mask 1: Pass

### 3.3.5.53. OWA Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0A20			Register Name: OWA_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	OWA_SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/N.
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO(4X) 01: PLL_32K(24.576MHz) 10:/ 11:/
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:0	/	/	/

### 3.3.5.54. OWA Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x0A2C			Register Name: OWA_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R/W	0x0	OWA_RST OWA Reset 0: Assert

			1: De-assert
15:1	/	/	/
0	R/W	0x0	OWA_GATING Gating Clock For OWA 0: Mask 1: Pass

### 3.3.5.55. DMIC Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0A40			Register Name: DMIC_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/N.
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO(4X) 01: PLL_32K(24.576MHz) 10:/ 11:/
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N Factor N 00: 1 01: 2 10: 4 11: 8
7:0	/	/	/

### 3.3.5.56. DMIC Bus Gating Reset Register (Default Value: 0x0000\_0000)

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

0	R/W	0x0	DMIC_GATING Gating Clock for DMIC 0: Mask 1: Pass
---	-----	-----	--

### 3.3.5.57. AUDIO CODEC 1X Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0A50			Register Name: AUDIO_CODEC_1X_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M.
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO(4X) 01: PLL_32K(24.576MHz) 10:/ 11:/
23:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 15.

### 3.3.5.58. AUDIO CODEC 4X Clock Register (Default Value: 0x0000\_0000)

Offset: 0x0A54			Register Name: AUDIO_CODEC_4X_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SCLK_GATING Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source/M.
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL Clock Source Select 00: PLL_AUDIO(4X) 01: PLL_32K(24.576MHz) 10:/ 11:/

23:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 15.

### 3.3.5.59. AUDIO CODEC Bus Gating Reset Register (Default Value: 0x0000\_0000)

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

### 3.3.5.60. USB0 Clock Register (Default Value: 0x0003\_0000)

Offset: 0x0A70			Register Name: USB0_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SCLK_GATING_OHCI0. Gating Special Clock For OHCI0 0: Clock is OFF 1: Clock is ON
30	R/W	0x0	USBPHY0_RST. USB PHY0 Reset 0: Assert 1: De-assert
29	R/W	0x0	SCLK_GATING_USBPHY0. Gating Special Clock For USBPHY0 0: Clock is OFF 1: Clock is ON SCLK is from OSC24M
28:26	/	/	/
25:24	R/W	0x0	OHCIO_12M_SRC_SEL OHCIO 12M Source Select 00: 12M divided from OHCIO_48M 01: 12M divided from HOSC

			10: LOSC 11: RC16M
23:18	/	/	/
17	R/W	0x0	PYH_SCLK_SRC_SEL 0:HOSC 1:/
16	R/W	0x1	PYH_SCLK_CLK_DIV_Bypass 0:Div2 1:Bypass
15:0	/	/	/

### 3.3.5.61. USB Bus Gating Reset Register (Default Value: 0x0000\_0000)

Offset: 0x0A8C			Register Name: USB_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	USBOTG_RST. USBOTG Reset. 0: Assert 1: De-assert
23:22	/	/	/
21	R/W	0x0	Reserved
20	R/W	0x0	USBEHCI0_RST. USBEHCI0 Reset. 0: Assert 1: De-assert
19:18	/	/	/
17	R/W	0x0	Reserved
16	R/W	0x0	USBOHCI0_RST. USBOHCI0 Reset. 0: Assert 1: De-assert
15:9	/	/	/
8	R/W	0x0	USBOTG_GATING. Gating Clock For USBOTG 0: Mask 1: Pass
7:6	/	/	/
5	R/W	0x0	Reserved
4	R/W	0x0	USBEHCI0_GATING. Gating Clock For USBEHCI0 0: Mask 1: Pass

3:2	/	/	/
1	R/W	0x0	Reserved
0	R/W	0x0	USBOHCI0_GATING. Gating Clock For USBOHCI0 0: Mask 1: Pass

### 3.3.5.62. VAD Bus Gating Reset Register (Default: 0x0000\_0000)

Offset: 0x0ACC			Register Name: VAD_BGR_REG
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/
18	R/W	0x0	VAD_CFG_RST. VAD Reset. 0: Assert 1: De-assert
17	R/W	0x0	VAD_AD_RST. VAD Reset. 0: Assert 1: De-assert
16	R/W	0x0	VAD_RST. VAD Reset. 0: Assert 1: De-assert
15:3	/	/	/
2	R/W	0x0	VAD_CFG_GATING. Gating Clock For VAD 0: Mask 1: Pass
1	R/W	0x0	VAD_AD_GATING. Gating Clock For VAD 0: Mask 1: Pass
0	R/W	0x0	VAD_GATING. Gating Clock For VAD 0: Mask 1: Pass

### 3.3.5.63. LPSD Clock Register(Default: 0x0000\_0000)

Offset: 0x0AD0			Register Name: LPSD_CLK_REG
Bit	Read/Write	Default/Hex	Description

31	R/W	0x0	SCLK_GATING. Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source
30:26	/	/	/
25:24	R/W	0x0	CLK_SRC_SEL. Clock Source Select 00: OSC24M 01: PLL_AUDIO(1X) 10: PLL_32K(24.576MHz)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N. Factor N 00: 1 01: 2 10: 4 11: 8
7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 15

### 3.3.5.64. LEDC Clock Register(Default: 0x0000\_0000)

Offset: 0x0BF0			Register Name: LEDC_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SCLK_GATING. Gating Special Clock 0: Clock is OFF 1: Clock is ON SCLK = Clock Source
30:25	/	/	/
24	R/W	0x0	CLK_SRC_SEL. Clock Source Select 0: OSC24M 1: PLL_PERIO(1X)
23:10	/	/	/
9:8	R/W	0x0	FACTOR_N. Factor N 00: 1 01: 2 10: 4 11: 8

7:4	/	/	/
3:0	R/W	0x0	FACTOR_M Factor M.(M= FACTOR_M +1) FACTOR_M is from 0 to 15

### 3.3.5.65. LEDC Bus Gating Reset Register (Default: 0x0000\_0000)

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

### 3.3.5.66. CCU Security Switch Register (Default Value: 0x0000\_0000)

Offset: 0x0F00			Register Name: CCU_SEC_SWITCH_REG
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2	R/W	0x0	MBUS_SEC MBUS Clock Register Security 0: Secure 1: Non-secure
1	R/W	0x0	BUS_SEC Bus Relevant Registers Security 0: Secure 1: Non-secure
0	R/W	0x0	PLL_SEC PLL Relevant Registers Security 0: Secure 1: Non-secure

### 3.3.5.67. PLL Lock Control Register (Default Value: 0x0000\_0000)

Offset: 0x0F04	Register Name: PLL_LOCK_CTRL_REG
----------------	----------------------------------

Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	DBG_EN Debug Enable 0: Disable 1: Enable
30:25	/	/	/
24:20	R/W	0x0	DBG_SEL Debug Select 0000: PLL_CPUX 00001: / 00010: PLL_DDR 00011: / 00100: PLL_PERIO(2X) 00101: PLL_PERI1(2X) 00110: / 00111: / 01000: PLL_VIDEO0(4X) 01001: / 01010: / 01011: / 01100: / 01101: / 01110: / 01111: PLL_AUDIO 10000: / 10001: / 10010: / 10011: / 10100: / 10101: / 10110: / 10111: / 11000: / 11001: / 11010: / 11011: PLL_32K Others: /
19	/	/	/
18:17	R/W	0x0	UNLOCK_LEVEL Unlock Level 00: 21-29 Clock Cycles 01: 22-28 Clock Cycles 1X: 20-30 Clock Cycles
16	R/W	0x0	LOCK_LEVEL Lock Level

			0: 24-26 Clock Cycles 1: 23-27 Clock Cycles
15:0	/	/	/

### 3.3.5.68. PLL\_CPUX Hardware FM Register (Default Value: 0x0000\_0000)

Offset: 0x0F20			Register Name: 24M_CLK_OUTPUT_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>PLL_CPUX_MODE 0:Normal Mode 1:Hardware FM Mode</p> <p>The frequency of PLL_CPUX in normal mode is decided by the configuration of PLL_CPUX_CTRL_REG.</p> <p>In hardware FM mode, PLL_CPUX uses hardware FM.</p>
30	R	0x0	<p>PMU_FLAG 0:Unlock 1:Lock</p>
29:18	/	/	/
17:16	R/W	0x0	<p>PLL_OUT_EXT_DIV P PLL Output external divider P 00:1 01:2 10:4 11:/</p>
15:8	R/W	0x0	<p>PLL_FACTOR_N PLL Factor N N=PLL_FACTOR_N +1 PLL_FACTOR_N is from 0 to 254</p> <p>In application, PLL_FACTOR_N shall be more than or equal to 11.</p>
7:2	/	/	/
1:0	R/W	0x0	<p>PLL_FACTOR_M PLL Factor M M=PLL_FACTOR_M+1 PLL_FACTOR_M is from 0 to 3.</p> <p>Factor M does not influence the frequency of CPU.</p>

### 3.3.5.69. Module Special Clock Register(Default Value: 0x0007\_0000)

Offset: 0x0F30			Register Name: MOD_SPE_CLK_REG
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/

17	R/W	0x1	THS\GPADC_24M_CLK_GATING 0:Mask 1:Pass
16	R/W	0x1	ACodec_24M_CLK_GATING 0:Mask 1:Pass
15:0	/	/	/

### 3.3.5.70. HOSC Output Control Register (Default Value: 0x0000\_0002)

Offset: 0x0F40			Register Name: HOSC_OUTPUT_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
1	R/W	0x1	CLK32K_LOSC Select Signal 0: Select external 32K oscillator source 1: Select the 32.768kHz source divided by PLL_32K(24.576MHz)
0	R/W	0x0	/

DCXO part:

### 3.3.5.71. L OSC Control Register(Default Value: 0x0000\_4010)

Offset:0x0000			Register Name: L OSC_CTRL_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 and bit 1 can be written with the new value.
15	R/W	0x0	L OSC_AUTO_SWT_FUNCTION L OSC auto switch function disable 0: Disable 1: Enable
14	R/W	0x1	L OSC_AUTO_SWT_PENDING_EN L OSC auto switch pending enable 0: Disable 1: Enable
13:6	/	/	/
5	R/W	0x1	RC 16M enable
4	R/W	0x1	EXT_LOSC_EN External 32768Hz Crystal Enable 0: disable 1: enable
3:2	R/W	0x0	EXT_LOSC_GSM

			External 32768Hz Crystal GSM 00: Low 01: / 10: / 11: High
1	R/W	0x0	RTC_SRC_SEL Low Frequency Clock Source Select 0: CLK32_LOSC 1: CLK32_HOSC
0	R/W	0x0	LOSC_SRC_SEL LOSC Clock source Select. 'N' is the value of Internal OSC Clock Prescalar register. 0: Low Frequency Clock from 16M RC 1: External 32.768kHz OSC Internal OSC is about 16MHz.

### 3.3.5.72. LOSC Auto Switch Status Register (Default Value: 0x0000\_0000)

Offset:0x0004			Register Name: LOSC_AUTO_SWT_STA_REG
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2	R	0x0	EXT_LOSC_STA 0: External 32.768k OSC work normally 1: External 32.768k OSC work abnormally
0	R	0x0	LOSC_SRC_SEL_STA Checking LOSC Clock Source Status. 'N' is the value of Internal OSC Clock Prescalar register. 0: Low Frequency Clock from 16M RC 1: External 32.768kHz OSC

### 3.3.5.73. Internal OSC Clock Prescalar Register (Default Value: 0x0000\_000F)

Offset:0x0008			Register Name: INTOSC_CLK_PRESCAL_REG
Bit	Read/Write	Default/Hex	Description
31:5	/	/	/
4:0	R/W	0xF	INTOSC_32K_CLK_PRESCAL Internal OSC 32K Clock Prescalar value N.The clock output = Internal RC/32 / N. 00000: 1 00001: 2 00002: 3 ..... 11111: 32

### 3.3.5.74. LOSC Output Gating Register (Default Value: 0x0000\_0000)

Offset:0x0060			Register Name: LOSC_OUT_GATING_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	LOSC_OUT_GATING Configuration of LOSC output, and no LOSC output by default. 0: Mask LOSC output gating 1: Enable LOSC output gating

### 3.3.5.75. General Purpose Register (Default Value: 0x0000\_0000)

Offset:0x0100+N *0x4 (N=0~7)			Register Name: GP_DATA_REGn
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	GP_DATA Data [31:0].

### 3.3.5.76. XO Control Register (Default Value: 0x083F\_10F2)

Offset:0x0160			Register Name:XO_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:24	R/W	0x8	DCXO_ICTRL DCXO current control value
23	/	/	/
22:16	R/W	0x3F	DCXO_TRIM DCXO cap array value cap cell is 55ff
15:13	/	/	/
12:8	R/W	0x10	DCXO_BG DCXO bandgap output voltage
7	R/W	0x1	DCXO_LDO_INRUSHB DCXO LDO driving capacity signal, active high
6	R/W	0x1	XTAL_MODE XTAL mode enable signal, active high; 0x0 for external clk input mode. 0x1 for normal mode.
5:4	R/W	0x3	DCXO_RFCLK_ENHANCE DCXO rfclk enhance Enhance driving capacity of output OUT_RF_REFCLK, 0x0 for 5pF, 0x1 for

			10pF, 0x2 for 15pF, 0x3 for 20pF
2	/	/	/
1	R/W	0x0	DCXO_EN DCXO enable 1: enable 0: disable
0	/	/	/

### 3.3.5.77. IC Characteristic Register (Default Value: 0x0000\_0000)

Offset:0x01F0			Register Name: IC_CHARA_REG
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x0	IC_CHARA Key Field Should be written at value 0x16AA. Writing any other value in this field aborts the write operation.
15:0	R/W	0x0	ID_DATA Return 0x16aa only if the KEY_FIELD is set as 0x16aa when read those bits, otherwise return 0x0.

### 3.3.5.78. Crypt Configuration Register (Default Value: 0x0000\_0000)

Offset: 0x0210			Register Name: CRY_CONFIG_REG
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0	KEY_FIELD Key Field If you want to read or write Crypt Key Register/Crypt Enable Register , you should write 0x1689 in these bits.

### 3.3.5.79. Crypt Key Register (Default Value: 0x0000\_0000)

Offset: 0x0214			Register Name: CRY_KEY_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	CRY_KEY Crypt Key

### 3.3.5.80. Crypt Enable Register (Default Value: 0x0000\_0000)

Offset: 0x0218			Register Name: CRY_EN_REG
Bit	Read/Write	Default/Hex	Description

31:1	/	/	/
0	R/W	0x0	CRY_EN Crypt enable

### 3.3.5.81. VDD\_SYS Power Off Gating Register (Default Value: 0x0000\_0000)

Offset: 0x0220			Register Name: VDD_SYS_PWROFF_GATING_REG
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9	R/W	0x0	VDD_ADDA_OFF_GATING Gating RES Cal modules to the HV power when VDD_SYS power off. 0: Invalid 1: Valid
8:3	/	/	/
2	R/W	0x0	AVCC_A_GATING Gating the corresponding modules to the AVCC_A Power Domain when VDD_SYS power off. 0: Invalid 1: Valid This bit should be set to 1 before VDD_SYS power off while it should be set to 0 after the VDD_SYS power on.
1	R/W	0x0	DRAM_ZQ_PAD_HOLD Hold the pad of DRAM channel 0: not hold 1: hold dram ZQ Pad This bit should be set to 1 before VDD_SYS power off while it should be set to 0 after the VDD_SYS power on.
0	R/W	0x0	DRAM_CH_PAD_HOLD Hold the pad of DRAM channel 0: not hold 1: hold dram Pad This bit should be set to 1 before VDD_SYS power off while it should be set to 0 after the VDD_SYS power on.

### 3.3.5.82. PLL Control Register 0 (Default Value: 0x0000\_0007)

Offset: 0x0224			Register Name: PLL_CTRL_REG0
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0	TEST_CLK_SEL Test Clock Select 0: XTAL

			1: External Clock
23:3	/	/	/
2	R/W	0x1	GM1 XTAL Gain Control Bit1
1	R/W	0x1	GM0 XTAL Gain Control Bit0
0	R/W	0x1	PLL_BIAS_EN PLL Bias Enable 0: Disable 1: Enable

### 3.3.5.83. PLL Control Register 1 (Default Value: 0x0004\_0005)

Offset: 0x0228			Register Name: PLL_CTRL_REG1
Bit	Read/Write	Default/Hex	Description
31:24	R/W	0x0	KEY_FIELD Key Field for LDO Enable bit. If the key field value is 0xA7, the bit[23:0] can be modified.
23:19	/	/	/
18:16	R/W	0x4	PLLVDD_LDO_OUT_CTRL PLLVDD LDO Output Control 000: 0.90V 001: 0.94V 010: 0.98V 011: 1.02V 100: 1.06V 101: 1.10V 110: 1.14V 111: 1.18V
15:5	/	/	/
4	R/W	0	MBIAS_EN Chip Master Bias Enable 0:From Internal Bias 1: From ADDA Bias
3	R/W	0	PLLTEST_EN For Verify (Back door clock PLLTEST enable).
2	R/W	1	OSC24M_EN External Crystal OSC24M Enable
1	/	/	/
0	R/W	1	LDO_EN 0: Disable 1: Enable PLL Power enable (power source from VCC_PLL).

### 3.3.5.84. Resistor Calibration Control Register (Default Value: 0x0033\_0003)

Offset: 0x0230			Register Name: RES_CAL_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	MANU_CAL_CLK Manual Calibration Clock.
23	/	/	/
21:16	R/W	0x33	MANU_CAL_OVER_DRV Manual Calibration Over Drive Bit
15:9	/	/	
8	R/W	0x0	DDR_RES240_SEL DDR 240ohms Resistor Manual Control Enable 0: Disable 1: Enable
7:6	/	/	/
5	R/W	0x0	Reserved
4	R/W	0x0	USBPHY0_RES200_SEL USBPHY0 200ohms Resistor Manual Control Enable 0: Disable 1: Enable
3	/	/	/
2	R/W	0x0	CAL_OVER_DRV_EN Calibration Over Drive Enable 0: Auto Calibration 1: Manual Calibration
1	R/W	0x1	CAL_ANALOG_EN Calibration Circuits Analog Enable 0: Disable 1: Enable
0	R/W	0x1	CAL_EN Calibration Enable 0: Disable 1: Enable

### 3.3.5.85. 200ohms Resistor Manual Control Register (Default Value: 0x0000\_3333)

Offset: 0x0234			Register Name: RES200_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13:8	R/W	0x33	Reserved
7:6	/	/	/
5:0	R/W	0x33	USBPHY0_RES200_CTRL

			USBPHY0 200ohms Resistor Manual Control Bits
--	--	--	--

### 3.3.5.86. 240ohms Resistor Manual Control Register (Default Value: 0x0000\_0033)

Offset: 0x0238			Register Name: RES240_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:6	/	/	/
5:0	R/W	0x33	DDR_RES240_CTRL DDR 240ohms resistor manual control bits.

### 3.3.5.87. Resistor Calibration Status Register (Default Value: 0x0000\_0000)

Offset: 0x023C			Register Name: RES_CAL_STATUS_REG
Bit	Read/Write	Default/Hex	Description
31:9	/	/	
8	R	0x0	CAL_ANALOG_CMP_OPT Calibration Circuits Analog Compare Output
7:6	/	/	/
5:0	R	0x0	AUTO_CAL_OPT Auto Calibration Results Output

### 3.3.5.88. NMI Control Register(Default Value: 0x0000\_0000)

Offset: 0x0240			Register Name: RES_CAL_STATUS_REG
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16	R	0x0	NMI interrupt status
15:9	/	/	/
8	W	0x0	Write 1 to clear interrupt.
7:3	/	/	/
2	R/W	0x0	Interrupt enable 0: Disable 1: Enable
1:0	R/W	0x0	Interrupt type selection 00: Low level 01: Falling edge 10: High level 11: Rising edge

### 3.3.5.89. LDOA and LDOB Control Register (Default Value: 0x0000\_0000)

Offset:0x0254			Register Name: LDO_CTRL_REG
Bit	Read/write	Default/Hex	Description
31:8	R	0x0	Reserved
7:5	R/W	0x0	LDOB Output Voltage Step 000: 0mV 001: 25mV 010: 50mV 011: 75mV 100: 100mV 101: -75mV 110: -50mV 111: -25mV
4:3	R/W	0x0	LDOB Voltage Selection 00: 1.35V 01: 1.5V 10: 1.8V 11: Non-configurable. Otherwise LDO output is abnormal.
2:0	R/W	0x0	LDOA Step Voltage Adjust 000: 1.8V 001: 1.825V 010: 1.85V 011: 1.875V 100: 1.9V 101: 1.725V 110: 1.75V 111: 1.775V

### 3.3.5.90. Boot CPU1 Software Entry Register (Default Value: 0x0000\_0000)

Offset:0x0258			Register Name: CPU1_SOFT_ENT_REG
Bit	Read/write	Default/Hex	Description
31:0	R/W	0x0	Boot cpu1 software entry when acting from cpu1 reset.

### 3.3.5.91. EMPTY0 Register (Default Value: 0x0000\_0000)

Offset:0x0260			Register Name: EMPO_REG
Bit	Read/write	Default/Hex	Description
31:0	R/W	0x0	Reserved register It is used for staging information.

### 3.3.5.92. EMPTY1 Register (Default Value: 0x0000\_0000)

Offset:0x0264			Register Name: EMP1_REG
Bit	Read/write	Default/Hex	Description
31:0	R/W	0x0	Reserved register It is used for staging information.

### 3.3.5.93. EMPTY2 Register (Default Value: 0x0000\_0000)

Offset:0x0268			Register Name: EMP2_REG
Bit	Read/write	Default/Hex	Description
31:0	R/W	0x0	Reserved register It is used for staging information.

## 3.4. BROM System

### 3.4.1. Overview

The BROM system has several ways to boot. It has an integrated on-chip Boot ROM (BROM) which could be considered the primary program-loader. On startup process, the R328-S3 starts to fetch the first instruction from address 0x0, where is the BROM located at.

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

The BROM system includes the following features:

- CPU0 boot process and NON\_CPU0 boot process
- Mandatory upgrade process through USB
- Supports eFuse to select the kind of boot media to boot
- Supports normal boot and secure boot
- Secure BROM : loads only certified firmware
- Secure BROM : ensures that the Secure Boot is a trusted environment

### 3.4.2. Operations and Functional Descriptions

#### 3.4.2.1. Boot Media Select

The BROM system supports the following boot media:

- SPI NOR FLASH
- SPI NAND FLASH

On startup,The BROM will read the state of BOOT\_MODE ,according to the state of BOOT\_MODE to decide whether the Try or eFuse to select the kind of boot media to boot. The BOOT\_MODE is actually a bit in the eFuse. Table 3-3 shows BOOT\_MODE Setting.

Table 3- 3. BOOT\_MODE Setting

BOOT_MODE	Boot Select Type
0	Try Media boot
1	eFuse select

If the state of the BOOT\_MODE is 0, that is to choose the Try Media boot: SPI NOR->SPI NAND.

If the state of the BOOT\_MODE is 1, that is to choose the eFuse type .eFuse type has one 12 bits configuration, every 3 bits is divided into a group of the Boot Select, so it has four groups of boot\_select .Table 3-4 shows eFuse Boot Select

Configure.

**Table 3- 4. eFuse Boot Select Configure**

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

Table 3-5 describes each group of the eFuse Boot Select Setting. The first group to the third group are the same settings, but the fourth group need to be careful. If eFuse\_Boot\_Select\_4 is set to 111, that means the way of the *Try*. The way of *Try* is followed by SPI NOR,SPI NAND.

**Table 3- 5. eFuse Boot Select Setting**

eFuse_Boot_Select_n	Boot Media
000	<i>Try</i>
001	Reserved
010	Reserved
011	SPI NOR
100	SPI NAND
101	Reserved
110	Reserved
111	The next a group of the eFuse_Boot_Select. But when the n is equal to 4,it will be a way of <i>Try</i> .

### 3.4.2.2. Normal BROM and Secure BROM

If the System on Chip (SoC) has implemented the ARM TrustZone technology,when the Secure Enable Bit is enabled, the SoC will enable the the ARM TrustZone technology. The minimal security functionality an ARM TrustZone technology based system must implement for its Trusted Boot.

So the BROM is divided into Normal Brom and Secure BROM. the Secure BROM protects against the potential threat of attackers modifying areas of code or data in programmable memory .

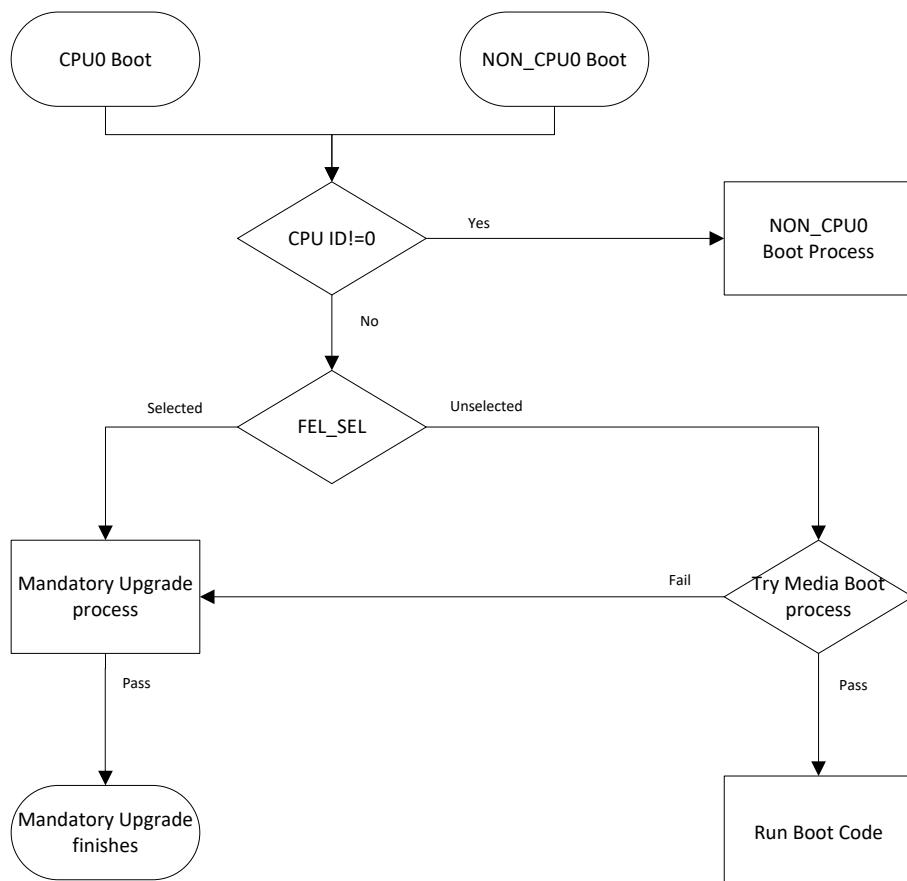
On startup,the SOC will read the Secure Enable Bit,if the bit is 0,then mapping Normal Brom code to address 0x0, or mapping Secure Brom code to address 0x0.

#### 3.4.2.2.1. Normal BROM Process

In Normal boot mode,the system boot will start from CPU0 or NON\_CPU0, BROM will read CPU ID number to distinguish CPU0 or NON\_CPU0, then BROM will read the state of the FEL Pin, if the FEL Pin signal is detected to pull to high level, then the system will jump to the Try Media Boot process,or jump to the mandatory upgrade process. Figure 3-5 shows the BROM Process.

**NOTE**

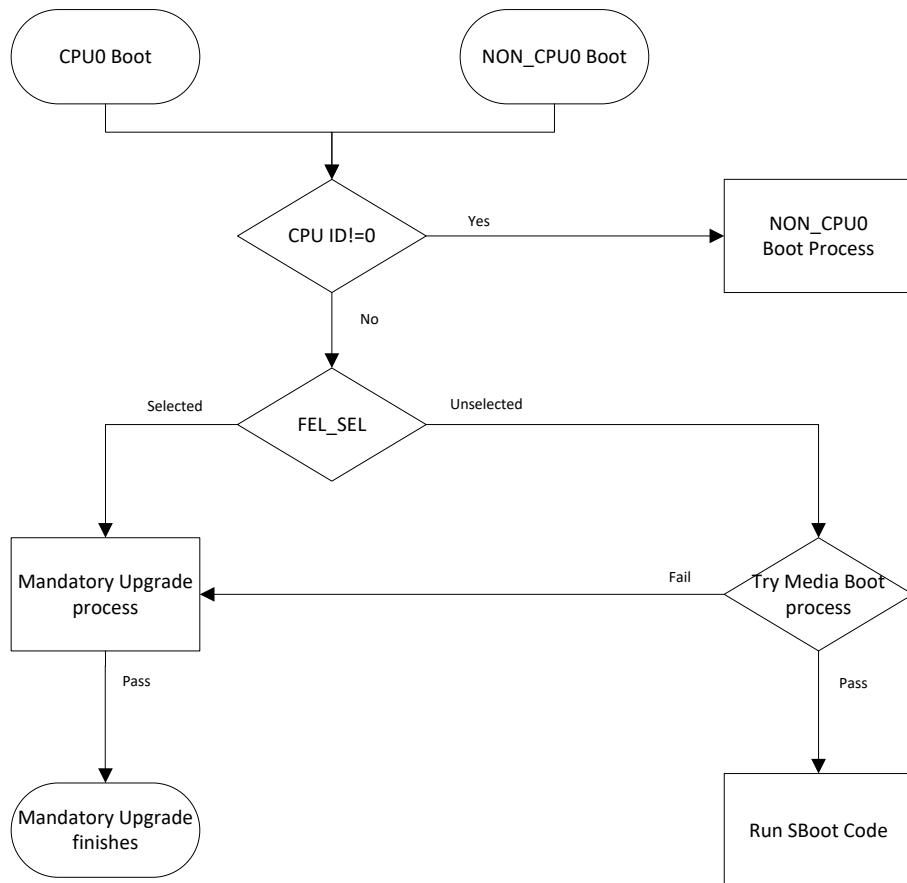
**NON\_CPU0** means the CPU core n(n>0).



**Figure 3- 5. Normal Mode Boot Process**

#### 3.4.2.2.2. Secure BROM Process

In Secure Boot mode, by comparison with Normal BROM, after the Try Media Boot Process finishes, the system will go to run Secure BROM Software. Figure 3-6 shows the Secure BROM Process.



**Figure 3- 6. Secure BROM Process Diagram**

### 3.4.2.2.3. Secure BROM Requirement

The Secure BROM has the following some requirements:

- Supports X509 certificate
- Supports cryptographic algorithms
  - SHA-256
  - RSA-2048
- Supports OTP/eFuse

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

### 3.4.2.3. BROM System Process Description

#### 3.4.2.3.1. NON\_CPU0 Boot Process

If CPU ID is greater than 0, the system boot from boot from NON\_CPU0, BROM will read the Soft Entry Address Register, then jump the Soft Entry Address, and run NON\_CPU0 boot code. Figure 3-7 shows the NON\_CPU0 Boot Process.

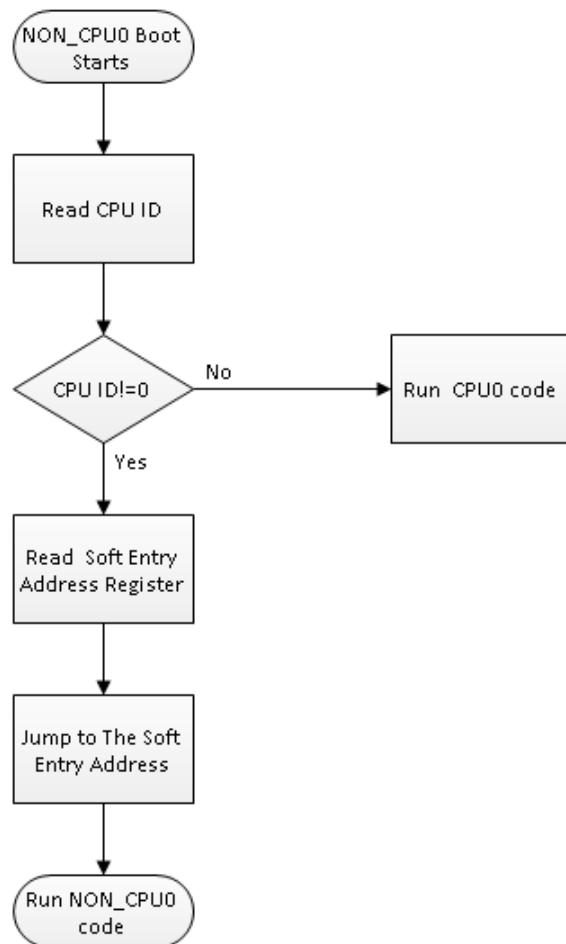


Figure 3- 7. NON\_CPU0 Boot Process



#### NOTE

Each processor owns the Soft Entry Address Register.

#### 3.4.2.3.2. Mandatory Upgrade Process

If the FEL Pin signal is detected to pull low, then the system will jump to mandatory upgrade process. Figure 3-8 shows the mandatory upgrade process.

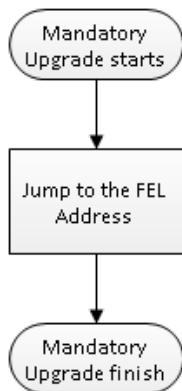


Figure 3- 8. Mandatory Upgrade Process



#### NOTE

The FEL address of the Normal BROM is 0x20.

#### 3.4.2.3.3. FEL Process

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

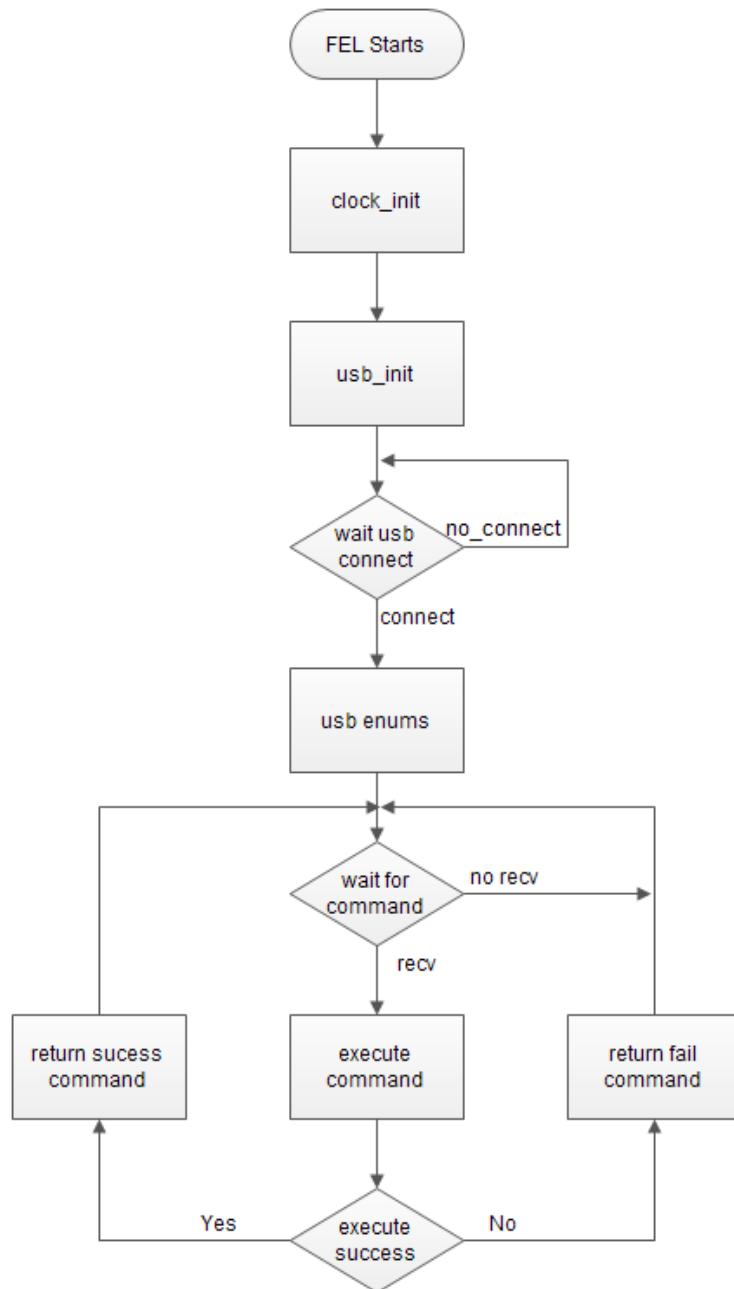
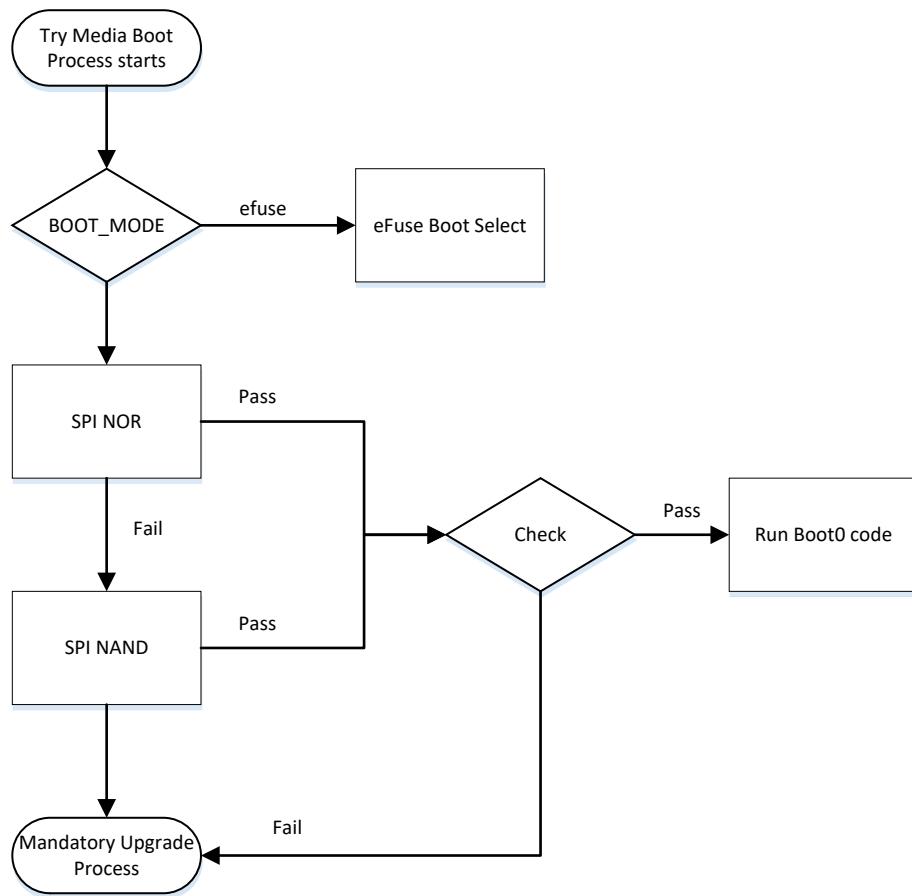


Figure 3- 9. USB FEL Process

#### 3.4.2.3.4. Try Media Boot Process

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

Try Media Boot Process will read the state of BOOT\_MODE register, the state of BOOT\_MODE decides whether select the efuse boot select. Figure 3-10 shows try media boot process.



**Figure 3- 10. Try Media Boot Process**

Figure 3-11 shows BROM eFuse boot select process.

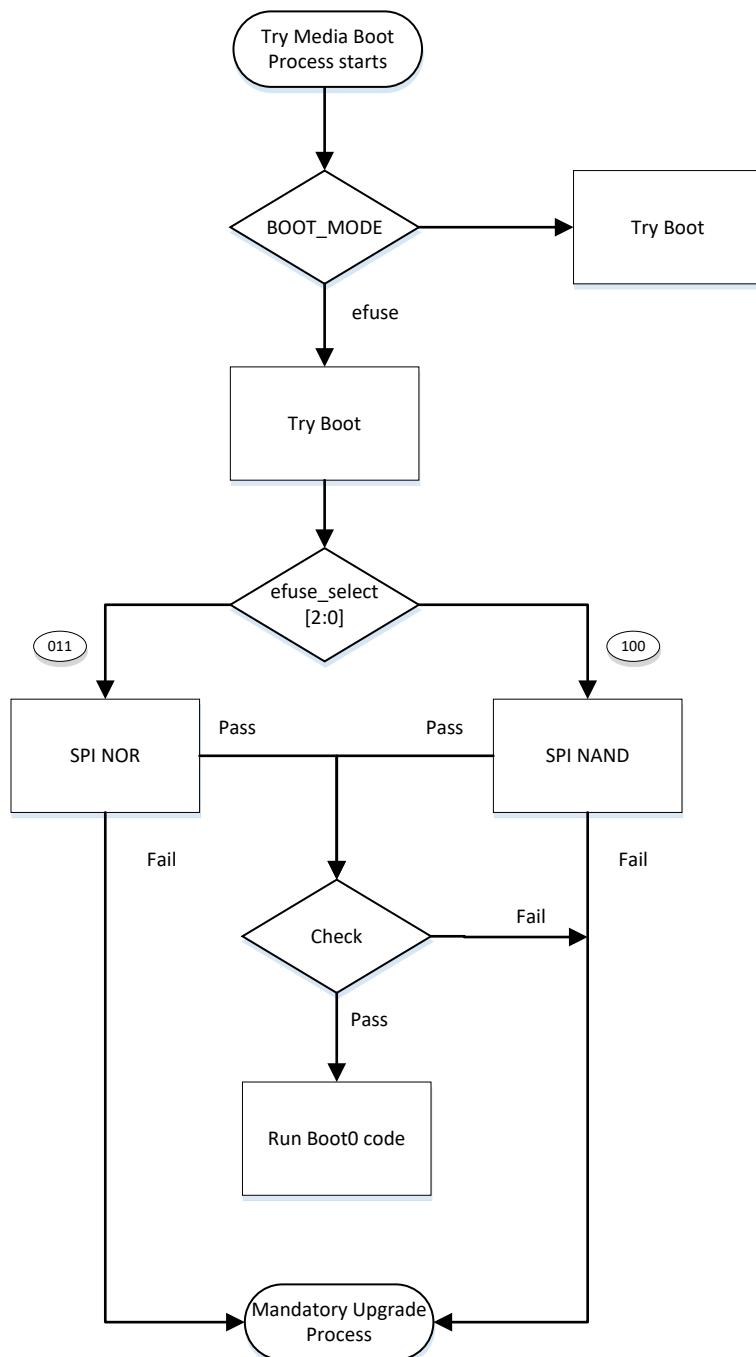


Figure 3- 11. eFuse Boot Select Process

## 3.5. System Configuration

### 3.5.1. Overview

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

The address range of SRAM is as follows.

Area	Address	Size
SRAM C	0x0002 0000---0x0003 FFFF VAD: 0x0002 0000---0x0003 FFFF	128KB(Borrow VAD, clock source is AHB1)

### 3.5.2. Register List

Module Name	Base Address
SYS_CFG	0x03000000

Register Name	Offset	Description
CLK_REG	0x0030	Clock Register

### 3.5.3. Register Description

#### 3.5.3.1. Clock Register (Default Value: 0x0005\_8000)

Offset:0x0030			Register Name: CLK_REG
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	BPS_EFUSE
27:20	R/W	0x0	Reserved
19	R/W	0x0	BIST_CLK_EN 0: BIST clk disable 1: BIST clk enable
18	R/W	0x1	CLK_SEL 0:25MHz 1:24MHz
17	R/W	0x0	LED_POL 0:High active 1:Low active
16	R/W	0x1	SHUTDOWN 0:Power up 1:Shut down

15	R/W	0x1	Reserved
14	/	/	/
13:0	R/W	0x0	Reserved

## 3.6. Timer

### 3.6.1. Overview

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

The timer0 and timer1 are completely consistent. The timer0/1 has the following features:

- Configurable count clock: LOSC and OSC24M. LOSC is the internal low-frequency clock or the external low-frequency clock by setting LOSC\_SRC\_SEL. The external low-frequency has much accuracy.
- Configurable 8 prescale factor
- Programmable 32-bit down timer
- Two working modes: continue mode and single count mode
- Generates an interrupt when the count is decreased to 0

The watchdog is used to transmit a reset signal to reset the entire system after an exception occurs in the system. The watchdog has the following features:

- Single clock source: HOSC\_32K(OSC24M/750, or OSC19.2M/600, or OSC38.4M/1200, selecting oscillator depends on PAD\_SEL)
- 12 initial values to configure
- Generation of timeout interrupts
- Generation of reset signal
- Watchdog restart the timing

### 3.6.2. Block Diagram

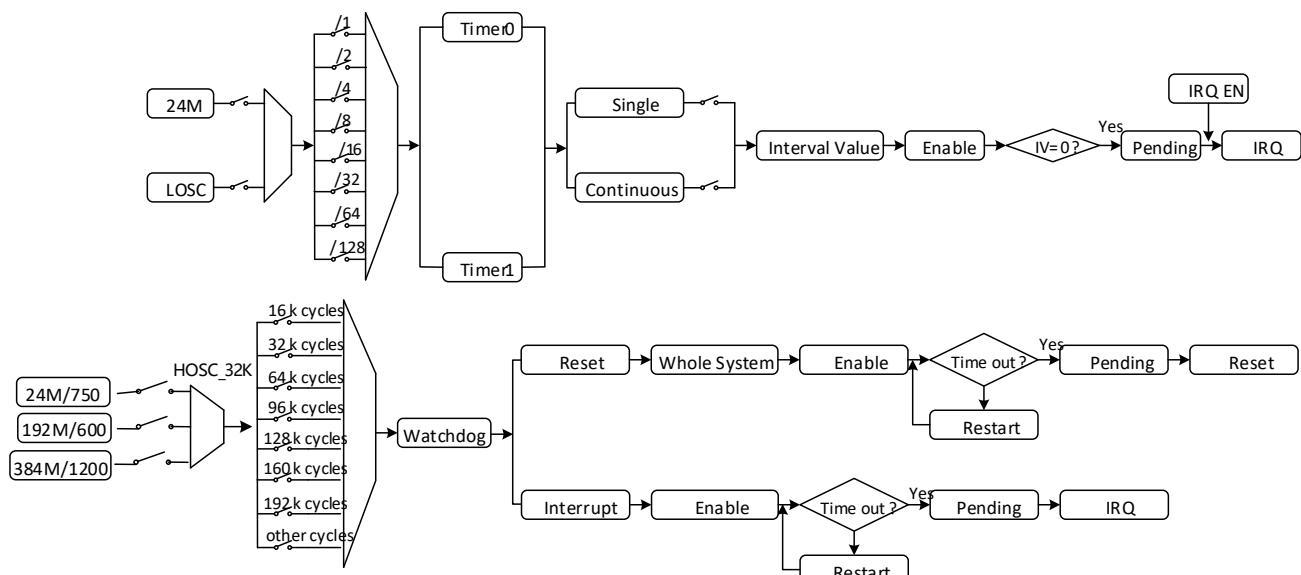


Figure 3- 12. Timer Block Diagram

### 3.6.3. Operations and Functional Descriptions

#### 3.6.3.1. Timer Formula

Using Timer0 as an example.

$$T_{\text{timer}0} = \frac{\text{TMRO\_INTV\_VALUE\_REG} - \text{TMRO\_CUR\_VALUE\_REG}}{\text{TMRO\_CLK\_SRC}} \times \text{TMRO\_CLK\_PRES}$$

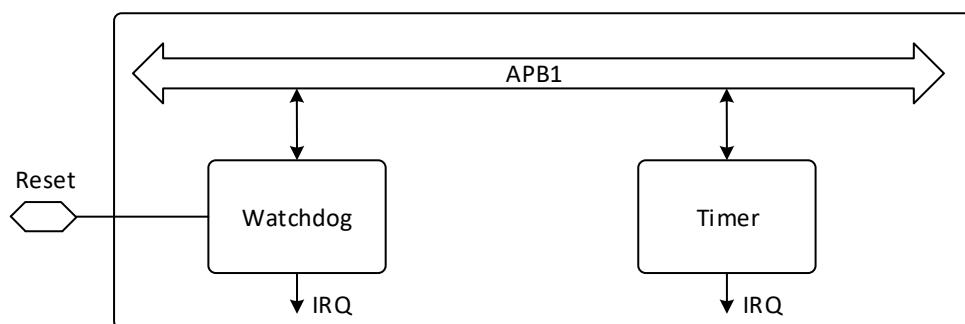
TMRO\_INTV\_VALUE\_REG: timer initial value;

TMRO\_CUR\_VALUE\_REG: timer current counter;

TMRO\_CLK\_SRC: timer clock source;

TMRO\_CLK\_PRES: timer clock prescale ratio.

#### 3.6.3.2. Typical Application



**Figure 3- 13. Timer Application Diagram**

Timer and watchdog configure register by APB1 bus. Timer and watchdog have interrupt mode.

The system configures the time of watchdog, if the system has no timing for restart watchdog(such as bus hang dead), then watchdog sends out watchdog reset external signal to reset system; meanwhile watchdog outputs signal to RESET pad to reset PMIC.

#### 3.6.3.3. Function Implementation

##### 3.6.3.3.1. Timer

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

The timer has two operating modes.

- **Continuous mode**

The bit7 of the **TMRn\_CTRL\_REG** is set to the continuous mode, when the count value is decreased to 0, the timer module reloads data from **TMRn\_INTV\_VALUE\_REG** then continues to count.

- **Single mode**

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

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

### 3.6.3.3.2. Watchdog

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

The watchdog has two operating modes.

- **Interrupt mode**

The **WDOG0\_CFG\_REG** is set to 0x2, when the counter value reaches 0 and **WDOG0\_IRQ\_EN\_REG** is enabled, the watchdog generates an interrupt.

- **Reset mode**

The **WDOG0\_CFG\_REG** is set to 0x1, when the counter value reaches 0, the watchdog generates a reset signal to reset the entire system.

The clock source of the watchdog is **HOSC\_32K**. There are 12 configurable initial count values.

The watchdog can restart to count by setting the **WDOG0\_CTRL\_REG**: write 0xA57 to bit[12:1], then write 1 to bit[0].

### 3.6.3.4. Operating Mode

#### 3.6.3.4.1. Timer Initial

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

#### 3.6.3.4.2. Timer Interrupt

- (1) Enable interrupt: write corresponding interrupt enable bit of **TMR\_IRQ\_EN\_REG**, when timer counter time reaches,

the corresponding interrupt generates.

- (2) After enter interrupt process, write **TMR\_IRQ\_STA\_REG** to clear the interrupt pending, and execute the process of waiting for the interrupt.
- (3) Resume the interrupt and continue to execute the interrupted process.

#### 3.6.3.4.3. Watchdog Initial

- (1) Write **WDOGO\_CFG\_REG** to configure the generation of the interrupts and the output of reset signal.
- (2) Write **WDOGO\_MODE\_REG** to configure the initial count value.
- (3) Write **WDOGO\_MODE\_REG** to enable the watchdog.

#### 3.6.3.4.4. Watchdog Interrupt

Watchdog interrupt is only used for the counter.

- (1) Write **WDOGO\_IRQ\_EN\_REG** to enable the interrupt.
- (2) After enter the interrupt process, write **WDOGO\_IRQ\_STA\_REG** to clear the interrupt pending, and execute the process of waiting for the interrupt.
- (3) Resume the interrupt and continue to execute the interrupted process.

### 3.6.4. Programming Guidelines

#### 3.6.4.1. Timer

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

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

#### 3.6.4.2. Watchdog Reset

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

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

### 3.6.4.3. Watchdog Restart

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

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

### 3.6.5. Register List

Module Name	Base Address
Timer	0x03009000

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

### 3.6.6. Register Description

#### 3.6.6.1. Timer IRQ Enable Register(Default Value: 0x0000\_0000)

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

31:2	/	/	/
1	R/W1S	0x0	TMR1_IRQ_EN Timer 1 Interrupt Enable 0: No effect 1: Timer 1 Interval Value reached interrupt enable
0	R/W1S	0x0	TMR0_IRQ_EN Timer 0 Interrupt Enable 0: No effect 1: Timer 0 Interval Value reached interrupt enable

### 3.6.6.2. Timer IRQ Status Register(Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: TMR_IRQ_STA_REG
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W1C	0x0	TMR1_IRQ_PEND Timer 1 IRQ Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending, timer 1 interval value is reached
0	R/W1C	0x0	TMR0_IRQ_PEND Timer 0 IRQ Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending, timer 0 interval value is reached

### 3.6.6.3. Timer 0 Control Register(Default Value: 0x0000\_0004)

Offset: 0x0010			Register Name: TMR0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	TMR0_MODE Timer 0 mode 0: Continuous mode. When interval value reached, the timer will not disable automatically. 1: Single mode. When interval value reached, the timer will disable automatically.
6:4	R/W	0x0	TMR0_CLK_PRES Select the pre-scale of timer 0 clock source. 000: /1 001: /2 010: /4 011: /8 100: /16 101: /32

			110: /64 111: /128
3:2	R/W	0x1	TMR0_CLK_SRC 00:LOSC 01: OSC24M 10: / 11: /
1	R/W	0x0	TMR0_RELOAD Timer 0 Reload 0: No effect 1: Reload timer 0 Interval value After the bit is set, it can not be written again before it is cleared automatically.
0	R/W	0x0	TMR0_EN Timer 0 Enable 0: Stop/Pause 1: Start If the timer is started, it will reload the interval value to internal register, and the current counter will count from interval value to 0. If the current counter does not reach the zero, the timer enable bit is set to "0", the current value counter will pause. At least wait for 2 cycles, the start bit can be set to 1. In timer pause state, the interval value register can be modified. If the timer is started again, and the software hopes the current value register to down-count from the new interval value, the reload bit and the enable bit should be set to 1 at the same time.

#### 3.6.6.4. Timer 0 Interval Value Register(Default Value: 0x0000\_0000)

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



#### NOTE

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

#### 3.6.6.5. Timer 0 Current Value Register(Default Value: 0x0000\_0000)

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

### 3.6.6.6. 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	<p>TMR1_MODE Timer 1 mode 0: Continuous mode. When interval value reached, the timer will not disable automatically. 1: Single mode. When interval value reached, the timer will disable automatically.</p>
6:4	R/W	0x0	<p>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</p>
3:2	R/W	0x1	<p>TMR1_CLK_SRC 00:LOSC 01: OSC24M 10: / 11: /</p>
1	R/W	0x0	<p>TMR1_RELOAD Timer 1 Reload 0: No effect 1: Reload timer 1 Interval value After the bit is set, it can not be written again before it is cleared automatically.</p>
0	R/W	0x0	<p>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, the timer enable bit is set to "0", the current value counter will pause. At least wait for 2 cycles, the start bit can be set to 1. In timer pause state, the interval value register can be modified. If the timer is started again, and the software hopes the current value register to down-count from the new interval value, the reload bit and the enable bit should be set to 1.</p>

			1 at the same time.
--	--	--	---------------------

### 3.6.6.7. Timer 1 Interval Value Register(Default Value: 0x0000\_0000)

Offset: 0x0024			Register Name: TMR1_INTV_VALUE_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	TMR1_INTV_VALUE Timer 1 Interval Value


**NOTE**

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

### 3.6.6.8. Timer 1 Current Value Register(Default Value: 0x0000\_0000)

Offset: 0x0028			Register Name: TMR1_CUR_VALUE_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	TMR1_CUR_VALUE Timer 1 Current Value Timer 1 current value is a 32-bit down-counter (from interval value to 0).

### 3.6.6.9. Watchdog IRQ Enable Register(Default Value: 0x0000\_0000)

Offset: 0x00A0			Register Name: WDOG_IRQ_EN_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W1S	0x0	WDOG_IRQ_EN Watchdog Interrupt Enable 0: No effect 1: Watchdog interrupt enable

### 3.6.6.10. Watchdog Status Register (Default Value: 0x0000\_0000)

Offset: 0x00A4			Register Name: WDOG_IRQ_STA_REG
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W1C	0x0	WDOG_IRQ_PEND Watchdog IRQ Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending. Watchdog interval value is reached.

### 3.6.6.11. Watchdog Control Register(Default Value: 0x0000\_0000)

Offset: 0x00B0			Register Name:WDOG_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:13	/	/	/
12:1	R/W	0x0	WDOG_KEY_FIELD Watchdog Key Field It should be written to 0xA57. Writing any other value in this field aborts the write operation.
0	R/W1S	0x0	WDOG_RESTART Watchdog Restart 0: No effect 1: Restart the Watchdog

### 3.6.6.12. Watchdog Configuration Register (Default Value: 0x0000\_0001)

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

### 3.6.6.13. Watchdog Mode Register (Default Value: 0x0000\_0000)

Offset: 0x00B8			Register Name:WDOG_MODE_REG
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:4	R/W	0x0	WDOG_INTV_VALUE Watchdog Interval Value Watchdog clock source is HOSC_32K. If the clock source is turned off, Watchdog will not work. 0000: 16000 cycles (0.5s) 0001: 32000 cycles (1s) 0010: 64000 cycles (2s) 0011: 96000 cycles (3s) 0100: 128000 cycles (4s) 0101: 160000 cycles (5s) 0110: 192000 cycles (6s) 0111: 256000 cycles (8s) 1000: 320000 cycles (10s)

			1001: 384000 cycles (12s) 1010: 448000 cycles (14s) 1011: 512000 cycles (16s) Others:Reserved
3:1	/	/	/
0	R/W1S	0x0	WDOG_EN Watchdog Enable 0: No effect 1: Enable the Watchdog

## 3.7. High Speed Timer

### 3.7.1. Overview

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

The HSTimer has the following features:

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

### 3.7.2. Block Diagram

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

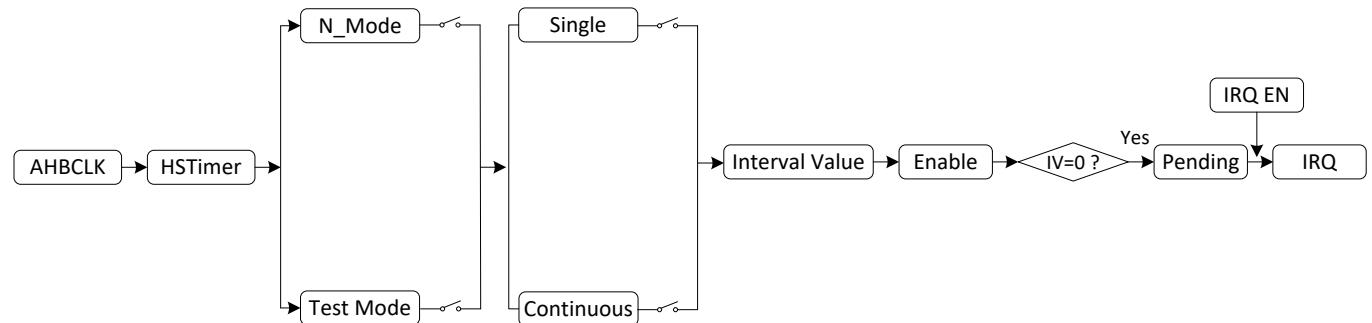


Figure 3- 14. HSTimer Block Diagram

### 3.7.3. Operations and Functional Description

#### 3.7.3.1. HSTimer Formula

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

HS\_TMR\_INTV\_HI\_REG: Initial of Counter Higher Bit

HS\_TMR\_INTV\_LO\_REG: Initial of Counter Lower Bit

HS\_TMR\_CURNT\_HI\_REG: Current Value of Counter Higher Bit

HS\_TMR\_CURNT\_LO\_REG: Current Value of Counter Lower Bit

AHB1CLK: AHB1 Clock Frequency

HS\_TMR\_CLK: Time Prescale Ratio of Counter

### 3.7.3.2. Typical Application

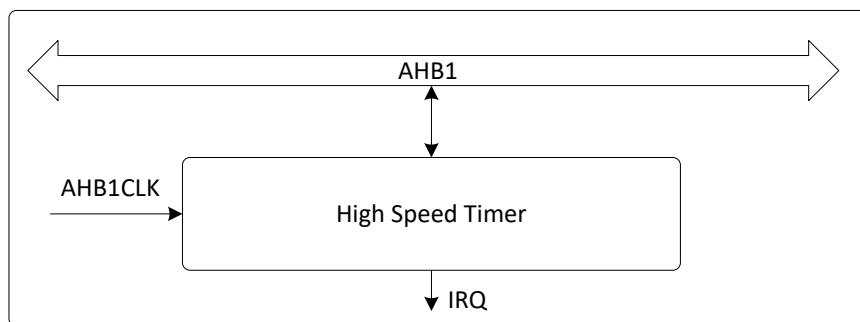


Figure 3- 15. HSTimer Application Diagram

The high speed timer is on AHB1, and the high speed timer controls registers by AHB1.

The high speed timer has single clock source: AHB. The high speed timer can generate interrupt.

### 3.7.3.3. Function Implementation

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

The high speed timer has two timing modes.

- Continuous mode : The bit7 of **HS\_TMR0\_CTRL\_REG** is set to the continuous mode, when the count value is decreased to 0, the high speed timer module reloads data from **HS\_TMR\_INTV\_LO\_REG** and **HS\_TMR\_INTV\_HI\_REG** then continues to count.
- Single mode : The bit7 of **HS\_TMR0\_CTRL\_REG** is set to the single mode, when the count value is decreased to 0, the high speed timer stops counting. The high speed timer starts to count again only when a new initial value is loaded.

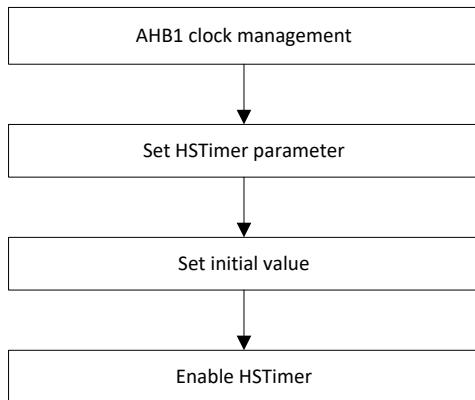
The high speed timer has two operating modes.

- Normal mode: When the bit31 of **HS\_TMR0\_CTRL\_REG** is set to the normal mode, the high speed timer is used as 56-bit down counter, which can finish continuous timing and single timing.
- Test mode: When the bit31 of **HS\_TMR0\_CTRL\_REG** is set to the test mode, then **HS\_TMR\_INTV\_LO\_REG** must be set to 0x1, the high speed timer is used as 24-bit down counter, and **HS\_TMR\_INTV\_HI\_REG** is the initial value of the high speed timer.

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

### 3.7.3.4. Operating Mode

#### 3.7.3.4.1. HSTimer Initial



**Figure 3- 16. HSTimer Initialization Process**

- (1) AHB1 clock management: Open the clock gating of AHB1 and de-assert the soft reset of AHB1 in CCU.
- (2) Configure the corresponding parameters of the high speed timer: clock source, prescaler factor, working mode, counting mode. These parameters that are written to **HS\_TMR0\_CTRL\_REG** have no sequences.
- (3) Write the initial value: Firstly write the low-bit register **HS\_TMR\_INTV\_LO\_REG**, then write the high-bit register **HS\_TMR\_INTV\_HI\_REG**. Write the bit1 of **HS\_TMR0\_CTRL\_REG** to load the initial value. If in timing stop stage of high speed timer, write the bit1 and bit0 of **HS\_TMR0\_CTRL\_REG** to reload the initial value.
- (4) Enable high speed timer: Write the bit[0] of **HS\_TMR0\_CTRL\_REG** to enable high speed timer to count.
- (5) Reading **HS\_TMR\_CURNT\_LO\_REG** and **HS\_TMR\_CURNT\_HI\_REG** can get current counting value.

#### 3.7.3.4.2. HSTimer Interrupt

- (1) Enable interrupt: Write the corresponding interrupt enable bit of **HS\_TMR\_IRQ\_EN\_REG**, when the counting time of high speed timer reaches , the corresponding interrupt generates.
- (2) After enter the interrupt process, write **HS\_TMR\_IRQ\_STAS\_REG** to clear the interrupt pending.
- (3) Resume the interrupt and continue to execute the interrupted process.

### 3.7.4. Programming Guidelines

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

```

writel(0x0, HS_TMR0_INTV_HI);           //Set interval value Hi 0x0
writel(0x32, HS_TMR0_INTV_LO);          //Set interval value Lo 0x32
writel(0x90, HS_TMR0_CTRL);             //Select n_mode,2 pre-scale,single mode
writel(readl(HS_TMR0_CTRL)|(1<<1), HS_TMR0_CTRL); //Set Reload bit
writel(readl(HS_TMR0_CTRL)|(1<<0), HS_TMR0_CTRL); //Enable HSTimer0
  
```

```

while(!(readl(HS_TMR IRQ_STAS)&1));           //Wait for HSTimer0 to generate pending
writel(1,HS_TMR IRQ_STAS);                    //Clear HSTimer0 pending
    
```

### 3.7.5. Register List

Module Name	Base Address
High Speed Timer	0x03005000

Register Name	Offset	Description
HS_TMR IRQ_EN_REG	0x0000	HS Timer IRQ Enable Register
HS_TMR IRQ_STAS_REG	0x0004	HS Timer Status Register
HS_TMR0_CTRL_REG	0x0020	HS Timer 0 Control Register
HS_TMR0_INTV_LO_REG	0x0024	HS Timer 0 Interval Value Low Register
HS_TMR0_INTV_HI_REG	0x0028	HS Timer 0 Interval Value High Register
HS_TMR0_CURNT_LO_REG	0x002C	HS Timer 0 Current Value Low Register
HS_TMR0_CURNT_HI_REG	0x0030	HS Timer 0 Current Value High Register
HS_TMR1_CTRL_REG	0x0040	HS Timer 1 Control Register
HS_TMR1_INTV_LO_REG	0x0044	HS Timer 1 Interval Value Low Register
HS_TMR1_INTV_HI_REG	0x0048	HS Timer 1 Interval Value High Register
HS_TMR1_CURNT_LO_REG	0x004C	HS Timer 1 Current Value Low Register
HS_TMR1_CURNT_HI_REG	0x0050	HS Timer 1 Current Value High Register

### 3.7.6. Register Description

#### 3.7.6.1. HS Timer IRQ Enable Register (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: HS_TMR IRQ_EN_REG
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W1S	0x0	HS_TMR1_INT_EN High Speed Timer 1 Interrupt Enable 0: No effect 1: High Speed Timer1 interval value reached interrupt enable
0	R/W1S	0x0	HS_TMR0_INT_EN High Speed Timer 0 Interrupt Enable 0: No effect 1: High Speed Timer0 interval value reached interrupt enable

### 3.7.6.2. HS Timer IRQ Status Register(Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: HS_TMR_IRQ_STAS_REG
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W1C	0x0	<p>HS_TMR1_IRQ_PEND High Speed Timer 1 IRQ Pending Setting 1 to the bit will clear it. 0: No effect 1: Pending, High speed timer 1 interval value is reached.</p>
0	R/W1C	0x0	<p>HS_TMR0_IRQ_PEND High Speed Timer 0 IRQ Pending Setting 1 to the bit will clear it. 0: No effect 1: Pending, High speed timer 0 interval value is reached.</p>

### 3.7.6.3. HS Timer 0 Control Register(Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: HS_TMR0_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<p>HS_TMR0_TEST High Speed Timer 0 Test Mode In test mode, the low register should be set to 0x1, the high register will down counter. The counter needs to be reloaded. 0: Normal mode 1: Test mode</p>
30:8	/	/	/
7	R/W	0x0	<p>HS_TMR0_MODE High Speed Timer 0 Mode 0: Continuous mode. When interval value reached, the timer will not disable automatically. 1: Single mode. When interval value reached, the timer will disable automatically.</p>
6:4	R/W	0x0	<p>HS_TMR0_CLK Select the pre-scale of the high speed timer 0 clock sources. 000: /1 001: /2 010: /4 011: /8 100: /16 101: / 110: / 111: /</p>
3:2	/	/	/

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

#### 3.7.6.4. HS Timer 0 Interval Value Lo Register(Default Value: 0x0000\_0000)

Offset: 0x0024			Register Name: HS_TMR0_INTV_LO_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	HS_TMR0_INTV_VALUE_LO High Speed Timer 0 Interval Value [31:0]

#### 3.7.6.5. HS Timer 0 Interval Value Hi Register(Default Value: 0x0000\_0000)

Offset: 0x0028			Register Name: HS_TMR0_INTV_HI_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:0	R/W	0x0	HS_TMR0_INTV_VALUE_HI High Speed Timer 0 Interval Value [55:32]



#### NOTE

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

#### 3.7.6.6. HS Timer 0 Current Value Lo Register(Default Value: 0x0000\_0000)

Offset: 0x002C			Register Name: HS_TMR0_CURNT_LO_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	HS_TMR0_CUR_VALUE_LO

			High Speed Timer 0 Current Value [31:0]
--	--	--	---

### 3.7.6.7. HS Timer 0 Current Value Hi Register(Default Value: 0x0000\_0000)

Offset: 0x0030			Register Name: HS_TMR0_CURNT_HI_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:0	R/W	0x0	HS_TMR0_CUR_VALUE_HI High Speed Timer 0 Current Value [55:32]


**NOTE**

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

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

### 3.7.6.8. HS Timer 1 Control Register(Default Value: 0x0000\_0000)

Offset: 0x0040			Register Name: HS_TMR1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	HS_TMR1_TEST High Speed Timer 1 Test Mode In test mode, the low register should be set to 0x1, the high register will down counter. The counter needs to be reloaded. 0: Normal mode 1: Test mode
30:8	/	/	/
7	R/W	0x0	HS_TMR1_MODE High Speed Timer 1 Mode 0: Continuous mode. When interval value reached, the timer will not disable automatically. 1: Single mode. When interval value reached, the timer will disable automatically.
6:4	R/W	0x0	HS_TMR1_CLK Select the pre-scale of the high speed timer 1 clock sources. 000: /1 001: /2 010: /4 011: /8 100: /16 101: / 110: / 111: /
3:2	/	/	/

1	R/W1S	0x0	<b>HS_TMR1_RELOAD</b> High Speed Timer 1 Reload 0: No effect 1: Reload High Speed Timer 1 Interval Value
0	R/W	0x0	<b>HS_TMR1_EN</b> High Speed 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, the timer enable bit is set to "0", the current value counter will pause. At least wait for 2 cycles, the start bit can be set to 1. In timer pause state, the interval value register can be modified. If the timer is started again, and the software hopes the current value register to down-count from the new interval value, the reload bit and the enable bit should be set to 1 at the same time.

### 3.7.6.9. HS Timer 1 Interval Value Lo Register(Default Value: 0x0000\_0000)

Offset: 0x0044			Register Name: HS_TMR1_INTV_LO_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	HS_TMR1_INTV_VALUE_LO High Speed Timer 1 Interval Value [31:0]

### 3.7.6.10. HS Timer 1 Interval Value Hi Register(Default Value: 0x0000\_0000)

Offset: 0x0048			Register Name: HS_TMR1_INTV_HI_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:0	R/W	0x0	HS_TMR1_INTV_VALUE_HI High Speed Timer 1 Interval Value [55:32]



#### NOTE

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

### 3.7.6.11. HS Timer 1 Current Value Lo Register(Default Value: 0x0000\_0000)

Offset: 0x004C			Register Name: HS_TMR1_CURNT_LO_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	HS_TMR1_CUR_VALUE_LO

			High Speed Timer 1 Current Value [31:0]
--	--	--	---

### 3.7.6.12. HS Timer 1 Current Value Hi Register(Default Value: 0x0000\_0000)

Offset: 0x0050			Register Name: HS_TMR1_CURNT_HI_REG
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:0	R/W	0x0	HS_TMR1_CUR_VALUE_HI High Speed Timer 1 Current Value [55:32]



#### NOTE

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

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

## 3.8. GIC

### 3.8.1. Interrupt Source

Interrupt Number	Interrupt Source	Interrupt Vector	Description
0	SGI 0	0x0000	SGI 0 interrupt
1	SGI 1	0x0004	SGI 1 interrupt
2	SGI 2	0x0008	SGI 2 interrupt
3	SGI 3	0x000C	SGI 3 interrupt
4	SGI 4	0x0010	SGI 4 interrupt
5	SGI 5	0x0014	SGI 5 interrupt
6	SGI 6	0x0018	SGI 6 interrupt
7	SGI 7	0x001C	SGI 7 interrupt
8	SGI 8	0x0020	SGI 8 interrupt
9	SGI 9	0x0024	SGI 9 interrupt
10	SGI 10	0x0028	SGI 10 interrupt
11	SGI 11	0x002C	SGI 11 interrupt
12	SGI 12	0x0030	SGI 12 interrupt
13	SGI 13	0x0034	SGI 13 interrupt
14	SGI 14	0x0038	SGI 14 interrupt
15	SGI 15	0x003C	SGI 15 interrupt
16	PPI 0	0x0040	PPI 0 interrupt
17	PPI 1	0x0044	PPI 1 interrupt
18	PPI 2	0x0048	PPI 2 interrupt
19	PPI 3	0x004C	PPI 3 interrupt
20	PPI 4	0x0050	PPI 4 interrupt
21	PPI 5	0x0054	PPI 5 interrupt
22	PPI 6	0x0058	PPI 6 interrupt
23	PPI 7	0x005C	PPI 7 interrupt
24	PPI 8	0x0060	PPI 8 interrupt
25	PPI 9	0x0064	PPI 9 interrupt
26	PPI 10	0x0068	PPI 10 interrupt
27	PPI 11	0x006C	PPI 11 interrupt
28	PPI 12	0x0070	PPI 12 interrupt
29	PPI 13	0x0074	PPI 13 interrupt
30	PPI 14	0x0078	PPI 14 interrupt
31	PPI 15	0x007C	PPI 15 interrupt
32~63	/	0x0080~0x00FC	Reserved
64	NMI	0x0100	NMI interrupt
65	/	0x0104	/
66	DMA	0x0108	DMA interrupt

Interrupt Number	Interrupt Source	Interrupt Vector	Description
67	VAD_WAKE	0x010C	VAD wake interrupt
68	VAD_DATA_REQ	0x0110	VAD data request interrupt
69	USB2.0_DRD_DEVICE	0x0114	USB2.0_DRD_DEVICE interrupt
70	USB2.0_DRD_EHCI	0x0118	USB2.0_DRD_EHCI interrupt
71	USB2.0_DRD_OHCI	0x011C	USB2.0_DRD_OHCI interrupt
72	/	0x0120	/
73	/	0x0124	/
74	/	0x0128	/
75	GPIOB	0x012C	GPIOB interrupt
76	GPIOE	0x0130	GPIOE interrupt
77	/	0x0134	/
78	GPIOG	0x0138	GPIOG interrupt
79	GPIOH	0x013C	GPIOH interrupt
80	GPADC	0x0140	GPADC interrupt
81	THERMAL	0x0144	THERMAL interrupt
82	LRADC	0x0148	LRADC interrupt
83	OWA	0x014C	OWA interrupt
84	DMIC	0x0150	DMIC interrupt
85	/	0x0154	/
86	MSI	0x0158	MSI interrupt
87	SMC	0x015C	SMC interrupt
88	WDOG	0x0160	Watchdog interrupt
89	CCU_FERR	0x0164	CCU_FERR interrupt
90	BUS_TIMEOUT	0x0168	BUS_TIMEOUT interrupt
91	PSI	0x016C	PSI interrupt
92	LEDC	0x0170	LEDC interrupt
93	AUDIO_CODEC DAC	0x0174	AUDIO_CODEC DAC interrupt
94	AUDIO_CODEC ADC	0x0178	AUDIO_CODEC ADC interrupt
95	Reserved	0x017C	Reserved
96	/	0x0180	/
97	CE_NS	0x0184	CE_NS interrupt
98	CE	0x0188	CE interrupt
99	I2S/PCM0	0x018C	I2S/PCM0 interrupt
100	I2S/PCM1	0x0190	I2S/PCM1 interrupt
101	I2S/PCM2	0x0194	I2S/PCM2 interrupt
102	TWI0	0x0198	TWI0 interrupt
103	TWI1	0x019C	TWI1 interrupt
104	/	/	/
105	/	0x01A4	/
106	SMHC1	0x01A8	SMHC1 interrupt
107	/	0x01AC	/
108	UART0	0x01B0	UART0 interrupt

Interrupt Number	Interrupt Source	Interrupt Vector	Description
109	UART1	0x01B4	UART1 interrupt
110	UART2	0x01B8	UART2 interrupt
111	UART3	0x01BC	UART3 interrupt
112	/	0x01C0	/
113	SPI0	0x01C4	SPI0 interrupt
114	SPI1	0x01C8	SPI1 interrupt
115	HSTIMER0	0x01CC	HSTIMER0 interrupt
116	HSTIMER1	0x01D0	HSTIMER1 interrupt
117	TIMER0	0x01D4	TIMER0 interrupt
118	TIMER1	0x01D8	TIMER1 interrupt
119	/	0x01DC	/
120	/	0x01E0	/
121~159	/	/	Resevred
160	C0_CTL0	0x0430	C0_CTL0 interrupt
161	C0_CTL1	0x0434	C0_CTL1 interrupt
162	C0_COMMTX0	0x0440	C0_COMMTX0 interrupt
163	C0_COMMTX1	0x0444	C0_COMMTX1 interrupt
164	C0_COMMRX0	0x0450	C0_COMMRX0 interrupt
165	C0_COMMRX1	0x0454	C0_COMMRX1 interrupt
166	C0_PMU0	0x0460	C0_PMU0 interrupt
167	C0_PMU1	0x0464	C0_PMU1 interrupt
168	C0_AXI_ERROR	0x0470	C0_AXI_ERROR interrupt

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

## 3.9. DMA

### 3.9.1. Overview

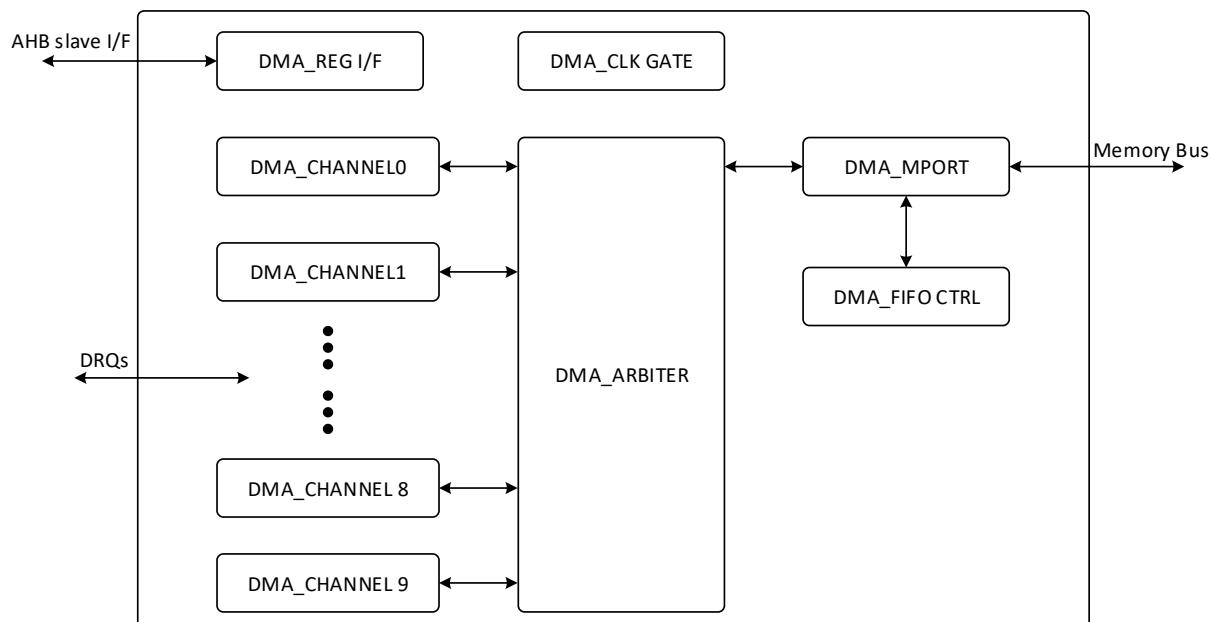
The direct memory access (DMA) is used to transfer data between a peripheral and a memory, between peripherals, or between memories. DMA is a high-speed data transfer operation that reduces the CPU resources.

The DMA has the following features:

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

### 3.9.2. Block Diagram

The following figure shows a block diagram of DMA.



**Figure 3- 17. DMA Block Diagram**

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

**DMA\_CHANNEL:** DMA transform engine.Each channel is independent. The priorities of DMA channels uses polling mechanism. When the DMA requests from two peripherals are valid simultaneously, if DMA\_ARBITER is non-idle ,the

next channel of the current channel has the higher priority; if DMA\_ARBTER is idle, the channel0 has the highest priority, whereas the channel 9 has the lowest priority.

DMA\_MPORT: Receive read/write requirement of DMA\_ARBTER ,and convert to the corresponding MBUS access.

DMA\_FIFOCTL: Internal FIFO cell control module.

DMA\_REGIF: Common register module, mainly used to resolve AHB1 demand.

DMA\_CLKGATE: Hardware auto clock gating control module.

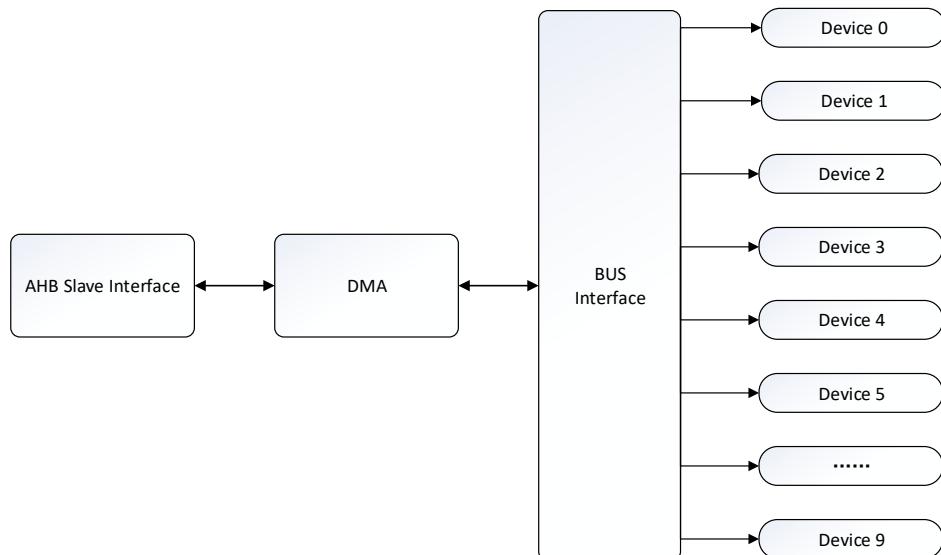
DMA integrates 10 independent DMA channels. When DMA channel starts, DMA gets DMA descriptor by DMA\_DESC\_ADDR\_REG to use for the configuration information of the current DMA package transfer ,and DMA can transfer data between the specified peripherals through the configuration information. When a package transfer finished, DMA judges if the current channel transfer finished through the linked information in descriptor.

### 3.9.3. Operations and Functional Description

#### 3.9.3.1. Clock and Reset

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

#### 3.9.3.2. Typical Application



**Figure 3- 18. DMA Typical Application Diagram**

#### 3.9.3.3. DRQ Type

**Table 3- 6. DMA DRQ Table**

Source DRQ Type		Destination DRQ Type	
port0	SRAM	port0	SRAM

port1	DRAM	port1	DRAM
port2	OWA-RX	port2	OWA-TX
port3	I2S/PCM0-RX	port3	I2S/PCM0-TX
port4	I2S/PCM1-RX	port4	I2S/PCM1-TX
port5	I2S/PCM2-RX	port5	I2S/PCM2-TX
port6	Audio Codec	port6	Audio Codec
port7	DMIC	port7	
port8		port8	
port9		port9	
port10		port10	
port11		port11	
port12	GPADC	port12	
port13		port13	
port14	UART0-RX	port14	UART0-TX
port15	UART1-RX	port15	UART1-TX
port16	UART2-RX	port16	UART2-TX
port17	UART3-RX	port17	UART3-TX
port18	UART4-RX	port18	UART4-TX
port19		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		Port30	
Port31		Port31	
Port32		Port32	
Port33		Port33	
Port34		Port34	
Port35		Port35	
Port36		Port36	
Port37		Port37	
Port38		Port38	
Port39		Port39	
Port40		Port40	
Port41		Port41	
Port42		Port42	
Port43	LEDC	Port43	
Port44	VAD_RX	Port44	VAD_TX

Port45	TWI0	Port45	TWI0
Port46	TWI1	Port46	TWI1

### 3.9.3.4. DMA Descriptor



Figure 3- 19. DMA Descriptor

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

**Configuration :** Configure the following information by DMA\_CFG\_REG.

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

**Source Address:** Configure the transferred source address.

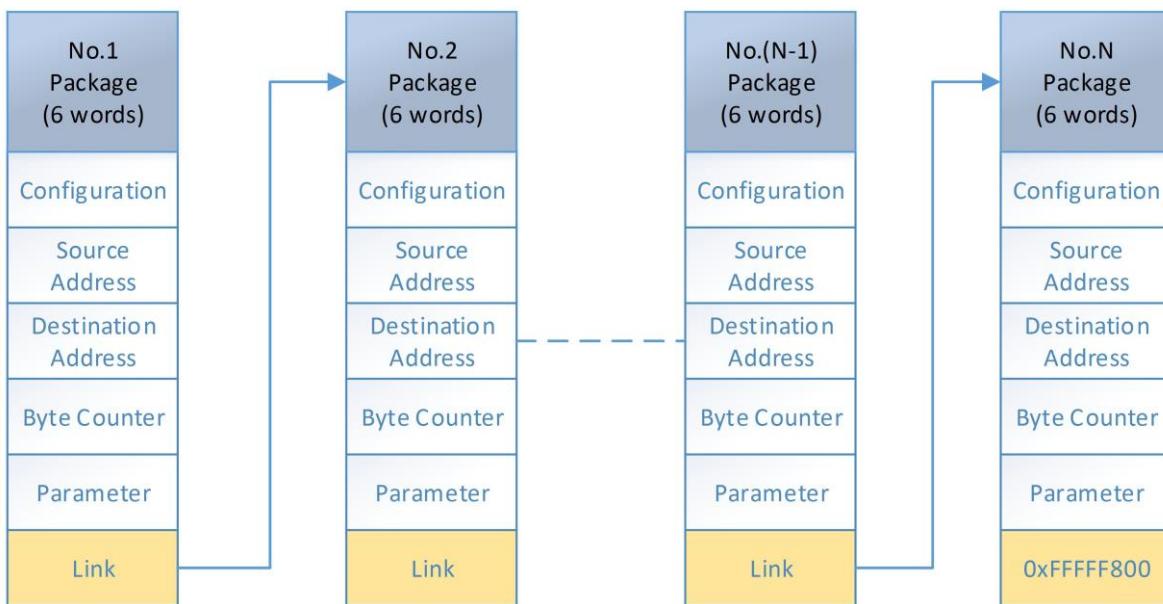
**Destination Address:** Configure the transferred destination address.

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

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

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

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

**Figure 3- 20. DMA Chain Transfer**

### 3.9.3.5. Interrupt

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

### 3.9.3.6. Security

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

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

### 3.9.3.7. Clock Gating

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

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

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

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

### 3.9.3.8. Transfer Mode

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

#### (1) Wait Mode

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

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

#### (2) Handshake Mode

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

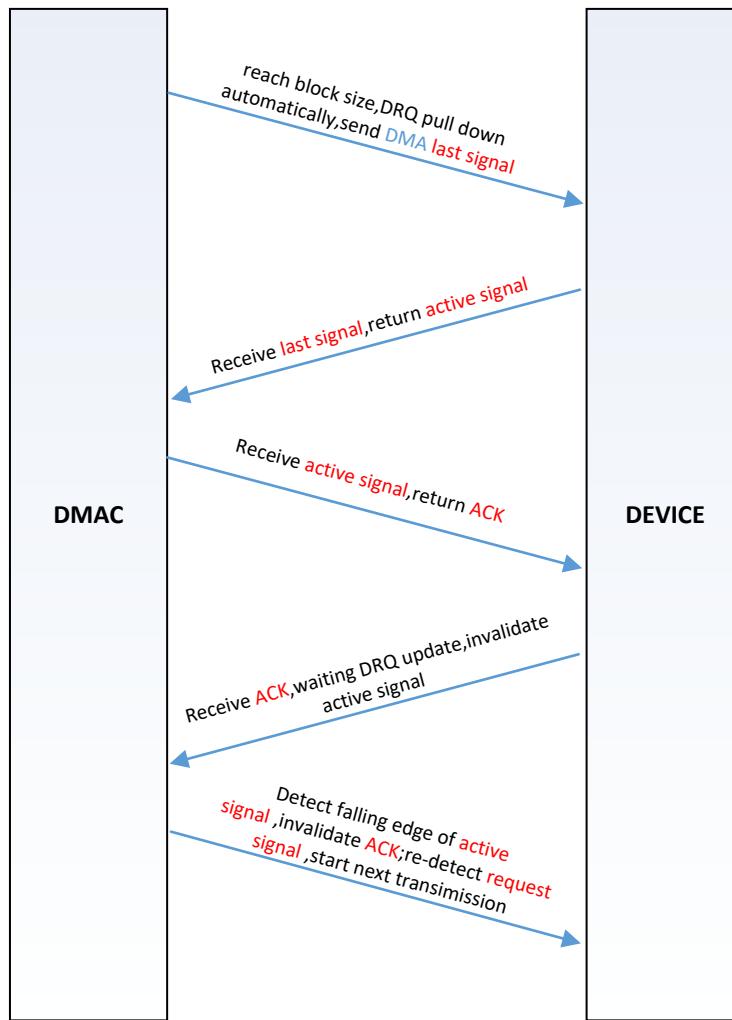


Figure 3- 21. DMA Transfer Mode

### 3.9.3.9. Auto-alignment Function

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

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

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

### 3.9.3.10. Operating Mode

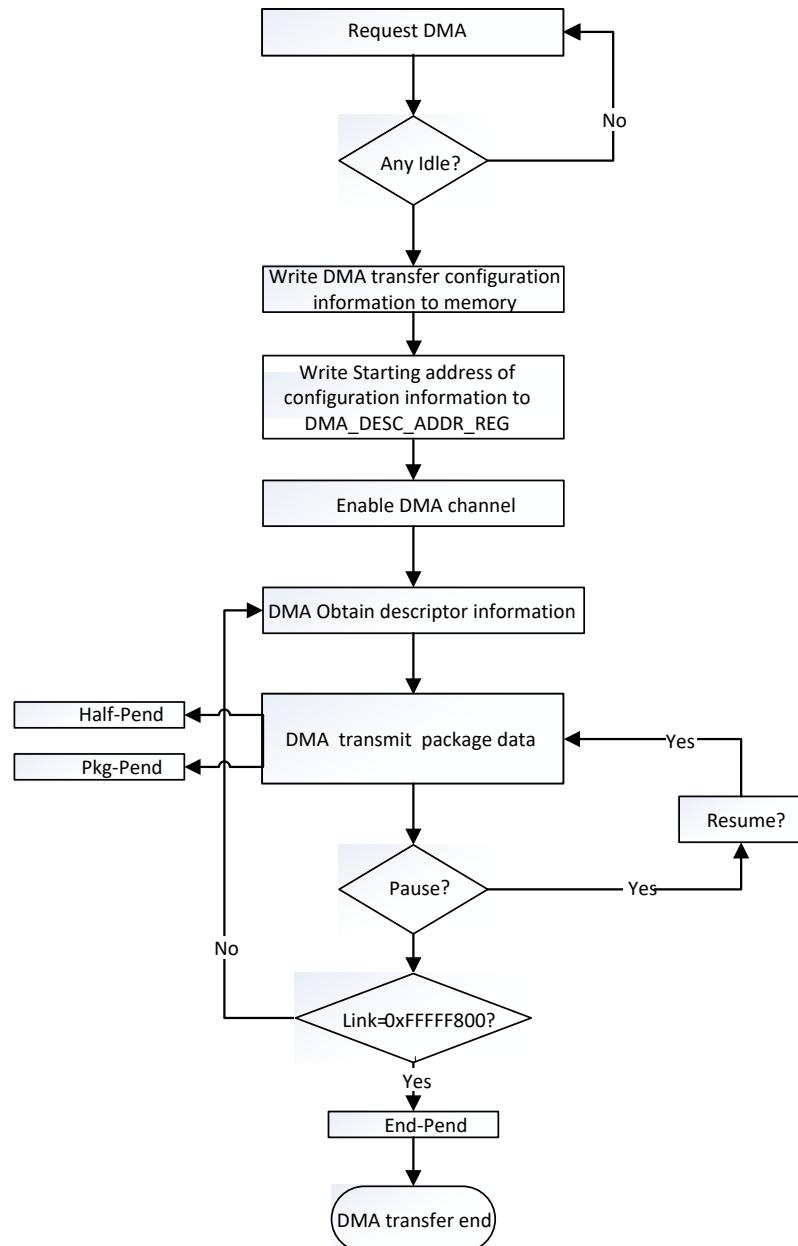
#### 3.9.3.10.1. DMA Clock Control

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

#### 3.9.3.10.2. DMA Transfer Process

The DMA transfer process is as follows.

- (1) Request DMA channel, and judge the idle state of the channel by the enable or disable of DMA channel.
- (2) Write the descriptor with 6-word into memory, the descriptor must be word-aligned. Refer to 3.9.3.4 DMA descriptor in detail.
- (3) Write the start address of storing descriptor to **DMA\_DESC\_ADDR\_REG**.
- (4) Enable DMA channel, and write the corresponding channel to **DMA\_EN\_REG**.
- (5) DMA obtains the descriptor information.
- (6) Start to transmit a package ,when half package is completed, DMA sends **Half Package Transfer Interrupt**; when total package is completed, DMA sends **Package End Transfer Interrupt**. These interrupt status can be read by **DMA\_IRQ\_PEND\_REG**.
- (7) Set **DMA\_PAU\_REG** to pause or resume the data transfer.
- (8) After completed the total package transfer, DMA decides to start the next package transfer or end the transfer by the link of the descriptor. If the link is 0xFFFFF800, the transfer ends; if the link is other value, the next package starts to transmit. When the transfer ends, DMA sends **Queue End Transfer Interrupt**.
- (9) Disable the DMA channel.



**Figure 3- 22. DMA Transfer Process**

### 3.9.3.10.3. DMA Interrupt

- (1) Enable interrupt: write the corresponding interrupt enable of **DMA\_IRQ\_EN\_REG**, when the corresponding interrupt condition is satisfied, the corresponding interrupt generates.
- (2) After enter the interrupt process, write **DMA\_IRQ\_PEND\_REG** to clear the interrupt pending, and execute the process of waiting for the interrupt.
- (3) Resume the interrupt and continue to execute the interrupted process.

### 3.9.4. Programming Guidelines

- (1) The transfer width of IO type device is consistent with the offset of start address.
- (2) MBUS protocol does not support read operation of non-integer word, so for non-integer word read operation, device must ignore redundant inconsistent data between data width and configuration, that is, the device of non-integer word must interpret DMA demand through its FIFO width instead of read demand width.
- (3) When the DMA transfer is paused, this is equivalent to invalid DRQ. Because DMA transfer command has a certain time delay, DMA will not stop transfer immediately until the current command and the command in Arbiter finished, at most 32byte data.

DMA application example :

```
writel(0x00000000, mem_address + 0x00); //Setting configuration, mem_address must be word-aligned  
writel(0x00001000, mem_address + 0x04); // Setting the start address for the source device  
writel(0x20000000, mem_address + 0x08); //Setting the start address for the destination device  
writel(0x00000020, mem_address + 0x0C); // Setting data package size  
writel(0x00000000, mem_address + 0x10); //Setting parameter  
writel(0xFFFFF800, mem_address + 0x14); //Setting the start address for the next descriptor  
writel(mem_address, 0x01C02000+ 0x100 + 0x08); //Setting the start address for the DMA channel0 descriptor  
do{  
    If(mem_address == readl(0x01C02000 + 0x100 + 0x08));  
    break;  
}while(1); //Make sure writing operation valid  
writel(0x00000001, 0x01C02000 + 0x100 + 0x00); // Enable DMA channel0 transfer
```

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

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

### 3.9.5. Register List

Module Name	Base Address
DMA	0x03002000

Register Name	Offset	Description
DMA_IRQ_EN_REG0	0x0000	DMA IRQ Enable Register 0
DMA_IRQ_EN_REG1	0x0004	DMA IRQ Enable Register 1
DMA_IRQ_PEND_REG0	0x0010	DMA IRQ Pending Register 0
DMA_IRQ_PEND_REG1	0x0014	DMA IRQ Pending Register 1
DMA_SEC_REG	0x0020	DMA Security Register
DMA_AUTO_GATE_REG	0x0028	DMA Auto Gating Register
DMA_STA_REG	0x0030	DMA Status Register
DMA_EN_REG	0x0100+N*0x0040	DMA Channel Enable Register (N=0~9)
DMA_PAU_REG	0x0100+N*0x0040+0x0004	DMA Channel Pause Register(N=0~9)
DMA_DESC_ADDR_REG	0x0100+N*0x0040+0x0008	DMA Channel Start Address Register(N=0~9)
DMA_CFG_REG	0x0100+N*0x0040+0x000C	DMA Channel Configuration Register(N=0~9)
DMA_CUR_SRC_REG	0x0100+N*0x0040+0x0010	DMA Channel Current Source Register(N=0~9)
DMA_CUR_DEST_REG	0x0100+N*0x0040+0x0014	DMA Channel Current Destination Register(N=0~9)
DMA_BCNT_LEFT_REG	0x0100+N*0x0040+0x0018	DMA Channel Byte Counter Left Register(N=0~9)
DMA_PARA_REG	0x0100+N*0x0040+0x001C	DMA Channel Parameter Register(N=0~9)
DMA_MODE_REG	0x0100+N*0x0040+0x0028	DMA Mode Register(N=0~9)
DMA_FDESC_ADDR_REG	0x0100+N*0x0040+0x002C	DMA Former Descriptor Address Register(N=0~9)
DMA_PKG_NUM_REG	0x0100+N*0x0040+0x0030	DMA Package Number Register(N=0~9)

### 3.9.6. Register Description

#### 3.9.6.1. 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
28	R/W	0x0	DMA7_HLAF_IRQ_EN DMA 7 Half Package Transfer Interrupt Enable

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

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

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

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

### 3.9.6.3. DMA IRQ Pending Status Register 0 (Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: DMA_IRQ_PEND_REG0
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30	R/W1C	0x0	DMA7_QUEUE_IRQ_PEND DMA 7 Queue End Transfer Interrupt Pending. Setting 1 to the bit will

			clear it. 0: No effect 1: Pending
29	R/W1C	0x0	DMA7_PKG_IRQ_PEND DMA 7 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
28	R/W1C	0x0	DMA7_HLAF_IRQ_PEND DMA 7 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
27	/	/	/
26	R/W1C	0x0	DMA6_QUEUE_IRQ_PEND DMA 6 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
25	R/W1C	0x0	DMA6_PKG_IRQ_PEND DMA 6 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
24	R/W1C	0x0	DMA6_HLAF_IRQ_PEND DMA 6 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
23	/	/	/
22	R/W1C	0x0	DMA5_QUEUE_IRQ_PEND DMA 5 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
21	R/W1C	0x0	DMA5_PKG_IRQ_PEND DMA 5 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
20	R/W1C	0x0	DMA5_HLAF_IRQ_PEND DMA 5 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending

19	/	/	/
18	R/W1C	0x0	DMA4_QUEUE_IRQ_PEND DMA 4 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
17	R/W1C	0x0	DMA4_PKG_IRQ_PEND DMA 4 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
16	R/W1C	0x0	DMA4_HLAF_IRQ_PEND DMA 4 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
15	/	/	/
14	R/W1C	0x0	DMA3_QUEUE_IRQ_PEND DMA 3 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending.
13	R/W1C	0x0	DMA3_PKG_IRQ_PEND DMA 3 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
12	R/W1C	0x0	DMA3_HLAF_IRQ_PEND DMA 3 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
11	/	/	/
10	R/W1C	0x0	DMA2_QUEUE_IRQ_PEND DMA 2 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
9	R/W1C	0x0	DMA2_PKG_IRQ_PEND DMA 2 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
8	R/W1C	0x0	DMA2_HLAF_IRQ_PEND DMA 2 Half Package Transfer Interrupt Pending. Setting 1 to the bit will

			clear it. 0: No effect 1: Pending
7	/	/	/
6	R/W1C	0x0	DMA1_QUEUE_IRQ_PEND DMA 1 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
5	R/W1C	0x0	DMA1_PKG_IRQ_PEND DMA 1 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
4	R/W1C	0x0	DMA1_HLA_F_IRQ_PEND DMA 1 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
3	/	/	/
2	R/W1C	0x0	DMA0_QUEUE_IRQ_PEND DMA 0 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
1	R/W1C	0x0	DMA0_PKG_IRQ_PEND DMA 0 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
0	R/W1C	0x0	DMA0_HLA_F_IRQ_PEND DMA 0 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending

#### 3.9.6.4. DMA IRQ Pending Status Register 1 (Default Value: 0x0000\_0000)

Offset:0x0014			Register Name: DMA_IRQ_PEND_REG0
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6	R/W1C	0x0	DMA9_QUEUE_IRQ_PEND DMA 9 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it.

			0: No effect 1: Pending
5	R/W1C	0x0	DMA9_PKG_IRQ_PEND DMA 9 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
4	R/W1C	0x0	DMA9_HLAF_IRQ_PEND DMA 9 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
3	/	/	/
2	R/W1C	0x0	DMA8_QUEUE_IRQ_PEND DMA 8 Queue End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
1	R/W1C	0x0	DMA8_PKG_IRQ_PEND DMA 8 Package End Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending
0	R/W1C	0x0	DMA8_HLAF_IRQ_PEND DMA 8 Half Package Transfer Interrupt Pending. Setting 1 to the bit will clear it. 0: No effect 1: Pending

### 3.9.6.5. DMA Security Register (Default Value: 0x0000\_0000)

Offset:0x0020			Register Name: DMA_SEC_REG
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9	R/W	0x0	DMA9_SEC DMA channel 9 security 0: Secure 1: Non-secure
8	R/W	0x0	DMA8_SEC DMA channel 8 security 0: Secure 1: Non-secure
7	R/W	0x0	DMA7_SEC DMA channel 7 security

			0: Secure 1: Non-secure
6	R/W	0x0	DMA6_SEC DMA channel 6 security 0: Secure 1: Non-secure
5	R/W	0x0	DMA5_SEC DMA channel 5 security 0: Secure 1: Non-secure
4	R/W	0x0	DMA4_SEC DMA channel 4 security 0: Secure 1: Non-secure
3	R/W	0x0	DMA3_SEC DMA channel 3 security 0: Secure 1: Non-secure
2	R/W	0x0	DMA2_SEC DMA channel 2 security 0: Secure 1: Non-secure
1	R/W	0x0	DMA1_SEC DMA channel 1 security 0: Secure 1: Non-secure
0	R/W	0x0	DMA0_SEC DMA channel 0 security 0: Secure 1: Non-secure

### 3.9.6.6. DMA Auto Gating Register (Default Value: 0x0000\_0000)

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

0	R/W	0x0	DMA_CHAN_CIRCUIT DMA channel circuit auto gating bit 0: Auto gating enable 1: Auto gating disable
---	-----	-----	--

**NOTE**

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

### 3.9.6.7. DMA Status Register (Default Value: 0x0000\_0000)

Offset:0x0030			Register Name: DMA_STA_REG
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30	R	0x0	MBUS FIFO Status 0: Empty 1: Not Empty
29:10	/	/	/
9	R	0x0	DMA9_STATUS DMA Channel 9 Status 0: Idle 1: Busy
8	R	0x0	DMA8_STATUS DMA Channel 8 Status 0: Idle 1: Busy
7	R	0x0	DMA7_STATUS DMA Channel 7 Status 0: Idle 1: Busy
6	R	0x0	DMA6_STATUS DMA Channel 6 Status 0: Idle 1: Busy
5	R	0x0	DMA5_STATUS DMA Channel 5 Status 0: Idle 1: Busy
4	R	0x0	DMA4_STATUS DMA Channel 4 Status 0: Idle 1: Busy
3	R	0x0	DMA3_STATUS DMA Channel 3 Status 0: Idle

			1: Busy
2	R	0x0	DMA2_STATUS DMA Channel 2 Status 0: Idle 1: Busy
1	R	0x0	DMA1_STATUS DMA Channel 1 Status 0: Idle 1: Busy
0	R	0x0	DMA0_STATUS DMA Channel 0 Status 0: Idle 1: Busy

### 3.9.6.8. DMA Channel Enable Register (Default Value: 0x0000\_0000)

Offset:0x0100+N*0x0040 (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

### 3.9.6.9. 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

### 3.9.6.10. 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
31:0	R/W	0x0	DMA_DESC_ADDR DMA Channel Descriptor Address The Descriptor Address must be word-aligned.

### 3.9.6.11. 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
31:27	/	/	/
26:25	R	0x0	DMA_DEST_DATA_WIDTH DMA Destination Data Width 00: 8-bit 01: 16-bit 10: 32-bit 11: 64-bit
24	R	0x0	DMA_ADDR_MODE DMA Destination Address Mode 0: Linear Mode 1: IO Mode
23:22	R	0x0	DMA_DEST_BLOCK_SIZE DMA Destination Block Size 00: 1 01: 4 10: 8 11: 16
21:16	R	0x0	DMA_DEST_DRQ_TYPE DMA Destination DRQ Type The details in DRQ Type and Port Corresponding Relation.
15:11	/	/	/
10:9	R	0x0	DMA_SRC_DATA_WIDTH DMA Source Data Width 00: 8-bit 01: 16-bit 10: 32-bit 11: 64-bit
8	R	0x0	DMA_SRC_ADDR_MODE DMA Source Address Mode 0: Linear Mode 1: IO Mode
7:6	R	0x0	DMA_SRC_BLOCK_SIZE DMA Source Block Size 00: 1 01: 4 10: 8 11: 16
5:0	R	0x0	DMA_SRC_DRQ_TYPE

			DMA Source DRQ Type The details in DRQ Type and Port Corresponding Relation.
--	--	--	---

### 3.9.6.12. 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.

### 3.9.6.13. 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.

### 3.9.6.14. 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.

### 3.9.6.15. DMA Channel Parameter Register (Default Value: 0x0000\_0000)

Offset:0x0100+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

### 3.9.6.16. DMA Mode Register (Default Value: 0x0000\_0000)

Offset:0x0100+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	/	/	/

### 3.9.6.17. 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.

### 3.9.6.18. 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.

## 3.10. Thermal Sensor Controller

### 3.10.1. Overview

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

The Thermal Sensor Controller(THS) embeds one thermal sensor located in CPU. The thermal sensor can generate interrupt to SW to lower temperature via DVFS, on reaching a certain thermal threshold.

The THS has the following features:

- Temperature Accuracy :  $\pm 3^{\circ}\text{C}$  from  $0^{\circ}\text{C}$  to  $+100^{\circ}\text{C}$ ,  $\pm 5^{\circ}\text{C}$  from  $-25^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$
- Power supply voltage: 1.8V
- Averaging filter for thermal sensor reading
- Supports over-temperature protection interrupt and over-temperature alarm interrupt

### 3.10.2. Block Diagram

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

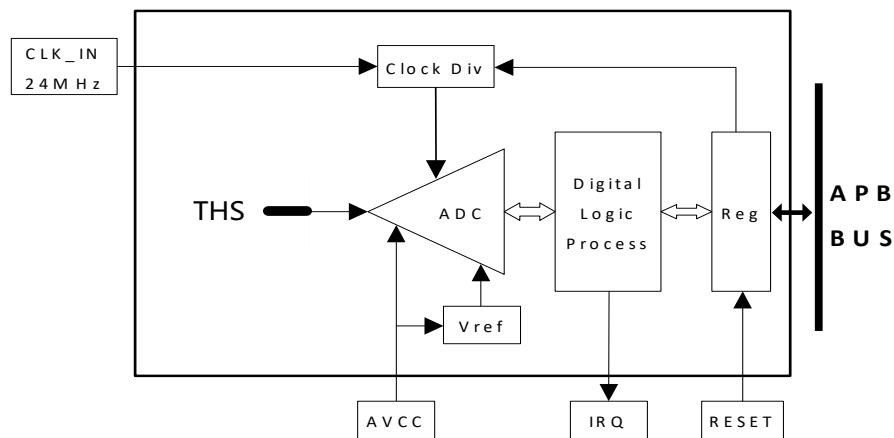


Figure 3- 23. Thermal Sensor Controller Block Diagram

### 3.10.3. Operations and Functional Descriptions

#### 3.10.3.1. Clock Sources

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

**Table 3- 9. Thermal Sensor Controller Clock Sources**

Clock Sources	Description
HOSC	HOSC
PLL_24MHz	PLL_24MHz(default)

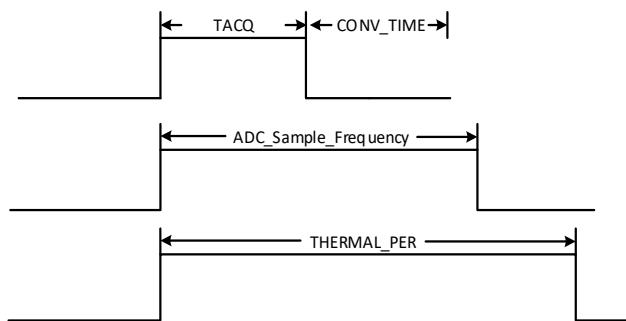
### 3.10.3.2. Timing Requirements

CLK\_IN = 24MHz

CONV\_TIME(Conversion Time) =  $1/(24\text{MHz}/14\text{Cycles}) = 0.583 \text{ (us)}$

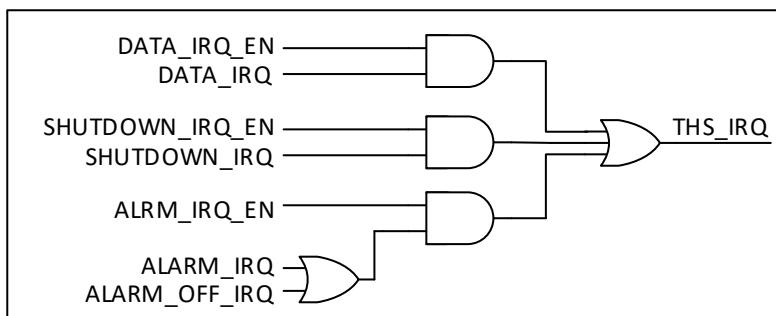
TACQ >  $1/(24\text{MHz}/24\text{Cycles})$

THERMAL\_PER > ADC Sample Frequency > TACQ+CONV\_TIME

**Figure 3- 24. Thermal Sensor Time Requirement**

### 3.10.3.3. Interrupt

The THS has four interrupt sources, such as DATA\_IRQ , SHUTDOWN\_IRQ, ALARM\_IRQ and ALARM\_OFF\_IRQ. Figure 3-25 shows the thermal sensor interrupt sources.

**Figure 3- 25. Thermal Sensor Controller Interrupt Source**

When temperature is higher than Alarm\_Threshold, ALARM\_IRQ is generated. When temperature is lower than Alarm\_Off\_Threshold, ALARM\_OFF\_IRQ is generated. ALARM\_OFF\_IRQ is fall edge trigger.

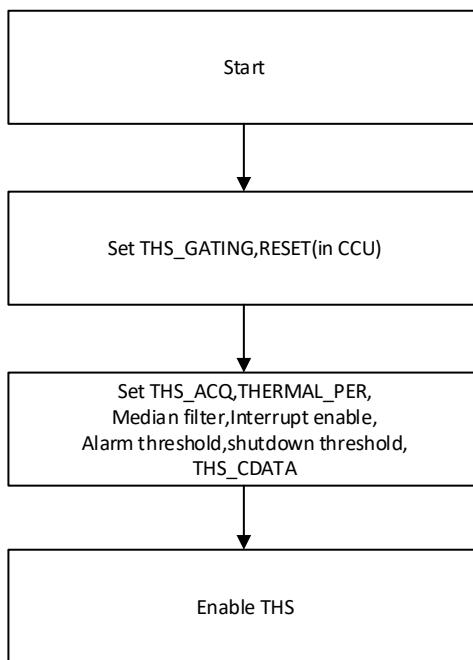
### 3.10.3.4. THS Temperature Conversion Formula

$T = (\text{sensor\_data} - 2794) / (-14.882)$ , the unit of T is Celsius.

sensor\_data: read from sensor data register.

### 3.10.4. Programming Guidelines

The initial process of the THS is as follows.



**Figure 3- 26. THS Initial Process**

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

Before enabling THS, read EFUSE value and write the value to **THS\_CDATA**.

#### (1).Query Mode

Step1: Write 0x1 to the bit16 of **THS\_BGR\_REG** to dessert reset.

Step2: Write 0x1 to the bit0 of **THS\_BGR\_REG** to open THS clock.

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

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

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

Step6: Write 0x1 to the bit2 of **THS\_FILTER** to enable temperature convert filter.

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

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

Step9: Write 0x1 to the bit[0] Of **THS\_EN** to enable THS.

Step10: Read the bit[0] of **THS\_DATA\_INTS**, if is 1, temperature conversion is complete.

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

## (2). Interrupt Mode

Step1: Write 0x1 to the bit16 of **THS\_BGR\_REG** to dessert reset.

Step2: Write 0x1 to the bit0 of **THS\_BGR\_REG** to open THS clock.

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

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

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

Step6: Write 0x1 to the bit2 of **THS\_FILTER** to enable temperature convert filter.

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

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

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

Step10: Set GIC interface based on IRQ 81, write the bit[17] of the **0x03021108** register to 0x1.

Step11: Put interrupt handler address into interrupt vector table.

Step12: Write 0x1 to the bit[0] of **THS\_EN** to enable THS.

Step13: Read the bit[0] of **THS\_DATA\_INTS**, if is 1, temperature conversion is complete.

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

## 3.10.5. Register List

Module Name	Base Address
Thermal Sensor	0x05070400

Register Name	Offset	Description
THS_CTRL	0x0000	THS Control Register
THS_EN	0x0004	THS Enable Register
THS_PER	0x0008	THS Period Control Register
THS_DATA_INTC	0x0010	THS Data Interrupt Control Register
THS_SHUT_INTC	0x0014	THS Shut Interrupt Control Register
THS_ALARM_INTC	0x0018	THS Alarm Interrupt Control Register
THS_DATA_INTS	0x0020	THS Data Interrupt Status Register
THS_SHUT_INTS	0x0024	THS Shut Interrupt Status Register
THS_ALARMO_INTS	0x0028	THS Alarm off Interrupt Status Register
THS_ALARM_INTS	0x002C	THS Alarm Interrupt Status Register
THS_FILTER	0x0030	THS Median Filter Control Register
THS_ALARM_CTRL	0x0040	THS Alarm Threshold Control Register
THS_SHUTDOWN_CTRL	0x0080	THS Alarm Threshold Control Register
THS_CDATA	0x00A0	THS Calibration Data
THS_DATA	0x00C0	THS Data Register

### 3.10.6. Register Description

#### 3.10.6.1. THS Control Register(Default Value : 0x01DF\_002F)

Offset: 0x0000			Register Name: THS_CTRL
Bit	Read/Write	Default/Hex	Description
31:16	R/W	0x1DF	FS_DIV ADC Sample Frequency Divider CLK_IN/(N+1) , N > 0x17 The default value indicates 50 kHz.
15:0	R/W	0x2F	TACQ ADC Acquire Time CLK_IN/(n+1) The default value indicates 2us.

#### 3.10.6.2. THS Enable Register(Default Value : 0x0000\_0000)

Offset: 0x0004			Register Name: THS_EN
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	THS0_EN Enable temperature measurement sensor0 0:Disable 1:Enable

#### 3.10.6.3. THS Period Control Register(Default Value: 0x0003\_A000)

Offset: 0x0008			Register Name: THS_PER
Bit	Read/Write	Default/Hex	Description
31:12	R/W	0x3A	THERMAL_PER 4096*(n+1)/CLK_IN The default value indicates 10ms.
11:0	/	/	/

#### 3.10.6.4. THS Data Interrupt Control Register(Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: THS_DATA_INTC
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	THS_DATA_IRQ_EN Selects temperature measurement data of sensor

			0:Disable 1:Enable
--	--	--	-----------------------

### 3.10.6.5. THS Shut Interrupt Control Register(Default Value: 0x0000\_0000)

Offset: 0x0014			Register Name: THS_SHUT_INTC
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	SHUT_INT_EN Selects shutdown interrupt for sensor 0:Disable 1:Enable

### 3.10.6.6. THS Alarm Interrupt Control Register(Default Value: 0x0000\_0000)

Offset: 0x0018			Register Name: THS_ALARM_INTC
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x0	ALARM_INT_EN Selects alarm interrupt for sensor 0:Disable 1:Enable

### 3.10.6.7. THS Data Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: THS_DATA_INTS
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W1C	0x0	THS_DATA_IRQ_STS Data interrupt status for sensor Write '1' to clear this interrupt.

### 3.10.6.8. THS Shut Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0024			Register Name: THS_SHUT_INTS
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W1C	0x0	SHUT_INT_STS Shutdown interrupt status for sensor0 Write '1' to clear this interrupt .

### 3.10.6.9. THS Alarm Off Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0028			Register Name: THS_ALARM_INTS
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W1C	0x0	ALARM_OFF_STS Alarm interrupt off pending for sensor Write '1' to clear this interrupt.

### 3.10.6.10. THS Alarm Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x002C			Register Name: THS_ALARM_INTS
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W1C	0x0	ALARM_INT_STS Alarm interrupt pending for sensor Write '1' to clear this interrupt.

### 3.10.6.11. Median Filter Control Register(Default Value: 0x0000\_0001)

Offset: 0x0030			Register Name: THS_FILTER
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2	R/W	0x0	FILTER_EN Filter Enable 0: Disable 1: Enable
1:0	R/W	0x1	FILTER_TYPE Average Filter Type 00: 2 01: 4 10: 8 11: 16

### 3.10.6.12. THS Alarm Threshold Control Register(Default Value: 0x05A0\_0684)

Offset: 0x0040			Register Name: THS_ALARM_CTRL
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/

27:16	R/W	0x5A0	ALARM_T_HOT Thermal Sensor alarm threshold for hot temperature
15:12	/	/	/
11:0	R/W	0x684	ALARM_T_HYST Thermal Sensor alarm threshold for hysteresis temperature

### 3.10.6.13. THS Shutdown Threshold Control Register (Default Value: 0x04E9\_04E9)

Offset: 0x0080			Register Name: THS_SHUTDOWN_CTRL
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
11:0	R/W	0x4E9	SHUT_T_HOT Thermal Sensor shutdown threshold for hot temperature

### 3.10.6.14. THS Calibration Data Register (Default Value: 0x0800\_0800)

Offset: 0x00A0			Register Name: THS_CDATA
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:0	R/W	0x800	THS_CDATA Thermal Sensor calibration data

### 3.10.6.15. THS Data Register(Default Value: 0x0000\_0000)

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

## 3.11. PSI

### 3.11.1. Overview

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

### 3.11.2. Block Diagram

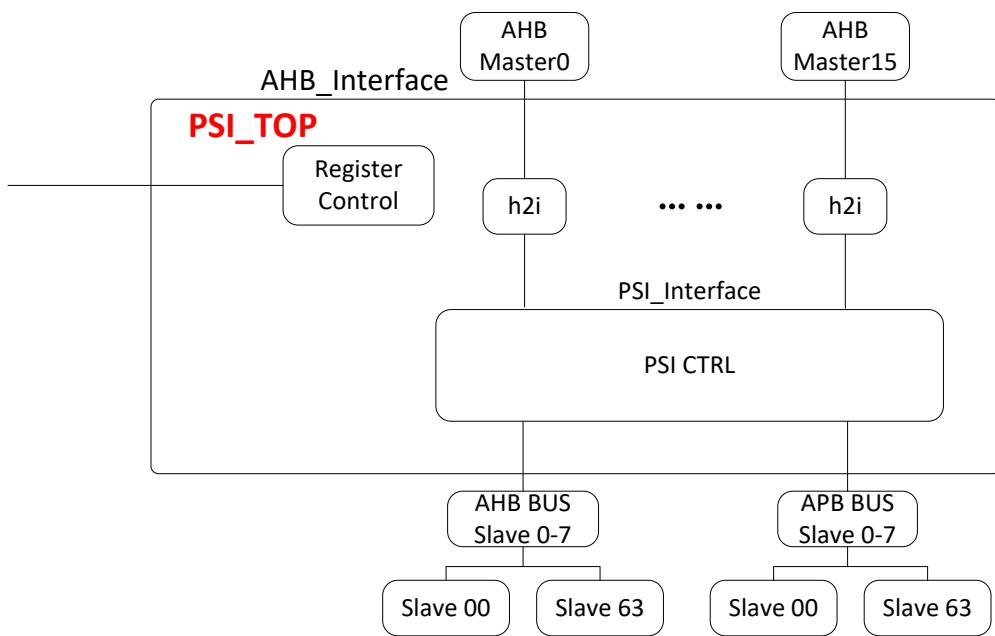


Figure 3- 27. PSI Block Diagram

## 3.12. VAD

### 3.12.1. Overview

The voice activity detector(VAD) can reduce the power consumption of voice ICs at idle status while not effecting voice processing performance.

The VAD has the following features:

- Supports 3 I2S interfaces, 1 DMIC PDM audio transfer interface
- Supported sample rate: 16 kHz, 48 kHz
- Supports voice detection module based on energy recognition
- Supports 4 working status: idle, wait, run, normal
- Supports voice activity: idle-> energy wake-up
- Supports 128KB SRAM used to store audio data
- Supports hardware interpolation for audio data channel numbers, including 5 modes: 2chs-4chs, 2chs-6chs, 2chs-8chs, 4chs-6chs, 4chs-8chs
- Supports secure protection for register configuration
- Supports secure protection of VAD\_SRAM
- Supports VAD bypass function, after bypass, VAD\_SRAM can be used as on-chip SRAM to work individual

### 3.12.2. Block Diagram

The overall design of VAD consists of two major aspects, one is the generation of wake-up interrupts, and the other is the transmission of voice data after wake-up. The former is to wake up the processor, and the later is to allow the processor to acquire audio data in time for keyword recognition.

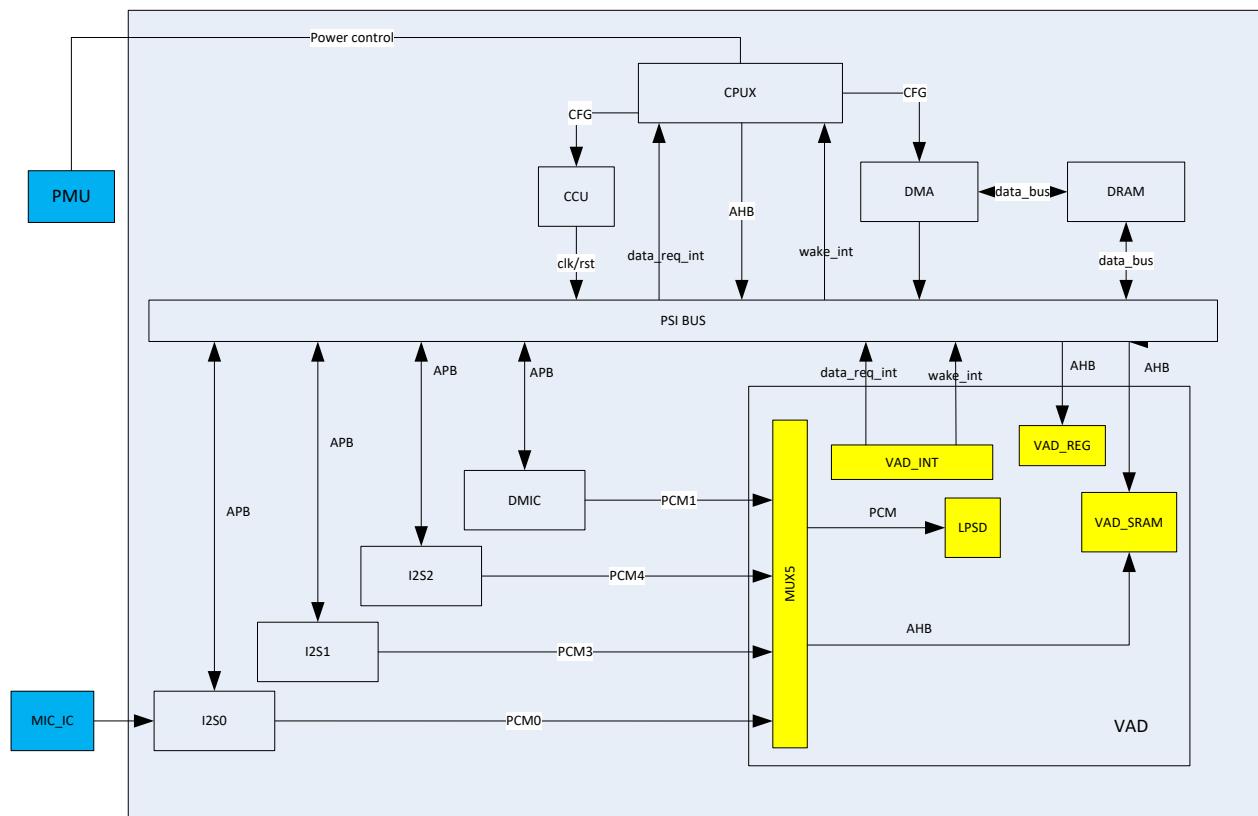
The data source of VAD may be multiple mics, such as the I2S interface of the microphone array IC, the digital mic of DMIC. The VAD can currently support the input of 4 pcm codes, and the VAD can perform software configuration to select which audio source is used as input, it also supports one AHB configuration interface. When the VAD recognizes the human voice successfully in CPU low-power mode, an interrupt can be generated and reported to CPU.

SRAM is used to store the audio data buffer during power consumption. It is 128KB in size and supports byte read and write. The data source is in parallel with the VAD's audio data source and is also the audio data after the MUX. At the same time, SRAM can be read and written by other master bytes of PSI, so there is a competition relationship between the above two interfaces, which needs to be punched. Here, the ready-made AHB arbitration is used.

After CPU wakes up, CPU here is initially CPUS, and after CPUS wakes up, the SRAM is directly read for speech recognition. After the speech recognition is correct, CPUS can wake up the CPUX, during which the audio data is stored in the SRAM. After waking up, CPUX configures DMA to carry audio data into the dram executive.

**NOTE**

Here, the size of the SRAM is required to cache the voice data in CPUX wakeup process in addition to the pre-awake voice data. VAD, MUX5, SRAM will be packaged together to form a VAD\_SUBSYS.



**Figure 3- 28. VAD System Architeture Diagram**

VAD\_REG: configures AHB register, register read/write, configuration value distribution.

LPSD: voice wake-up module. It can reach threshold according to audio data size, and over a period of time, wake-up interrupt will generate to wake-up CPU.

VAD\_INT: interrupt handle.

VAD\_SRAM: buffering data for audio data transmitted by VAD\_AD\_MUX.

VAD\_AD\_MUX: MUX selecting for audio input of VAD external, and protocol transformation for SRAM transferred data.

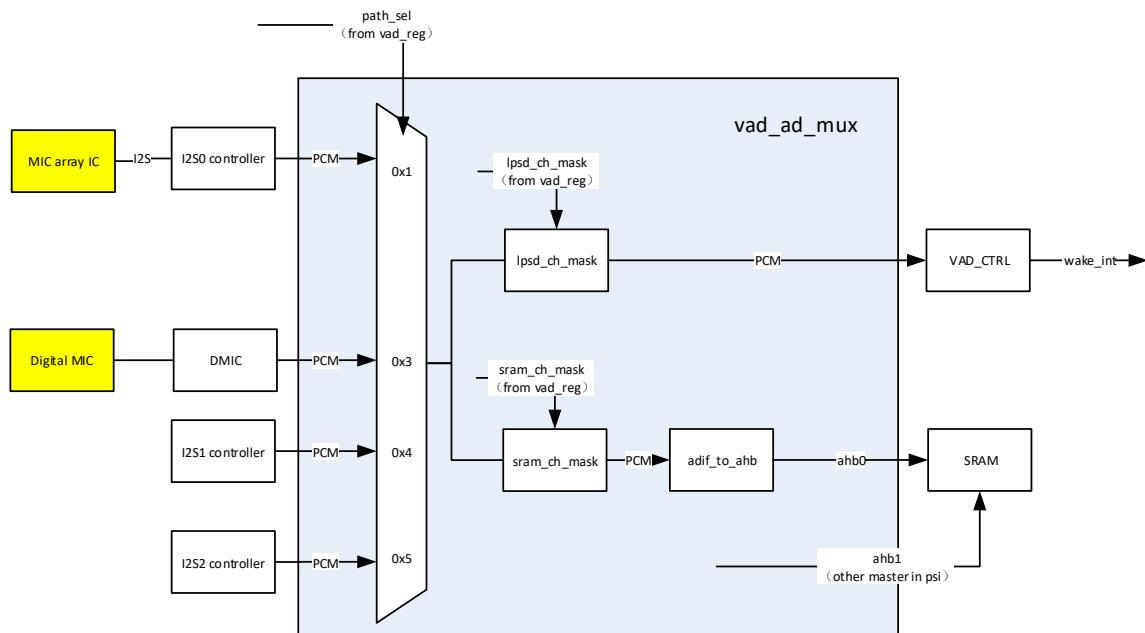
### 3.12.3. Operations and Functional Descriptions

#### 3.12.3.1. Audio Data Flow

The transmission path of voice data is as follows, and there are four audio input channels.

- The microphone array IC, DMIC and other four audio input sources are connected to one MUX5, and the software configuration selects which way to use as the audio input.

- After the Audio data source is selected, it is transmitted to the VAD in parallel for voice detection, and simultaneously transmitted to the SRAM for voice storage.
- After the VAD generates an interrupt, it is transferred to CPU, and CPU goes to the SRAM to acquire the voice data for processing.
- Reserved I2S1, I2S2 corresponds to PCM interface for subsequent expansion.



**Figure 3- 29. VAD Data Flow**

### 3.12.3.2. Clock and Reset

The VAD includes 2 VAD reset and AHB bus reset.

The VAD supports 4 clocks: vad\_clk, hclk, lpsd\_clk, test\_clk. Data path uses vad\_clk and lpsd\_clk clock domain, control logic uses hclk clock domain.

The specific clock domain control is divided as follows.

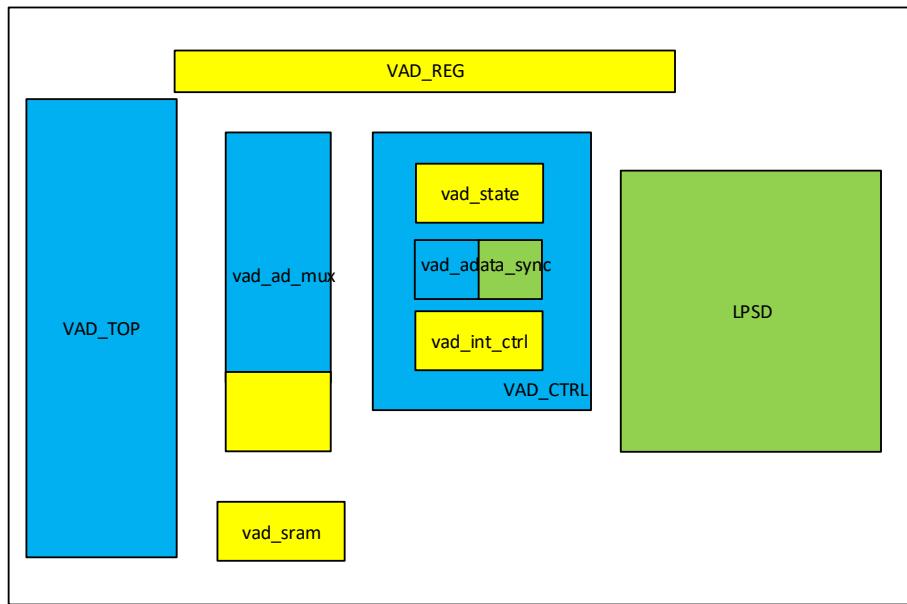
- Yellow part: hclk clock domain, 200MHz
- Blue part: vad\_clk clock domain, 100MHz, vad\_clk and pclk has the same frequency and phase, and different gating control.
- Green part: lpsd\_clk clock domain, 512kHz.

vad\_clk, lpsd\_clk, and hclk all support individual gating control. Each function clock supports mux selecting for test\_clk in scan\_mode.

The specific reset domain control is divided as follows.

- For reset, VAD\_REG divides into hrestn and vad\_rstn, SRAM configuration-related register is controlled by hrestn, other parts is controlled by vad\_rstn after hclk synchronization control.
- vad\_sram and pcm2ahb(yellow part in vad\_ad\_mux) is controlled by hrestn.

- The logic of vad\_clk clock domain and lpsd\_clk clock domain is controlled by vad\_rstn.



**Figure 3- 30. VAD Clock System**

### 3.12.4. Programming Guidelines

The initial process of VAD is as follows.

#### Step1 Configure LPSD related register

- (1) Configure VAD\_SRAM\_WAKE\_BACK\_DA\_REG(offset address: 0x0020) to determine how long LPSD generates wake\_int after the first energy sample. The recommended configuration is 0x00000005.
- (2) Configure VAD\_LPSD\_AD\_SYNC\_FC\_REG(offset address: 0x0028) to determine audio sample rate received by LPSD. The default value is 0x20, configured according to 512kHz clock frequency and 16kHz sample rate.
- (3) Configure VAD\_LPSD\_TH\_REG(offset address: 0x002C) to determine energy threshold detected by LPSD. The recommended configuration is 0x000004B0.
- (4) Configure VAD\_LPSD\_RRUN\_REG(offset address: 0x0030) to determine the start time of voice wake-up audio detection. The recommended configuration is 0x00000091.
- (5) Configure VAD\_LPSD\_RSTOP\_REG(offset address: 0x0034) to determine the stop time of voice wake-up audio detection. The recommended configuration is 0x000000AA.
- (6) Configure VAD\_LPSD\_ECNT\_REG(offset address: 0x0038) to determine how long LPSD generates wake\_int after the last energy sample. The recommended configuration is 0x00000032.

#### Step2 Configure SRAM related control register

- (1) Configure VAD\_SRAM\_POINT\_REG(offset address: 0x0004) and VAD\_SRAM\_SIZE\_REG(offset address: 0x0008) to determine the start address of VAD\_SRAM storage address and total size. The default start address is 0x0000, and the default SRAM size is 0x100.
- (2) Configure VAD\_SRAM\_STORE\_TH\_REG(offset address: 0x0014) to determine storage threshold at which an

interrupt is generated.

- (3) Configure VAD\_SRAM\_SEC\_REGION(offset address: 0x0044) to determine secure domain and non-secure domain of VAD\_SRAM. All are non-secure domain by default.
- (4) Configure VAD\_CTRL\_REG(offset address: 0x0000), setting the bit[6] to 1 to reset SRAM, after initialization, setting the bit[6] to 0 to de-assert SRAM. Reset can perform after SRAM basic initialization, only then the initialization configuration can load into SRAM control logic.
- (5) If using DMA, VAD\_SRAM\_AHB1\_TX\_TH\_REG(offset address: 0x0018) and VAD\_SRAM\_AHB1\_RX\_TH\_REG(offset address: 0x001C) need be configured. The threshold of DMA RX and TX channel is usually set to 1,4,8,16, which need align with block\_size of DMA.

### Step3 Configure VAD and audio data related register

- (1) Configure VAD\_AD\_PATH\_SEL\_REG(offset address: 0x0024) to determine which way the audio data source selects. Neither of them is selected.
- (2) Configure VAD\_SRAM\_CH\_MASK\_REG(offset address: 0x003C),
  - Configure the bit[25:21] to determine how many channels of data is the audio data source;
  - Configure the bit[20:16] to determine how many audio data channels VAD\_SRAM receives;
  - Configure the bit[15:0] to determine which channels are respectively . Suggest configuring receiving 2 channels data during voice wake-up, receiving 8 channels data after voice wake-up. Choose according to customer needs in application.
- (3) Configure VAD\_LPSD\_CH\_MASK\_REG(offset address: 0x0040),
  - Configure the bit[19:16] to determine how many audio data channels VAD\_LPSD receives;
  - Configure the bit[15:0] to determine which channels are respectively. Suggest receiving 1 channel data at any time.
- (4) In channel switching and hardware interpolation, configure VAD\_SRAM\_CH\_MASK\_REG(offset address: 0x003C),
  - Configure the bit[20:16] to determine how many audio data channels VAD\_SRAM receives;
  - Configure the bit[15:0] to determine which channels are respectively, the above two steps can proceed simultaneously;
  - Lastly, configure the bit[30] to enable channel switching;
- (5) Configure VAD\_INT\_MASK\_REG(offset address: 0x0054), configure the bit[1:0] to determine which interrupt to mask. All is non-masked by default.
- (6) Configure VAD\_CTRL\_REG(offset address: 0x0000),
  - Configure the bit[4] to determine whether DMA is used;
  - Configure the bit[5] to determine which mode DMA uses, suggest using DMA after CPUX wake-up;
  - Configure the bit[3:1] to set VAD state machine jump;
  - Configure the bit[0] to enable VAD.

#### 3.12.5. Register List

Module Name	Base Address	
VAD	0x05400000	

Register Name	Offset	Description
VAD_CTRL_REG	0x0000	VAD Control Register

VAD_SRAM_POINT_REG	0x0004	VAD SRAM Start Address Register
VAD_SRAM_SIZE_REG	0x0008	VAD SRAM Size Register
VAD_SRAM_RD_POINT_REG	0x000C	VAD SRAM Available Data Start Address Register
VAD_SRAM_RD_SIZE_REG	0x0010	VAD SRAM Available Data Size Register
VAD_SRAM_STORE_TH_REG	0x0014	VAD SRAM Store Threshold Register
VAD_SRAM_AHB1_TX_TH_REG	0x0018	VAD SRAM AHB1 TX Threshold Register
VAD_SRAM_AHB1_RX_TH_REG	0x001C	VAD SRAM AHB1 RX Threshold Register
VAD_SRAM_WAKE_BACK_DA_REG	0x0020	VAD SRAM Wake Up Data Register
VAD_AD_PATH_SEL_REG	0x0024	VAD SRAM Audio Path Select Register
VAD_LPSD_AD_SYNC_FC_REG	0x0028	VAD LPSD Audio Data Sync Frequency Register
VAD_LPSD_TH_REG	0x002C	VAD LPSD Threshold Register
VAD_LPSD_RRUN_REG	0x0030	VAD LPSD Ratio Run Register
VAD_LPSD_RSTOP_REG	0x0034	VAD LPSD Ratio Stop Register
VAD_LPSD_ECNT_REG	0x0038	VAD LPSD End Count Register
VAD_SRAM_CH_MASK_REG	0x003C	VAD SRAM Store Audio Channel Mask Register
VAD_LPSD_CH_MASK_REG	0x0040	VAD LPSD Get Audio Channel Mask Register
VAD_SRAM_SEC_REGION_REG	0x0044	VAD SRAM Secure Region Register
VAD_SRAM_PREF_DSIZ_REG	0x0048	VAD SRAM Pre Data Size Register
VAD_SRAM_DMA_TF_SIZE_REG	0x004C	VAD SRAM DMA Transfer Data Size Register
VAD_SRAM_DMA_TF_LAST_SIZE_REG	0x0050	VAD SRAM DMA Transfer Last Data Size Register
VAD_INT_ST_CLR_REG	0x0060	VAD Interrupt State/Interrupt Clear Register
VAD_INT_MASK_REG	0x0064	VAD Interrupt Mask Register
VAD_STAT_REG	0x0068	VAD State Register
VAD_DEBUG_REG	0x006C	VAD Debug Register

### 3.12.6. Register Description

#### 3.12.6.1. VAD Control Register(Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: VAD_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	AUDIO_DATA_SYNC_FRC Read/Write 1: Use 48kHz audio data sample rate Read/Write 0: Use 16kHz audio data sample rate The bit cannot support dynamic configuration, that is , the configuration is not allowed to change during data reception.
6	R/W	0x0	SRAM_RST Read/Write 1: Reset SRAM, clear SRAM-related pointer information, storage starts at the starting address. In reset status, SRAM-related pointer information does not count. Read/Write 0: De-assert reset DRAM, SRAM continues to transmit.

5	R/W	0x0	DMA_TYPE Read/Write 1: Use IO mode when external access to the master of VAD_SRAM Read/Write 0: Use MEM mode when external access to the master of VAD_SRAM The bit cannot support dynamic configuration, that is , the configuration is not allowed to change during DMA transfer.
4	R/W	0x0	DMA_EN Read/Write 1: Use DMA Read/Write 0: Not use DMA
3	R/W	0x0	CPUS_RD_DONE Read/Write 1: CPUS keyword not successful, the hardware can auto-cleared after WAIT. Read/Write 0: CPUS keyword successful When VAD_EN is closed, the hardware can clear configuration, that is, cpus_rd_done is 0. Each configuration takes effect only once. The configuration is invalid in WAIT status.
2	R/W	0x0	GO_ON_SLEEP Read/Write 1: CPUX need resume sleep, the hardware can auto-cleared after WAIT. Read/Write 0: CPUX need not resume sleep. When VAD_EN is closed, the hardware can clear configuration, that is, cpus_rd_done is 0. Each configuration takes effect only once. The configuration is invalid in WAIT status.
1	R/W	0x0	KEY_WORD_OK Read/Write 1: CPUS keyword recognition successful, the hardware can auto-cleared after NORMAL. Read/Write 0: CPUS keyword recognition has no result. When VAD_EN is closed, the hardware can clear configuration, that is, key_word_ok is 0. Each configuration takes effect only once. The configuration is invalid in NORMAL status.
0	R/W	0x0	VAD_EN Read/Write 1: VAD enable, all functions start. vad_en is 1, vad_done is 0. Read/Write 0: VAD disable, all functions close. vad_en is 0, vad_done is 1.

### 3.12.6.2. VAD SRAM Point Register(Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: VAD_SRAM_POINT_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	VAD_SRAM_POINT Starting address of audio data in sram The configuration is relevant with sram memory map, sram storage rollback starts from the address. The field must be word-aligned, and does not support dynamic

			configuration, that is, the configuration is not allowed to change during data reception.
--	--	--	---

### 3.12.6.3. VAD SRAM Size Register(Default Value: 0x0000\_0000)

Offset: 0x0008			Register Name: VAD_SRAM_SIZE_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x100	VAD_SRAM_SIZE SRAM size, 0KB to 256KB, default value is 256KB. Read/Write 0: 0KB Read/Write 256: 256KB The field does not support dynamic configuration, that is, the configuration is not allowed to change during data reception.

### 3.12.6.4. VAD SRAM RD Point Register (Default Value:0x0000\_0000)

Offset: 0x000C			Register Name: VAD_SRAM_RD_POINT_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	VAD_SRAM_RD_POINT Starting address of voice recognition The field indicates the starting address of CPU for voice recognition is informed after current VAD generates interrupt.

### 3.12.6.5. VAD SRAM RD Size Register (Default Value:0x0000\_0000)

Offset: 0x0010			Register Name: VAD_SRAM_RD_SIZE_REG
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	VAD_SRAM_RD_SIZE Data volume indication of voice recognition The field indicates how much valid data is stored by current VAD from starting address. (unit: bytes)

### 3.12.6.6. VAD SRAM Store Threshold Register (Default Value:0x0000\_0000)

Offset: 0x0014			Register Name: VAD_SRAM_STORE_TH_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	VAD_SRAM_STORE_TH SRAM data threshold The threshold determines when data_req_int is generated. When SRAM data is more than or equal to the threshold, data_req_int will

			<p>be generated.</p> <p>Here said that SRAM data indicates the counting data when detected human voice.</p> <p>In VAD disable, data_req_int can be generated based on threshold; in VAD enable, data_req_int can be generated based on threshold after wake_int generated.</p>
--	--	--	--

### 3.12.6.7. VAD SRAM AHB1 TX Threshold Register (Default Value:0x0000\_0000)

Offset: 0x0018			Register Name: VAD_SRAM_AHB1_TX_TH_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>VAD_SRAM_AHB1_TX_TH</p> <p>The field indicates channel of TX data on the basic of VAD_SRAM.</p> <p>The threshold determines when dma_tx_req is transmitted.</p> <p>When SRAM data is more than or equal to the threshold, dma_tx_req will be generated. (unit: byte)</p>

### 3.12.6.8. VAD SRAM AHB1 RX Threshold Register (Default Value:0x0000\_0000)

Offset: 0x001C			Register Name: VAD_SRAM_AHB1_RX_TH_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>VAD_SRAM_AHB1_RX_TH</p> <p>The field indicates channel of RX data on the basic of VAD_SRAM.</p> <p>The threshold determines when dma_rx_req is transmitted.</p> <p>When SRAM data is more than or equal to the threshold, dma_rx_req will be generated. (unit: byte)</p>

### 3.12.6.9. VAD SRAM Wake Back Data/LPSD Start\_Count Register (Default Value:0x0000\_0000)

Offset: 0x0020			Register Name: VAD_SRAM_WAKE_BACK_DA_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>VAD_SRAM_WAKE_BACK_DA</p> <p>The field indicates how many data before SRAM current storage address is used as CPU processing starting data in wake_int generation time.(Unit: audio frame number)</p> <p>The field also indicates how long interrupt is pulled-up after voice wakeup recognizes the first audio sample point. (Unit: audio frame number)</p> <p> <b>NOTE</b></p> <p><b>The configuration cannot exceed vad_sram area, the last configuration value need be converted to halfword unit.</b></p>

### 3.12.6.10. VAD Audio Path Select Register (Default Value:0x0000\_0000)

Offset: 0x0024			Register Name: VAD_AD_PATH_SEL_REG
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3:0	R/W	0x0	VAD_AD_PATH_SEL Audio data source select 0x0: No select any channel, non audio input 0x1: (MIC IC)connect to I2S/PCM 0 interface 0x2: audio codec, PCM interface 0x3: DMIC, PCM interface 0x4:connect to I2S/PCM1 interface 0x5: connect to I2S/PCM2 interface Other: No select any channel, non audio input The field cannot support dynamic configuration.

### 3.12.6.11. VAD LPSD Audio Data SYNC Frequency Register (Default Value: 0x0000\_0020)

Offset: 0x0028			Register Name: VAD_LPSD_AD_SYNC_FC_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x20	VAD_LPSD_AD_SYNC_FC LPSD receives sample rate of audio_data, its unit is based on LPSD_CLK magnification. The field is 512kHz LPSD_CLK, 16kHz sample rate by default. (32 times = 0x20) The field cannot support dynamic configuration.

### 3.12.6.12. VAD LPSD Threshold Register (Default Value: 0x0000\_0000)

Offset: 0x002C			Register Name: VAD_LPSD_TH_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	VAD_LPSD_TH Voice wakeup energy threshold, that is, how much energy is reached to generate wake_int. The lower 16bits only is valid. The field cannot support dynamic configuration.

### 3.12.6.13. VAD LPSD Ratio\_Run Register (Default Value: 0x0000\_0000)

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

31:0	R/W	0x0	VAD_LPSD_RRUN Starting time of voice activity detection (unit: LPSD clock cycle). The lower 8bits is valid. The field cannot support dynamic configuration.
------	-----	-----	--

### 3.12.6.14. VAD LPSD Ratio\_Stop Register (Default Value: 0x0000\_0000)

Offset: 0x0034			Register Name: VAD_LPSD_RSTOP_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	VAD_LPSD_RSTOP Ending time of voice activity detection (unit: LPSD clock cycle). The lower 8bits is valid. The field cannot support dynamic configuration.

### 3.12.6.15. VAD LPSD End\_Count Register (Default Value: 0x0000\_0000)

Offset: 0x0038			Register Name: VAD_LPSD_ECNT_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	VAD_LPSD_ECNT The field indicates how long interrupt is pulled down after voice wakeup recognizes the last audio sample point. (Unit: audio frame number) The field cannot support dynamic configuration.

### 3.12.6.16. VAD SRAM\_Store Audio Channel Mask Register (Default Value: 0x0000\_0000)

Offset: 0x003C			Register Name: VAD_SRAM_CH_MASK_REG
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30	R/W	0x0	VAD_CH_CHANGE_EN Channel interpolation enable 0x0: No channel interpolation 0x1: Start to perform channel interpolation, the bit can be configured along with VAD_SRAM_CH_MASK,VAD_SRAM_CH_NUM. When channel interpolation finished, that is, software reads all interpolation data, hardware will auto-clear the bit to 0.
29:26	R/W	0x0	VAD_CH_COM_NUM Audio data hardware interpolation mode configuration when DMA transfers SRAM data. 0000: No hardware interpolation 0001: convert 2-ch to 4-ch 0010: convert 2-ch to 6-ch 0011: convert 2-ch to 8-ch

			0100: convert 4-ch to 6-ch 0101: convert 4-ch to 8-ch Other: Reserved
25:21	R/W	0x0	VAD_AD_SRC_CH_NUM  How many channels of audio data is VAD audio source. Here supports 0x0 to 0x10, that is, 0-ch to 16-ch.
20:16	R/W	0x0	VAD_SRAM_CH_NUM  How many channels of audio data is received by VAD_SRAM. Here supports any channel number configuration from 0 to 16.
15: 0	R/W	0x0	VAD_SRAM_CH_MASK  16 audio channels mask enable.  Any channel, or any of 16 channels can be set to non-masked.  Xbit read/write 1: audio channel x enable, that is, channel x of I2S TDM protocol enable  Xbit read/write 0: audio channel x mask, that is, channel x of I2S TDM protocol mask

### 3.12.6.17. VAD\_LPSD\_GET Audio Channel Mask Register (Default Value: 0x0010\_0000)

Offset: 0x0040			Register Name: VAD_LPSD_CH_MASK_REG
Bit	Read/Write	Default/Hex	Description
31:21	/	/	/
20	R/W	0x1	VAD_LPSD_DCBLOCK_EN  Current LPSD eliminate DC element enable Read/write 1: Eliminate DC element enable Read/write 0: Eliminate DC element disable
19:16	R/W	0x0	VAD_LPSD_CH_NUM  Audio data channel number of current VAD_LPSD, only support 0 channel and 1 channel, and 0x0, 0x1 configuration value
15: 0	R/W	0x0	VAD_LPSD_CH_MASK  16 audio channels mask enable.  Any 0/1 of 16 channels can be set to non-masked.  Xbit read/write 1: audio channel x enable, that is, channel x of I2S TDM protocol enable  Xbit read/write 0: audio channel x mask, that is, channel x of I2S TDM protocol mask

### 3.12.6.18. VAD SRAM Secure Region Register (Default Value: 0x0000\_0000)

Offset: 0x0044			Register Name: VAD_SRAM_SEC_REGION_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R/W	0x0	VAD_SRAM_SEC_REGION  VAD SRAM secure domain configuration

			<p>The register divides VAD_SRAM into 2 parts. The range from address 0 to the configuration value is secure domain, the remaining domain is non-secure domain.</p> <p>The register can only be secure accessed.</p> <p>The default value is 0x0, that is, all domain of VAD_SRAM is non-secure. The maximum secure domain can be configured to 0xFFFF (256KB).</p>
--	--	--	---

### 3.12.6.19. VAD SRAM PRE Data Size Register (Default Value: 0x0000\_0000)

Offset: 0x0048			Register Name: VAD_SRAM_PRE_DSIZREG
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x0	<p>VAD_SRAM_PRE_DADA_SIZE</p> <p>VAD SRAM stores the amount of data that has been stored in the previous SRAM when switching the number of audio storage channels.</p> <p>The field is used to software integrate audio data with different channel count. (Unit: halfword)</p>

### 3.12.6.20. VAD DMA Transfer Data Size Register (Default Value: 0x0000\_0000)

Offset: 0x004C			Register Name: VAD_DMA_TF_SIZE_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x0	<p>VAD_DMA_TF_SIZE</p> <p>DMA transfer data size in VAD SRAM</p> <p>Only for DMA read VAD SRAM.(Unit: Byte)</p>

### 3.12.6.21. VAD DMA Transfer Last Data Size Register (Default Value: 0x0000\_0000)

Offset: 0x0050			Register Name: VAD_DMA_TF_LAST_SIZE_REG
Bit	Read/Write	Default/Hex	Description
31: 0	R	0x0	<p>VAD_DMA_TF_LAST_SIZE</p> <p>Remain data size in VAD SRAM after DMA transfer</p> <p>Only for DMA read VAD SRAM. (Unit: Byte)</p>

### 3.12.6.22. VAD Interrupt\_Status/Interrupt\_Clear Register (Default Value: 0x0000\_0000)

Offset: 0x0060			Register Name: VAD_INT_ST_CLR_REG
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W	0x0	<p>DATA_REQ_INT</p> <p>CPU read SRAM enable interrupt(the status is not controlled by mask)</p>

			Read 0: Not enable interrupt generated Read 1: Enable interrupt generated Write 0: No effect Write 1: Clear current enable interrupt
0	R/W	0x0	WAKE_INT Voice wakeup interrupt(the status is not controlled by mask) Read 0: Not wakeup interrupt generated Read 1: Wakeup interrupt generated Write 0: No effect Write 1: Clear current wakeup interrupt

### 3.12.6.23. VAD Interrupt Mask Register (Default Value: 0x0000\_0000)

Offset: 0x0064			Register Name: VAD_INT_MASK_REG
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1	R/W	0x0	DATA_REQ_INT_MASK CPU read sram interrupt output Read/Write 0: mask Read/Write 1: enable
0	R/W	0x0	WAKE_INT_MASK Wakeup interrupt output Read/Write 0: mask Read/Write 1: enable

### 3.12.6.24. VAD State Register (Default Value: 0x0000\_0002)

Offset: 0x0068			Register Name: VAD_STATE_REG
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:24	R	0x0	Winding count indicator about AHB1 write VAD_SRAM (0x0~0xF) When the count is 0xF, the next winding count starts from 0x0.
23:20	R	0x0	Winding count indicator about AHB0 write VAD_SRAM (0x0~0xF) When the count is 0xF, the next winding count starts from 0x0.
19:18	/	/	/
17:12	R	0x0	VAD_COM_STAT Channel interpolation function status [17]: ahb1_old_rd_done, ahb1 read interpolation data finish [16]: vad_ch_com_done, interpolation finish [15]: rd_sram_done, read sram finish [14]: wr_fifo_done, write fifo finish [13:12]: com_stat channel interpolation state machine,

			2'b00: IDLE status 2'b01: RD_SRAM status 2'b10: WR_FIFO status
11:8	R	0x0	VAD_LPSD_STAT VAD_LPSD status [9:8]: the status of LPSP internal state machine [10]: the result of LPSP internal summation algorithm [11]: the result of LPSP internal subtract algorithm
7:4	R	0x0	VAD_STAT The status of VAD state machine 4'b0000: IDLE status 4'b0001: WAIT status 4'b0010: RUN status 4'b0100: NORMAL status
3	/	/	/
2	R	0x0	VAD_SRAM_FULL It indicates whether valid data storage is full or overflow when VAD SRAM is as VAD. 0: The valid data storage of VAD does not fill full or overflow 1: The valid data storage of VAD has been filled full or overflowed
1	R	0x1	VAD_SRAM_EMPTY It indicates whether valid data storage is empty when VAD SRAM is as VAD. 0: The valid data storage of VAD is not empty 1: The valid data storage of VAD is empty
0	R	0x0	VAD_RUN It indicates whether VAD has started to work 0: VAD has not started to work 1: VAD has started to work

### 3.12.6.25. VAD Debug Register (Default Value: 0x0000\_0000)

Offset: 0x006C			Register Name: VAD_DEBUG_REG
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:4	R/W	0x0	VAD_CFG_ERR It indicates whether the software configuration of VAD has configuration error. Read 0: No VAD software configuration errors occurred Bit[4]=1: VAD occurs an error about lpsd side channel mask configuration Bit[5]=1: VAD occurs an error about sram side channel mask configuration Bit[6]=1: VAD occurs an error of channel interpolation enable configuration (For example, when no data need interpolation, channel interpolation is enabled ) Write 0: No effect

			Bit[4]=1: Clear Ipsd side channel mask configuration error Bit[5]=1: Clear sram side channel mask configuration error Bit[6]=1: Clear channel interpolation enable configuration error
3	R/W	0x0	VAD_SRAM_FULL_ERR It indicates that the access of VAD SRAM results writing overflow error. Read 0: The access of VAD SRAM does not result writing overflow error Read 1: The access of VAD SRAM results writing overflow error Write 0: No effect Write 1: Clear VAD SRAM access writing overflow error
2	R/W	0x0	VAD_SRAM_EMPTY_ERR It indicates that the access of VAD SRAM results read empty error. Read 0: The access of VAD SRAM does not result read empty error Read 1: The access of VAD SRAM results read empty error Write 0: No effect Write 1: Clear VAD SRAM access read empty error
1	R/W	0x0	VAD_SRAM_ADDR_ERR It indicates that the access of VAD SRAM results address detection error. Read 0: The access of VAD SRAM does not result address detection error Read 1: The access of VAD SRAM results address detection error Write 0: No effect Write 1: Clear VAD SRAM access address detection error
0	R/W	0x0	VAD_SRAM_SEC_ERR It indicates that the access of VAD SRAM results secure detection error. Read 0: The access of VAD SRAM does not result secure detection error Read 1: The access of VAD SRAM results secure detection error Write 0: No effect Write 1: Clear VAD SRAM access secure detection error

## Contents

---

Chapter 4 Memory.....	182
4.1. SDRAM Controller(DRAMC) .....	182

## Chapter 4 Memory

---

### 4.1. SDRAM Controller(DRAMC)

- Embedded with 1Gbit DDR3
- Supports clock frequency up to 792MHz for DDR3

## Contents

---

Chapter 5 Audio.....	186
5.1. I2S/PCM.....	186
5.1.1. Overview .....	186
5.1.2. Block Diagram .....	186
5.1.3. Operations and Functional Descriptions .....	187
5.1.4. Programming Guidelines.....	192
5.1.5. Register List .....	193
5.1.6. Register Description .....	194
5.2. DMIC.....	224
5.2.1. Overview .....	224
5.2.2. Block Diagram .....	224
5.2.3. Operations and Functional Descriptions .....	224
5.2.4. Register List .....	226
5.2.5. Register Description .....	227
5.3. OWA .....	238
5.3.1. Overview .....	238
5.3.2. Block Diagram .....	238
5.3.3. Operations and Functional Descriptions .....	239
5.3.4. Register List .....	243
5.3.5. Register Description .....	243
5.4. Audio Codec .....	256
5.4.1. Overview .....	256
5.4.2. Block Diagram .....	256
5.4.3. Operations and Functional Descriptions .....	257
5.4.4. Programming Guidelines.....	265
5.4.5. Register List .....	266
5.4.6. Register Description .....	269
5.4.7. Audio Codec Analog Part Configuration IO .....	300

## Figures

Figure 5- 1. I2S/PCM Interface System Block Diagram .....	187
Figure 5- 2. I2S Standard Mode Timing.....	188
Figure 5- 3. Left-Justified Mode Timing .....	188
Figure 5- 4. Right-Justified Mode Timing .....	189
Figure 5- 5. PCM Long Frame Mode Timing.....	189
Figure 5- 6. PCM Short Frame Mode Timing.....	189
Figure 5- 7. DMIC Block Diagram .....	224
Figure 5- 8. DMIC Operation Mode.....	225
Figure 5- 9. OWA Block Diagram.....	238
Figure 5- 10. OWA Biphasic-Mark Code .....	240
Figure 5- 11. OWA Sub-Frame Format.....	240
Figure 5- 12. OWA Frame/Block Format.....	241
Figure 5- 13. OWA Operation Flow .....	242
Figure 5- 14. Audio Codec Block Diagram.....	257
Figure 5- 15. Audio Codec Clock Diagram.....	258
Figure 5- 16. Audio Codec Digital Part Reset System.....	259
Figure 5- 17. Audio Codec Analog Part Reset System.....	259
Figure 5- 18. Audio Codec Data Path Diagram.....	260
Figure 5- 19. Audio Codec Interrupt System.....	262
Figure 5- 20. DAP Data Flow .....	262
Figure 5- 21. HPF Function .....	263

## Tables

---

Table 5- 1. I2S/PCM External Signals.....	187
Table 5- 2. I2S/PCM Clock Sources.....	188
Table 5- 3. Rx Mapping Encoder .....	190
Table 5- 4. DMIC External Signals.....	224
Table 5- 5. DMIC Clock Sources.....	225
Table 5- 6. OWA External Signals .....	239
Table 5- 7. OWA Clock Sources .....	239
Table 5- 8. Biphase-Mark Encoder .....	240
Table 5- 9. Preamble Codes .....	241

# Chapter 5 Audio

## 5.1. I2S/PCM

### 5.1.1. 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
- Compliant with Left-justified, Right-justified, PCM mode, and TDM (Time Division Multiplexing) format
- Full-duplex synchronous work mode
- Master/slave mode
- Adjustable interface voltage
- Clock up to 24.576MHz
- Adjustable audio sample resolution from 8-bit to 32-bit
- Up to 16 channel( $f_s = 48\text{kHz}$ ) which has adjustable width from 8-bit to 32-bit
- Sample rate from 8kHz to 384kHz(CHAN = 2)
- Supports 8-bit u-law and 8-bit A-law companded sample
- One 128 depth x 32-bit width TXFIFO for data transmit, one 64 depth x 32-bit width RXFIFO for data receive
- Programmable PCM frame width: 1 BCLK width (short frame) and 2 BCLKs width (long frame)
- Programmable FIFO thresholds
- Interrupt and DMA support
- Supports loop back mode for test

### 5.1.2. Block Diagram

The block diagram of I2S/PCM interface is shown as follows.

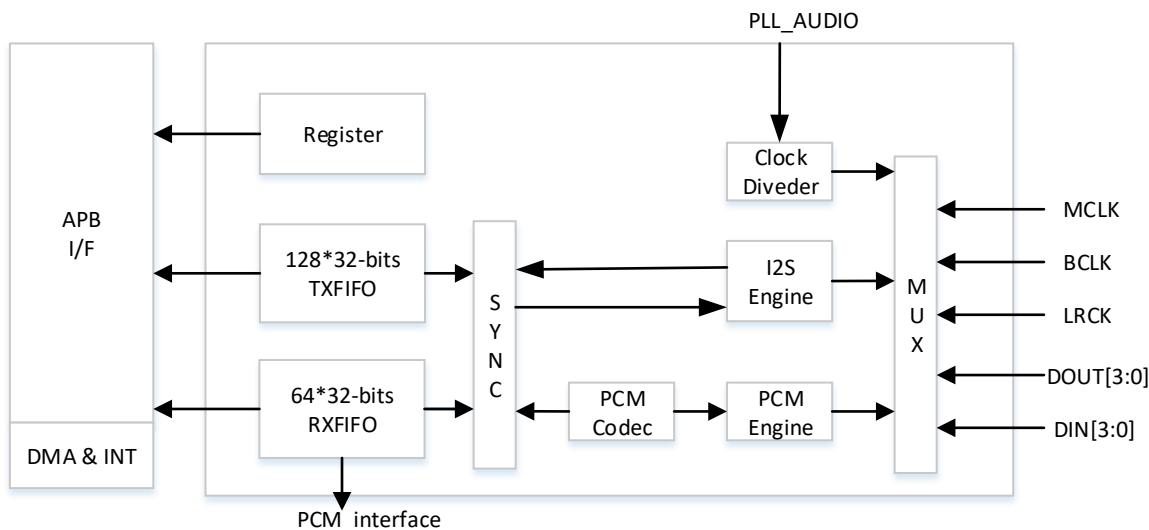


Figure 5-1. I2S/PCM Interface System Block Diagram

### 5.1.3. Operations and Functional Descriptions

#### 5.1.3.1. External Signals

Table 5-1 describes the external signals of I2S/PCM interface. LRCK and BCLK are bidirectional I/O, when I2S/PCM interface is configured as Master device, LRCK and BCLK is output pin; when I2S/PCM interface is configured as slave device, LRCK and BCLK is input pin. MCLK is an output pin for external device. DOUT is always the serial data output pin, and DIN is the serial data input pin. For information about General Purpose I/O port, see Port Controller.

Table 5-1. I2S/PCM External Signals

Signal Name	Description	Type
I2S0-MCLK	I2S0 Master Clock	O
I2S0-BCLK	I2S0/PCM0 Sample Rate Serial Clock	I/O
I2S0-LRCK	I2S0 Sample Rate Left and Right Channel Select Clock/PCM0 Sync	I/O
I2S0-DIN[3:0]	I2S0/PCM0 Serial Data Input[3:0]	I
I2S0-DOUT[3:0]	I2S0/PCM0 Serial Data Output[3:0]	O
I2S1-MCLK	I2S1 Master Clock	O
I2S1-BCLK	I2S1/PCM1 Sample Rate Serial Clock	I/O
I2S1-LRCK	I2S1 Sample Rate Left and Right Channel Select Clock/PCM1 Sync	I/O
I2S1-DIN[1:0]	I2S1/PCM1 Serial Data Input[1:0]	I
I2S1-DOUT[1:0]	I2S1/PCM1 Serial Data Output[1:0]	O
I2S2-MCLK	I2S2 Master Clock	O
I2S2-BCLK	I2S2/PCM2 Sample Rate Serial Clock	I/O
I2S2-LRCK	I2S2 Sample Rate Left and Right Channel Select Clock/PCM2 Sync	I/O
I2S2-DIN[1:0]	I2S2/PCM2 Serial Data Input[1:0]	I
I2S2-DOUT[1:0]	I2S2/PCM2 Serial Data Output[1:0]	O

### 5.1.3.2. Clock Sources

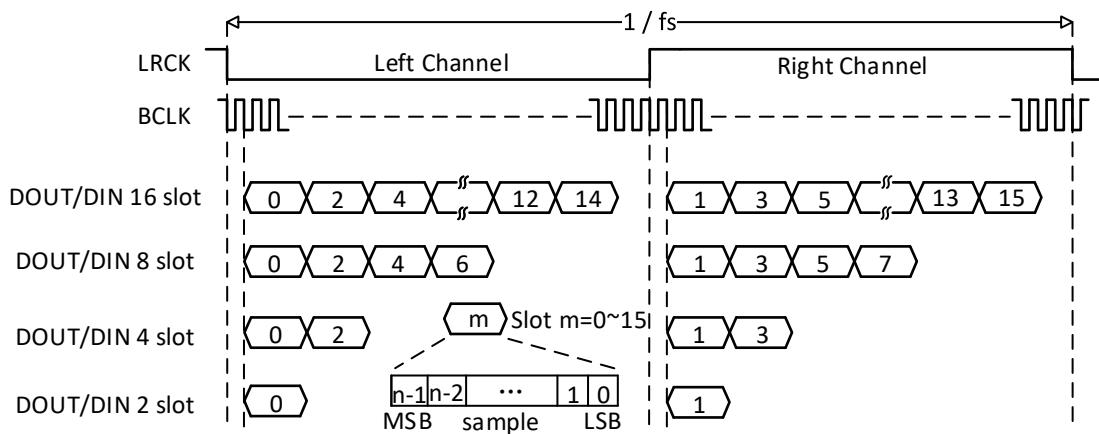
Table 5-2 describes the clock sources for I2S/PCM. Users can see **Chapter 3.3.CCU** for clock setting, configuration and gating information.

**Table 5- 2. I2S/PCM Clock Sources**

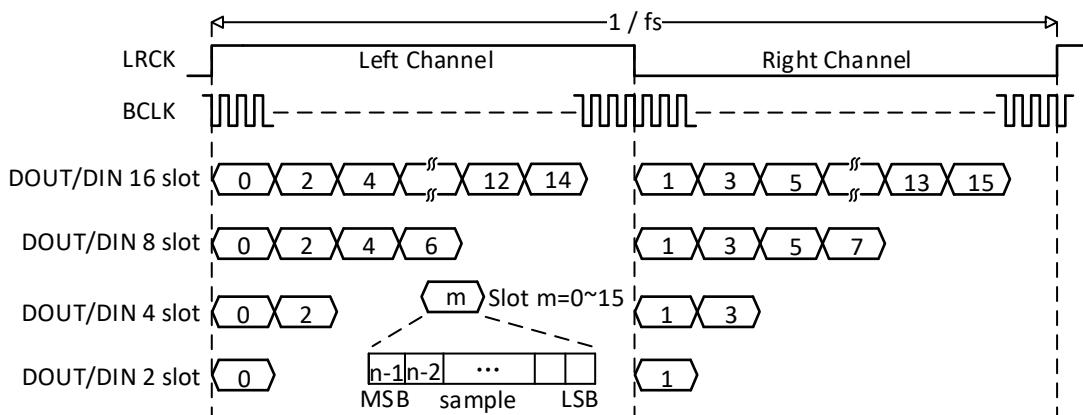
Clock Name	Description
PLL_AUDIO	24.576 MHz or 22.5792 MHz generated by PLL_AUDIO to produce 48 kHz or 44.1 kHz sample frequency

### 5.1.3.3. Timing Diagram

The I2S/PCM supports standard I2S mode, Left-justified I2S mode, Right-justified I2S mode, PCM mode and TDM mode. Software can select any modes by setting the **I2S/PCM Control Register**. Figure 5-2 to Figure 5-6 describe the waveforms for SYNC, BCLK and DOUT, DIN.



**Figure 5- 2. I2S Standard Mode Timing**



**Figure 5- 3. Left-Justified Mode Timing**

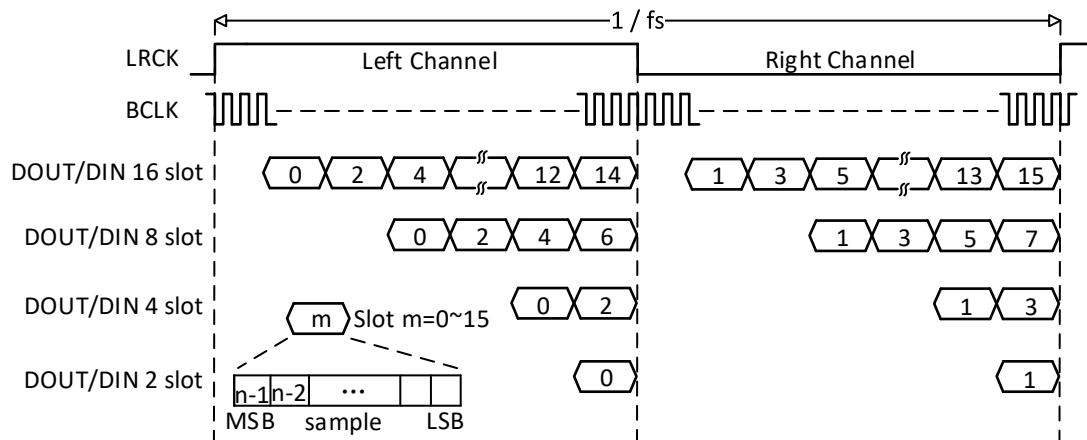


Figure 5-4. Right-Justified Mode Timing

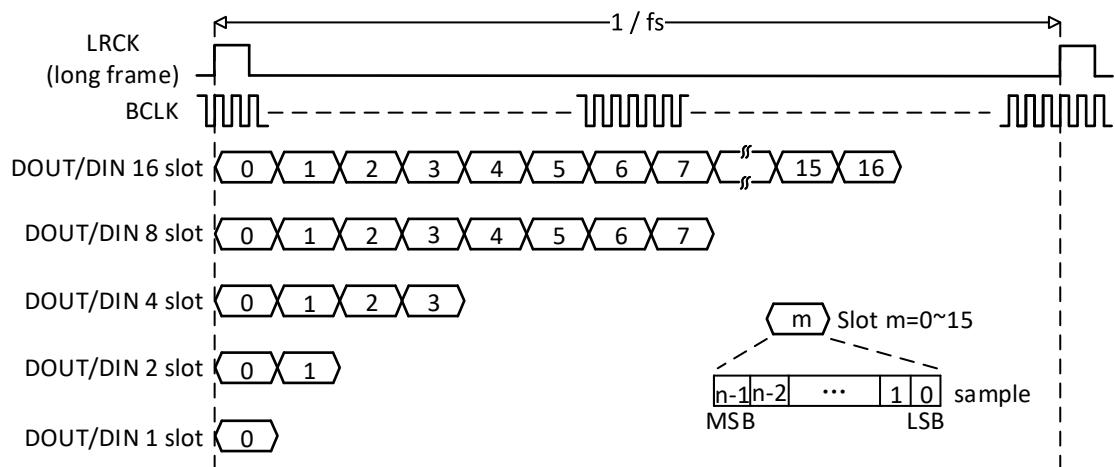


Figure 5-5. PCM Long Frame Mode Timing

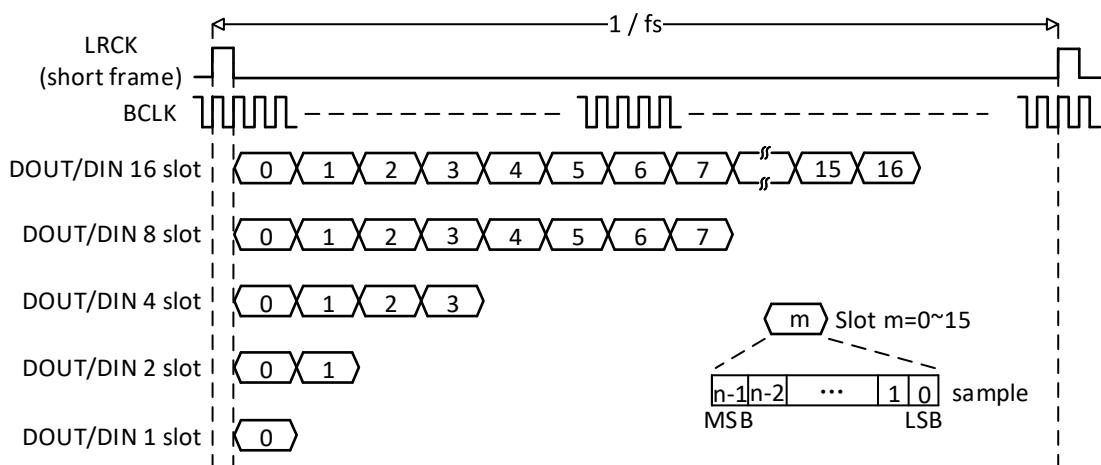


Figure 5-6. PCM Short Frame Mode Timing

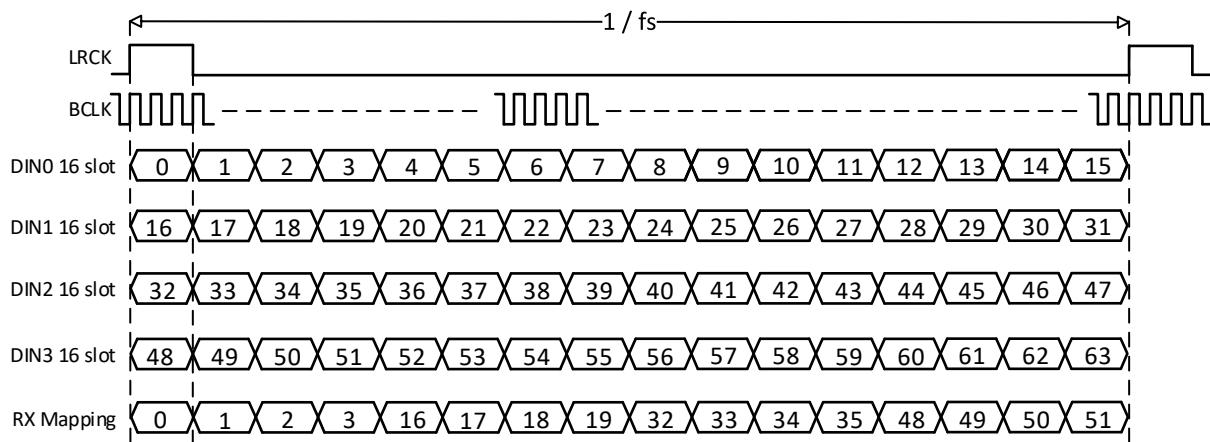
### 5.1.3.4. DIN Slot Mapping

The 4-wire DIN has 64 slots, each wire DIN has 16 slots, but RX is only 16 channels valid, the relationship between slot id and encoder is as follows .

**Table 5- 3. DIN Slot ID and Encoder**

DIN0 Slot ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DIN1 Slot ID	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
DIN2 Slot ID	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
DIN3 Slot ID	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63

There are 16 channels mapping configuration, each wire is selected four slots into RX.



**Figure 5- 7. 16 Channels Mapping Configuration**

### 5.1.3.5. Operation Modes

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. These five steps are described in detail in the following sections.

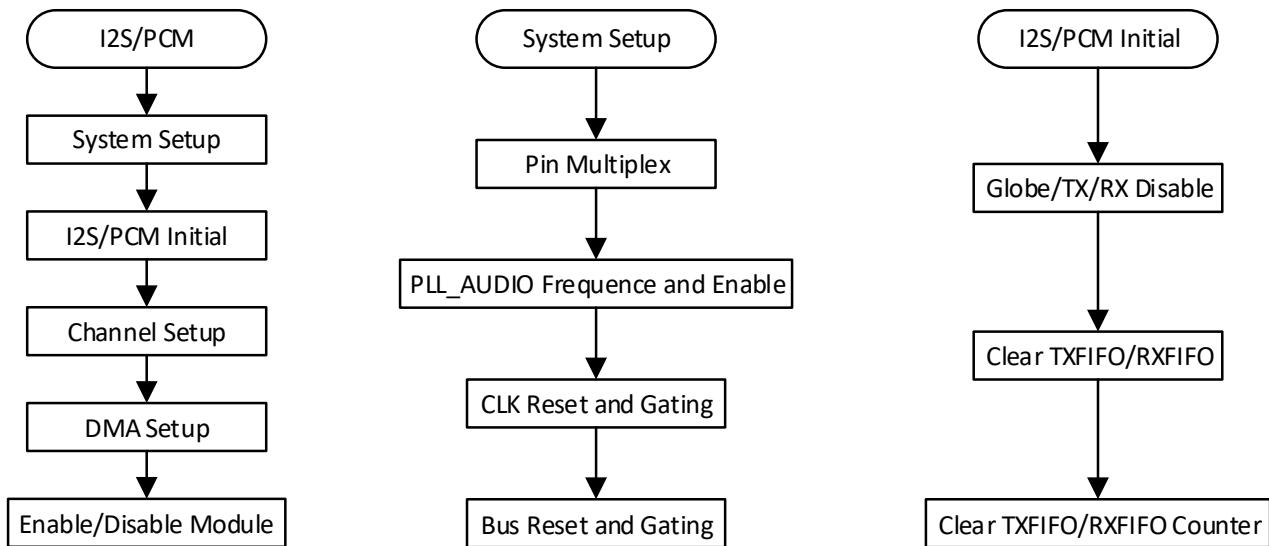


Figure 5-8. I2S/PCM Operation Flow

### (1). System Setup and I2S/PCM Initialization

The clock source for the I2S/PCM should be followed. At first you must disable the PLL\_AUDIO through the **PLL\_ENABLE** bit of **PLL\_AUDIO\_CTRL\_REG** in the CCU. The second step, you must set up the frequency of the PLL\_AUDIO in the **PLL\_AUDIO\_CTRL\_REG**. After that, you must open the I2S/PCM gating through the **I2S/PCM\_CLK\_REG** when you checkout that the LOCK bit of **PLL\_AUDIO\_CTRL\_REG** becomes to 1. At last, you must reset and open the I2S/PCM bus gating in the **CCU\_I2S\_BGR\_REG**.

After the system setup, the register of I2S/PCM can be setup. At first, you should initialization 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 to it. After that, you must clear the TX/RX FIFO by writing 0 to the bit[25:24] of **I2S/PCM\_FCTL**. At last, you can clear the TX FIFO and RX FIFO counter by writing 0 to **I2S/PCM\_TXCNT** and **I2S/PCM\_RXCNT**.

### (2). Channel Setup and DMA Setup

First, you can setup the I2S/PCM of mater and slave. The configuration can be referred to the protocol of I2S/PCM. Then, you can set up the translation mode, the sample resolution, the wide of slot, the channel slot number and the trigger level and so on. The setup of register can be found in the specification.

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.

### (3). Enable and Disable I2S/PCM

To enable the function, you can enable TX/RX by writing the **I2S/PCM\_CTL[2:1]**. After that, you must enable I2S/PCM

by writing 1 to the **Globe Enable** bit in the I2S/PCM\_CTL. Write 0 to the **Globe Enable** bit to disable I2S/PCM.

#### 5.1.4. Programming Guidelines

The following example assumes that the audio channels are stereo channels in I2S mode, the sampling rate is 48 kHz, the sampling precision is 16 bits. The recording and playback processes are as follows.

##### -----GPIO configuration-----

Step1: Ensure that I2S/PCM0 GPIO has power supply.

Step2: Configure GPIOB4 as Function4, GPIOB2/GPIOB3/GPIOB5/GPIOB13 as Function5.

##### -----Clock configuration-----

Step1: Configure PLL\_AUDIO as 24.576MHz, that is, set **PLL\_AUDIO Control Register** to 0xA90B1701, set **PLL\_AUDIO Pattern0 Register** to 0xC00126E9 (If PLL\_AUDIO is set as 22.5792MHz, that is, set **PLL\_AUDIO Control Register** to 0xA90B1501, set **PLL\_AUDIO Pattern0 Register** to 0xC001288D).

Step2: Check whether **PLL\_AUDIO Control Register[PLL\_AUDIO\_LOCK]** is 0x1. If is 1, set **I2S/PCM0 Clock Register** to 0x80000000.

Step3: Write 0x1 to the bit16 of **I2S/PCM0 Bus Gating Reset Register** to dessert I2S/PCM0 reset.

Step4: Write 0x1 to the bit0 of **I2S/PCM0 Bus Gating Reset Register** to open I2S/PCM0 gating.



##### NOTE

**Step3 and Step4 is set separately.**

##### -----Initialization I2S/PCM-----

Step1: Set the bit[2:0] of **I2S/PCM Control Register** to 0 to close TXEN,RXEN and GEN.

Step2: Set the bit[25:24] of **I2S/PCM FIFO Control Register** to 0x3 to clear TXFIFO and RXFIFO.

Step3: Set **I2S/PCM TX Counter Register** to 0 to clear TX counter, set **I2S/PCM RX Counter Register** to 0 to clear RX counter.

##### -----Format configuration-----

Step1: Master/slave configuration. In master mode, the bit[18:17] of **I2S/PCM RX Counter Register** is set to 0x3; in slave mode, the bit[18:17] of **I2S/PCM RX Counter Register** is set to 0.

Step2: Configure the[5:4] of **I2S/PCM Control Register** to 0x1 to set standard I2S mode, configure the bit[21:20] of **I2S/PCM TX0 Channel Select Register** to 0x1, configure the bit[21:20] of **I2S/PCM RX Channel Select Register** to 0x1.

Step3: Configure the bit[6:4] of **I2S/PCM Format Register0** to 0x3 to set sample resolution, configure the bit[2:0] of **I2S/PCM Format Register0** to 0x3 to set channel width.

Step4: Configure the bit[7:4] of **I2S/PCM Channel Configuration Register** to 0x1 to set RX channel number, configure the bit[3:0] of **I2S/PCM Channel Configuration Register** to 0x1 to set TX channel number. Configure the bit[19:16] of **I2S/PCM TX0 Channel Select Register** to 0x1, configure the bit[1:0] of **I2S/PCM TX0 Channel Select Register** to 0x3. Configure the bit[19:16] of **I2S/PCM RX Channel Select Register** to 0x1.

Step5: Configure the bit[7:0] of **I2S/PCM TX0 Channel Mapping Register 1** to 0x10, configure the bit[5:0] of **I2S/PCM RX Channel Mapping Register 3** to 0x0, configure the bit[13:8] of **I2S/PCM RX Channel Mapping Register 3** to 0x1.

---

-----Clock divider configuration-----

Step1: Set MCLK divider. Configure the bit[3:0] of **I2S/PCM Clock Divide Register** to 0x1, that is, MCLK=24.576MHz.

Configure the bit8 of **I2S/PCM Clock Divide Register** to 0x1 to enable MCLK.

Step2: Set BCLK divider. Configure the bit[7:4] of **I2S/PCM Clock Divide Register** to 0xF, that is, BCLK=Sample ratio\*Slot\_Width\*Slot\_Num=48K\*16\*2=1.536MHz.

Step3: Set LRCK divider. Configure the bit[17:8] of **I2S/PCM Format Register** to 0xF, that is, N-1=BCLK/Sample ratio/Slot\_Num =16,N=15.

---

-----DMA configuration-----

Step1: Set data width of both DMA\_SRC and DMA\_DEST to 16-bit.

Step2: Set DMA BLOCK SIZE,DMA\_SRC BLOCK SIZE and DMA\_DEST BLOCK SIZE to 8.

Step3: TX DMA configuration. Set DMA\_SRC\_DRQ\_TYPE to DRAM, set DMA\_SRC\_ADDR\_MODE to Linear Mode, set DMA\_DEST\_DRQ\_TYPE to I2S/PCM0-TX, set DMA\_DEST\_ADDR\_MODE to IO Mode, set DMA\_SRC\_ADDR to DRAM address of storing data, set DMA\_DEST\_ADDR to **I2S/PCM TXFIFO**(address: 0x05090020).

Step4: RX DMA configuration. Set DMA\_SRC\_DRQ\_TYPE to I2S/PCM0-RX, set DMA\_SRC\_ADDR\_MODE to IO Mode, set DMA\_DEST\_DRQ\_TYPE to DRAM, set DMA\_DEST\_ADDR\_MODE to Linear Mode, set DMA\_SRC\_ADDR to **I2S/PCM RXFIFO**(address: 0x05090010), set DMA\_DEST\_ADDR to DRAM address of storing data.

For more details about DMA, please see the description of DMA in section 3.9.



**NOTE**

If data is stored in SRAM, then DRAM is modified to SRAM.

---

-----Recording/playback/pause-----

Step1: Enable globe, set the bit0 of **I2S/PCM Control Register** to 0x1. Enable DOUT0\_EN, set the bit8 of **I2S/PCM Control Register** to 0x1.

Step2: Recording start: set the bit1 of **I2S/PCM Control Register** to 0x1, set the bit3 of **I2S/PCM DMA & Interrupt Control Register** to 0x1.

Step3: Playback start: set the bit2 of **I2S/PCM Control Register** to 0x1, set the bit7 of **I2S/PCM DMA & Interrupt Control Register** to 0x1.

Step4: Recording pause: set the bit1 of **I2S/PCM Control Register** to 0, set the bit3 of **I2S/PCM DMA & Interrupt Control Register** to 0.

Step5: Playback pause: set the bit2 of **I2S/PCM Control Register** to 0, set the bit7 of **I2S/PCM DMA & Interrupt Control Register** to 0.

### 5.1.5. Register List

Module Name	Base Address
I2S/PCM0	0x05090000
I2S/PCM1	0x05091000
I2S/PCM2	0x05092000

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/PCMISTA	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 & 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_TX0CHCFG	0x0034	I2S/PCM TX0 Channel Configuration Register
I2S/PCM_TX1CHCFG	0x0038	I2S/PCM TX1 Channel Configuration Register
I2S/PCM_TX2CHCFG	0x003C	I2S/PCM TX2 Channel Configuration Register
I2S/PCM_TX3CHCFG	0x0040	I2S/PCM TX3 Channel Configuration Register
I2S/PCM_TX0CHMAP0	0x0044	I2S/PCM TX0 Channel Mapping Register0
I2S/PCM_TX0CHMAP1	0x0048	I2S/PCM TX0 Channel Mapping Register1
I2S/PCM_TX1CHMAP0	0x004C	I2S/PCM TX1 Channel Mapping Register0
I2S/PCM_TX1CHMAP1	0x0050	I2S/PCM TX1 Channel Mapping Register1
I2S/PCM_TX2CHMAP0	0x0054	I2S/PCM TX2 Channel Mapping Register0
I2S/PCM_TX2CHMAP1	0x0058	I2S/PCM TX2 Channel Mapping Register1
I2S/PCM_TX3CHMAP0	0x005C	I2S/PCM TX3 Channel Mapping Register0
I2S/PCM_TX3CHMAP1	0x0060	I2S/PCM TX3 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

### 5.1.6. Register Description

#### 5.1.6.1. I2S/PCM Control Register(Default Value: 0x0006\_0000)

Offset: 0x0000			Register Name: I2S/PCM_CTL
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/
18	R/W	0x1	BCLK_OUT 0: Input 1: Output

17	R/W	0x1	LRCK_OUT 0: Input 1: Output
16:12	/	/	/
11	R/W	0x0	DOUT3_EN 0: Disable, Hi-Z State 1: Enable
10	R/W	0x0	DOUT2_EN 0: Disable, Hi-Z State 1: Enable
9	R/W	0x0	DOUT1_EN 0: Disable, Hi-Z State 1: Enable
8	R/W	0x0	DOUT0_EN 0: Disable, Hi-Z State 1: Enable
7	R/W	0x0	VAD_DATA_EN 0: Disable 1: Enable
6	R/W	0x0	OUT_MUTE 0: Normal Transfer 1: Force DOUT to output 0
5:4	R/W	0x0	MODE_SEL Mode Selection 00: PCM Mode (offset 0: Long Frame; offset 1: Short Frame) 01: Left Mode (offset 0: LJ Mode; offset 1: I2S Mode) 10: Right-Justified Mode 11: Reserved
3	R/W	0x0	LOOP Loop Back Test 0: Normal Mode 1: Loop Back Test When setting to '1' , the bit indicates that the DOUT connects to the DIN.
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 0: Disable

			1: Enable
--	--	--	-----------

### 5.1.6.2. 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	<p>LRCK_WIDTH LRCK Width(only apply in PCM mode ) 0: LRCK = 1 BCLK Width (Short Frame) 1: LRCK = 2 BCLK Width (Long Frame)</p>
29:20	/	/	/
19	R/W	0x0	<p>LRCK_POLARITY In I2S/Left-Justified/Right-Justified mode: 0: Left Channel when LRCK is low 1: Left channel when LRCK is high In PCM mode: 0: PCM LRCK asserted at the negative edge 1: PCM LRCK asserted at the positive edge</p>
18	/	/	/
17:8	R/W	0x0	<p>LRCK_PERIOD It is used to program the number of BCLKs per channel of sample frame. This value is interpreted as follows. <b>PCM mode:</b> Number of BCLKs within (Left + Right) channel width. <b>I2S/Left-Justified/Right-Justified mode:</b> Number of BCLKs within each individual channel width (Left or Right) . For example: N = 7: 8 BCLKs width ... N = 1023: 1024 BCLKs width</p>
7	R/W	0x0	<p>BCLK_POLARITY 0: Normal mode, DOUT drives data at negative edge 1: Invert mode, DOUT drives data at positive edge</p>
6:4	R/W	0x3	<p>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</p>
3	R/W	0x0	EDGE_TRANSFER

			<p>0: DOUT drives data and DIN sample data at the different BCLK edge          1: DOUT drives 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.</p>
2:0	R/W	0x3	<p>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</p>

#### 5.1.6.3. 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	<p>RX MLS          MSB/LSB First Select          0: MSB First          1: LSB First</p>
6	R/W	0x0	<p>TX MLS          MSB/LSB First Select          0: MSB First          1: LSB First</p>
5:4	R/W	0x3	<p>SEXT          Sign Extend in Slot [Sample Resolution &lt; 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</p>
3:2	R/W	0x0	<p>RX_PDM          PCM Data Mode          00: Linear PCM          01: Reserved</p>

			10: 8-bit u-law 11: 8-bit A-law
1:0	R/W	0x0	TX_PDM PCM Data Mode 00: Linear PCM 01: Reserved 10: 8-bit u-law 11: 8-bit A-law

#### 5.1.6.4. I2S/PCM Interrupt Status Register(Default Value: 0x0000\_0010)

Offset: 0x000C			Register Name: I2S/PCMISTA
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/W	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
--	--	--	---------------

#### 5.1.6.5. I2S/PCM RXFIFO Register(Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: I2S/PCM_RXFIFO
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>RX_DATA RX Sample</p> <p>Host can get one sample by reading this register. The left channel sample data is first and then the right channel sample.</p>

#### 5.1.6.6. 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	<p>HUB_EN Audio Hub Enable 0 : Disable 1 : Enable</p>
30:26	/	/	/
25	R/WAC	0x0	<p>FTX Write '1' to flush TXFIFO, self clear to '0'.</p>
24	R/WAC	0x0	<p>FRX Write '1' to flush RXFIFO, self clear to '0'.</p>
23:19	/	/	/
18:12	R/W	0x40	<p>TXTL TXFIFO Empty Trigger Level Interrupt and DMA request trigger level for TXFIFO normal condition. Trigger Level = TXTL</p>
11:10	/	/	/
9:4	R/W	0xF	<p>RXTL RXFIFO Trigger Level Interrupt and DMA request trigger level for RXFIFO normal condition. Trigger Level = RXTL + 1</p>
3	/	/	/
2	R/W	0x0	<p>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-bit 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}</p>

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-bit 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]}
-----	-----	-----	--

#### 5.1.6.7. I2S/PCM FIFO Status Register(Default Value: 0x1080\_0000)

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	VAD_DATA_ALIGN 0: misalign 1: align When data changes from vad output to apd output, software starts to send the command of apb read data in vad_data_align =1.
6:0	R	0x0	RXA_CNT RXFIFO available sample word counter

#### 5.1.6.8. I2S/PCM DMA & 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	<b>TX_DRQ</b> TXFIFO Empty DRQ Enable 0: Disable 1: Enable
6	R/W	0x0	<b>TXUI_EN</b> TXFIFO Underrun Interrupt Enable 0: Disable 1: Enable
5	R/W	0x0	<b>TXOI_EN</b> TXFIFO Overrun Interrupt Enable 0: Disable 1: Enable When setting to '1', an interrupt happens when writing new audio data if TXFIFO is full.
4	R/W	0x0	<b>TXEI_EN</b> TXFIFO Empty Interrupt Enable 0: Disable 1: Enable
3	R/W	0x0	<b>RX_DRQ</b> RXFIFO Data Available DRQ Enable 0: Disable 1: Enable When setting to '1', RXFIFO DMA request line is asserted if data is available in RXFIFO.
2	R/W	0x0	<b>RXUI_EN</b> RXFIFO Underrun Interrupt Enable 0: Disable 1: Enable
1	R/W	0x0	<b>RXOI_EN</b> RXFIFO Overrun Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	<b>RXAI_EN</b> RXFIFO Data Available Interrupt Enable 0: Disable 1: Enable

#### 5.1.6.9. 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	<b>TX_DATA</b> TX Sample Transmitting left, right channel sample data should be written to this register one by one. The left channel sample data is first and then the right channel sample.
------	---	-----	---

#### 5.1.6.10. 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	<b>MCLKO_EN</b> 0: Disable MCLK Output 1: Enable MCLK Output  <b>NOTE</b> <b>Whether in slave or master mode, when this bit is set to '1', MCLK should be output.</b>
7:4	R/W	0x0	<b>BCLKDIV</b> BCLK Divide Ratio from PLL_AUDIO 0000: Reserved 0001: Divide by 1 0010: Divide by 2 0011: Divide by 4 0100: Divide by 6 0101: Divide by 8 0110: Divide by 12 0111: Divide by 16 1000: Divide by 24 1001: Divide by 32 1010: Divide by 48 1011: Divide by 64 1100: Divide by 96 1101: Divide by 128 1110: Divide by 176 1111: Divide by 192
3:0	R/W	0x0	<b>MCLKDIV</b> 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
--	--	--	---

#### 5.1.6.11. I2S/PCM TX Counter Register(Default Value: 0x0000\_0000)

Offset: 0x0028			Register Name: I2S/PCM_TXCNT
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<b>TX_CNT</b> 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.

#### 5.1.6.12. I2S/PCM RX Counter Register(Default Value: 0x0000\_0000)

Offset: 0x002C			Register Name: I2S/PCM_RXCNT
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<b>RX_CNT</b> 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.

#### 5.1.6.13. 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

			0: Normal mode for the last half cycle of BCLK in the slot 1: Turn to Hi-Z state for the last half cycle of BCLK in the slot
8	R/W	0x0	TX_STATE 0: Transfer level 0 in non-transferring slot 1: Turn to Hi-Z State (TDM) in non-transferring slot
7:4	R/W	0x0	RX_SLOT_NUM RX Channel/Slot number between CPU/DMA and RXFIFO 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ... 1111:16 channels or slots
3:0	R/W	0x0	TX_SLOT_NUM TX Channel/Slot number between CPU/DMA and TXFIFO 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ... 1111:16 channels or slots

#### 5.1.6.14. 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 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ... 1111:16 channels or slots
15:0	R/W	0x0	TX0_CHEN TX0 Channel (Slot) Enable The 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 the disable state. 0: Disable

			1: Enable
--	--	--	-----------

#### 5.1.6.15. I2S/PCM TX1 Channel Select Register(Default Value: 0x0000\_0000)

Offset: 0x0038			Register Name: I2S/PCM_TX1CHSEL
Bit	Read/Write	Default/Hex	Description
31:22	/	/	/
21:20	R/W	0x0	<p>TX1_OFFSET TX1 Offset Tune(TX1 Data offset to LRCK) 0: No offset n: Data is offset by n BCLKs to LRCK</p>
19:16	R/W	0x0	<p>TX1_CHSEL TX1 Channel (Slot) Number Select for Each Output 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ... 1111:16 channels or slots</p>
15:0	R/W	0x0	<p>TX1_CHEN TX1 Channel (Slot) Enable The 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 the disable state. 0: Disable 1: Enable</p>

#### 5.1.6.16. I2S/PCM TX2 Channel Select Register(Default Value: 0x0000\_0000)

Offset: 0x003C			Register Name: I2S/PCM_TX2CHSEL
Bit	Read/Write	Default/Hex	Description
31:22	/	/	/
21:20	R/W	0x0	<p>TX2_OFFSET TX2 Offset Tune(TX2 Data offset to LRCK) 0: No offset n: Data is offset by n BCLKs to LRCK</p>
19:16	R/W	0x0	<p>TX2_CHSEL TX2 Channel (Slot) Number Select for Each Output 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ...</p>

			1111:16 channels or slots
15:0	R/W	0x0	<p>TX2_CHEN TX2 Channel (Slot) Enable The 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 the disable state. 0: Disable 1: Enable</p>

#### 5.1.6.17. I2S/PCM TX3 Channel Select Register(Default Value: 0x0000\_0000)

Offset: 0x0040			Register Name: I2S/PCM_TX3CHSEL
Bit	Read/Write	Default/Hex	Description
31:22	/	/	/
21:20	R/W	0x0	<p>TX3_OFFSET TX3 Offset Tune(TX3 Data offset to LRCK) 0: No offset n: Data is offset by n BCLKs to LRCK</p>
19:16	R/W	0x0	<p>TX3_CHSEL TX3 Channel (Slot) Number Select for Each Output 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ... 1111:16 channels or slots</p>
15:0	R/W	0x0	<p>TX3_CHEN TX3 Channel (Slot) Enable The 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 the disable state. 0: Disable 1: Enable</p>

#### 5.1.6.18. I2S/PCM TX0 Channel Mapping0 Register 0(Default Value: 0x0000\_0000)

Offset: 0x0044			Register Name: I2S/PCM_TX0CHMAP0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	<p>TX0_CH15_MAP TX0 Channel 15 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ...</p>

			1111: 16th Sample
27:24	R/W	0x0	TX0_CH14_MAP TX0 Channel 14 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th sample
23:20	R/W	0x0	TX0_CH13_MAP TX0 Channel 13 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
19:16	R/W	0x0	TX0_CH12_MAP TX0 Channel 12 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
15:12	R/W	0x0	TX0_CH11_MAP TX0 Channel 11 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th sample
11:8	R/W	0x0	TX0_CH10_MAP TX0 Channel 10 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
7:4	R/W	0x0	TX0_CH9_MAP TX0 Channel 9 Mapping 0000: 1st Sample ...

			0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
3:0	R/W	0x0	TX0_CH8_MAP TX0 Channel 8 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample

#### 5.1.6.19. I2S/PCM TX0 Channel Mapping1 Register 1(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 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
27:24	R/W	0x0	TX0_CH6_MAP TX0 Channel 6 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
23:20	R/W	0x0	TX0_CH5_MAP TX0 Channel 5 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
19:16	R/W	0x0	TX0_CH4_MAP TX0 Channel 4 Mapping 0000: 1st Sample

			<p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>
15:12	R/W	0x0	<p>TX0_CH3_MAP</p> <p>TX0 Channel 3 Mapping</p> <p>0000: 1st Sample</p> <p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>
11:8	R/W	0x0	<p>TX0_CH2_MAP</p> <p>TX0 Channel 2 Mapping</p> <p>0000: 1st Sample</p> <p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>
7:4	R/W	0x0	<p>TX0_CH1_MAP</p> <p>TX0 Channel 1 Mapping</p> <p>0000: 1st Sample</p> <p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>
3:0	R/W	0x0	<p>TX0_CH0_MAP</p> <p>TX0 Channel 0 Mapping</p> <p>0000: 1st Sample</p> <p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>

#### 5.1.6.20. I2S/PCM TX1 Channel Mapping0 Register 0(Default Value: 0x0000\_0000)

Offset: 0x004C			Register Name: I2S/PCM_TX1CHMAP0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	<p>TX1_CH15_MAP</p> <p>TX1 Channel 15 Mapping</p>

			0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
27:24	R/W	0x0	TX1_CH14_MAP TX1 Channel 14 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th sample
23:20	R/W	0x0	TX1_CH13_MAP TX1 Channel 13 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
19:16	R/W	0x0	TX1_CH12_MAP TX1 Channel 12 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
15:12	R/W	0x0	TX1_CH11_MAP TX1 Channel 11 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th sample
11:8	R/W	0x0	TX1_CH10_MAP TX1 Channel 10 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ...

			1111: 16th Sample
7:4	R/W	0x0	TX1_CH9_MAP TX1 Channel 9 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
3:0	R/W	0x0	TX1_CH8_MAP TX1 Channel 8 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample

#### 5.1.6.21. I2S/PCM TX1 Channel Mapping1 Register 1(Default Value: 0x0000\_0000)

Offset: 0x0050			Register Name: I2S/PCM_TX1CHMAP1
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	TX1_CH7_MAP TX1 Channel 7 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
27:24	R/W	0x0	TX1_CH6_MAP TX1 Channel 6 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
23:20	R/W	0x0	TX1_CH5_MAP TX1 Channel 5 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample

			...
			1111: 16th Sample
19:16	R/W	0x0	<p>TX1_CH4_MAP</p> <p>TX1 Channel 4 Mapping</p> <p>0000: 1st Sample</p> <p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>
15:12	R/W	0x0	<p>TX1_CH3_MAP</p> <p>TX1 Channel 3 Mapping</p> <p>0000: 1st Sample</p> <p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>
11:8	R/W	0x0	<p>TX1_CH2_MAP</p> <p>TX1 Channel 2 Mapping</p> <p>0000: 1st Sample</p> <p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>
7:4	R/W	0x0	<p>TX1_CH1_MAP</p> <p>TX1 Channel 1 Mapping</p> <p>0000: 1st Sample</p> <p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>
3:0	R/W	0x0	<p>TX1_CH0_MAP</p> <p>TX1 Channel 0 Mapping</p> <p>0000: 1st Sample</p> <p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>

**5.1.6.22. I2S/PCM TX2 Channel Mapping0 Register 0(Default Value: 0x0000\_0000)**

Offset: 0x0054			Register Name: I2S/PCM_TX2CHMAP0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	TX2_CH15_MAP TX2 Channel 15 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
27:24	R/W	0x0	TX2_CH14_MAP TX2 Channel 14 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th sample
23:20	R/W	0x0	TX2_CH13_MAP TX2 Channel 13 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
19:16	R/W	0x0	TX2_CH12_MAP TX2 Channel 12 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
15:12	R/W	0x0	TX2_CH11_MAP TX2 Channel 11 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th sample
11:8	R/W	0x0	TX2_CH10_MAP

			TX2 Channel 10 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
7:4	R/W	0x0	TX2_CH9_MAP TX2 Channel 9 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
3:0	R/W	0x0	TX2_CH8_MAP TX2 Channel 8 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample

#### 5.1.6.23. I2S/PCM TX2 Channel Mapping1 Register 1(Default Value: 0x0000\_0000)

Offset: 0x0058			Register Name: I2S/PCM_TX2CHMAP1
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	TX2_CH7_MAP TX2 Channel 7 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
27:24	R/W	0x0	TX2_CH6_MAP TX2 Channel 6 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample

23:20	R/W	0x0	TX2_CH5_MAP TX2 Channel 5 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
19:16	R/W	0x0	TX2_CH4_MAP TX2 Channel 4 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
15:12	R/W	0x0	TX2_CH3_MAP TX2 Channel 3 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
11:8	R/W	0x0	TX2_CH2_MAP TX2 Channel 2 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
7:4	R/W	0x0	TX2_CH1_MAP TX2 Channel 1 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
3:0	R/W	0x0	TX2_CH0_MAP TX2 Channel 0 Mapping 0000: 1st Sample ... 0111: 8th Sample

			1000: 9th Sample ... 1111: 16th Sample
--	--	--	--

#### 5.1.6.24. I2S/PCM TX3 Channel Mapping0 Register 0(Default Value: 0x0000\_0000)

Offset: 0x005C			Register Name: I2S/PCM_TX3CHMAP0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	TX3_CH15_MAP TX3 Channel 15 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
27:24	R/W	0x0	TX3_CH14_MAP TX3 Channel 14 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th sample
23:20	R/W	0x0	TX3_CH13_MAP TX3 Channel 13 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
19:16	R/W	0x0	TX3_CH12_MAP TX3 Channel 12 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
15:12	R/W	0x0	TX3_CH11_MAP TX3 Channel 11 Mapping 0000: 1st Sample ...

			0111: 8th Sample 1000: 9th Sample ... 1111: 16th sample
11:8	R/W	0x0	TX3_CH10_MAP TX3 Channel 10 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
7:4	R/W	0x0	TX3_CH9_MAP TX3 Channel 9 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
3:0	R/W	0x0	TX3_CH8_MAP TX3 Channel 8 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample

#### 5.1.6.25. I2S/PCM TX3 Channel Mapping1 Register 1(Default Value: 0x0000\_0000)

Offset: 0x0060			Register Name: I2S/PCM_TX3CHMAP1
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	TX3_CH7_MAP TX3 Channel 7 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
27:24	R/W	0x0	TX3_CH6_MAP TX3 Channel 6 Mapping 0000: 1st Sample

			<p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>
23:20	R/W	0x0	<p>TX3_CH5_MAP</p> <p>TX3 Channel 5 Mapping</p> <p>0000: 1st Sample</p> <p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>
19:16	R/W	0x0	<p>TX3_CH4_MAP</p> <p>TX3 Channel 4 Mapping</p> <p>0000: 1st Sample</p> <p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>
15:12	R/W	0x0	<p>TX3_CH3_MAP</p> <p>TX3 Channel 3 Mapping</p> <p>0000: 1st Sample</p> <p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>
11:8	R/W	0x0	<p>TX3_CH2_MAP</p> <p>TX3 Channel 2 Mapping</p> <p>0000: 1st Sample</p> <p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>
7:4	R/W	0x0	<p>TX3_CH1_MAP</p> <p>TX3 Channel 1 Mapping</p> <p>0000: 1st Sample</p> <p>...</p> <p>0111: 8th Sample</p> <p>1000: 9th Sample</p> <p>...</p> <p>1111: 16th Sample</p>

3:0	R/W	0x0	TX3_CH0_MAP TX3 Channel 0 Mapping 0000: 1st Sample ... 0111: 8th Sample 1000: 9th Sample ... 1111: 16th Sample
-----	-----	-----	---

#### 5.1.6.26. 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	<b>RX_OFFSET</b> 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	<b>RX_CHSEL</b> RX Channel (Slot) Number Select for Input 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ... 1111: 16 channels or slots
15:0	/	/	/

#### 5.1.6.27. 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	/	/	/
29:24	R/W	0x0	<b>RX_CH15_MAP</b> RX Channel 15 Mapping 000000: 1st Sample ... 000111: 8th Sample 001000: 9th Sample ... 111111: 64th Sample
23:22	/	/	/

21:16	R/W	0x0	RX_CH14_MAP RX Channel 14 Mapping 000000: 1st Sample ... 000111: 8th Sample 001000: 9th Sample ... 111111: 64th Sample
15:14	/	/	/
13:8	R/W	0x0	RX_CH13_MAP RX Channel 13 Mapping 000000: 1st Sample ... 000111: 8th Sample 001000: 9th Sample ... 111111: 64th Sample
7:6	/	/	/
5:0	R/W	0x0	RX_CH12_MAP RX Channel 12 Mapping 000000: 1st Sample ... 000111: 8th Sample 001000: 9th Sample ... 111111: 64th Sample

#### 5.1.6.28. 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	/	/	/
29:24	R/W	0x0	RX_CH11_MAP RX Channel 11 Mapping 000000: 1st Sample ... 000111: 8th Sample 001000: 9th Sample ... 111111: 64th Sample
23:22	/	/	/
21:16	R/W	0x0	RX_CH10_MAP RX Channel 10 Mapping 000000: 1st Sample

			<p>...</p> <p>000111: 8th Sample</p> <p>001000: 9th Sample</p> <p>...</p> <p>111111: 64th Sample</p>
15:14	/	/	/
13:8	R/W	0x0	<p>RX_CH9_MAP</p> <p>RX Channel 9 Mapping</p> <p>000000: 1st Sample</p> <p>...</p> <p>000111: 8th Sample</p> <p>001000: 9th Sample</p> <p>...</p> <p>111111: 64th Sample</p>
7:6	/	/	/
5:0	R/W	0x0	<p>RX_CH8_MAP</p> <p>RX Channel 8 Mapping</p> <p>000000: 1st Sample</p> <p>...</p> <p>000111: 8th Sample</p> <p>001000: 9th Sample</p> <p>...</p> <p>111111: 64th Sample</p>

#### 5.1.6.29. 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	/	/	/
29:24	R/W	0x0	<p>RX_CH7_MAP</p> <p>RX Channel 7 Mapping</p> <p>000000: 1st Sample</p> <p>...</p> <p>000111: 8th Sample</p> <p>001000: 9th Sample</p> <p>...</p> <p>111111: 64th Sample</p>
23:22	/	/	/
21:16	R/W	0x0	<p>RX_CH6_MAP</p> <p>RX Channel 6 Mapping</p> <p>000000: 1st Sample</p> <p>...</p> <p>000111: 8th Sample</p> <p>001000: 9th Sample</p>

			... 111111: 64th Sample
15:14	/	/	/
13:8	R/W	0x0	RX_CH5_MAP RX Channel 5 Mapping 000000: 1st Sample ... 000111: 8th Sample 001000: 9th Sample ... 111111: 64th Sample
7:6	/	/	/
5:0	R/W	0x0	RX_CH4_MAP RX Channel 4 Mapping 000000: 1st Sample ... 000111: 8th Sample 001000: 9th Sample ... 111111: 64th Sample

#### 5.1.6.30. 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	/	/	/
29:24	R/W	0x0	RX_CH3_MAP RX Channel 3 Mapping 000000: 1st Sample ... 000111: 8th Sample 001000: 9th Sample ... 111111: 64th Sample
23:22	/	/	/
21:16	R/W	0x0	RX_CH2_MAP RX Channel 2 Mapping 000000: 1st Sample ... 000111: 8th Sample 001000: 9th Sample ... 111111: 64th Sample
15:14	/	/	/

13:8	R/W	0x0	<p>RX_CH1_MAP</p> <p>RX Channel 1 Mapping</p> <p>000000: 1st Sample</p> <p>...</p> <p>000111: 8th Sample</p> <p>001000: 9th Sample</p> <p>...</p> <p>111111: 64th Sample</p>
7:6	/	/	/
5:0	R/W	0x0	<p>RX_CH0_MAP</p> <p>RX Channel 0 Mapping</p> <p>000000: 1st Sample</p> <p>...</p> <p>000111: 8th Sample</p> <p>001000: 9th Sample</p> <p>...</p> <p>111111: 64th Sample</p>

## 5.2. DMIC

### 5.2.1. Overview

The DMIC controller supports one 8-channels digital microphone interface, the DMIC controller can output 128fs or 64fs (fs= ADC sample rate).

The DMIC controller includes the following features:

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

### 5.2.2. Block Diagram

Figure 5-9 shows a block diagram of the DMIC.

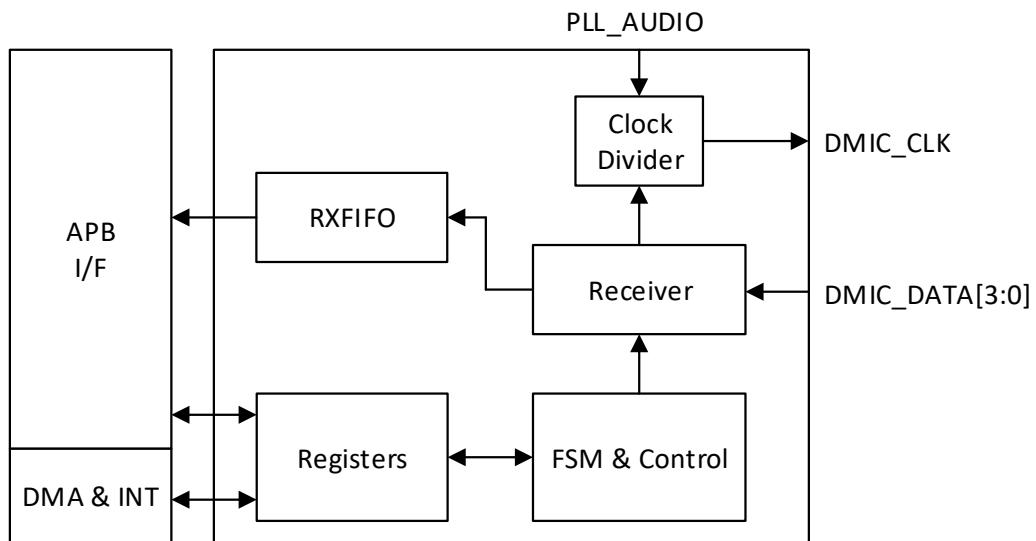


Figure 5- 9. DMIC Block Diagram

### 5.2.3. Operations and Functional Descriptions

#### 5.2.3.1. External Signals

Table 5-4 describes the external signals of DMIC.

Table 5- 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

### 5.2.3.2. Clock Sources

Table 5-5 describes the clock source for DMIC. Users can see **Clock Controller Unit(CCU)** for clock setting, configuration and gating information.

Table 5- 5. DMIC Clock Sources

Clock Sources	Description
PLL_AUDIO	24.576 MHz or 22.5792 MHz generated by PLL_AUDIO to produce 48 kHz or 44.1 kHz serial frequency.

### 5.2.3.3. Operation Mode

The software operation of the DMIC is divided into five steps: system setup, DMIC initialization, channel setup, DMA setup and Enable/Disable module. Five steps are described in detail in the following sections.

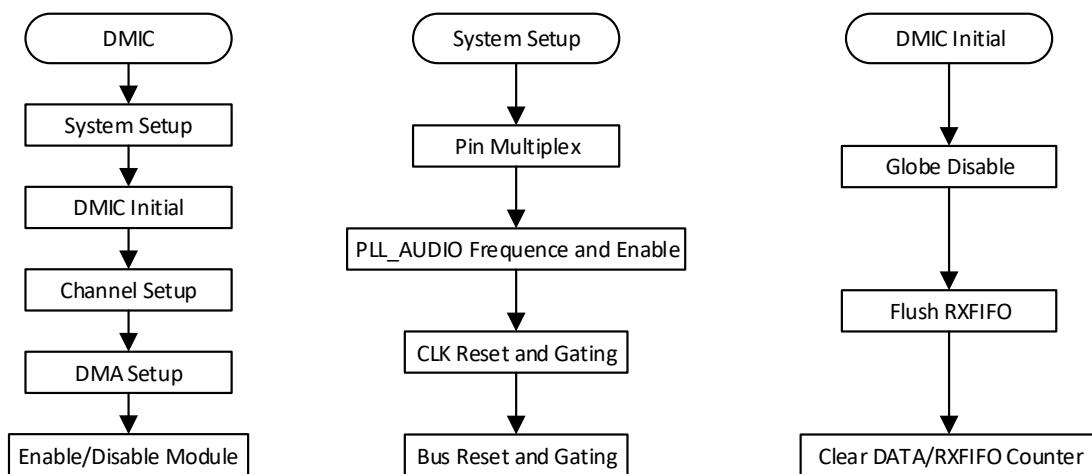


Figure 5- 10. DMIC Operation Mode

#### 5.2.3.3.1. System Setup and DMIC Initialization

The first step in the system setup is properly programming the GPIO. Because the DMIC port is a multiplex pin. You can find the function in the pin multiplex specification.

The clock source for the DMIC should be followed. At first you must disable the PLL\_AUDIO through the PLL\_ENABLE

bit of **PLL\_AUDIO\_CTRL\_REG** in the CCU. The second step, you must set up the frequency of the PLL\_AUDIO in the **PLL\_AUDIO\_CTRL\_REG**. Then enable PLL\_AUDIO. After that, you must open the DMIC gating through the **DMIC\_CLK\_REG** when you checkout that the LOCK bit of **PLL\_AUDIO\_CTRL\_REG** becomes 1. At last, you must reset and open the DMIC bus gating in the **CCU\_DMIC\_BGR\_REG**.

After the system setup, the register of DMIC can be setup. At first, you should initialize the DMIC. You should close the **globe enable bit(DMIC\_EN[8])**, **data channel enable bit(DMIC\_EN[7:0])** by writing 0 to it. After that, you must flush the RXFIFO by writing 1 to register **DMIC\_RXFIFO\_CTR[31]**. At last, you can clear the Data/RXFIFO counter by writing 1 to **DMIC\_RXFIFO\_STA,DMIC\_CNT**.

#### 5.2.3.3.2. Channel Setup and DMA Setup

You can set up the sample rate, the sample resolution, the over sample rate, the channel number, the RXFIFO output mode and the RXFIFO trigger level and so on. The setup of register can be found in the specification.

The DMIC supports two methods to transfer the data. The most common way is DMA, the setup of DMA can be found in the DMA specification. In this module, you just enable the DRQ.

#### 5.2.3.3.3. Enable and Disable DMIC

To enable the function, you can enable **data channel enable bit(DMIC\_EN[7:0])** by writing 1 to it. After that, you must enable DMIC by writing 1 to the **Globe Enable (DMIC\_EN[8])**. Write 0 to **Globe Enable** to disable DMIC.

### 5.2.4. Register List

Module Name	Base Address
DMIC	0x05095000

Register Name	Offset	Description
DMIC_EN	0x0000	DMIC Enable Control Register
DMIC_SR	0x0004	DMIC Sample Rate Register
DMIC_CTR	0x0008	DMIC Control Register
DMIC_DATA	0x0010	DMIC Data Register
DMIC_INTC	0x0014	MIC Interrupt Control Register
DMIC_INTS	0x0018	DMIC Interrupt Status Register
DMIC_RXFIFO_CTR	0x001C	DMIC RXFIFO Control Register
DMIC_RXFIFO_STA	0x0020	DMIC RXFIFO Status Register
DMIC_CH_NUM	0x0024	DMIC Channel Numbers Register
DMIC_CH_MAP	0x0028	DMIC Channel Mapping Register
DMIC_CNT	0x002C	DMIC Counter Register

DATA0_DATA1_VOL_CTR	0x0030	Data0 and Data1 Volume Control Register
DATA2_DATA3_VOL_CTR	0x0034	Data2 And Data3 Volume Control Register
HPF_EN_CTR	0x0038	High Pass Filter Enable Control Register
HPF_COEF_REG	0x003C	High Pass Filter Coef Register
HPF_GAIN_REG	0x0040	High Pass Filter Gain Register

## 5.2.5. Register Description

### 5.2.5.1. DMIC Enable Control Register (Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: DMIC_EN
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	VAD_DATA_EN 0: Disable 1: Enable
30:9	/	/	/
8	R/W	0x0	GLOBE_EN DMIC Globe Enable 0: Disable 1: Enable
7	R/W	0x0	DATA3_CHR_EN DATA3 Right Channel Enable 0: Disable 1: Enable
6	R/W	0x0	DATA3_CHL_EN DATA3 Left Channel Enable 0: Disable 1: Enable
5	R/W	0x0	DATA2_CHR_EN DATA2 Right Channel Enable 0: Disable 1: Enable
4	R/W	0x0	DATA2_CHL_EN DATA2 Left Channel Enable 0: Disable 1: Enable
3	R/W	0x0	DATA1_CHR_EN DATA1 Right Channel Enable 0: Disable 1: Enable
2	R/W	0x0	DATA1_CHL_EN DATA1 Left Channel Enable 0: Disable

			1: Enable
1	R/W	0x0	DATA0_CHR_EN DATA0 Right Channel Enable 0: Disable 1: Enable
0	R/W	0x0	DATA0_CHL_EN DATA0 Left Channel Enable 0: Disable 1: Enable

#### 5.2.5.2. DMIC Sample Rate Register (Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: DMIC_SR
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2:0	R/W	0x0	DMIC_SR Sample Rate of DMIC 000: 48 kHz 010: 24 kHz 100: 12 kHz 110: Reserved 001: 32 kHz 011: 16 kHz 101: 8 kHz 111: Reserved 44.1 kHz/22.05 kHz/11.025 kHz can be supported by Audio PLL Configure Bit.

#### 5.2.5.3. DMIC Control Register (Default Value: 0x0000\_0000)

Offset: 0x0008			Register Name: DMIC_CTR
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:9	R/W	0x0	DMICFDT DMIC RXFIFO Delay Time for Writing Data after GLOBE_EN 00: 5ms 01: 10ms 10: 20ms 11: 30ms
8	R/W	0x0	DMICDFEN DMIC RXFIFO Delay Function for Writing Data after GLOBE_EN 0: Disable 1: Enable
7	R/W	0x0	DATA3 Left Data and Right Data Swap Enable

			0: Disable 1: Enable
6	R/W	0x0	DATA2 Left Data and Right Data Swap Enable 0: Disable 1: Enable
5	R/W	0x0	DATA1 Left Data and Right Data Swap Enable 0: Disable 1: Enable
4	R/W	0x0	DATA0 Left Data and Right Data Swap Enable 0: Disable 1: Enable
3:1	/	/	/
0	R/W	0x0	DMIC Oversample Rate 0: 128 (Support 8 kHz ~ 24 kHz) 1: 64 (Support 16 kHz ~ 48 kHz)

#### 5.2.5.4. 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

#### 5.2.5.5. DMIC Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x0014			Register Name: DMIC_INTC
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2	R/W	0x0	RXFIFO_DRQ_EN DMIC RXFIFO Data Available DRQ Enable 0: Disable 1: Enable
1	R/W	0x0	RXFIFO_OVERRUN_IRQ_EN DMIC RXFIFO Overrun IRQ Enable 0: Disable 1: Enable
0	R/W	0x0	DATA_IRQ_EN DMIC RXFIFO Data Available IRQ Enable 0: Disable 1: Enable

### 5.2.5.6. 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	<p>RXFIFO_OVERRUN_IRQ_PENDING DMIC RXFIFO Overrun Pending Interrupt 0: No pending IRQ 1: RXFIFO overrun pending IRQ Writing '1' to clear this interrupt or automatically clear if interrupt condition fails.</p>
0	R/W1C	0x0	<p>RXFIFO_DATA_IRQ_PENDING DMIC RXFIFO Data Available Pending Interrupt 0: No pending IRQ 1: Data available pending IRQ Writing '1' to clear this interrupt or automatically clear if interrupt condition fails.</p>

### 5.2.5.7. DMIC RXFIFO Control Register (Default Value: 0x0000\_0040)

Offset: 0x001C			Register Name: DMIC_FIFO_CTR
Bit	Read/Write	Default/Hex	Description
31	R/W1C	0x0	<p>DMIC_RXFIFO_FLUSH DMIC RXFIFO Flush Writing '1' to flush RXFIFO, self clear to '0'</p>
30:10	/	/	/
9	R/W	0x0	<p>RXFIFO_MODE RXFIFO Output Mode (Mode 0, 1) 0: Expanding '0' at LSB of RXFIFO register 1: Expanding received sample sign bit at MSB of RXFIFO register  For 24-bit received audio sample: Mode 0: RXDATA[31:0] = {RXFIFO_O[20:0], 11'h0} Mode 1: RXDATA[31:0] = {8{RXFIFO_O[20]}, RXFIFO_O[20:0], 3'h0}  For 16-bit received audio sample: Mode 0: RXDATA[31:0] = {RXFIFO_O[20:5], 16'h0} Mode 1: RXDATA[31:0] = {16{RXFIFO_O[20]}}, RXFIFO_O[20:5]}</p>
8	R/W	0x0	<p>Sample_Resolution 0: 16-bit 1: 24- bit</p>
7:0	R/W	0x40	<p>RXFIFO_TRG_LEVEL RXFIFO Trigger Level (TRLV[7:0]) Interrupt and DMA request trigger level for DMIC RXFIFO normal condition</p>

			IRQ/DRQ Generated when WLEVEL > TRLV[7:0]) WLEVEL represents the number of valid samples in the DMIC RXFIFO
--	--	--	--

#### 5.2.5.8. DMIC RXFIFO Status Register (Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: DMIC_RXFIFO_STA
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8	R	0x1	VAD_DATA_ALIGN 0: misalign 1: align When data changes from vad output to apb output, software need send the command of apb read data in vad_data_align = 1.
7:0	R/W	0x0	DMIC_DATA_CNT DMIC RXFIFO Available Sample Word Counter

#### 5.2.5.9. DMIC Channel Numbers Register (Default Value: 0x0000\_0001)

Offset: 0x0024			Register Name: DMIC_CH_NUM
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2:0	R/W	0x1	DMIC_CH_NUM DMIC enable channel numbers are (N+1).

#### 5.2.5.10. DMIC Channel Mapping Register (Default Value: 0x7654\_3210)

Offset: 0x0028			Register Name: DMIC_CH_MAP
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x7	DMIC_CH7_MAP DMIC Channel 7 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
27:24	R/W	0x6	DMIC_CH6_MAP DMIC Channel 6 Mapping 0000: DATA0 Left Channel

			0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
23:20	R/W	0x5	DMIC_CH5_MAP DMIC Channel 5 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
19:16	R/W	0x4	DMIC_CH4_MAP DMIC Channel 4 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
15:12	R/W	0x3	DMIC_CH3_MAP DMIC Channel 3 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
11:8	R/W	0x2	DMIC_CH2_MAP DMIC Channel 2 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel

			0110: DATA3 Left Channel 0111: DATA3 Right Channel
7:4	R/W	0x1	DMIC_CH1_MAP DMIC Channel 1 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
3:0	R/W	0x0	DMIC_CH0_MAP DMIC Channel0 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel

#### 5.2.5.11. DMIC Counter Register (Default Value: 0x0000\_0000)

Offset: 0x002C			Register Name: DMIC_CNT
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	<p>DMIC_CNT RX Sample Counter The audio sample number of writing into RXFIFO. When one sample is written by Digital Audio Engine, the RX sample counter register increases by one. The RX sample counter register can be set to any initial value at any time. After been updated by the initial value, the counter register should count on base of this initial value.</p> <p> NOTE</p> <p><b>It is used for Audio/ Video Synchronization.</b></p>

#### 5.2.5.12. DATA0 and DATA1 Volume Control Register (Default Value: 0xA0A0\_A0A0)

Offset: 0x0030			Register Name: DATA0_DATA1_VOL_CTR
Bit	Read/Write	Default/Hex	Description
31:24	R/W	0xA0	DATA1L_VOL

			(-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ..... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ..... 0xFF: 71.25 dB
23:16	R/W	0xA0	DATA1R_VOL (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ..... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ..... 0xFF: 71.25 dB
15:8	R/W	0xA0	DATA0L_VOL (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ..... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ..... 0xFF: 71.25 dB
7:0	R/W	0xA0	DATA0R_VOL (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ..... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ..... 0xFF: 71.25 dB

#### 5.2.5.13. DATA2 and DATA3 Volume Control Register (Default Value: 0xA0A0\_A0A0)

Offset: 0x0034		Register Name: DATA2_DATA3_VOL_CTR	
Bit	Read/Write	Default/Hex	Description

31:24	R/W	0xA0	DATA3L_VOL (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ..... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ..... 0xFF: 71.25 dB
23:16	R/W	0xA0	DATA3R_VOL (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ..... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ..... 0xFF: 71.25 dB
15:8	R/W	0xA0	DATA2L_VOL (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ..... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ..... 0xFF: 71.25 dB
7:0	R/W	0xA0	DATA2R_VOL (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ..... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ..... 0xFF: 71.25 dB

#### 5.2.5.14. High Pass Filter Enable Control Register (Default Value: 0x0000\_0000)

<b>Offset: 0x0038</b>	<b>Register Name: HPF_EN_CTR</b>
-----------------------	----------------------------------

Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	<b>HPF_DATA3_CHR_EN</b> High Pass Filter DATA3 Right Channel Enable 0: Disable 1: Enable
6	R/W	0x0	<b>HPF_DATA3_CHL_EN</b> High Pass Filter DATA3 Left Channel Enable 0: Disable 1: Enable
5	R/W	0x0	<b>HPF_DATA2_CHR_EN</b> High Pass Filter DATA2 Right Channel Enable 0: Disable 1: Enable
4	R/W	0x0	<b>HPF_DATA2_CHL_EN</b> High Pass Filter DATA2 Left Channel Enable 0: Disable 1: Enable
3	R/W	0x0	<b>HPF_DATA1_CHR_EN</b> High Pass Filter DATA1 Right Channel Enable 0: Disable 1: Enable
2	R/W	0x0	<b>HPF_DATA1_CHL_EN</b> High Pass Filter DATA1 Left Channel Enable 0: Disable 1: Enable
1	R/W	0x0	<b>HPF_DATA0_CHR_EN</b> High Pass Filter DATA0 Right Channel Enable 0: Disable 1: Enable
0	R/W	0x0	<b>HPF_DATA0_CHL_EN</b> High Pass Filter DATA0 Left Channel Enable 0: Disable 1: Enable

#### 5.2.5.15. High Pass Filter Coef Register (Default Value: 0x00FF\_AA45)

Offset: 0x003C			Register Name: HPF_COEF_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x00FFAA45	High Pass Filter Coefficient

#### 5.2.5.16. High Pass Filter Gain Register (Default Value: 0x00FF\_D522)

Offset: 0x0040		Register Name: HPF_GAIN_REG	
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x00FFD522	High Pass Filter Gain

## 5.3. OWA

### 5.3.1. Overview

The OWA(One Wire Audio) provides a serial bus interface for audio data between system. This interface is widely used for consumer audio.

#### Features:

- IEC-60958 transmitter and receiver functionality
- Compliance with S/PDIF Interface
- Supports channel status insertion for the transmitter
- Supports channel status capture for the receiver
- Hardware parity generation on the transmitter
- Hardware parity checking on the receiver
- One 128x24bits TXFIFO and 64x24bits RXFIFO for audio data transfer
- Programmable FIFO thresholds
- Interrupt and DMA support
- Supports 16-bit,20-bit and 24-bit data formats

### 5.3.2. Block Diagram

The OWA block diagram is shown as follows.

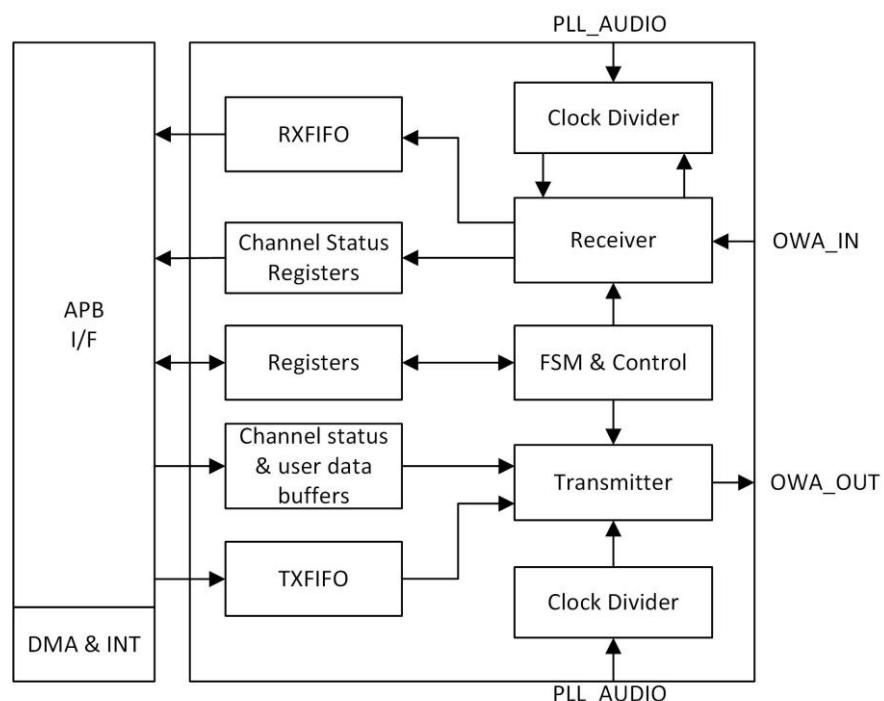


Figure 5- 11. OWA Block Diagram

### 5.3.3. Operations and Functional Descriptions

#### 5.3.3.1. External Signals

OWA is a Biphase-Mark Encoding Digital Audio Transfer protocol. In this protocol, the CLK signal and data signals are transferred in the same line. Table 5-6 describes the external signals of OWA. OWA\_DOUT is output pin for output CLK and DATA, and OWA\_DIN is input pin for input CLK and DATA.

**Table 5- 6. OWA External Signals**

Signal Name	Description	Type
OWA-OUT	OWA output	O
OWA-IN	OWA input	I

#### 5.3.3.2. Clock Sources

Table 5-7 describes the clock sources for OWA. Users can see **Chapter 3.3.CCU** for clock setting, configuration and gating information.

**Table 5- 7. OWA Clock Sources**

Clock Name	Description
PLL_AUDIO	24.576 MHz or 22.5792 MHz generated by PLL_AUDIO to produce 48 kHz or 44.1 kHz serial frequency

#### 5.3.3.3. Biphase-Mark Code (BMC)

In OWA format, the digital signal is coded using the biphase-mark code (BMC). The clock, frame, and data are embedded in only one signal—the data pin. In the BMC system, each data bit is encoded into two logical states (00, 01, 10, or 11) at the pin. Figure 5-12 and Table 5-8 show how data is encoded to the BMC format.

As shown in Figure 5-12, the frequency of the clock is twice the data bit rate. In addition, the clock is always programmed to 128xfs, where fs is the sample rate. The device receiving in OWA format can recover the clock and frame information from the BMC signal.

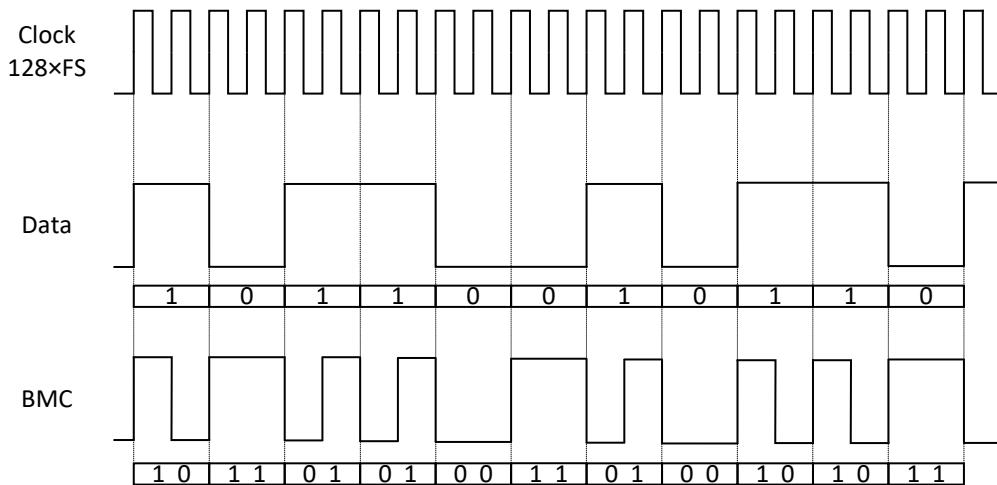


Figure 5- 12. OWA Biphasic-Mark Code

Table 5- 8. Biphasic-Mark Encoder

Data	Previous State	BMC
0	0	11
0	1	00
1	0	10
1	1	01

#### 5.3.3.4. OWA Transmit Format

The OWA supports digital audio data transfer and receive. And it supports full-duplex synchronous work mode. Software can set the work mode by the OWA Control Register.

Every audio sample transmitted in a subframe consists 32-bit, numbered from 0 to 31. Figure 5-13 shows a subframe. The OWA supports the transfer of digital audio data. And it supports full-duplex synchronous work mode.

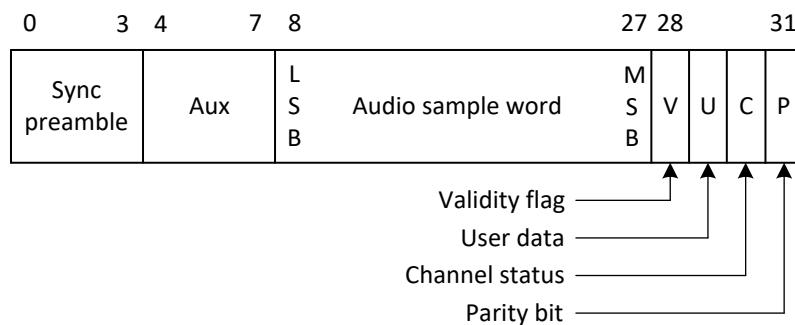


Figure 5- 13. OWA Sub-Frame Format

**Bit 0-3** carry one of the four permitted preambles to signify the type of audio sample in the current subframe. The preamble is not encoded in BMC format, and therefore the preamble code can contain more than two consecutive 0

or 1 logical states in a row. See Table 5-9.

**Bit 4-27** carry the audio sample word in linear 2s-complement representation. The most-significant bit (MSB) is carried by bit 27. When a 24-bit coding range is used, the least-significant bit (LSB) is in bit 4. When a 20-bit coding range is used, bit 8-27 carry the audio sample word with the LSB in bit 8. Bit 4-7 may be used for other applications and are designated auxiliary sample bits.

If the source provides fewer bits than the interface allows (either 20 or 24), the unused LSBs are set to logical 0. For a nonlinear PCM audio application or a data application, the main data field may carry any other information.

**Bit 28** carries the validity bit (V) associated with the main data field in the subframe.

**Bit 29** carries the user data channel (U) associated with the main data field in the subframe.

**Bit 30** carries the channel status information (C) associated with the main data field in the subframe. The channel status indicates if the data in the subframe is digital audio or some other type of data.

**Bit 31** carries a parity bit (P) such that bit 4-31 carry an even number of 1s and an even number of 0s (even parity). As shown in Table 5-9, the preambles (bit 0-3) are also defined with even parity.

Table 5- 9. Preamble Codes

Preamble Code	Previous Logical State	Logical State	Description
B(or Z)	0	1110 1000	Start of a block and subframe 1
M(or X)	0	1110 0010	Subframe 1
W(or Y)	0	1110 0100	Subframe 2

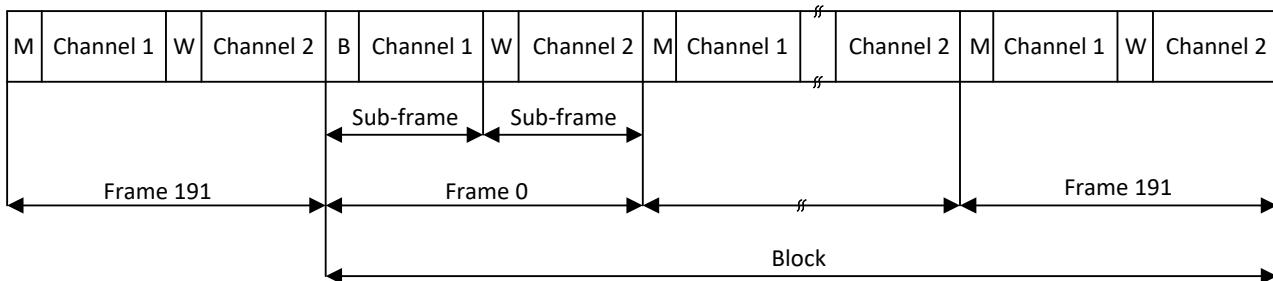


Figure 5- 14. OWA Frame/Block Format

### 5.3.3.5. Operation Modes

The software operation of the OWA is divided into five steps: system setup, OWA initialization, channel setup, DMA setup and enable/disable module. These five steps are described in detail in the following sections.

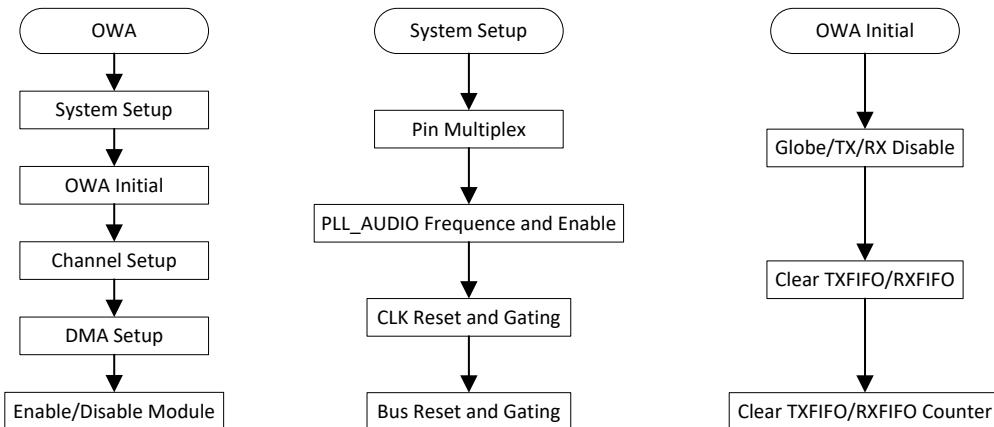


Figure 5- 15. OWA Operation Flow

### (1) System Setup and OWA Initialization

The first step in the OWA initialization is properly programming the GPIO. Because the OWA port is a multiplex pin. You can find the function in the **Port Controller**.

The clock source for the OWA should be followed. At first you must reset the audio PLL in the CCU\_PLL\_AUDIO\_CTRL. The second step, you must setup the frequency of the Audio PLL in the CCU\_PLL\_AUDIO\_CTRL. After that, you must open the OWA gating. At last, you must open the OWA bus gating.

After the system setup, the register of OWA can be setup. At first, you should reset the OWA by writing 1 to **OWA\_CTL[0]** and clear the TX/RX FIFO by writing 1 to **OWA\_FCTL[17:16]**. After that you should enable the globe enable bit by writing 1 to **OWA\_CTL[1]** and clear the interrupt and TX/RX counter by the **OWAISTA** and **OWATX\_CNT/OWARX\_CNT**.

### (2) Channel Setup and DMA Setup

You can set up the audio type, clock divide ratio, the sample format and the trigger level and so on. The setup of 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 the **DMA**. In this module, you just enable the DRQ in the **OWA\_INT[7]**.

### (3) Enable and Disable OWA

To enable the function, you can enable TX/RX by writing the **OWA\_TX\_CFIG[31]/OWA\_RX\_CFIG[0]**. After that, you must enable OWA by writing 1 to the **GEN** bit in the **OWA\_CTL** register. Writing 0 to the **GEN** bit to disable process.

### 5.3.4. Register List

Module Name	Base Address
OWA	0x05093000

Register Name	Offset	Description
OWA_GEN_CTL	0x0000	OWA General Control Register
OWA_TX_CFIG	0x0004	OWA TX Configuration Register
OWA_RX_CFIG	0x0008	OWA RX Configuration Register
OWAISTA	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

### 5.3.5. Register Description

#### 5.3.5.1. OWA General Control Register (Default Value: 0x0000\_0080)

Offset: 0x0000			Register Name: OWA_CTL
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9:5	R/W	0x4	Reserved
4	/	/	/
3	R/W	0x0	Reserved
2	R/W	0x0	LOOP Loop Back Test 0: Normal Mode 1: Loop Back Test When setting to '1', DOUT and DIN need be connected.
1	R/W	0x0	GEN Global Enable A disable on this bit overrides any other block or channel enables and flushes all FIFOs. 0: Disable 1: Enable

0	R/W	0x0	RST Reset 0: Normal 1: Reset Self clear to 0.
---	-----	-----	---

### 5.3.5.2. OWA TX Configure Register (Default Value: 0x0000\_00F0)

Offset: 0x0004			Register Name: OWA_TX_CFIG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	TX_SINGLE_MODE Tx Single Channel Mode 0: Disable 1: Enable
30:18	/	/	/
17	R/W	0x0	ASS Audio Sample Select with TX FIFO Underrun when 0: Sending 0 1: Sending the last audio   <b>NOTE</b> <b>This bit is only valid in PCM mode.</b>
16	R/W	0x0	TX_AUDIO TX Data Type 0: Linear PCM (Valid bit of both sub-frame set to 0 ) 1: Non-audio(Valid bit of both sub-frame set to 1)
15:9	/	/	/
8:4	R/W	0xF	TX_RATIO TX Clock Divide Ratio Clock divide ratio = TX_TATIO +1 Fs= PLL_AUDIO/[(TX_TATIO +1)*64*2]
3:2	R/W	0x0	TX_SF TX Sample Format 00: 16 bits 01: 20 bits 10: 24 bits 11: Reserved
1	R/W	0x0	TX_CHM CHSTMODE 0: Channel status A&B set to 0 1: Channel status A&B generated from TX_CHSTA
0	R/W	0x0	TXEN 0: Disabled 1: Enabled

### 5.3.5.3. 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 0: Unlock 1: Lock
3	R/W	0x0	RX_CHST_SRC 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 When setting to '1', the channel status information is capturing, the bit will clear to '0' after captured.
0	R/W	0x0	RXEN 0: Disabled 1: Enabled

### 5.3.5.4. OWA Interrupt Status Register (Default Value: 0x0000\_0010)

Offset: 0x000C			Register Name: OWAISTA
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/
18	R/W1C	0x0	RX_LOCK_INT 0: No Pending IRQ 1: RX Lock Pending Interrupt (RX_LOCK_FLAG 0→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 1→0) 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.
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 interrupt condition fails.

#### 5.3.5.5. 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 Host can get one sample by reading this register, the A channel data is first and then the B channel data.

#### 5.3.5.6. OWA\_FIFO Control Register (Default Value: 0x0004\_0200)

Offset: 0x0014			Register Name: OWA_FCTL
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	HUB_EN Audio Hub Enable

			0: Disable 1: Enable
30	R/W1C	0x0	FTX Write '1' to flush TXFIFO, self clear to '0'.
29	R/W1C	0x0	FRX Write '1' to flush RXFIFO, self clear to '0'.
28:20	/	/	/
19:12	R/W	0x40	TXTL TX FIFO Empty Trigger Level Interrupt and DMA request trigger level for TX FIFO normal condition. Trigger Level = TXTL
11	/	/	/
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 sign 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]}

### 5.3.5.7. OWA FIFO Status Register (Default Value: 0x8080\_0000)

Offset: 0x0018			Register Name: OWA_FSTA
Bit	Read/Write	Default/Hex	Description
31	R	0x1	TXE TXFIFO Empty (indicate TXFIFO is not full) 0: No room for new sample in TXFIFO

			1: More than one room for new sample in TXFIFO ( >=1 Word )
30:24	/	/	/
23:16	R	0x80	TXE_CNT TXFIFO Empty Space Word Counter
15	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

### 5.3.5.8. 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: Disable 1: Enable
17	R/W	0x0	RX_UNLOCKI_EN RX UNLOCK Interrupt Enable 0: Disable 1: Enable
16	R/W	0x0	RX_PARERRI_EN RX PARITY ERROR Interrupt Enable 0: Disable 1: Enable
15:8	/	/	/
7	R/W	0x0	TX_DRQ TXFIFO Empty DRQ Enable 0: Disable 1: Enable
6	R/W	0x0	TXUI_EN TXFIFO Underrun Interrupt Enable 0: Disable 1: Enable
5	R/W	0x0	TXOI_EN TXFIFO Overrun Interrupt Enable 0: Disable 1: Enable
4	R/W	0x0	TXEI_EN

			TXFIFO Empty Interrupt Enable 0: Disable 1: Enable
3	/	/	/
2	R/W	0x0	RX_DRQ RXFIFO Data Available DRQ Enable When setting to '1', RXFIFO DMA Request is asserted if data is available in RXFIFO. 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

### 5.3.5.9. OWA TX FIFO Register (Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: OWA_TXFIFO
Bit	Read/Write	Default/Hex	Description
31:0	W	0x0	TX_DATA Transmitting A, B channel data should be written this register one by one. The A channel data is first and then the B channel data.

### 5.3.5.10. 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 base of this initial value.

### 5.3.5.11. 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	<p>RX_CNT RX Sample Counter The audio sample number of writing into RXFIFO.</p> <p>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 been updated by the initial value, the counter register should count on base of this value.</p>

### 5.3.5.12. 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	<p>CA Clock Accuracy 00: Level 2 01: Level 1 10: Level 3 11: Not matched</p>
27:24	R/W	0x0	<p>FREQ Sampling Frequency 0000: 44.1kHz 0001: Not indicated 0010: 48kHz 0011: 32kHz 0100: 22.05kHz 0101: Reserved 0110: 24kHz 0111: Reserved 1000: Reserved 1001: 768kHz 1010: 96kHz 1011: Reserved 1100: 176.4kHz 1101: Reserved 1110: 192kHz 1111: Reserved</p>
23:20	R/W	0x0	CN Channel Number
19:16	R/W	0x0	SN Source Number
15:8	R/W	0x0	CC Category Code

			Indicates the kind of equipment that generates the digital audio interface signal.
7:6	R/W	0x0	<p>MODE</p> <p>Mode</p> <p>00: Default Mode</p> <p>01~11: Reserved</p>
5:3	R/W	0x0	<p>EMP</p> <p>Emphasis</p> <p>Additional format information</p> <p>For bit 1 = "0", Linear PCM audio mode:</p> <p>000: 2 audio channels without pre-emphasis</p> <p>001: 2 audio channels with 50 µs / 15 µs pre-emphasis</p> <p>010: Reserved (for 2 audio channels with pre-emphasis)</p> <p>011: Reserved (for 2 audio channels with pre-emphasis)</p> <p>100~111: Reserved</p> <p>For bit 1 = "1", other than Linear PCM applications:</p> <p>000: Default state</p> <p>001~111: Reserved</p>
2	R/W	0x0	<p>CP</p> <p>Copyright</p> <p>0: Copyright is asserted</p> <p>1: No copyright is asserted</p>
1	R/W	0x0	<p>TYPE</p> <p>Audio Data Type</p> <p>0: Linear PCM samples</p> <p>1: Non-linear PCM audio</p>
0	R/W	0x0	<p>PRO</p> <p>Application Type</p> <p>0: Consumer application</p> <p>1: Professional application</p> <p>This bit must be fixed to "0".</p>

### 5.3.5.13. OWA TX Channel Status Register1 (Default Value: 0x0000\_0000)

Offset: 0x0030			Register Name: OWA_TX_CHSTA1
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9:8	R/W	0x0	<p>CGMS_A</p> <p>00: Copying is permitted without restriction</p> <p>01: One generation of copies may be made</p> <p>10: Condition not be used</p> <p>11: No copying is permitted</p>
7:4	R/W	0x0	ORIG_FREQ

			Original Sampling Frequency 0000: Not indicated 0001: 192kHz 0010: 12kHz 0011: 176.4kHz 0100: Reserved 0101: 96kHz 0110: 8kHz 0111: 88.2kHz 1000: 16kHz 1001: 24kHz 1010: 11.025kHz 1011: 22.05kHz 1100: 32kHz 1101: 48kHz 1110: Reserved 1111: 44.1kHz
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

#### 5.3.5.14. 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	<p>CA Clock Accuracy 00: Level 2 01: Level 1 10: Level 3 11: Not Matched</p>
27:24	R/W	0x0	<p>FREQ Sampling Frequency 0000: 44.1kHz 0001: Not Indicated 0010: 48kHz 0011: 32kHz 0100: 22.05kHz 0101: Reserved 0110: 24kHz 0111: Reserved 1000: Reserved 1001: 768kHz 1010: 96kHz 1011: Reserved 1100: 176.4kHz 1101: Reserved 1110: 192kHz 1111: Reserved</p>
23:20	R/W	0x0	CN Channel Number
19:16	R/W	0x0	SN Source Number
15:8	R/W	0x0	CC Category Code Indicates the Kind of Equipment that Generates the digital audio interface Signal.
7:6	R/W	0x0	MODE Mode 00: Default mode 01~11: Reserved
5:3	R/W	0x0	EMP Emphasis Additional Format Information For bit 1 = '0', Linear PCM Audio mode: 000: 2 Audio channels without pre-emphasis 001: 2 Audio channels with 50 µs/15 µs pre-emphasis 010: Reserved (For 2 Audio channels with pre-emphasis)

			011: Reserved (For 2 Audio channels with pre-emphasis) 100~111: Reserved  For bit 1 = '1', Other than Linear PCM applications: 000: Default state 001~111: Reserved
2	R/W	0x0	CP Copyright 0: Copyright is asserted 1: No Copyright is asserted
1	R/W	0x0	TYPE Audio Data Type 0: Linear PCM samples 1: Non-linear PCM audio
0	R/W	0x0	PRO Application Type 0: Consumer application 1: Professional application

#### 5.3.5.15. 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
7:4	R/W	0x0	ORIG_FREQ Original Sampling Frequency 0000: Not indicated 0001: 192kHz 0010: 12kHz 0011: 176.4kHz 0100: Reserved 0101: 96kHz 0110: 8kHz 0111: 88.2kHz 1000: 16kHz 1001: 24kHz 1010: 11.025kHz 1011: 22.05kHz 1100: 32kHz

			1101: 48kHz 1110: Reserved 1111: 44.1kHz
3:1	R/W	0x0	<p>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</p> <p>For bit 0 = '1': 000: Not indicated 001: 20 bits 010: 22 bits 100: 23 bits 101: 24 bits 110: 21 bits 111: Reserved</p>
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

## 5.4. Audio Codec

### 5.4.1. Overview

The Audio Codec has 2 I2S/PCM interfaces, 1-ch DAC and 3-ch ADC with a high level of mixed-signal integration which ideal for smart phone and other portable devices. The DRC with integrated hardware DAP engine can be used in record and playback paths .

The Audio Codec has the following features:

- One audio digital-to-analog(DAC) channel
  - Supports 8 kHz to 192 kHz DAC sample rate
  - $95\pm2$ dB SNR@A-weight,  $-80\pm3$ dB THD+N
  - Supports 16-bit and 20-bit audio sample resolution
- One audio output
  - One differential LINEOUTP/N or single-end LINEOUTL output
  - Full-scale output level is 1.1Vrms@0dBFS differential LINEOUTP/N
  - Full-scale output level is 0.55Vrms@0dBFS single-ended LINEOUTL
- Three audio analog-to-digital(ADC) channels
  - Supports 8 kHz to 48 kHz ADC sample rate
  - $95\pm2$ dB SNR@A-weight,  $-80\pm3$ dB THD+N
  - Supports 16-bit and 20-bit audio sample resolution
- Three audio inputs
  - Three differential microphone inputs(MICIN1P and MICIN1N, MICIN2P and MICIN2N, MICIN3P and MICIN3N)
  - Full-scale input level is 1.8Vpp and maximum undistorted input level is 1.7Vpp
- Supports Dynamic Range Controller(DRC) adjusting the ADC recording and DAC playback
- One low-noise analog microphone bias output
- One 128x24-bits FIFO for DAC data transmit, one 128x24-bits FIFO for ADC data receive
- Programmable FIFO thresholds
- Interrupt and DMA support

### 5.4.2. Block Diagram

Figure 5-16 shows the block diagram of Audio Codec.

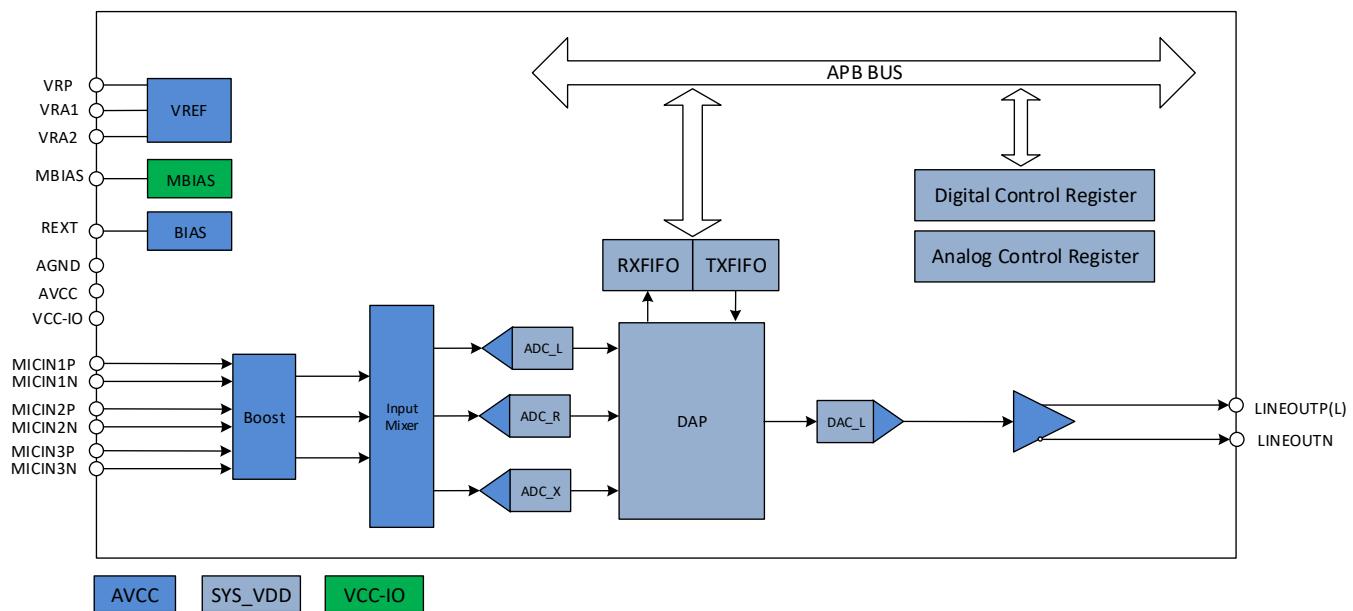


Figure 5-16. Audio Codec Block Diagram

### 5.4.3. Operations and Functional Descriptions

#### 5.4.3.1. External Signals

##### 5.4.3.1.1. Analog I/O Pins

Signal	Type	Description
MICIN1P	AI	Positive differential input for MIC1
MICIN1N	AI	Negative differential input for MIC1
MICIN2P	AI	Positive differential input for MIC2
MICIN2N	AI	Negative differential input for MIC2
MICIN3P	AI	Positive differential input for MIC3
MICIN3N	AI	Negative differential input for MIC3
LINEOUTP(L)	AO	Differential mono positive output(or left single-end output for lineout)
LINEOUTN	AO	Differential mono negative output

##### 5.4.3.1.2. Reference

Signal	Type	Description
MBIAS	AO	First bias voltage output for main microphone
VRA1	AO	Internal reference voltage
VRA2	AO	Internal reference voltage
VRP	AO	Internal reference voltage
REXT	AO	External reference pin

### 5.4.3.1.3. Power/Ground

Signal	Type	Description
AVCC	P	Analog power
AGND	G	Analog ground

### 5.4.3.2. Clock Sources

Figure 5-17 describes the Audio Codec clock source. Users can see **CCU** for clock setting, configuration and gating information.

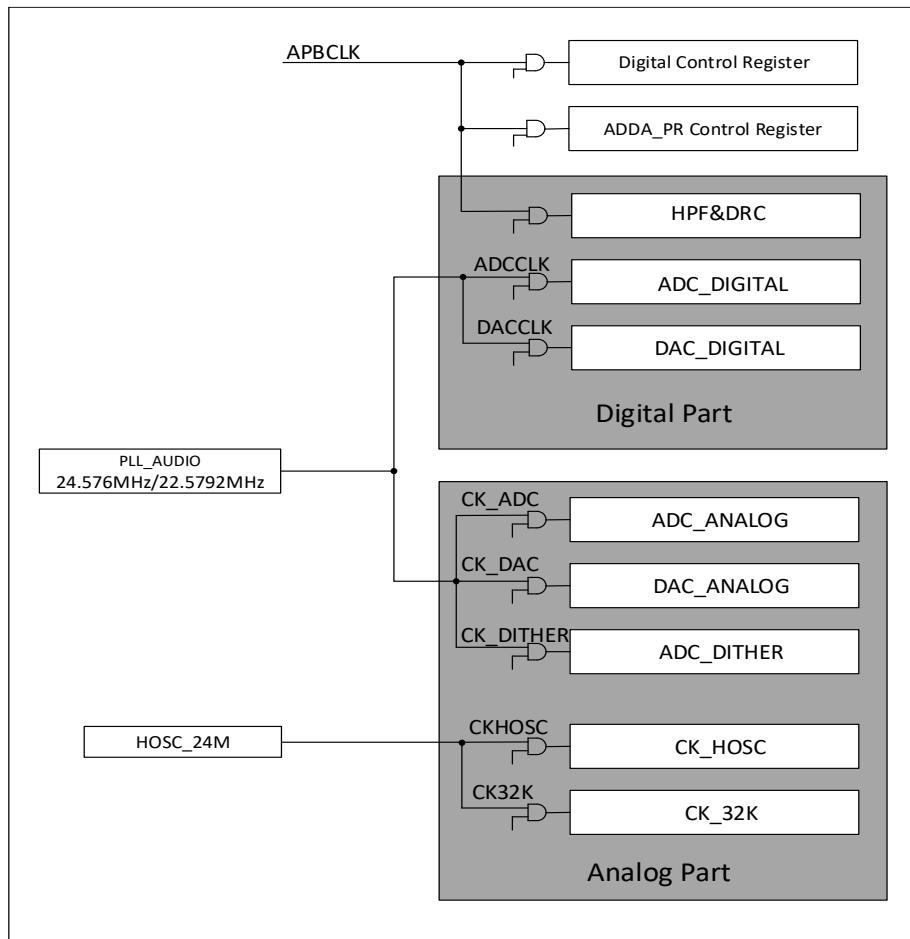


Figure 5- 17. Audio Codec Clock Diagram

The clock of digital part is from PLL\_AUDIO(1X). The clock of analog part includes CK\_ADC, CK\_DAC, CK\_DITHER, CKHOSC, CK32K. CK\_ADC, CK\_DAC, CK\_DITHER is provided by PLL\_AUDIO(1X); CKHOSC, CK32K is provided by 24M system oscillator.

### 5.4.3.3. Reset System

#### 5.4.3.3.1. Digital Part Reset System

The SYS\_RST will be provided by the VDD-SYS domain, which comes from VDD-SYS domain and is produced by RTC domain. Each domain has the de-bounce to confirm the reset system is strong. The codec register part, MIX will be reset by the SYS\_RST during the power on or the system soft writing the reset control logic. The other parts will be reset by the soft configure through writing register.

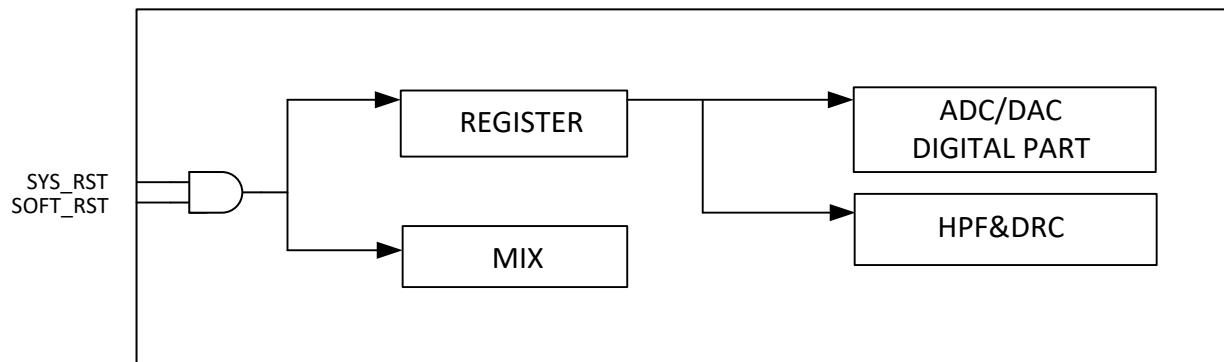


Figure 5- 18. Audio Codec Digital Part Reset System

#### 5.4.3.3.2. Analog Part Reset System

When AVCC is powered on, it will send the AVCC\_POR signal. And the AVCC\_POR signal passes the level shift and RC filter part to ADDA logic core.

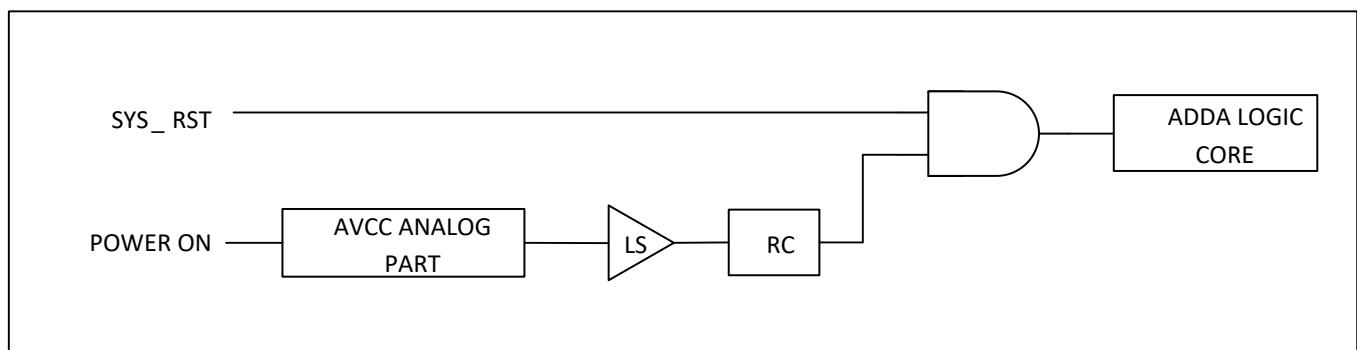
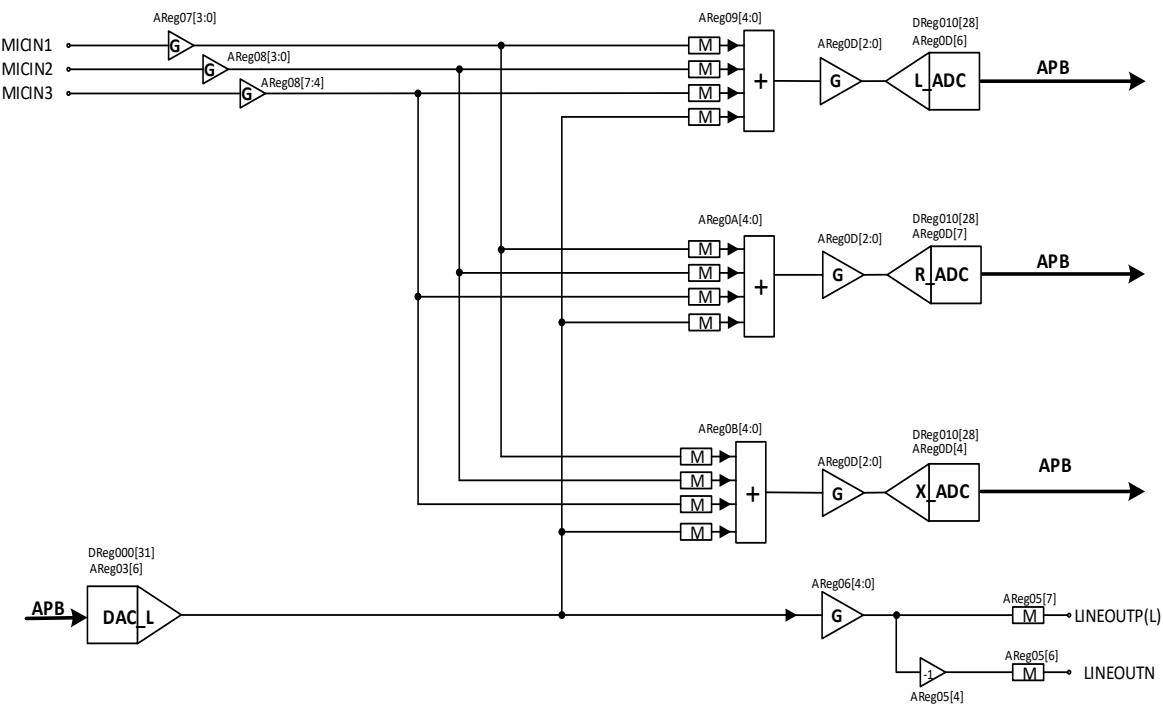


Figure 5- 19. Audio Codec Analog Part Reset System

#### 5.4.3.4. Data Path Diagram

Figure 5-20 shows a data path of the Audio Codec.



Note: Areg indicates analog register, Dreg indicates digital register

**Figure 5- 20. Audio Codec Data Path Diagram**

#### 5.4.3.5. Three ADC

The three ADC is used for recording stereo sound and a reference signal. The sample rate of the three ADC is independent of DAC sample rate. In order to save power, the left and right analog ADC part can be enabled/disabled separately by setting the bit[7:6] of the **AC\_ADC\_CTRL** register. The digital ADC part can be enabled/disabled by the bit28 of the **AC\_ADC\_FIFOC** register.

#### 5.4.3.6. Mono DAC

The mono DAC sample rate can be configured by setting the register. In order to save power, the DAC can be enabled/disabled by setting the bit[6] of the **MIX\_DAC\_CTRL** register. The digital DAC part can be enabled/disabled by the bit[31] of the **AC\_DAC\_DPC** register.

#### 5.4.3.7. Mixer

The Audio Codec supports two mixers for all function requirements:

- 3 channels ADC Record mixers

The ADC record mixer is used to mix analog signals as input to the ADC for recording. The following signals can be mixed into the input mixer.

- MICIN1P/N

- MICIN2P/N
- MICIN3P/N
- Mono DAC output

#### 5.4.3.8. Analog Audio Input Path

The Audio Codec supports three analog audio input paths:

- MICIN1P/N
- MICIN2P/N
- MICIN3P/N

MICIN1P/N, MICIN2P/N and MICIN3P/N provide differential input that can be mixed into the ADC record mixer. MICIN is high impedance, low capacitance input suitable for connection to a wide range of differential microphones of different dynamics and sensitive. The gain for each pre-amplifier can be set independently. MBIAS provides reference voltage for electret condenser type(ECM) microphones.

#### 5.4.3.9. Analog Audio Output Path

The Audio Codec has one type analog output port:

- LINEOUTP/N or LINEOUTL

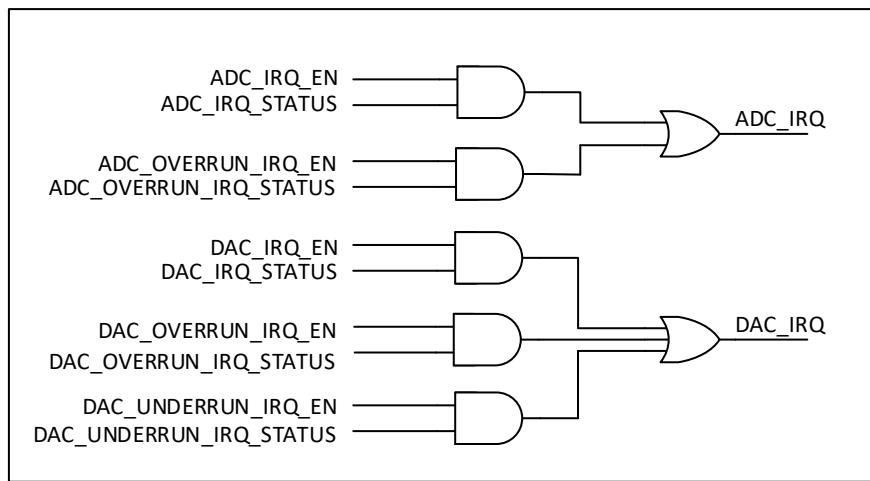
The LINEOUT provides one differential output to drive line level signals to external audio equipment. The LINEOUTL(P) output source is from DACL. The LINEOUTR output source is from DAC differential output. The volume control is logarithmic with an 43.5dB rang in 1.5dB step from -43.5dB to 0dB. The LINEOUT output buffer is powered up or down by the bit[7:6] of **LINEOUT\_CTRL0**.

#### 5.4.3.10. Microphone BIAS

The MBIAS output provides a low noise reference voltage suitable for biasing electrets type microphones and the associated external resistor biasing network. Refer to the Applications Information section for recommended external components.

#### 5.4.3.11. Interrupt

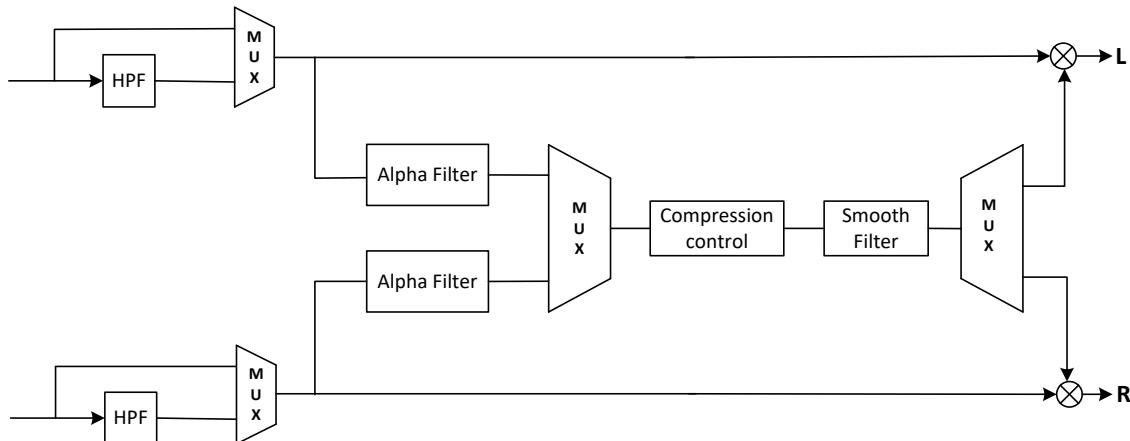
The Audio Codec has two interrupts. Figure 5-21 describes the Audio Codec interrupt system.



**Figure 5- 21. Audio Codec Interrupt System**

#### 5.4.3.12. DAP

##### 5.4.3.12.1. DAP Data Flow



**Figure 5- 22. DAP Data Flow**

##### 5.4.3.12.2. HPF Function

The DAP has individual channel high pass filter (HPF, -3dB cutoff < 1Hz) that can be enabled and disabled. The filter cutoff frequency is less than 1 Hz that can be removed DC offset from ADC recording. The HPF can also be bypassed.

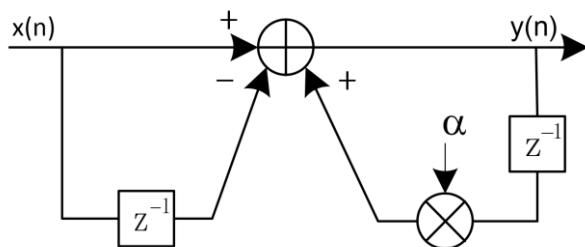


Figure 5- 23. HPF Function

#### 5.4.3.12.3. DRC Function

The DRC scheme has three thresholds, three offset, and four slope (all programmable). There is one ganged DRC for the left/right channels. The diagram of DRC input/output is as follows.

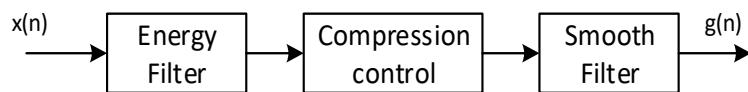


Figure 5- 24. DRC Block Diagram

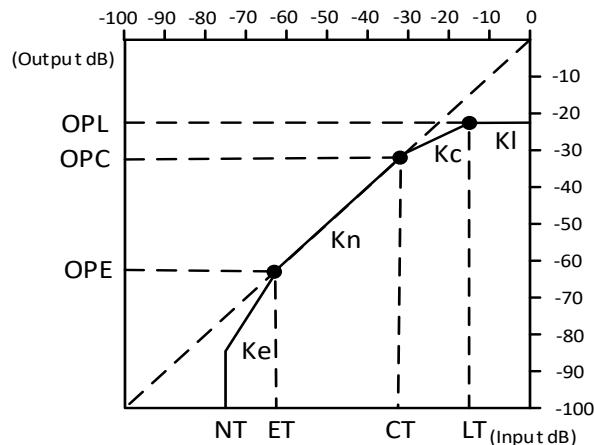


Figure 5- 25. DRC Static Curve Parameters

Professional-quality dynamic range compression automatically adjusts volume to flatten volume level.

One DRC for left/right and one DRC for subwoofer.

Each DRC has adjustable threshold, offset, and compression levels, programmable energy, attack, and decay time constants.

**Transparent compression:** Compressors can attack fast enough to avoid apparent clipping before engaging, and decay times can be set slow enough to avoid pumping.

#### DRC parameter setting:

- Number format

The Number format is N.M which means there are N bits to the left of the decimal point including the sign bit and M bits to the right of the decimal point. For example, Numbers formatted 9.23 means that there are 9 bits at the left of the decimal point and 23 bits at the right decimal point.

- Energy Filter

The Energy Filter is to estimate the RMS value of the audio data stream into DRC, and has two parameters, which determine the time window over which RMS to be made. The parameter is computed by  $\alpha = 1 - e^{-2.2Ts/ta}$ .

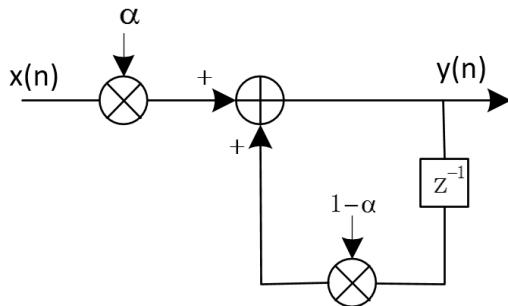


Figure 5- 26. Energy Filter Structure

### Compression Control

This element has ten parameters ( ET, CT, LT, Ke, Kn, Kc, KI, OPL, OPC, OPE), which are all programmable, and the computation will be explained as follows.

- Threshold Parameter Computation(T parameter)

The threshold is the value that determines the signal to be compressed or not. When the signal's RMS is larger than the threshold, the signal will be compressed. The value of threshold input to the coefficient register is computed by

$$Tin = -\frac{T_{dB}}{6.0206}$$

There,  $T_{dB}$  must less than zero, the positive value is illegal.

For example,it is desired to set CT=-40dB,then the Tin require to set CT to -40dB is  $CT_{in} = -(-40dB)/6.0206 = 6.644$ ,  $CT_{in}$  is entered as a 32-bit number in 8.24 format.

Therefore,  $CT_{in} = 6.644 = 0000\ 0110.1010\ 0100\ 1101\ 0011\ 1100\ 0000 = 0x06A4\ D3C0$  in 8.24 format.

- Slope Parameter Computation (K parameter)

The K is the slope within compression region. For example, a n:1 compression means that an output increase 1dB is for n dB RMS input. The k input to the coefficient ram is computed by  $K = \frac{1}{n}$

There, n is from 1 to 50, and must be integer.

For example,it is desired to set 2:1,then the Kc require to set to 2:1 is  $Kc = 1/2 = 0.5$ , Kc is entered as a 32-bit number in 8.24 format.

Therefore,  $Kc = 0.5 = 0000\ 0000.1000\ 0000\ 0000\ 0000\ 0000 = 0x0080\ 0000$  in 8.24 format.

- Gain Smooth Filter

The Gain Smooth Filter is to smooth the gain and control the ratio of gain increase and decrease. The decay time and attack is shown in Figure 5-27. The structure of the Gain Smooth filter is also the Alpha filter, so the rise time computation is the same as the Energy filter which is  $\alpha = 1 - e^{-2.2Ts/ta}$ .

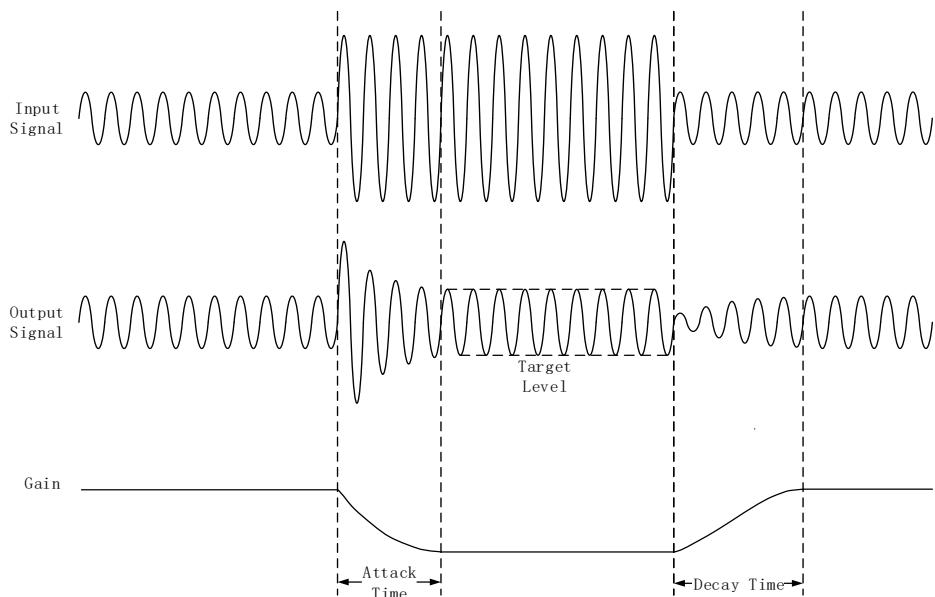


Figure 5- 27. Gain Smooth Filter

#### 5.4.4. Programming Guidelines

##### 5.4.4.1. Playback Process

- (1) Codec initial: Open audio codec bus clock gating and de-assert bus reset through **AUDIO\_CODEC\_BGR\_REG**, configure PLL\_Audio frequency and enable PLL\_Audio through **PLL\_AUDIO\_CTRL\_REG**. Please refer to CCU in chapter 3.3 about detail.
- (2) Configure MIX path.
- (3) Set sample rate,configure data transfer format, open DAC.
- (4) DMA configure and DMA request.
- (5) Enable DAC DRQ and DMA.

##### 5.4.4.2. Record Process

- (1) Codec initial: Open audio codec bus clock gating and de-assert bus reset through **AUDIO\_CODEC\_BGR\_REG**, configure PLL\_Audio frequency and enable PLL\_Audio through **PLL\_AUDIO\_CTRL\_REG**.Please refer to CCU in chapter 3.3 about detail.
- (2) Configure MIX path.
- (3) Set sample rate,configure data transfer format, open ADC.
- (4) DMA configure and DMA request.
- (5) Enable ADC DRQ and DMA.

#### 5.4.5. Register List

Module Name	Base Address
Audio Codec	0x05096000

Register Name	Offset	Description
AC_DAC_DPC	0x0000	DAC Digital Part Control Register
AC_DAC_FIFOC	0x0010	DAC FIFO Control Register
AC_DAC_FIFOS	0x0014	DAC FIFO Status Register
AC_DAC_TXDATA	0x0020	DAC TX DATA Register
AC_DAC_CNT	0x0024	DAC TX FIFO Counter Register
AC_DAC_DG	0x0028	DAC Debug Register
AC_ADC_FIFOC	0x0030	ADC FIFO Control Register
AC_ADC_FIFOS	0x0034	ADC FIFO Status Register
AC_ADC_RXDATA	0x0040	ADC RX Data Register
AC_ADC_CNT	0x0044	ADC RX Counter Register
AC_ADC_DG	0x004C	ADC Debug Register
AC_DAC_DAP_CTRL	0x00F0	DAC DAP Control Register
AC_ADC_DAP_CTR	0x00F8	ADC DAP Control Register
AC_DAC_DRC_HHPFC	0x0100	DAC DRC High HPF Coef Register
AC_DAC_DRC_LHPFC	0x0104	DAC DRC Low HPF Coef Register
AC_DAC_DRC_CTRL	0x0108	DAC DRC Control Register
AC_DAC_DRC_LPFHAT	0x010C	DAC DRC Left Peak Filter High Attack Time Coef Register
AC_DAC_DRC_LPFLAT	0x0110	DAC DRC Left Peak Filter Low Attack Time Coef Register
AC_DAC_DRC_RPFHAT	0x0114	DAC DRC Right Peak Filter High Attack Time Coef Register
AC_DAC_DRC_RPFLAT	0x0118	DAC DRC Peak Filter Low Attack Time Coef Register
AC_DAC_DRC_LPFHRT	0x011C	DAC DRC Left Peak Filter High Release Time Coef Register
AC_DAC_DRC_LPFLRT	0x0120	DAC DRC Left Peak Filter Low Release Time Coef Register
AC_DAC_DRC_RPFHRT	0x0124	DAC DRC Right Peak filter High Release Time Coef Register
AC_DAC_DRC_RPFLRT	0x0128	DAC DRC Right Peak filter Low Release Time Coef Register
AC_DAC_DRC_LRMSHAT	0x012C	DAC DRC Left RMS Filter High Coef Register
AC_DAC_DRC_LRMSLAT	0x0130	DAC DRC Left RMS Filter Low Coef Register
AC_DAC_DRC_RRMSHAT	0x0134	DAC DRC Right RMS Filter High Coef Register
AC_DAC_DRC_RRMSLAT	0x0138	DAC DRC Right RMS Filter Low Coef Register
AC_DAC_DRC_HCT	0x013C	DAC DRC Compressor Threshold High Setting Register
AC_DAC_DRC_LCT	0x0140	DAC DRC Compressor Slope High Setting Register
AC_DAC_DRC_HKC	0x0144	DAC DRC Compressor Slope High Setting Register
AC_DAC_DRC_LKC	0x0148	DAC DRC Compressor Slope Low Setting Register
AC_DAC_DRC_HOPC	0x014C	DAC DRC Compressor High Output at Compressor Threshold Register
AC_DAC_DRC_LOPC	0x0150	DAC DRC Compressor Low Output at Compressor Threshold

		Register
AC_DAC_DRC_HLT	0x0154	DAC DRC Limiter Threshold High Setting Register
AC_DAC_DRC_LLT	0x0158	DAC DRC Limiter Threshold Low Setting Register
AC_DAC_DRC_HKI	0x015C	DAC DRC Limiter Slope High Setting Register
AC_DAC_DRC_LKI	0x0160	DAC DRC Limiter Slope Low Setting Register
AC_DAC_DRC_HOPL	0x0164	DAC DRC Limiter High Output at Limiter Threshold
AC_DAC_DRC_LOPL	0x0168	DAC DRC Limiter Low Output at Limiter Threshold
AC_DAC_DRC_HET	0x016C	DAC DRC Expander Threshold High Setting Register
AC_DAC_DRC LET	0x0170	DAC DRC Expander Threshold Low Setting Register
AC_DAC_DRC_HKE	0x0174	DAC DRC Expander Slope High Setting Register
AC_DAC_DRC_LKE	0x0178	DAC DRC Expander Slope Low Setting Register
AC_DAC_DRC_HOPE	0x017C	DAC DRC Expander High Output at Expander Threshold
AC_DAC_DRC_LOPE	0x0180	DAC DRC Expander Low Output at Expander Threshold
AC_DAC_DRC_HKN	0x0184	DAC DRC Linear Slope High Setting Register
AC_DAC_DRC_LKN	0x0188	DAC DRC Linear Slope Low Setting Register
AC_DAC_DRC_SFHAT	0x018C	DAC DRC Smooth filter Gain High Attack Time Coef Register
AC_DAC_DRC_SFLAT	0x0190	DAC DRC Smooth filter Gain Low Attack Time Coef Register
AC_DAC_DRC_SFVRT	0x0194	DAC DRC Smooth filter Gain High Release Time Coef Register
AC_DAC_DRC_SFLRT	0x0198	DAC DRC Smooth filter Gain Low Release Time Coef Register
AC_DAC_DRC_MXGHS	0x019C	DAC DRC MAX Gain High Setting Register
AC_DAC_DRC_MXGLS	0x01A0	DAC DRC MAX Gain Low Setting Register
AC_DAC_DRC_MNGHS	0x01A4	DAC DRC MIN Gain High Setting Register
AC_DAC_DRC_MNGLS	0x01A8	DAC DRC MIN Gain Low Setting Register
AC_DAC_DRC_EPSHC	0x01AC	DAC DRC Expander Smooth Time High Coef Register
AC_DAC_DRC_EPSLC	0x01B0	DAC DRC Expander Smooth Time Low Coef Register
AC_DAC_DRC_HPFHGAIN	0x01B8	DAC DRC HPF Gain High Coef Register
AC_DAC_DRC_HPFLGAIN	0x01BC	DAC DRC HPF Gain Low Coef Register
AC_ADC_DRC_HHPFC	0x0200	ADC DRC High HPF Coef Register
AC_ADC_DRC_LHPFC	0x0204	ADC DRC Low HPF Coef Register
AC_ADC_DRC_CTRL	0x0208	ADC DRC Control Register
AC_ADC_DRC_LPFHAT	0x020C	ADC DRC Left Peak Filter High Attack Time Coef Register
AC_ADC_DRC_LPFLAT	0x0210	ADC DRC Left Peak Filter Low Attack Time Coef Register
AC_ADC_DRC_RPFHAT	0x0214	ADC DRC Right Peak Filter High Attack Time Coef Register
AC_ADC_DRC_RPFLAT	0x0218	ADC DRC Peak Filter Low Attack Time Coef Register
AC_ADC_DRC_LPFHRT	0x021C	ADC DRC Left Peak Filter High Release Time Coef Register
AC_ADC_DRC_LPFLRT	0x0220	ADC DRC Left Peak Filter Low Release Time Coef Register
AC_ADC_DRC_RPFHRT	0x0224	ADC DRC Right Peak filter High Release Time Coef Register
AC_ADC_DRC_RPFLRT	0x0228	ADC DRC Right Peak filter Low Release Time Coef Register
AC_ADC_DRC_LRMSHAT	0x022C	ADC DRC Left RMS Filter High Coef Register
AC_ADC_DRC_LRMSLAT	0x0230	ADC DRC Left RMS Filter Low Coef Register
AC_ADC_DRC_RRMSHAT	0x0234	ADC DRC Right RMS Filter High Coef Register
AC_ADC_DRC_RRMSLAT	0x0238	ADC DRC Right RMS Filter Low Coef Register
AC_ADC_DRC_HCT	0x023C	ADC DRC Compressor Threshold High Setting Register

AC_ADC_DRC_LCT	0x0240	ADC DRC Compressor Slope High Setting Register
AC_ADC_DRC_HKC	0x0244	ADC DRC Compressor Slope High Setting Register
AC_ADC_DRC_LKC	0x0248	ADC DRC Compressor Slope Low Setting Register
AC_ADC_DRC_HOPC	0x024C	ADC DRC Compressor High Output at Compressor Threshold Register
AC_ADC_DRC_LOPC	0x0250	ADC DRC Compressor Low Output at Compressor Threshold Register
AC_ADC_DRC_HLT	0x0254	ADC DRC Limiter Threshold High Setting Register
AC_ADC_DRC_LLT	0x0258	ADC DRC Limiter Threshold Low Setting Register
AC_ADC_DRC_HKI	0x025C	ADC DRC Limiter Slope High Setting Register
AC_ADC_DRC_LKI	0x0260	ADC DRC Limiter Slope Low Setting Register
AC_ADC_DRC_HOPL	0x0264	ADC DRC Limiter High Output at Limiter Threshold
AC_ADC_DRC_LOPL	0x0268	ADC DRC Limiter Low Output at Limiter Threshold
AC_ADC_DRC_HET	0x026C	ADC DRC Expander Threshold High Setting Register
AC_ADC_DRC LET	0x0270	ADC DRC Expander Threshold Low Setting Register
AC_ADC_DRC_HKE	0x0274	ADC DRC Expander Slope High Setting Register
AC_ADC_DRC_LKE	0x0278	ADC DRC Expander Slope Low Setting Register
AC_ADC_DRC_HOPE	0x027C	ADC DRC Expander High Output at Expander Threshold
AC_ADC_DRC_LOPE	0x0280	ADC DRC Expander Low Output at Expander Threshold
AC_ADC_DRC_HKN	0x0284	ADC DRC Linear Slope High Setting Register
AC_ADC_DRC_LKN	0x0288	ADC DRC Linear Slope Low Setting Register
AC_ADC_DRC_SFHT	0x028C	ADC DRC Smooth filter Gain High Attack Time Coef Register
AC_ADC_DRC_SFLAT	0x0290	ADC DRC Smooth filter Gain Low Attack Time Coef Register
AC_ADC_DRC_SFVRT	0x0294	ADC DRC Smooth filter Gain High Release Time Coef Register
AC_ADC_DRC_SFLRT	0x0298	ADC DRC Smooth filter Gain Low Release Time Coef Register
AC_ADC_DRC_MXGHS	0x029C	ADC DRC MAX Gain High Setting Register
AC_ADC_DRC_MXGLS	0x02A0	ADC DRC MAX Gain Low Setting Register
AC_ADC_DRC_MNGLS	0x02A4	ADC DRC MIN Gain High Setting Register
AC_ADC_DRC_MXGLS	0x02A8	ADC DRC MIN Gain Low Setting Register
AC_ADC_DRC_EPSHC	0x02AC	ADC DRC Expander Smooth Time High Coef Register
AC_ADC_DRC_EPSLC	0x02B0	ADC DRC Expander Smooth Time Low Coef Register
AC_ADC_DRC_HPFHGAIN	0x02B8	ADC DRC HPF Gain High Coef Register
AC_ADC_DRC_HPFLGAIN	0x02BC	ADC DRC HPF Gain Low Coef Register
Analog Domain Register		
LINEOUT_CTRL0	0x05	LINEOUT Control Register 0
LINEOUT_CTRL1	0x06	LINEOUT Control Register 1
MIC1_CTRL	0x07	MIC1 Control Register
MIC2_MIC3_CTRL	0x08	MIC2 MIC3 Control Register
L_ADCMIX_SRC	0x09	Left ADC Mixer Control Register
R_ADCMIX_SRC	0x0A	Right ADC Mixer Control Register
X_ADCMIX_SRC	0x0B	X ADC Mixer Control Register
ADC_CTRL	0x0D	ADC Control Register
MBIAS_CTRL	0x0E	Microphone Bias Control Register

APT_REG	0x0F	Analog Performance Tuning Register
OP_BIAS_CTRL0	0x10	OP BIAS Control Register0
OP_BIAS_CTRL1	0x11	OP BIAS Control Register1
ZC_VOL_CTRL	0x12	ZERO Cross &USB Bias Control Register
BIAS_CAL_CTRL	0x15	Bias Calibration Control Register

#### 5.4.6. Register Description

##### 5.4.6.1. DAC Digital Part Control Register(Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: AC_DAC_DPC
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	EN_DA DAC Digital Part Enable 0: Disable 1: Enable
30:29	/	/	/
28:25	R/W	0x0	MODQU Internal DAC Quantization Levels Levels=[7*(21+MODQU[3:0])]/128 Default levels=7*21/128=1.15
24	R/W	0x0	DWA DWA Function Disable 0: Enable 1: Disable
23:19	/	/	/
18	R/W	0x0	HPF_EN High Pass Filter Enable 0: Disable 1: Enable
17:12	R/W	0x0	DVOL Digital volume control: DVC, ATT=DVC[5:0]*(-1.16dB) 64 steps, -1.16dB/step
11:1	/	/	/
0	R/W	0x0	HUB_EN Audio Hub Enable 0: Disable 1: Enable

##### 5.4.6.2. DAC FIFO Control Register(Default Value: 0x0000\_4000)

Offset: 0x0010	Register Name: AC_DAC_FIFOC
----------------	-----------------------------

Bit	Read/Write	Default/Hex	Description
31:29	R/W	0x0	<p>DAC_FS Sample Rate of DAC 000: 48 kHz 010: 24 kHz 100: 12 kHz 110: 192 kHz 001: 32 kHz 011: 16 kHz 101: 8 kHz 111: 96 kHz 44.1 kHz/22.05 kHz/11.025 kHz can be supported by Audio PLL Configure Bit</p>
28	R/W	0x0	<p>FIR_VER FIR Version 0: 64-Tap FIR 1: 32-Tap FIR</p>
27	/	/	/
26	R/W	0x0	<p>SEND_LASAT Audio sample select when TX FIFO underrun 0: Sending zero 1: Sending last audio sample</p>
25:24	R/W	0x0	<p>FIFO_MODE For 20-bit transmitted audio sample: 00/10: FIFO_I[19:0] = {TXDATA[31:12]} 01/11: FIFO_I[19:0] = {TXDATA[19:0]} For 16-bit transmitted audio sample: 00/10: FIFO_I[19:0] = {TXDATA[31:16], 4'b0} 01/11: FIFO_I[19:0] = {TXDATA[15:0], 4'b0}</p>
23	/	/	/
22:21	R/W	0x0	<p>DAC_DRQ_CLR_CNT When TX FIFO available room is less than or equal N, DRQ Request will be de-asserted. N is defined here: 00: IRQ/DRQ De-asserted when WLEVEL &gt; TXTL 01: 4 10: 8 11: 16</p>
20:15	/	/	/
14:8	R/W	0x40	<p>TX_TRIG_LEVEL TX FIFO Empty Trigger Level (TXTL[12:0]) Interrupt and DMA request trigger level for TX FIFO normal condition. IRQ/DRQ generated when WLEVEL ≤ TXTL</p> <p> <b>NOTE</b></p> <p><b>WLEVEL represents the number of valid samples in the TX FIFO.</b> <b>Only TXTL[6:0] valid when TXMODE = 0</b></p>

7	/	/	/
6	R/W	0x0	DAC_MONO_EN DAC Mono Enable 0: Stereo, 64 levels FIFO 1: mono, 128 levels FIFO When enabled, L & R channel send same data.
5	R/W	0x0	TX_SAMPLE_BITS Transmitting Audio Sample Resolution 0: 16 bits 1: 20 bits
4	R/W	0x0	DAC_DRQ_EN DAC FIFO Empty DRQ Enable 0: Disable 1: Enable
3	R/W	0x0	DAC_IRQ_EN DAC FIFO Empty IRQ Enable 0: Disable 1: Enable
2	R/W	0x0	FIFO_UNDERRUN_IRQ_EN DAC FIFO Underrun IRQ Enable 0: Disable 1: Enable
1	R/W	0x0	FIFO_OVERRUN_IRQ_EN DAC FIFO Overrun IRQ Enable 0: Disable 1: Enable
0	R/WC	0x0	FIFO_FLUSH DAC FIFO Flush Write '1' to flush TX FIFO, self clear to '0'

#### 5.4.6.3. DAC FIFO Status Register(Default Value: 0x0080\_8008)

Offset: 0x0014			Register Name: AC_DAC_FIFOS
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23	R	0x1	TX_EMPTY TX FIFO Empty 0: No room for new sample in TX FIFO 1: More than one room for new sample in TX FIFO (>= 1 word)
22:8	R	0x80	TXE_CNT TX FIFO Empty Space Word Counter
7:4	/	/	/
3	R/W1C	0x1	TXE_INT

			TX FIFO Empty Pending Interrupt 0: No Pending IRQ 1: FIFO Empty Pending Interrupt Write '1' to clear this interrupt or automatically clear if interrupt condition fails.
2	R/W1C	0x0	TXU_INT TX FIFO Underrun Pending Interrupt 0: No Pending Interrupt 1: FIFO Underrun Pending Interrupt Write '1' to clear this interrupt
1	R/W1C	0x0	TXO_INT TX FIFO Overrun Pending Interrupt 0: No Pending Interrupt 1: FIFO Overrun Pending Interrupt Write '1' to clear this interrupt
0	/	/	/

#### 5.4.6.4. DAC TX DATA Register(Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: AC_DAC_TXDATA
Bit	Read/Write	Default/Hex	Description
31:0	W	0x0	TX_DATA Transmitting left, right channel sample data should be written this register one by one. The left channel sample data is first and then the right channel sample.

#### 5.4.6.5. DAC TX Counter Register(Default Value: 0x0000\_0000)

Offset: 0x0024			Register Name: AC_DAC_CNT
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	TX_CNT TX Sample Counter The audio sample number of sending into TXFIFO. When one sample is put into TXFIFO by DMA or by host IO, the TX sample counter register increases by one. The TX sample counter register can be set to any initial value at any time. After been updated by the initial value, the counter register should count on base of this initial value.   <b>NOTE</b> <b>It is used for Audio/Video Synchronization</b>

#### 5.4.6.6. DAC Debug Register(Default Value: 0x0000\_0000)

Offset: 0x0028			Register Name: AC_DAC_DG
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11	R/W	0x0	DAC_MODU_SELECT DAC Modulator Debug 0: DAC Modulator Normal Mode 1: DAC Modulator Debug Mode
10:9	R/W	0x0	DAC_PATTERN_SELECT DAC Pattern Select 00: Normal (Audio Sample from TX FIFO) 01: -6 dB Sin wave 10: -60 dB Sin wave 11: Silent wave
8	R/W	0x0	CODEC_CLK_SELECT CODEC Clock Source Select 0: CODEC Clock from PLL 1: CODEC Clock from OSC (for Debug)
7	/	/	/
6	R/W	0x0	DA_SWP DAC Output Channel Swap Enable 0:Disable 1:Enable
5:3	/	/	/
1:0	R/W	0x0	ADDA_LOOP_MODE ADDA Loop Mode Select 00: Disable 01: ADDA LOOP MODE DACL connect to ADCL 10: ADDA LOOP MODE DACL connect to ADCX 11:Reserved

#### 5.4.6.7. ADC FIFO Control Register(Default Value: 0x0000\_0400)

Offset: 0x0030			Register Name: AC_ADC_FIFOC
Bit	Read/Write	Default/Hex	Description
31:29	R/W	0x0	ADFS Sample Rate of ADC 000: 48 kHz 010: 24 kHz 100: 12 kHz 110: Reserved 001: 32 kHz

			011: 16 kHz 101: 8 kHz 111: Reserved 44.1 kHz/22.05 kHz/11.025 kHz can be supported by Audio PLL Configure Bit
28	R/W	0x0	EN_AD ADC Digital Part Enable 0: Disable 1: Enable
27:26	R/W	0x0	ADCFDT ADC FIFO delay time for writing data after EN_AD 00:5ms 01:10ms 10:20ms 11:30ms
25	R/W	0x0	ADCDFEN ADC FIFO delay function for writing data after EN_AD 0: Disable 1: Enable
24	R/W	0x0	RX_FIFO_MODE RX FIFO Output Mode (Mode 0, 1) 0: Expanding '0' at LSB of TX FIFO register 1: Expanding received sample sign bit at MSB of TX FIFO register For 20-bit received audio sample: Mode 0: RXDATA[31:0] = {FIFO_O[19:0], 12'h0} Mode 1: RXDATA[31:0] = {12{FIFO_O[19]}, FIFO_O[19:0]}  For 16-bit received audio sample: Mode 0: RXDATA[31:0] = {FIFO_O[19:4], 16'h0} Mode 1: RXDATA[31:0] = {16{FIFO_O[19]}, FIFO_O[19:4]}
23:17	/	/	/
16	R/W	0x0	RX_SAMPLE_BITS Receiving Audio Sample Resolution 0: 16 bits 1: 20 bits
15	/	/	/
14:12	R/W	0x0	ADC_CHANNEL_EN Bit 14: ADCX enable Bit 13: ADCR enable Bit 12: ADCL enable
11	/	/	/
10:4	R/W	0x40	RX_FIFO_TRG_LEVEL RX FIFO Trigger Level (RXTL[5:0]) Interrupt and DMA request trigger level for RX FIFO normal condition IRQ/DRQ generated when WLEVEL > RXTL[5:0]

			 <b>NOTE</b> <b>WLEVEL</b> represents the number of valid samples in the RX FIFO.
3	R/W	0x0	<b>ADC_DRQ_EN</b> ADC FIFO Data Available DRQ Enable 0: Disable 1: Enable
2	R/W	0x0	<b>ADC_IRQ_EN</b> ADC FIFO Data Available IRQ Enable 0: Disable 1: Enable
1	R/W	0x0	<b>ADC_OVERRUN_IRQ_EN</b> ADC FIFO Overrun IRQ Enable 0: Disable 1: Enable
0	R/WC	0x0	<b>ADC_FIFO_FLUSH</b> ADC FIFO Flush Write '1' to flush TX FIFO, self clear to '0'.

#### 5.4.6.8. ADC FIFO Status Register(Default Value: 0x0000\_0000)

Offset: 0x0038			Register Name: AC_ADC_FIFOS
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23	R	0x0	<b>RXA</b> RX FIFO Available 0: No available data in RX FIFO 1: More than one sample in RX FIFO (>= 1 word)
22:15	/	/	/
14:8	R	0x0	<b>RXA_CNT</b> RX FIFO Available Sample Word Counter
7:4	/	/	/
3	R/W1C	0x0	<b>RXA_INT</b> 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.
2	/	/	/
1	R/W1C	0x0	<b>RXO_INT</b> RX FIFO Overrun Pending Interrupt 0: No Pending IRQ 1: FIFO Overrun Pending IRQ Write '1' to clear this interrupt

0	/	/	/
---	---	---	---

#### 5.4.6.9. ADC RX DATA Register(Default Value: 0x0000\_0000)

Offset: 0x0040			Register Name: AC_ADC_RXDATA
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.

#### 5.4.6.10. ADC RX Counter Register(Default Value: 0x0000\_0000)

Offset: 0x0044			Register Name: AC_ADC_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 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.  <b>NOTE</b> <b>It is used for Audio/Video Synchronization.</b>

#### 5.4.6.11. ADC Debug Register (Default Value: 0x0000\_0000)

Offset: 0x004C			Register Name: AC_ADC_DG_REG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	AD_SWP ADC output channel swap enable (for digital filter) 0: Disable 1: Enable
23:0	/	/	/

#### 5.4.6.12. DAC DAP Control Register (Default Value: 0x0000\_0000)

Offset: 0x00F0	Register Name: AC_DAC_DAP_CTR
----------------	-------------------------------

Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	DDAP_EN DAP for DRC Enable 0 : Bypass 1 : Enable
30	/	/	/
29	R/W	0x0	DDAP_DRC_EN DRC enable control 0:Disable 1:Enable
28	R/W	0x0	DDAP_HPF_EN HPF enable control 0:Disable 1:Enable
27:0	/	/	/

#### 5.4.6.13. ADC DAP Control Register (Default Value: 0x0000\_0000)

Offset: 0x00F8			Register Name: AC_ADC_DAP_CTR
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	ADC_DAP0_EN DAP for ADC enable 0:Bypass 1:Enable
30	/	/	/
29	R/W	0x0	ADC_DRCO_EN ADC DRCO enable control 0:Disable 1:Enable
28	R/W	0x0	ADC_HPFO_EN ADC HPFO enable control 0:Disable 1:Enable
27	R/W	0x0	ADC_DAP1_EN The DAP controls the DAC of ADCX/Y.
26	/	/	/
25	R/W	0x0	ADC_DRC1_EN ADC DRC1 enable control 0:Disable 1:Enable
24	R/W	0x0	ADC_HPF1_EN ADC HPF1 enable control 0:Disable

			1:Enable
23:0	/	/	/

#### 5.4.6.14. DAC DRC High HPF Coef Register (Default Value: 0x0000\_00FF)

Offset: 0x0100			Register Name: AC_DAC_DRC_HHPFC
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0xFF	HPF coefficient setting and the data is 3.24 format.

#### 5.4.6.15. DAC DRC Low HPF Coef Register (Default Value: 0x0000\_FAC1)

Offset: 0x0104			Register Name: AC_DAC_DRC_LHPFC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFAC1	HPF coefficient setting and the data is 3.24 format.

#### 5.4.6.16. DAC DRC Control Register (Default Value: 0x0000\_0080)

Offset: 0x0108			Register Name: AC_DAC_DRC_CTRL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R	0x0	DRC delay buffer data output state when drc delay function is enabled and the drc funciton is disabled. After disabled drc function and this bit goes to 0, the user should write the drc delay function bit to 0. 0: Not completed 1: Completed
14:10	/	/	/
13:8	R/W	0x0	Signal delay time setting 6'h00 : (8x1)fs 6'h01 : (8x2)fs 6'h02 : (8x3)fs ----- 6'h2e : (8*47)fs 6'h2f : (8*48)fs 6'h30 -- 6'h3f : (8*48)fs Delay time = 8*(n+1)fs, n<6'h30; When the delay function is disabled, the signal delay time is unused.
7	R/W	0x1	The delay buffer use or not when the drc is disabled and the drc buffer data output completely. 0: Don't use the buffer

			1: Use the buffer
6	R/W	0x0	DRC gain max limit enable 0: Disable 1: Enable
5	R/W	0x0	DRC gain min limit enable When this function is enabled, it will overwrite the noise detect 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 selects Peak filter, the RMS parameter is unused. (AC_DRC_LRMASHAT/AC_DRC_LRMSLAT/AC_DRC_LRMASHAT/AC_DRC_LRMSLAT) When signal function selects RMS filter, the Peak filter parameter is unused.(AC_DRC_LPFHAT/AC_DRC_LPFLAT/AC_DRC_RPFHAT/AC_DRC_RPFLAT /AC_DRC_LPFHRT/AC_DRC_LPFLRT/AC_DRC_RPFHRT/AC_DRC_RPFLRT)
2	R/W	0x0	Delay function enable 0: Disable 1: Enable When the bit is disabled, the signal delay time is unused.
1	R/W	0x0	DRC LT enable 0: Disable 1: Enable When the bit is disabled, KI and OPL parameter is unused.
0	R/W	0x0	DRC ET enable 0: Disable 1: Enable When the bit is disabled, Ke and OPE parameter is unused.

#### 5.4.6.17. DAC DRC Left Peak Filter High Attack Time Coef Register (Default Value: 0x0000\_000B)

Offset: 0x010C			Register Name: AC_DAC_DRC_LPFHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x000B	The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/ta). The format is 3.24.( The default value is 1ms)

#### 5.4.6.18. DAC DRC Left Peak Filter Low Attack Time Coef Register (Default Value: 0x0000\_77BF)

Offset: 0x0110			Register Name: AC_DAC_DRC_LPFLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x77BF	The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/ta). The format is 3.24. (The default value is 1ms)

#### 5.4.6.19. DAC DRC Right Peak Filter High Attack Time Coef Register (Default Value: 0x0000\_000B)

Offset: 0x0114			Register Name: AC_DAC_DRC_RPFHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x000B	The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/ta). The format is 3.24. (The default value is 1ms)

#### 5.4.6.20. DAC DRC Left Peak Filter Low Attack Time Coef Register (Default Value: 0x0000\_77BF)

Offset: 0x0118			Register Name: AC_DAC_DRC_RPFLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x77BF	The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/ta). The format is 3.24. (The default value is 1ms)

#### 5.4.6.21. DAC DRC Left Peak Filter High Release Time Coef Register (Default Value: 0x0000\_00FF)

Offset: 0x011C			Register Name: AC_DAC_DRC_LPFHRT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x00FF	The left peak filter release time parameter setting, which is determined by the equation that RT = exp(-2.2Ts/tr). The format is 3.24. (The default value is 100ms)

#### 5.4.6.22. DAC DRC Left Peak Filter Low Release Time Coef Register (Default Value: 0x0000\_E1F8)

Offset: 0x0120			Register Name: AC_DAC_DRC_LPFLRT
Bit	Read/Write	Default/Hex	Description

31:16	/	/	/
15:0	R/W	0xE1F8	The left peak filter release time parameter setting, which is determined by the equation that $RT = \exp(-2.2Ts/tr)$ . The format is 3.24. (The default value is 100ms)

#### 5.4.6.23. DAC DRC Right Peak Filter High Release Time Coef Register (Default Value: 0x0000\_00FF)

Offset: 0x0124			Register Name: AC_DAC_DRC_RPFHRT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x00FF	The left peak filter attack time parameter setting, which is determined by the equation that $RT = \exp(-2.2Ts/tr)$ . The format is 3.24. (The default value is 100ms)

#### 5.4.6.24. DAC DRC Right Peak Filter Low Release Time Coef Register(Default Value: 0x0000\_E1F8)

Offset: 0x0128			Register Name: AC_DAC_DRC_RPFLRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
10:0	R/W	0xE1F8	The left peak filter release time parameter setting, which is determined by the equation that $AT = \exp(-2.2Ts/tr)$ . The format is 3.24. (The default value is 100ms)

#### 5.4.6.25. DAC DRC Left RMS Filter High Coef Register(Default Value: 0x0000\_0001)

Offset: 0x012C			Register Name: AC_DAC_DRC_LRMSHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0001	The left RMS filter average time parameter setting, which is determined by the equation that $AT = 1 - \exp(-2.2Ts/tav)$ . The format is 3.24. (The default value is 10ms)

#### 5.4.6.26. DAC DRC Left RMS Filter Low Coef Register(Default Value: 0x0000\_2BAF)

Offset: 0x0130			Register Name: AC_DAC_DRC_LRMSLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
10:0	R/W	0x2BAF	The left RMS filter average time parameter setting, which is determined by the equation that $AT = 1 - \exp(-2.2Ts/tav)$ . The format is 3.24. (The default value is 10ms)

#### 5.4.6.27. DAC DRC Right RMS Filter High Coef Register(Default Value: 0x0000\_0001)

Offset: 0x0134			Register Name: AC_DAC_DRC_RRMSHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0001	The right RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/tav). The format is 3.24. (The default value is 10ms)

#### 5.4.6.28. DAC DRC Right RMS Filter Low Coef Register(Default Value: 0x0000\_2BAF)

Offset: 0x0138			Register Name: AC_DAC_DRC_RRMSLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2BAF	The right RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/tav). The format is 3.24. (The default value is 10ms)

#### 5.4.6.29. DAC DRC Compressor Threshold High Setting Register(Default Value: 0x0000\_06A4)

Offset: 0x013C			Register Name: AC_DAC_DRC_HCT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x06A4	The compressor threshold setting, which is set by the equation that CTin = -CT/6.0206. The format is 8.24 (The default value is -40dB)

#### 5.4.6.30. DAC DRC Compressor Slope High Setting Register(Default Value: 0x0000\_D3C0)

Offset: 0x0140			Register Name: AC_DAC_DRC_LCT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xD3C0	The compressor threshold setting, which is set by the equation that CTin = -CT/6.0206. The format is 8.24 (The default value is -40dB)

#### 5.4.6.31. DAC DRC Compressor Slope High Setting Register(Default Value: 0x0000\_0080)

Offset: 0x0144			Register Name: AC_DAC_DRC_HKC
Bit	Read/Write	Default/Hex	Description

31:16	/	/	/
15:0	R/W	0x0080	The slope of the compressor, which is determined by the equation that $K_c = 1/R$ , there, R is the ratio of the compressor, which always is integer. The format is 8.24. (The default value is 2 : 1)

#### 5.4.6.32. DAC DRC Compressor Slope Low Setting Register(Default Value: 0x0000\_0000)

Offset: 0x0148			Register Name: AC_DAC_DRC_LKC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0000	The slope of the compressor, which is determined by the equation that $K_c = 1/R$ , there, R is the ratio of the compressor, which always is integer. The format is 8.24. (The default value is 2 : 1)

#### 5.4.6.33. DAC DRC Compressor High Output at Compressor Threshold Register (Default Value: 0x0000\_F95B)

Offset: 0x014C			Register Name: AC_DAC_DRC_HOPC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xF95B	The output of the compressor, which is determined by the equation $-OPC/6.0206$ . The format is 8.24 (The default value is -40dB)

#### 5.4.6.34. DAC DRC Compressor Low Output at Compressor Threshold Register(Default Value: 0x0000\_2C3F)

Offset: 0x0150			Register Name: AC_DAC_DRC_LOPC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2C3F	The output of the compressor, which is determined by the equation $OPC/6.0206$ . The format is 8.24 (The default value is -40dB)

#### 5.4.6.35. DAC DRC Limiter Threshold High Setting Register(Default Value: 0x0000\_01A9)

Offset: 0x0154			Register Name: AC_DAC_DRC_HLT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x01A9	The limiter threshold setting, which is set by the equation that $LTin = -LT/6.0206$ , The format is 8.24. (The default value is -10dB)

#### 5.4.6.36. DAC DRC Limiter Threshold Low Setting Register(Default Value: 0x0000\_34F0)

Offset: 0x0158			Register Name: AC_DAC_DRC_LLT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x34F0	The limiter threshold setting, which is set by the equation that $LTin = -LT/6.0206$ . The format is 8.24. (The default value is -10dB)

#### 5.4.6.37. DAC DRC Limiter Slope High Setting Register(Default Value: 0x0000\_0005)

Offset: 0x015C			Register Name: AC_DAC_DRC_HKI
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0005	The slope of the limiter which is determined by the equation that $KI = 1/R$ , there, R is the ratio of the limiter, which always is interger. The format is 8.24. (The default value is <50 :1>)

#### 5.4.6.38. DAC DRC Limiter Slope Low Setting Register(Default Value: 0x0000\_1EB8)

Offset: 0x0160			Register Name: AC_DAC_DRC_LKI
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x1EB8	The slope of the limiter, which is determined by the equation that $KI = 1/R$ , there, R is the ratio of the limiter, which always is interger. The format is 8.24. (The default value is <50 :1>)

#### 5.4.6.39. DAC DRC Limiter High Output at Limiter Threshold Register(Default Value: 0x0000\_FBD8)

Offset: 0x0164			Register Name: AC_DAC_DRC_HOPL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFBD8	The output of the limiter, which is determined by equation $OPT/6.0206$ . The format is 8.24 (The default value is -25dB)

#### 5.4.6.40. DAC DRC Limiter Low Output at Limiter Threshold Register(Default Value: 0x0000\_FBA7)

Offset: 0x0168			Register Name: AC_DAC_DRC_LOPL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFBA7	The output of the limiter, which is determined by equation $OPT/6.0206$ . The

			format is 8.24 (The default value is -25dB)
--	--	--	---

#### 5.4.6.41. DAC DRC Expander Threshold High Setting Register(Default Value: 0x0000\_0BA0)

Offset: 0x016C			Register Name: AC_DAC_DRC_HET
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0BA0	The expander threshold setting, which is set by the equation that ETin = -ET/6.0206, The format is 8.24. (The default value is -70dB)

#### 5.4.6.42. DAC DRC Expander Threshold Low Setting Register(Default Value: 0x0000\_7291)

Offset: 0x0170			Register Name: AC_DAC_DRC_LET
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x7291	The expander threshold setting, which is set by the equation that ETin = -ET/6.0206, The format is 8.24. (The default value is -70dB)

#### 5.4.6.43. DAC DRC Expander Slope High Setting Register(Default Value: 0x0000\_0500)

Offset: 0x0174			Register Name: AC_DAC_DRC_HKE
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13:0	R/W	0x0500	The slope of the expander, which is determined by the equation that Ke = 1/R, there, R is the ratio of the expander, which always is interger and the ke must larger than 50. The format is 8.24. (The default value is <1:5>)

#### 5.4.6.44. DAC DRC Expander Slope Low Setting Register(Default Value: 0x0000\_0000)

Offset: 0x0178			Register Name: AC_DAC_DRC_LKE
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0000	The slope of the expander, which is determined by the equation that Ke = 1/R, there, R is the ratio of the expander, which always is interger and the ke must larger than 50. The format is 8.24. (The default value is <1:5>)

#### 5.4.6.45. DAC DRC Expander High Output at Expander Threshold Register(Default Value: 0x0000\_F45F)

Offset: 0x017C		Register Name: AC_DAC_DRC_HOPE

Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xF45F	The output of the expander, which is determined by equation OPE/6.0206. The format is 8.24 (The default value is -70dB)

#### 5.4.6.46. DAC DRC Expander Low Output at Expander Threshold Register(Default Value: 0x0000\_8D6E)

Offset: 0x0180			Register Name: AC_DAC_DRC_LOPE
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x8D6E	The output of the expander which determine by equation OPE/6.0206. The format is 8.24 (The default value is -70dB)

#### 5.4.6.47. DAC DRC Linear Slope High Setting Register(Default Value: 0x0000\_0100)

Offset: 0x0184			Register Name: AC_DAC_DRC_HKN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0100	The slope of the linear, which is determined by the equation that $K_n = 1/R$ , there, R is the ratio of the linear, which always is interger . The format is 8.24. (The default value is <1:1>)

#### 5.4.6.48. DAC DRC Linear Slope Low Setting Register(Default Value: 0x0000\_0000)

Offset: 0x0188			Register Name: AC_DAC_DRC_LKN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0000	The slope of the linear, which is determined by the equation that $K_n = 1/R$ , there, R is the ratio of the linear, which always is interger . The format is 8.24. (The default value is <1:1>)

#### 5.4.6.49. DAC DRC Smooth Filter Gain High Attack Time Coef Register(Default Value: 0x0000\_0002)

Offset: 0x018C			Register Name: AC_DAC_DRC_SFHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0002	The smooth filter attack time parameter setting, which is determined by the equation that $AT = 1 - \exp(-2.2Ts/tr)$ . The format is 3.24. (The default value is 5ms)

#### 5.4.6.50. DAC DRC Smooth Filter Gain Low Attack Time Coef Register(Default Value: 0x0000\_5600)

Offset: 0x0190			Register Name: AC_DAC_DRC_SFLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x5600	The smooth filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 5ms)

#### 5.4.6.51. DAC DRC Smooth Filter Gain High Release Time Coef Register(Default Value: 0x0000\_0000)

Offset: 0x0194			Register Name: AC_DAC_DRC_SFHRT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0000	The gain smooth filter release time parameter setting, which is determined by the equation that RT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 200ms)

#### 5.4.6.52. DAC DRC Smooth Filter Gain Low Release Time Coef Register(Default Value: 0x0000\_OF04)

Offset: 0x0198			Register Name: AC_DAC_DRC_SFLRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xOF04	The gain smooth filter release time parameter setting, which is determined by the equation that RT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 200ms)

#### 5.4.6.53. DAC DRC MAX Gain High Setting Register(Default Value: 0x0000\_FE56)

Offset: 0x019C			Register Name: AC_DAC_DRC_MXGHS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFE56	The max gain setting, which is determined by equation MXG/6.0206. The format is 8.24 and must -20dB <MXG< 30dB (The default value is -10dB)

#### 5.4.6.54. DAC DRC MAX Gain Low Setting Register(Default Value: 0x0000\_CB0F)

Offset: 0x01A0			Register Name: AC_DAC_DRC_MXGLS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

15:0	R/W	0xCBOF	The max gain setting, which is determined by equation MXG/6.0206. The format is 8.24 and must -20dB < MXG < 30dB (The default value is -10dB)
------	-----	--------	---

#### 5.4.6.55. DAC DRC MIN Gain High Setting Register(Default Value: 0x0000\_F95B)

Offset: 0x01A4			Register Name: AC_DAC_DRC_MNGHS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xF95B	The min gain setting, which is determined by equation MXG/6.0206. The format is 8.24 and must -60dB ≤ MNG ≤ -40dB (The default value is -40dB)

#### 5.4.6.56. DAC DRC MIN Gain Low Setting Register(Default Value: 0x0000\_2C3F)

Offset: 0x01A8			Register Name: AC_DAC_DRC_MNGLS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2C3F	The min gain setting, which is determined by equation MNG/6.0206. The format is 8.24 and must -60dB ≤ MNG ≤ -40dB (The default value is -40dB)

#### 5.4.6.57. DAC DRC Expander Smooth Time High Coef Register(Default Value: 0x0000\_0000)

Offset: 0x01AC			Register Name: AC_DAC_DRC_EPSHC
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0000	The gain smooth filter release and attack time parameter setting in expander region, which are determined by the equation that RT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 30ms)

#### 5.4.6.58. DAC DRC Expander Smooth Time Low Coef Register(Default Value: 0x0000\_640C)

Offset: 0x01B0			Register Name: AC_DAC_DRC_EPSLC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x640C	The gain smooth filter release and attack time parameter setting in expander region, which are determined by the equation that RT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 30ms)

#### 5.4.6.59. DAC DRC HPF Gain High Coef Register(Default Value: 0x0000\_0100)

Offset: 0x01B8			Register Name: AC_DAC_DRC_HPFHGAIN
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x100	The gain of the hpf coefficient setting which format is 3.24.(gain = 1)

#### 5.4.6.60. DAC DRC HPF Gain Low Coef Register(Default Value: 0x0000\_0000)

Offset: 0x01BC			Register Name: AC_DAC_DRC_HPFLGAIN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0000	The gain of the hpf coefficient setting which format is 3.24.(gain = 1)

#### 5.4.6.61. ADC DRC High HPF Coef Register(Default Value: 0x0000\_00FF)

Offset: 0x0200			Register Name: AC_ADC_DRC_HHPFC
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0xFF	HPF coefficient setting and the data is 3.24 format.

#### 5.4.6.62. ADC DRC Low HPF Coef Register(Default Value: 0x0000\_FAC1)

Offset: 0x0204			Register Name: AC_ADC_DRC_LHPFC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFAC1	HPF coefficient setting and the data is 3.24 format.

#### 5.4.6.63. ADC DRC Control Register(Default Value: 0x0000\_0080)

Offset: 0x0208			Register Name: AC_ADC_DRC_CTRL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R	0x0	DRC delay buffer data output state when drc delay function is enabled and the drc funciton is disabled. After disabled drc function and this bit goes to 0, the user should write the drc delay function bit to 0. 0: Not completed 1: Completed
14:10	/	/	/
13:8	R/W	0x0	Signal delay time setting

			<p>6'h00 : (8x1)fs          6'h01 : (8x2)fs          6'h02 : (8x3)fs</p> <hr/> <p>6'h2e : (8*47)fs          6'h2f : (8*48)fs          6'h30 -- 6'h3f : (8*48)fs          Delay time = 8*(n+1)fs, n&lt;6'h30;          When the delay function is disabled, the signal delay time is unused.</p>
7	R/W	0x1	<p>The delay buffer use or not when the drc is disabled and the drc buffer data output completely.          0: Don't use the buffer          1: Use the buffer</p>
6	R/W	0x0	<p>DRC gain max limit enable          0: Disable          1: Enable</p>
5	R/W	0x0	<p>DRC gain min limit enable. When this fuction is enabled, it will overwrite the noise detect funciton.          0: Disable          1: Enable</p>
4	R/W	0x0	<p>Control the drc to detect noise when ET is enabled          0: Disable          1: Enable</p>
3	R/W	0x0	<p>Signal function select          0: RMS filter          1: Peak filter          When signal function selects Peak filter, the RMS parameter is unused.(AC_DRC_LRMSHAT/AC_DRC_LRMSLAT/AC_DRC_LRMSSHAT/AC_DRC_LRMSLAT)          When signal function selects RMS filter, the Peak filter parameter is unused.(AC_DRC_LPFHAT/AC_DRC_LPFLAT/AC_DRC_RPFHAT/AC_DRC_RPFLAT /AC_DRC_LPFHRT/AC_DRC_LPFLRT/AC_DRC_RPFHRT/AC_DRC_RPFLRT)</p>
2	R/W	0x0	<p>Delay function enable          0: Disable          1: Enable          When the bit is disabled, the signal delay time is unused.</p>
1	R/W	0x0	<p>DRC LT enable          0: Disable          1: Enable          When the bit is disabled, KI and OPL parameter is unused.</p>
0	R/W	0x0	<p>DRC ET enable          0: Disable          1: Enable          When the bit is disabled, Ke and OPE parameter is unused.</p>

**5.4.6.64. ADC DRC Left Peak Filter High Attack Time Coef Register(Default Value: 0x0000\_000B)**

Offset: 0x020C			Register Name: AC_ADC_DRC_LPFHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x000B	The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/ta). The format is 3.24. (The default value is 1ms)

**5.4.6.65. ADC DRC Left Peak Filter Low Attack Time Coef Register(Default Value: 0x0000\_77BF)**

Offset: 0x0210			Register Name: AC_ADC_DRC_LPFLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x77BF	The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/ta). The format is 3.24. (The default value is 1ms)

**5.4.6.66. ADC DRC Right Peak Filter High Attack Time Coef Register(Default Value: 0x0000\_000B)**

Offset: 0x0214			Register Name: AC_ADC_DRC_RPFHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x000B	The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/ta). The format is 3.24. (The default value is 1ms)

**5.4.6.67. ADC DRC Peak Filter Low Attack Time Coef Register(Default Value: 0x0000\_77BF)**

Offset: 0x0218			Register Name: AC_ADC_DRC_RPFLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x77BF	The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/ta). The format is 3.24. (The default value is 1ms)

**5.4.6.68. ADC DRC Left Peak Filter High Release Time Coef Register(Default Value: 0x0000\_00FF)**

Offset: 0x021C			Register Name: AC_ADC_DRC_LPFHRT
Bit	Read/Write	Default/Hex	Description

31:11	/	/	/
10:0	R/W	0x00FF	The left peak filter release time parameter setting, which is determined by the equation that $RT = \exp(-2.2Ts/tr)$ . The format is 3.24. (The default value is 100ms)

#### 5.4.6.69. ADC DRC Left Peak Filter Low Release Time Coef Register(Default Value: 0x0000\_E1F8)

Offset: 0x0220			Register Name: AC_ADC_DRC_LPFLRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xE1F8	The left peak filter release time parameter setting, which is determined by the equation that $RT = \exp(-2.2Ts/tr)$ . The format is 3.24. (The default value is 100ms)

#### 5.4.6.70. ADC DRC Right Peak Filter High Release Time Coef Register(Default Value: 0x0000\_00FF)

Offset: 0x0224			Register Name: AC_ADC_DRC_RPFHRT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x00FF	The left peak filter attack time parameter setting, which is determined by the equation that $RT = \exp(-2.2Ts/tr)$ . The format is 3.24. (The default value is 100ms)

#### 5.4.6.71. ADC DRC Right Peak Filter Low Release Time Coef Register(Default Value: 0x0000\_E1F8)

Offset: 0x0228			Register Name: AC_ADC_DRC_RPFLRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xE1F8	The left peak filter release time parameter setting, which is determined by the equation that $AT = \exp(-2.2Ts/tr)$ . The format is 3.24. (The default value is 100ms)

#### 5.4.6.72. ADC DRC Left RMS Filter High Coef Register(Default Value: 0x0000\_0001)

Offset: 0x022C			Register Name: AC_ADC_DRC_LRMSSHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0001	The left RMS filter average time parameter setting, which is determined by the equation that $AT = 1 - \exp(-2.2Ts/tav)$ . The format is 3.24. (The default value is 10ms)

#### 5.4.6.73. ADC DRC Left RMS Filter Low Coef Register(Default Value: 0x0000\_2BAF)

Offset: 0x0230			Register Name: AC_ADC_DRC_LRMSLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2BAF	The left RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/tav). The format is 3.24. (The default value is 10ms)

#### 5.4.6.74. ADC DRC Right RMS Filter High Coef Register(Default Value: 0x0000\_0001)

Offset: 0x0234			Register Name: AC_ADC_DRC_RRMSHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0001	The right RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/tav). The format is 3.24. (The default value is 10ms)

#### 5.4.6.75. ADC DRC Right RMS Filter Low Coef Register(Default Value: 0x0000\_2BAF)

Offset: 0x0238			Register Name: AC_ADC_DRC_RRMSLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2BAF	The right RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/tav). The format is 3.24. (The default value is 10ms)

#### 5.4.6.76. ADC DRC Compressor Threshold High Setting Register(Default Value: 0x0000\_06A4)

Offset: 0x023C			Register Name: AC_ADC_DRC_HCT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x06A4	The compressor threshold setting, which is set by the equation that CTin = -CT/6.0206. The format is 8.24. (The default value is -40dB)

#### 5.4.6.77. ADC DRC Compressor Slope High Setting Register(Default Value: 0x0000\_D3C0)

Offset: 0x0240	Register Name: AC_ADC_DRC_LCT
----------------	-------------------------------

Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xD3C0	The compressor threshold setting, which is set by the equation that $CTin = -CT/6.0206$ . The format is 8.24. (The default value is -40dB)

#### 5.4.6.78. ADC DRC Compressor Slope High Setting Register(Default Value: 0x0000\_0080)

Offset: 0x0244			Register Name: AC_ADC_DRC_HKC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0080	The slope of the compressor which is determined by the equation that $Kc = 1/R$ , there, R is the ratio of the compressor, which always is interger. The format is 8.24. (The default value is <2 : 1>)

#### 5.4.6.79. ADC DRC Compressor Slope Low Setting Register(Default Value: 0x0000\_0000)

Offset: 0x0248			Register Name: AC_ADC_DRC_LKC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0000	The slope of the compressor, which is determined by the equation that $Kc = 1/R$ , there, R is the ratio of the compressor, which always is interger. The format is 8.24. (The default value is <2 : 1>)

#### 5.4.6.80. ADC DRC Compressor High Output at Compressor Threshold Register(Default Value: 0x0000\_F95B)

Offset: 0x024C			Register Name: AC_ADC_DRC_HOPC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xF95B	The output of the compressor, which is determined by the equation $-OPC/6.0206$ The format is 8.24. (The default value is -40dB)

#### 5.4.6.81. ADC DRC Compressor Low Output at Compressor Threshold Register(Default Value: 0x0000\_2C3F)

Offset: 0x0250			Register Name: AC_ADC_DRC_LOPC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2C3F	The output of the compressor, which is determined by the equation $OPC/6.0206$ The format is 8.24 (The default value is -40dB)

#### 5.4.6.82. ADC DRC Limiter Threshold High Setting Register(Default Value: 0x0000\_01A9)

Offset: 0x0254			Register Name: AC_ADC_DRC_HLT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x01A9	The limiter threshold setting, which is set by the equation that LTin = -LT/6.0206, The format is 8.24. (The default value is -10dB)

#### 5.4.6.83. ADC DRC Limiter Threshold Low Setting Register(Default Value: 0x0000\_34F0)

Offset: 0x0258			Register Name: AC_ADC_DRC_LLT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x34F0	The limiter threshold setting, which is set by the equation that LTin = -LT/6.0206, The format is 8.24. (The default value is -10dB)

#### 5.4.6.84. ADC DRC Limiter Slope High Setting Register(Default Value: 0x0000\_0005)

Offset: 0x025C			Register Name: AC_ADC_DRC_HKI
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13:0	R/W	0x0005	The slope of the limiter, which is determined by the equation that KI = 1/R, there, R is the ratio of the limiter, which always is interger. The format is 8.24. (The default value is <50 :1>)

#### 5.4.6.85. ADC DRC Limiter Slope Low Setting Register(Default Value: 0x0000\_1EB8)

Offset: 0x0260			Register Name: AC_ADC_DRC_LKI
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x1EB8	The slope of the limiter, which is determined by the equation that KI = 1/R, there, R is the ratio of the limiter, which always is interger. The format is 8.24. (The default value is <50 :1>)

#### 5.4.6.86. ADC DRC Limiter High Output at Limiter Threshold Register(Default Value: 0x0000\_FBD8)

Offset: 0x0264			Register Name: AC_ADC_DRC_HOPL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFB8D	The output of the limiter, which is determined by equation OPT/6.0206. The

			format is 8.24. (The default value is -25dB)
--	--	--	--

#### 5.4.6.87. ADC DRC Limiter Low Output at Limiter Threshold Register(Default Value: 0x0000\_FBA7)

Offset: 0x0268			Register Name: AC_ADC_DRC_LOPL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFBA7	The output of the limiter which is determined by equation OPT/6.0206. The format is 8.24. (The default value is -25dB)

#### 5.4.6.88. ADC DRC Expander Threshold High Setting Register(Default Value: 0x0000\_0BA0)

Offset: 0x026C			Register Name: AC_ADC_DRC_HET
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0BA0	The expander threshold setting, which is set by the equation that ETin = -ET/6.0206, The format is 8.24. (The default value is -70dB)

#### 5.4.6.89. ADC DRC Expander Threshold Low Setting Register(Default Value: 0x0000\_7291)

Offset: 0x0270			Register Name: AC_ADC_DRC_LET
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x7291	The expander threshold setting, which is set by the equation that ETin = -ET/6.0206, The format is 8.24. (The default value is -70dB)

#### 5.4.6.90. ADC DRC Expander Slope High Setting Register(Default Value:0x0000\_0500)

Offset: 0x0274			Register Name: AC_ADC_DRC_HKE
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13:0	R/W	0x0500	The slope of the expander, which is determined by the equation that Ke = 1/R, there, R is the ratio of the expander, which always is interger and the ke must larger than 50. The format is 8.24. (The default value is <1:5>)

#### 5.4.6.91. ADC DRC Expander Slope Low Setting Register(Default Value: 0x0000\_0000)

Offset: 0x0278			Register Name: AC_ADC_DRC_LKE
Bit	Read/Write	Default/Hex	Description

31:16	/	/	/
15:0	R/W	0x0000	The slope of the expander, which is determined by the equation that $K_e = 1/R$ , there, R is the ratio of the expander, which always is interger and the ke must larger than 50. The format is 8.24. (The default value is <1:5>)

#### 5.4.6.92. ADC DRC Expander High Output at Expander Threshold Register(Default Value:0x0000\_F45F)

Offset: 0x027C			Register Name: AC_ADC_DRC_HOPE
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xF45F	The output of the expander, which is determined by equation OPE/6.0206. The format is 8.24. (The default value is -70dB)

#### 5.4.6.93. ADC DRC Expander Low Output at Expander Threshold Register(Default Value: 0x0000\_8D6E)

Offset: 0x0280			Register Name: AC_ADC_DRC_LOPE
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x8D6E	The output of the expander, which is determined by equation OPE/6.0206. The format is 8.24. (The default value is -70dB)

#### 5.4.6.94. ADC DRC Linear Slope High Setting Register(Default Value: 0x0000\_0100)

Offset: 0x0284			Register Name: AC_ADC_DRC_HKN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0100	The slope of the linear, which is determined by the equation that $K_n = 1/R$ , there, R is the ratio of the linear, which always is interger . The format is 8.24. (The default value is <1:1>)

#### 5.4.6.95. ADC DRC Linear Slope Low Setting Register(Default Value: 0x0000\_0000)

Offset: 0x0288			Register Name: AC_ADC_DRC_LKN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0000	The slope of the linear, which is determined by the equation that $K_n = 1/R$ , there, R is the ratio of the linear, which always is interger . The format is 8.24. (The default value is <1:1>)

**5.4.6.96. ADC DRC Smooth Filter Gain High Attack Time Coef Register(Default Value: 0x0000\_0002)**

Offset: 0x028C			Register Name: AC_ADC_DRC_SFHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0002	The smooth filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 5ms)

**5.4.6.97. ADC DRC Smooth Filter Gain Low Attack Time Coef Register(Default Value: 0x0000\_5600)**

Offset: 0x0290			Register Name: AC_ADC_DRC_SFLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x5600	The smooth filter attack time parameter setting, which is determined by the equation that AT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 5ms)

**5.4.6.98. ADC DRC Smooth Filter Gain High Release Time Coef Register(Default Value: 0x0000\_0000)**

Offset: 0x0294			Register Name: AC_ADC_DRC_SFVRT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0000	The gain smooth filter release time parameter setting, which is determined by the equation that RT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 200ms)

**5.4.6.99. ADC DRC Smooth Filter Gain Low Release Time Coef Register(Default Value: 0x0000\_OF04)**

Offset: 0x0298			Register Name: AC_ADC_DRC_SFLRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xOF04	The gain smooth filter release time parameter setting, which is determined by the equation that RT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 200ms)

**5.4.6.100. ADC DRC MAX Gain High Setting Register(Default Value: 0x0000\_FE56)**

Offset: 0x029C			Register Name: AC_ADC_DRC_MXGHS
Bit	Read/Write	Default/Hex	Description

31:16	/	/	/
15:0	R/W	0xFE56	The max gain setting, which is determined by equation MXG/6.0206. The format is 8.24 and must -20dB < MXG < 30dB (The default value is -10dB)

#### 5.4.6.101. ADC DRC MAX Gain Low Setting Register(Default Value: 0x0000\_CBOF)

Offset: 0x02A0			Register Name: AC_ADC_DRC_MXGLS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xCBOF	The max gain setting, which is determined by equation MXG/6.0206. The format is 8.24 and must -20dB < MXG < 30dB (The default value is -10dB)

#### 5.4.6.102. ADC DRC MIN Gain High Setting Register(Default Value: 0x0000\_F95B)

Offset: 0x02A4			Register Name: AC_ADC_DRC_MNGHS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xF95B	The min gain setting, which is determined by equation MXG/6.0206. The format is 8.24 and must -60dB ≤ MNG ≤ -40dB (The default value is -40dB)

#### 5.4.6.103. ADC DRC MIN Gain Low Setting Register(Default Value: 0x0000\_2C3F)

Offset: 0x02A8			Register Name: AC_ADC_DRC_MNGLS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2C3F	The min gain setting, which is determined by equation MNG/6.0206. The format is 8.24 and must -60dB ≤ MNG ≤ -40dB (The default value is -40dB)

#### 5.4.6.104. ADC DAP Expander Smooth Time High Coef Register(Default Value: 0x0000\_0000)

Offset: 0x02AC			Register Name: AC_ADC_DRC_EPSHC
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0000	The gain smooth filter release and attack time parameter setting in expander region, which is determined by the equation that RT = 1-exp(-2.2Ts/tr). The format is 3.24. (The default value is 30ms)

#### 5.4.6.105. ADC DRC Expander Smooth Time Low Coef Register(Default Value: 0x0000\_640C)

Offset: 0x02B0			Register Name: AC_ADC_DRC_EPSLC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x640C	The gain smooth filter release and attack time parameter setting in expander region, which is determined by the equation that $RT = 1 - \exp(-2.2Ts/tr)$ . The format is 3.24. (The default value is 30ms)

#### 5.4.6.106. ADC DRC HPF Gain High Coef Register(Default Value: 0x0000\_0100)

Offset: 0x02B8			Register Name: AC_ADC_DRC_HPFHGAIN
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x100	The gain of the hpf coefficient setting, which format is 3.24.(gain = 1)

#### 5.4.6.107. ADC DRC HPF Gain Low Coef Register(Default Value: 0x0000\_0000)

Offset: 0x02BC			Register Name: AC_ADC_DRC_HPFLGAIN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0000	The gain of the hpf coefficient setting, which format is 3.24.(gain = 1)

### 5.4.7. Audio Codec Analog Part Configuration IO

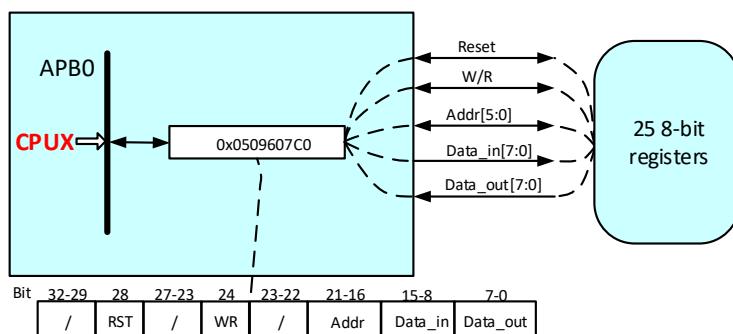
The analog part configuration of Audio Codec is set through the **AC\_PR\_CFG\_REG**, which is in PRCM spec. Configure the codec analog domain circuit by this register.

ADDR[5:0]: AC\_PR Address;

W/R: Write/Read Enable;

WDAT[7:0]: Write Data;

RDAT[7:0]: Read Data;



Offset: 0x0509607C0			Register Name: ADDA_PR_CFG_REG
Bit	Read/Write	Default/Hex	Description

31:29	/	/	/
28	R/W	0x1	ADDA_PR_RST ADDA_PR Reset 0: Assert 1: De-assert
27:25	/	/	/
24	R/W	0x0	ADDA_PR_RW ADDA_PR Read or Write 0: read 1: write
23:22	/	/	/
21:16	R/W	0x0	ADDA_PR_ADDR ADDA_PR Address[5:0]
15:8	R/W	0x0	ADDA_PR_WDAT ADDA_PR Write Data [7:0]
7:0	R/W	0x0	ADDA_PR_RDAT ADDA_PR Read Data[7:0]

#### 5.4.7.1. Lineout Control Register0(Default Value: 0x00)

Offset: 0x05			Register Name: LINEOUT_CTRL0
Bit	Read/Write	Default/Hex	Description
7	R/W	0x0	LINEOUTP(L) Enable 0: Disable 1: Enable
6	R/W	0x0	LINEOUTN Enable 0: Disable 1: Enable
5	R/W	0x0	LINEOUTP(L) Source Select 0: DACL 1: /
4	R/W	0x0	LINEOUTN Source Select 0: / 1: For differential output
3:0	/	/	/

#### 5.4.7.2. Lineout Control Register1(Default Value: 0x00)

Offset: 0x06			Register Name: LINEOUT_CTRL1
Bit	Read/Write	Default/Hex	Description
7:5	/	/	/
4:0	R/W	0x0	LINEOUT Volume Control

			Total 30 level from 0x1F to 0x02 with the volume 0dB to -43.5dB, 1.5dB/step, it indicates mute when the value is 00000 & 00001, 0dB when the value is 0x1f.
--	--	--	---

#### 5.4.7.3. MIC1 Control Register(Default Value: 0x04)

Offset: 0x07			Register Name: MIC1_CTRL
Bit	Read/Write	Default/Hex	Description
7:4	/	/	/
3	R/W	0x0	MIC1AMPEN MIC1 Boost AMP Enable 0: Disable 1: Enable
2:0	R/W	0x4	MIC1BOOST MIC1 Boost AMP Gain Control 0dB when 000, 15dB to 33dB when 001 to 111, 3dB/step, default is 24dB

#### 5.4.7.4. MIC2&MIC3 Control Register(Default Value: 0xB4)

Offset: 0x08			Register Name: MIC2_MIC3_CTRL
Bit	Read/Write	Default/Hex	Description
7	R/W	0x1	MIC3AMPEN MIC3 Boost AMP Enable 0: Disable 1: Enable
6:4	R/W	0x3	MIC3BOOST MIC3 Boost AMP Gain Control 0dB when 000, 15dB to 33dB when 001 to 111, 3dB/step, default is 24dB
3	R/W	0x0	MIC2AMPEN MIC2 Boost AMP Enable 0: Disable 1: Enable
2:0	R/W	0x4	MIC2BOOST MIC2 Boost AMP Gain Control 0dB when 000, 15dB to 33dB when 001 to 111, 3dB/step, default is 24dB

#### 5.4.7.5. Left ADC Mixer Control Register(Default Value: 0x03)

Offset: 0x09			Register Name: L_ADCMIX_SRC
Bit	Read/Write	Default/Hex	Description
7	/	/	/

6:0	R/W	0x3	LADCMIXMUTE Left ADC Mixer Mute Control 0: Mute; 1: On Bit 6: / Bit 4: MIC3 Boost Bit 3: MIC2 Boost Bit 2: MIC1 Boost Bit 1: DACL Bit 0: /
-----	-----	-----	--

#### 5.4.7.6. Right ADC Mixer Control Register(Default Value: 0x00)

Offset: 0x0A			Register Name: R_ADCMIX_SRC
Bit	Read/Write	Default/Hex	Description
7	/	/	/
6:0	R/W	0x0	RADCMIXMUTE Right ADC Mixer Mute Control 0: Mute; 1: On Bit 6: / Bit 4: MIC3 Boost Bit 3: MIC2 Boost Bit 2: MIC1 Boost Bit 1: / Bit 0: DACL

#### 5.4.7.7. X ADC Mixer Control Register(Default Value: 0x00)

Offset: 0x0B			Register Name: X_ADCMIX_SRC
Bit	Read/Write	Default/Hex	Description
7	/	/	/
6:0	R/W	0x0	XADCMIXMUTE X ADC Mixer Mute Control 0: Mute; 1: On Bit 6: / Bit 4: MIC3 Boost Bit 3: MIC2 Boost Bit 2: MIC1 Boost Bit 1: DACL Bit 0: /

#### 5.4.7.8. ADC Control Register(Default Value: 0x03)

Offset: 0x0D			Register Name: ADC_CTRL
Bit	Read/Write	Default/Hex	Description
7	R/W	0x0	ADCREN ADC Right Channel Enable 0: Disable 1: Enable
6	R/W	0x0	ADCLEN ADC Left Channel Enable 0: Disable 1: Enable
5	/	/	/
4	R/W	0x0	ADCXEN ADC X Channel Enable 0: Disable 1: Enable
3	R/W	0x0	Dither Select 0: New dither off 1: New dither on
2:0	R/W	0x3	ADCG ADC Input Gain Control From -4.5dB to 6dB, 1.5dB/step default is 0dB

#### 5.4.7.9. Microphone Bias Control Register(Default Value: 0x20)

Offset: 0x0E			Register Name: MBIAS_CTRL
Bit	Read/Write	Default/Hex	Description
7	R/W	0x0	MMICBIASEN Master Microphone Bias enable 0: Disable 1: Enable
6:5	R/W	0x1	MBIASSEL MMICBIAS voltage level select 00: 1.88V 01: 2.09V 10: 2.33V 11: 2.50V
4:0	/	/	/

#### 5.4.7.10. Analog Performance Tuning Register(Default Value: 0xD6)

Offset: 0x0F			Register Name: ART_REG
Bit	Read/Write	Default/Hex	Description
7	R/W	0x1	MMIC BIAS chopper enable 0: Disable 1: Enable
6:5	R/W	0x2	MMIC BIAS chopper clock select 00: 250kHz 01: 500kHz 10: 1MHz 11: 2MHz
4	R/W	0x1	DITHER ADC dither on/off control 0: dither off 1: dither on
3:2	R/W	0x1	DITHER_CLK_SELECT ADC dither clock select 00: ADC FS * (8/9), about 43kHz when FS=48kHz 01: ADC FS * (16/15), about 51kHz when FS=48kHz 10: ADC FS * (4/3), about 64kHz when FS=48kHz 11: ADC FS * (16/9), about 85kHz when FS=48kHz
1:0	R/W	0x2	BIHE_CTRL BIHE control 00: no BIHE 01: BIHE=7.5 HOSC 10: BIHE=11.5 HOSC 11: BIHE=15.5 HOSC

#### 5.4.7.11. OP BIAS Control Register0(Default Value: 0x55)

Offset: 0x10			Register Name: OP_BIAS_CTRL0
Bit	Read/Write	Default/Hex	Description
7:6	R/W	0x1	OPDRV_OPEAR_CUR. OPDRV/OPEAR output stage current setting
5:4	R/W	0x1	OPADC1_BIAS_CUR OPADC1 Bias Current Select
3:2	R/W	0x1	OPADC2_BIAS_CUR OPADC2 Bias Current Select
1:0	R/W	0x1	OPAAF_BIAS_CUR OPAAF in ADC Bias Current Select

#### 5.4.7.12. OP BIAS Control Register1(Default Value: 0x55)

Offset: 0x11			Register Name: OP_BIAS_CTRL1
Bit	Read/Write	Default/Hex	Description
7:6	R/W	0x1	OPMIC_BIAS_CUR OPMIC Bias Current Control
5:4	R/W	0x1	OPVR_BIAS_CUR. OPVR Bias Current Control
3:2	R/W	0x1	OPDAC_BIAS_CUR. OPDAC Bias Current Control
1:0	R/W	0x1	OPMIX_BIAS_CUR. OPMIX/OPLPF/OPDRV/OPEAR Bias Current Control

#### 5.4.7.13. ZERO Cross &USB Bias Control Register(Default Value: 0x02)

Offset: 0x12			Register Name: ZC_VOL_CTRL
Bit	Read/Write	Default/Hex	Description
7	R/W	0x0	Function enable for master volume change at zero cross over 0: Disable 1: Enable
6	R/W	0x0	Timeout control for master volume change at zero cross over 0: 32ms 1: 64ms
5:3	/	/	/
2:0	R/W	0x2	USB_BIAS_CUR. USB bias current tuning From 23uA to 30uA, Default is 25uA

#### 5.4.7.14. Bias Calibration Control Register(Default Value: 0x00)

Offset: 0x15			Register Name: BIAS_CAL_CTRL
Bit	Read/Write	Default/Hex	Description
7	/	/	/
6	R/W	0x0	CURRENT_TEST_SELECT Internal Bias Current sink test enable(from LINEOUTN pin) 0:Normal 1: for Debug
5:0	/	/	/

# Contents

Chapter 6 Interfaces.....	312
6.1. SD/MMC Host Controller(SMHC) .....	312
6.1.1. Overview .....	312
6.1.2. Block Diagram .....	312
6.1.3. Operations and Functional Descriptions .....	313
6.1.4. Programming Guidelines.....	316
6.1.5. Register List .....	320
6.1.6. Register Description .....	321
6.2. TWI.....	346
6.2.1. Overview .....	346
6.2.2. Block Diagram .....	346
6.2.3. Operations and Functional Descriptions .....	347
6.2.4. Programming Guidelines.....	349
6.2.5. Register List .....	353
6.2.6. Register Description .....	354
6.3. UART.....	365
6.3.1. Overview .....	365
6.3.2. Block Diagram .....	365
6.3.3. Operations and Functional Descriptions .....	366
6.3.4. Programming Guidelines.....	371
6.3.5. Register List .....	373
6.3.6. Register Description .....	373
6.4. SPI.....	391
6.4.1. Overview .....	391
6.4.2. Block Diagram .....	391
6.4.3. Operations and Functional Descriptions .....	392
6.4.4. Programming Guidelines.....	397
6.4.5. Register List .....	401
6.4.6. Register Description .....	402
6.5. USB2.0 OTG .....	417
6.5.1. Overview .....	417
6.5.2. Block Diagram .....	417
6.5.3. Operations and Functional Descriptions .....	418
6.6. Port Controller.....	419
6.6.1. Overview .....	419
6.6.2. Block Diagram .....	419
6.6.3. Operations and Functional Descriptions .....	420
6.6.4. Register List .....	423
6.6.5. Register Description .....	423
6.7. GPADC .....	471

6.7.1. Overview .....	471
6.7.2. Block Diagram .....	471
6.7.3. Operations and Functional Descriptions .....	472
6.7.4. Programming Guidelines.....	473
6.7.5. Register List .....	473
6.7.6. Register Description .....	474
6.8. LRADC.....	483
6.8.1. Overview .....	483
6.8.2. Block Diagram .....	483
6.8.3. Operations and Functional Descriptions .....	483
6.8.4. Programming Guidelines.....	485
6.8.5. Register List .....	485
6.8.6. Register Description .....	485
6.9. PWM .....	489
6.9.1. Overview .....	489
6.9.2. Block Diagram .....	490
6.9.3. Operations and Functional Descriptions .....	490
6.9.4. Register List .....	497
6.9.5. Register Description .....	498
6.10. LEDC .....	516
6.10.1. Overview .....	516
6.10.2. Block Diagram .....	516
6.10.3. Operations and Functional Descriptions .....	516
6.10.4. Programming Guidelines.....	520
6.10.5. Register List .....	524
6.10.6. Register Description .....	524

## Figures

Figure 6- 1. SMHC Block Diagram .....	313
Figure 6- 2. IDMAC Descriptor Structure Diagram.....	314
Figure 6- 3. TWI Block Diagram.....	346
Figure 6- 4. TWI Driver Write Packet Transmission .....	348
Figure 6- 5. TWI Driver Read Packet Transmission .....	349
Figure 6- 6. TWI Initial Flow .....	350
Figure 6- 7. TWI Engine Write Flow .....	351
Figure 6- 8. TWI Engine Read Flow .....	352
Figure 6- 9. TWI Driver Packet Transmission Flow.....	353
Figure 6- 10. UART Block Diagram .....	366
Figure 6- 11. UART Application Diagram.....	367
Figure 6- 12. UART Serial Data Format .....	367
Figure 6- 13. RTS/CTS Autoflow Control Timing .....	367
Figure 6- 14. Serial IrDA Data Format .....	368
Figure 6- 15. RS-485 Timing .....	368
Figure 6- 16. UART DRQ Flow Chart.....	371
Figure 6- 17. UART IRQ Flow Chart .....	372
Figure 6- 18. SPI Block Diagram .....	391
Figure 6- 19. SPI Application Block Diagram .....	393
Figure 6- 20. SPI Phase 0 Timing Diagram.....	394
Figure 6- 21. SPI Phase 1 Timing Diagram.....	394
Figure 6- 22. SPI 3-Wire Mode .....	395
Figure 6- 23. SPI Dual-Input/Dual-Output Mode .....	396
Figure 6- 24. SPI Dual I/O Mode.....	396
Figure 6- 25. SPI Quad-Input/Quad-Output Mode .....	397
Figure 6- 26. SPI Write/Read Data in CPU Mode .....	398
Figure 6- 27. SPI Write/Read Data in DMA Mode.....	399
Figure 6- 28. USB2.0 OTG Controller Block Diagram .....	418
Figure 6- 29. USB2.0 OTG Clock and Reset Description .....	418
Figure 6- 30. Port Controller Block Diagram .....	419
Figure 6- 31. Pull up/down Logic.....	421
Figure 6- 32. IO Buffer Strength Diagram .....	421
Figure 6- 33. GPADC Block Diagram.....	471
Figure 6- 34. GPADC Clock and Timing Requirement.....	473
Figure 6- 35. GPADC Initial Process.....	473
Figure 6- 36. LRADC Block Diagram.....	483
Figure 6- 37. LRADC Interrupt .....	484
Figure 6- 38. LRADC Initial Process .....	485
Figure 6- 39. PWM Block Diagram .....	490
Figure 6- 40. PWM01 Clock Controller Diagram .....	491

Figure 6- 41. PWM01 Output Logic Module Diagram.....	492
Figure 6- 42. PWM0 High Level Active State.....	492
Figure 6- 43. PWM0 Low Level Active State .....	493
Figure 6- 44. PWM0 Output Waveform in Pulse Mode and Cycle Mode .....	493
Figure 6- 45. PWM01 Complementary Pair Output.....	495
Figure 6- 46. Dead-time Output Waveform .....	495
Figure 6- 47. PWM01 Capture Logic Module Diagram .....	496
Figure 6- 48. PWM0 Channel Capture Timing.....	497
Figure 6- 49. LEDC Block Diagram .....	516
Figure 6- 50. LEDC Package Output Timing Diagram .....	517
Figure 6- 51. LEDC 1-frame Output Timing Diagram.....	517
Figure 6- 52. LEDC Input Data Structure .....	517
Figure 6- 53. LEDC Typical Circuit.....	518
Figure 6- 54. LEDC Data Input Code .....	518
Figure 6- 55. LEDC Data Transfer Mode .....	519
Figure 6- 56. LEDC Normal Configuration Process .....	521
Figure 6- 57. LEDC Timeout Abnormal Processing Flow .....	522
Figure 6- 58. FIFO Overflow Abnormal Processing Flow.....	523

## Tables

Table 6- 1. SMHC External Signals.....	313
Table 6- 2. SMHC Clock Sources.....	313
Table 6- 3. TWI External Signals .....	347
Table 6- 4. TWI Clock Sources .....	347
Table 6- 5. UART External Signals.....	366
Table 6- 6. UART Clock Sources.....	367
Table 6- 7. UART Mode Baud and Error Rates .....	369
Table 6- 8. IrDA Mode Baud and Error Rates .....	369
Table 6- 9. RS485 Mode Baud and Error Rates .....	369
Table 6- 10. SPI External Signals.....	392
Table 6- 11. SPI Clock Sources.....	393
Table 6- 12. SPI Transmit Format.....	394
Table 6- 13. SPI Sample Mode and Run Clock.....	400
Table 6- 14. USB2.0 OTG External Signals.....	418
Table 6- 15. Multi-function Port Table .....	420
Table 6- 16. Port Function.....	420
Table 6- 17. GPADC External Signals .....	472
Table 6- 18. GPADC Clock Sources .....	472
Table 6- 19. LRADC External Signals.....	483
Table 6- 20. LRADC Clock Sources .....	484
Table 6- 21. PWM External Signals .....	490
Table 6- 22. LEDC External Signals .....	517
Table 6- 23. LEDC Clock Sources .....	517

# Chapter 6 Interfaces

## 6.1. SD/MMC Host Controller(SMHC)

### 6.1.1. Overview

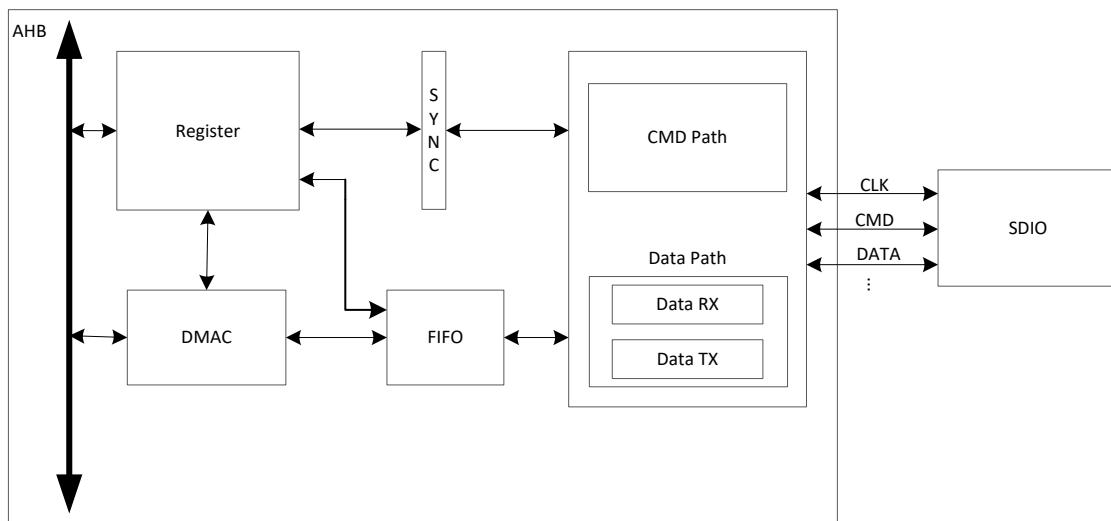
The SD-MMC Host Controller(SMHC) supports various extended devices based on the secure digital input/output(SDIO) protocol.

The SMHC has the following features:

- SMHC1 supports SDIO(Version1.1 to 3.0),4-bit bus width
  - SDR mode 50MHz@3.3V IO pad
  - SDR mode 150MHz@1.8V IO pad
  - DDR mode 50MHz@1.8V IO pad
- Hardware CRC generation and error detection
- Programmable baud rate
- Host pull-up control
- SDIO interrupts in 1-bit and 4-bit modes
- Block size of 1 to 65535 bytes
- Descriptor-based internal DMA controller
- Internal 1KB FIFO for data transfer

### 6.1.2. Block Diagram

Figure 6-1 shows a block diagram of the SMHC.



**Figure 6- 1. SMHC Block Diagram**

DMAC:Direct memory access controller

Register: SMHC FIFO register(0x200)

CMD Path: internal command path

Data Path: internal data path. Write data through Data TX, read data through Data RX.

### 6.1.3. Operations and Functional Descriptions

#### 6.1.3.1. External Signals

Table 6-1 describes the external signals of SMHC.

**Table 6- 1. SMHC External Signals**

Port Name	Width	Type	Description
SDC1-CLK	1	O	Clock output for SDIO Wi-Fi
SDC1-CMD	1	I/O,OD	CMD line for SDIO Wi-Fi
SDC1-D[i] (i=0~3)	4	I/O	Data line for SDIO Wi-Fi

#### 6.1.3.2. Clock Sources

Each SMHC gets three different clocks. User can select one of them to make SMHC clock source. Table 6-2 describes the clock sources of SMHC. Users can see CCU in chapter 3.3 for clock setting, configuration and gating information.

**Table 6- 2. SMHC Clock Sources**

Clock Sources	Description
OSC24M	24MHz Crystal
PLL_PERIO(2X)	Peripheral Clock, the default value is 1.2GHz

PLL_PERI1(2X)	Peripheral Clock, the default value is 1.2GHz
---------------	---

### 6.1.3.3. Timing Diagram

Please refer to relative specifications:

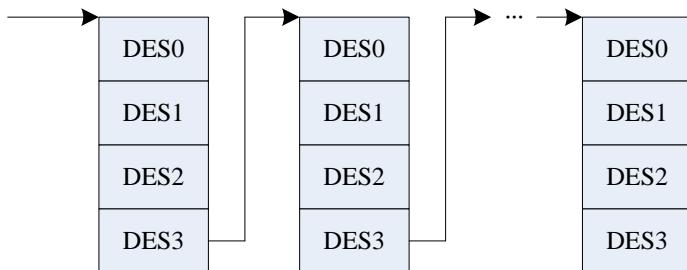
- Physical Layer Specification Ver3.00 Final
- SDIO Specification Ver2.00

### 6.1.3.4. Internal DMA Controller Description

SMHC has an internal DMA controller (IDMAC) to transfer data between host memory and SMHC port. With a descriptor, IDMAC can efficiently move data from source to destination by automatically loading next DMA transfer arguments, which need less CPU intervention. Before transfer data in IDMAC, host driver should construct a descriptor list, configure arguments of every DMA transfer, then launch the descriptor and start the DMA. IDMAC has an interrupt controller, when enabled, it can interrupt the HOST CPU in situations such as data transmission completed or some errors happened.

#### 6.1.3.4.1. IDMAC Descriptor Structure

The IDMAC uses a descriptor with a chain structure, and each descriptor points to a unique buffer and the next descriptor.



**Figure 6- 2. IDMAC Descriptor Structure Diagram**

This figure illustrates the internal formats of a descriptor. The descriptor address must be aligned to the bus width used for 32-bit buses. Each descriptor contains 16 bytes of control and status information.

DES0 is a notation used to denote the [31:0] bits, DES1 to denote [63:32] bits, DES2 to denote [95:64] bits, and DES3 to denote [127:96] bits in a descriptor.

#### 6.1.3.4.2. DES0 Definition

Bits	Name	Descriptor
31	HOLD	DES_own_flag When set, this bit indicates that the descriptor is owned by the IDMAC.

		When this bit is reset, it indicates that the descriptor is owned by the host. This bit is cleared when transfer is over.
30	ERROR	ERR_FLAG When some error happened in transfer, this bit will be set.
29:5	/	/
4	Chain Flag	CHAIM_MOD When set, this bit indicates that the second address in descriptor is the next descriptor address. Must be set 1.
3	First DES Flag	FIRST_FLAG When set, this bit indicates that this descriptor contains the first buffer of data. Must be set to 1 in first DES.
2	Last DES Flag	LAST_FLAG When set, this bit indicates that the buffers pointed to this descriptor are the last data buffer.
1	Disable Interrupt on completion	CUR_TXRX_OVER_INT_DIS When set, this bit will prevent the setting of the TX/RX interrupt bit of the IDMAC status register for data that ends in the buffer pointed to this descriptor.
0	/	/

#### 6.1.3.4.3. DES1 Definition

Bits	Name	Descriptor
31:16	/	/
15:0	Buffer size	BUFF_SIZE These bits indicate the data buffer byte size, which must be a multiple of 4 bytes. If this field is 0, the DMA ignores this buffer and proceeds to the next descriptor.

#### 6.1.3.4.4. DES2 Definition

Bits	Name	Descriptor
31:0	Buffer address pointer	BUFF_ADDR These bits indicate the physical address of data buffer. The IDMAC ignores DES2[1:0], corresponding to the bus width of 32.

#### 6.1.3.4.5. DES3 Definition

Bits	Name	Descriptor
31:0	Next descriptor address	NEXT_DESP_ADDR

		These bits indicate the pointer to the physical memory where the next descriptor is present.
--	--	--

### 6.1.3.5. Calibrate Delay Chain

The sample clock delay chain and Data Strobe delay chain are used to generate delay to make proper timing between internal card clock signals and data signals. Each delay chain is made up with 64 delay cells. The delay time of one delay cell can be estimated through delay chain calibration.

The steps to calibrate delay chain are as follows.

Step1: Enable SMHC. In order to calibrate delay chain by operation registers in SMHC, SMHC must be enabled through **SMHC Bus Gating Reset Register**.

Step2: Configure a proper clock for SMHC. Calibration delay chain is based on the clock for SMHC from Clock Control Unit(CCU). Calibration delay chain is an internal function in SMHC and does not need device. So, it is unnecessary to open clock signal for device. The recommended clock frequency is 200MHz.

Step3: Set proper initial delay value. Writing 0xA0 to **delay control register** and setting initial delay value 0x20 to **Delay chain**(bit[5:0]). Then write 0x0 to **delay control register** to clear the value.

Step4: Write 0x8000 to **delay control register** to start calibrate delay chain.

Step5: Wait until the flag(bit14 in **delay control register**) of calibration done is set. The number of delay cells is shown at bit[14:8] in **delay control register**. The delay time generated by these delay cells is equal to the cycle of SMHC's clock nearly. This value is the result of calibration.

Step6: Calculate the delay time of one delay cell according to the cycle of SMHC's clock and the result of calibration.

### 6.1.4. Programming Guidelines

#### 6.1.4.1. Initialization

Before data and command are exchanged between a card and the SMHC, the SMHC need to be initialized .The SMHC is initialized as follows.

Step1: Configure GPIO register as SMHC function by Port Controller module; reset clock by writing 1 to **SMHC\_BGR\_REG**[SMHC1\_RST], open clock gating by writing 1 to **SMHC\_BGR\_REG**[SMHC1\_GATING]; select clock sources and set division factor by configuring the **SMHC1\_CLK\_REG** register.

Step2: Configure **SMHC\_CTRL** to enable total interrupt; configure **SMHC\_INTMASK** to 0xFFCE to enable normal interrupt and error abnormal interrupt, and register interrupt function.

Step3: Configure **SMHC\_CLKDIV** to open clock for device; configure **SMHC\_CMD** as change clock command(for example 0x80202000); send update clock command to deliver clock to device.

Step4: Configure **SMHC\_CMDARG**, configure **SMHC\_CMD** to set response type, response length, etc, then command

can send. According to initial process in the protocol, you can finish SMHC initializing by sending corresponding command one by one.

#### 6.1.4.2. Writing a Single Data Block

To Write a single data block, perform the following steps:

Step1: Write 0x1 to **SMHC\_CTRL[DMA\_RST]** to reset internal DMA controller; write 0x82 to **SMHC\_IDMAC** to enable IDMAC interrupt, configure AHB master burst transfers; configure **SMHC\_IDIE** to enable transfer interrupt,receive interrupt, and abnormal interrupt.

Step2: Configure **SMHC\_FIFOTH** to determine burst size, TX/RX trigger level. For example, if **SMHC\_FIFOTH** is configured as 0x300F00F0, which indicates that Burst size is 16, TX\_TL is 15,RX\_TL is 240. Configure **SMHC\_DLBA** to determine the start address of DMA descriptor.

Step3: If writing 1 data block to the sector 1, then **SMHC\_BYCNT[BYTE\_CNT]** need be set to 0x200, the descriptor is set based on data size; set the data sector address of CMD24(Single Data Block Write) to 0x1, write 0x80002758 to **SMHC\_CMD**, send CMD24 command to write data to device.

Step4: Check whether **SMHC\_RINTSTS[CC]** is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.

Step5: Check whether **SMHC\_IDST\_REG[TX\_INT]** is 1. If yes, writing DMA data transfer is completed, then write 0x337 to **SMHC\_IDST\_REG** to clear interrupt flag; if no, continue to wait until timeout, then exit process.

Step6: Check whether **SMHC\_RINTSTS[DTC]** is 1. If yes, data transfer is completed and CMD24 writing operation is completed. If no, that is, abnormality exists. Read **SMHC\_RINTSTS,SMHC\_STATUS** to query existing abnormality.

Step7: Send CMD13 command to query whether device writing operation is completed and whether return to idle status. For example, device RCA is 0x1234, first set **SMHC\_CMDARG** to 0x12340000, write 0x8000014D to **SMHC\_CMD**, go to step4 to ensure command transfer completed, then check whether the highest bit of **SMHC\_RESP0**(CMD13 response) is 1. If yes, device is in Idle status, then the next command can be sent. If no, device is in busy status, then continue to send CMD13 to wait device in the idle status until timeout exit.

#### 6.1.4.3. Reading a Single Data Block

To read a single data block, perform the following steps:

Step1: Write 0x1 to **SMHC\_CTRL[DMA\_RST]** to reset internal DMA controller; write **SMHC\_IDMAC** to 0x82 to enable IDMAC interrupt and configure AHB master burst transfers; configure **SMHC\_IDIE** to enable transfer interrupt,receive interrupt, and abnormal interrupt.

Step2: Configure **SMHC\_FIFOTH** to determine burst size, TX/RX trigger level. For example, if **SMHC\_FIFOTH** is configured as 0x300F00F0, which indicates that Burst size is 16, TX\_TL is 15,RX\_TL is 240. Configure **SMHC\_DLBA** to determine the start address of DMA descriptor.

Step3: If reading 1 data block from the sector 1, then **SMHC\_BYCNT[BYTE\_CNT]** need be set to 0x200, the descriptor is set based on data size; set the data sector address of CMD17(Single Data Block Read) to 0x1, write 0x80002351 to **SMHC\_CMD** , send CMD17 command to read data from device to DRAM/SRAM.

Step4: Check whether **SMHC\_RINTSTS[CC]** is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.

Step5: Check whether [SMHC\\_IDST\\_REG\[RX\\_INT\]](#) is 1. If yes, writing DMA data transfer is completed, then write 0x337 to [SMHC\\_IDST\\_REG](#) to clear interrupt flag; if no, continue to wait until timeout, then exit process.

Step6: Check whether [SMHC\\_RINTSTS\[DTC\]](#) is 1. If yes, data transfer is completed and CMD17 reading operation is completed. If no, that is, abnormality exists. Read [SMHC\\_RINTSTS](#),[SMHC\\_STATUS](#) to query existing abnormality.

#### 6.1.4.4. Writing Open-ended Multiple Data Blocks(CMD25+Auto CMD12)

To write open-ended multiple data blocks, perform the following steps:

Step1: Write 0x1 to [SMHC\\_CTRL\[DMA\\_RST\]](#) to reset internal DMA controller; write [SMHC\\_IDMAC](#) to 0x82 to enable IDMAC interrupt and configure AHB master burst transfers; configure [SMHC\\_IDIE](#) to enable transfer interrupt, receive interrupt, and abnormal interrupt.

Step2: Configure [SMHC\\_FIFOTH](#) to determine burst size, TX/RX trigger level. For example, if [SMHC\\_FIFOTH](#) is configured as 0x300F00F0, which indicates that Burst size is 16, TX\_TL is 15, RX\_TL is 240. Configure [SMHC\\_DLBA](#) to determine the start address of DMA descriptor.

Step3: If writing 3 data blocks to the sector 0, then [SMHC\\_BYCNT\[BYTE\\_CNT\]](#) need be set to 0x600, the descriptor is set based on data size; set the data sector address of CMD25(Multiple Data Blocks Write) to 0x0, write 0x80003759 to [SMHC\\_CMD](#), send CMD25 command to write data to device, when data transfer is completed, CMD12 will be sent automatically.

Step4: Check whether [SMHC\\_RINTSTS\[CC\]](#) is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.

Step5: Check whether [SMHC\\_IDST\\_REG\[TX\\_INT\]](#) is 1. If yes, writing DMA data transfer is completed, then write 0x337 to [SMHC\\_IDST\\_REG](#) to clear interrupt flag; if no, continue to wait until timeout, then exit process.

Step6: Check whether [SMHC\\_RINTSTS\[ACD\]](#) and [SMHC\\_RINTSTS\[DTC\]](#) are all 1. If yes, data transfer is completed, CMD12 transfer is completed and CMD25 writing operation is completed. If no, that is, abnormality exists. Read [SMHC\\_RINTSTS](#),[SMHC\\_STATUS](#) to query existing abnormality.

Step7: Send CMD13 command to query whether device writing operation is complete and whether return to idle status. For example, device RCA is 0x1234, first set [SMHC\\_CMDARG](#) to 0x12340000, write 0x8000014D to [SMHC\\_CMD](#), go to step4 to ensure command transfer completed, then check whether the highest bit of [SMHC\\_RESP0](#)(CMD13 response) is 1. If yes, device is in Idle status, then the next command can be sent. If no, device is in busy status, then continue to send CMD13 to wait device in the idle status until timeout exit.

#### 6.1.4.5. Reading Open-ended Multiple Data Blocks(CMD18+Auto CMD12)

To read open-ended multiple data blocks, perform the following steps:

Step1: Write 0x1 to [SMHC\\_CTRL\[DMA\\_RST\]](#) to reset internal DMA controller; write [SMHC\\_IDMAC](#) to 0x82 to enable IDMAC interrupt and configure AHB master burst transfers; configure [SMHC\\_IDIE](#) to enable transfer interrupt, receive interrupt, and abnormal interrupt.

Step2: Configure [SMHC\\_FIFOTH](#) to determine burst size, TX/RX trigger level. For example, if [SMHC\\_FIFOTH](#) is configured as 0x300F00F0, which indicates that Burst size is 16, TX\_TL is 15, RX\_TL is 240. Configure [SMHC\\_DLBA](#) to determine the start address of DMA descriptor.

Step3: If reading 3 data blocks from the sector 0, then [SMHC\\_BYCNT\[BYTE\\_CNT\]](#) need be set to 0x600, the descriptor

is set based on data size; set the data sector address of CMD18(Multiple Data Blocks Read) to 0x0, write 0x80003352 to [SMHC\\_CMD](#), send CMD18 command to read data to device, when data transfer is completed, CMD12 will be sent automatically.

Step4: Check whether [SMHC\\_RINTSTS\[CC\]](#) is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.

Step5: Check whether [SMHC\\_IDST\\_REG\[RX\\_INT\]](#) is 1. If yes, writing DMA data transfer is complete, then write 0x337 to [SMHC\\_IDST\\_REG](#) to clear interrupt flag; if no, continue to wait until timeout, then exit process.

Step6: Check whether [SMHC\\_RINTSTS\[ACD\]](#) and [SMHC\\_RINTSTS\[DTC\]](#) are all 1. If yes, data transfer is completed, CMD12 transfer is completed and CMD18 reading operation is completed. If no, that is, abnormality exists. Read [SMHC\\_RINTSTS](#),[SMHC\\_STATUS](#) to query existing abnormality.

#### 6.1.4.6. Writing Pre-defined Multiple Data Blocks(CMD23+CMD25)

To write pre-defined multiple data blocks, perform the following steps:

Step1: Write 0x1 to [SMHC\\_CTRL\[DMA\\_RST\]](#) to reset internal DMA controller; write [SMHC\\_IDMAC](#) to 0x82 to enable IDMAC interrupt and configure AHB master burst transfers; configure [SMHC\\_IDIE](#) to enable transfer interrupt, receive interrupt, and abnormal interrupt.

Step2: Configure [SMHC\\_FIFOTH](#) to determine burst size, TX/RX trigger level. For example, if [SMHC\\_FIFOTH](#) is configured as 0x300F00F0, which indicates that Burst size is 16, TX\_TL is 15, RX\_TL is 240. Configure [SMHC\\_DLBA](#) to determine the start address of DMA descriptor.

Step3: If writing 3 data blocks, then set [SMHC\\_CMDARG](#) to 0x3 to ensure the block number to be operated, send CMD23 command by writing 0x80000157 to [SMHC\\_CMD](#). Check whether [SMHC\\_RINTSTS\[CC\]](#) is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.

Step4: [SMHC\\_BYCNT\[BYTE\\_CNT\]](#) need be set to 0x600, the descriptor is set based on data size; set the data sector address of CMD25(Multiple Data Blocks Write) to 0x0, write 0x80002759 to [SMHC\\_CMD](#), send CMD25 command to write data to device.

Step5: Check whether [SMHC\\_RINTSTS\[CC\]](#) is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.

Step6: Check whether [SMHC\\_IDST\\_REG\[TX\\_INT\]](#) is 1. If yes, writing DMA data transfer is completed, then write 0x337 to [SMHC\\_IDST\\_REG](#) to clear interrupt flag; if no, continue to wait until timeout, then exit process.

Step7: Check whether [SMHC\\_RINTSTS\[DTC\]](#) is 1. If yes, data transfer is completed and CMD25 writing operation is completed. If no, that is, abnormality exists. Read [SMHC\\_RINTSTS](#),[SMHC\\_STATUS](#) to query existing abnormality.

Step8: Send CMD13 command to query whether device writing operation is completed and whether return to idle status. For example, device RCA is 0x1234, first set [SMHC\\_CMDARG](#) to 0x12340000, write 0x8000014D to [SMHC\\_CMD](#), go to step4 to ensure command transfer completed, then check whether the highest bit of [SMHC\\_RESP0](#)(CMD13 response) is 1. If yes, device is in Idle status, then the next command can be sent. If no, device is in busy status, then continue to send CMD13 to wait device in the idle status until timeout exit.

#### 6.1.4.7. Reading Pre-defined Multiple Data Blocks(CMD23+CMD18)

To read pre-defined multiple data blocks, perform the following steps:

- Step1: Write 0x1 to [SMHC\\_CTRL](#)[DMA\_RST] to reset internal DMA controller; write [SMHC\\_IDMAC](#) to 0x82 to enable IDMAC interrupt and configure AHB master burst transfers; configure [SMHC\\_IDIE](#) to enable transfer interrupt, receive interrupt, and abnormal interrupt.
- Step2: Configure [SMHC\\_FIFOTH](#) to determine burst size, TX/RX trigger level. For example, if [SMHC\\_FIFOTH](#) is configured as 0x300F00F0, which indicates that Burst size is 16, TX\_TL is 15, RX\_TL is 240. Configure [SMHC\\_DLBA](#) to determine the start address of DMA descriptor.
- Step3: If reading 3 data blocks, then set [SMHC\\_CMDARG](#) to 0x3 to ensure the block number to be operated, send CMD23 command by writing 0x80000157 to [SMHC\\_CMD](#). Check whether [SMHC\\_RINTSTS](#)[CC] is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.
- Step4: [SMHC\\_BYCNT](#)[BYTE\_CNT] need be set to 0x600, the descriptor is set based on data size; set the data sector address of CMD18(Multiple Data Blocks Read) to 0x0, write 0x80002352 to [SMHC\\_CMD](#), send CMD18 command to read data from device to DRAM/SRAM.
- Step5: Check whether [SMHC\\_RINTSTS](#)[CC] is 1. If yes, command sends successful; if no, continue to wait until timeout, then exit process.
- Step6: Check whether [SMHC\\_IDST\\_REG](#)[TX\_INT] is 1. If yes, writing DMA data transfer is completed, then write 0x337 to [SMHC\\_IDST\\_REG](#) to clear interrupt flag; if no, continue to wait until timeout, then exit process.
- Step7: Check whether [SMHC\\_RINTSTS](#)[DTC] is 1. If yes, data transfer is completed and CMD18 writing operation is completed. If no, that is, abnormality exists. Read [SMHC\\_RINTSTS](#), [SMHC\\_STATUS](#) to query existing abnormality.

### 6.1.5. Register List

Module Name	Base Address
SMHC1	0x04021000

Register Name	Offset	Description
SMHC_CTRL	0x0000	Control Register
SMHC_CLKDIV	0x0004	Clock Control Register
SMHC_TMOUT	0x0008	Time Out Register
SMHC_CTYPE	0x000C	Bus Width Register
SMHC_BLKSIZ	0x0010	Block Size Register
SMHC_BYTCNT	0x0014	Byte Count Register
SMHC_CMD	0x0018	Command Register
SMHC_CMDARG	0x001C	Command Argument Register
SMHC_RESP0	0x0020	Response 0 Register
SMHC_RESP1	0x0024	Response 1 Register
SMHC_RESP2	0x0028	Response 2 Register
SMHC_RESP3	0x002C	Response 3 Register
SMHC_INTMASK	0x0030	Interrupt Mask Register
SMHC_MINTSTS	0x0034	Masked Interrupt Status Register
SMHC_RINTSTS	0x0038	Raw Interrupt Status Register
SMHC_STATUS	0x003C	Status Register
SMHC_FIFOTH	0x0040	FIFO Water Level Register

SMHC_FUNS	0x0044	FIFO Function Select Register
SMHC_TCBCNT	0x0048	Transferred Byte Count between Controller and Card
SMHC_TBBCNT	0x004C	Transferred Byte Count between Host Memory and Internal FIFO
SMHC_DBGC	0x0050	Current Debug Control Register
SMHC_A12A	0x0058	Auto Command 12 Argument Register
SMHC_NTSR	0x005C	SD New Timing Set Register
SMHC_HWRST	0x0078	Hardware Reset Register
SMHC_IDMAC	0x0080	IDMAC Control Register
SMHC_DLBA	0x0084	Descriptor List Base Address Register
SMHC_IDST	0x0088	IDMAC Status Register
SMHC_IDIE	0x008C	IDMAC Interrupt Enable Register
SMHC_THLD	0x0100	Card Threshold Control Register
EMMC_DDR_SBIT_DET	0x010C	eMMC4.5 DDR Start Bit Detection Control Register
SMHC_RES_CRC	0x0110	Response CRC from Device
SMHC_D7_CRC	0x0114	CRC in Data7 from Device
SMHC_D6_CRC	0x0118	CRC in Data6 from Device
SMHC_D5_CRC	0x011C	CRC in Data5 from Device
SMHC_D4_CRC	0x0120	CRC in Data4 from Device
SMHC_D3_CRC	0x0124	CRC in Data3 from Device
SMHC_D2_CRC	0x0128	CRC in Data2 from Device
SMHC_D1_CRC	0x012C	CRC in Data1 from Device
SMHC_D0_CRC	0x0130	CRC in Data0 from Device
SMHC_CRC_STA	0x0134	CRC Status from Device in Write Operation
SMHC_DRV_DL	0x0140	Drive Delay Control Register
SMHC_SMAP_DL	0x0144	Sample Delay Control Register
SMHC_FIFO	0x0200	Read/Write FIFO

## 6.1.6. Register Description

### 6.1.6.1. SMHC Global Control Register(Default Value: 0x0000\_0100)

Offset: 0x0000			Register Name: SMHC_CTRL
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	FIFO_AC_MOD FIFO Access Mode 0: DMA bus 1: AHB bus
30:13	/	/	/
12	R/W	0x0	TIME_UNIT_CMD Time unit for command line Time unit is used to calculate command line time out value defined in RTO_LMT.

			0: 1 card clock period 1: 256 card clock period
11	R/W	0x0	TIME_UNIT_DAT Time unit for data line Time unit is used to calculate data line time out value defined in DTO_LMT. 0: 1 card clock period 1: 256 card clock period
10	R/W	0x0	DDR_MOD_SEL DDR Mode Select Although eMMC's HS400 speed mode is 8-bit DDR, this field should be cleared when HS400_MD_EN is set. 0: SDR mode 1: DDR mode
9	/	/	/
8	R/W	0x1	CD_DBC_ENB Card Detect (Data[3] status) De-bounce Enable 0: Disable de-bounce 1: Enable de-bounce
7:6	/	/	/
5	R/W	0x0	DMA_ENB DMA Global Enable 0: Disable DMA to transfer data, using AHB bus 1: Enable DMA to transfer data
4	R/W	0x0	INT_ENB Global Interrupt Enable 0: Disable interrupts 1: Enable interrupts
3	/	/	/
2	R/W	0x0	DMA_RST DMA Reset
1	R/W	0x0	FIFO_RST FIFO Reset 0: No change 1: Reset FIFO This bit is auto-cleared after completion of reset operation.
0	R/W	0x0	SOFT_RST Software Reset 0: No change 1: Reset SD/MMC controller This bit is auto-cleared after completion of reset operation.

#### 6.1.6.2. SMHC Clock Control Register(Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: SMHC_CLKDIV
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	MASK_DATA0 0: Do not mask data0 when update clock 1: Mask data0 when update clock
30:18	/	/	/
17	R/W	0x0	CCLK_CTRL Card Clock Output Control 0: Card clock always on 1: Turn off card clock when FSM is in IDLE state
16	R/W	0x0	CCLK_ENB Card Clock Enable 0: Card Clock off 1: Card Clock on
15:8	/	/	/
7:0	R/W	0x0	CCLK_DIV Card Clock Divider n: Source clock is divided by 2*n.(n=0~255) when HS400_MD_EN is set, this field must be cleared.

#### 6.1.6.3. SMHC Timeout Register(Default Value:0xFFFF\_FF40)

Offset: 0x0008			Register Name: SMHC_TMOUT
Bit	Read/Write	Default/Hex	Description
31:8	R/W	0xffffffff	<p>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 <math>N_{AC}</math>. About the <math>N_{AC}</math>, the explanation is as follows:</p> <p>When Host read data, data transmission from the Device starts after the access time delay <math>N_{AC}</math> beginning from the end bit of the read command(ACMD51,CMD8,CMD17,CMD18).</p> <p>When Host read multiple block(CMD18), a next block's data transmission from the Device starts after the access time delay <math>N_{AC}</math> beginning from the end bit of the previous block.</p> <p>When Host write data, the value is no effect.</p>
7:0	R/W	0x40	RTO_LMT Response Timeout Limit

#### 6.1.6.4. 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

#### 6.1.6.5. 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

#### 6.1.6.6. 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; should be integer multiple of Block Size for block transfers.

#### 6.1.6.7. 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_ABТ Boot Abort Setting this bit will terminate the boot operation.
26	R/W	0x0	EXP_BOOT_ACK Expect Boot Acknowledge When software sets this bit along in mandatory boot operation, the controller expects a boot acknowledge start pattern of 0-1-0 from the selected card.
25:24	R/W	0x0	BOOT_MOD Boot Mode 00: Normal command 01: Mandatory Boot operation 10: Alternate Boot operation 11: Reserved
23:22	/	/	/
21	R/W	0x0	PRG_CLK Change Clock 0: Normal command 1: Change Card Clock When this bit is set, controller will change clock domain and clock output. No command will be sent.
20:16	/	/	/
15	R/W	0x0	SEND_INIT_SEQ Send Initialization 0: Normal command sending 1: Send initialization sequence before sending this command.
14	R/W	0x0	STOP_ABТ_CMD Stop Abort Command 0: Normal command sending 1: Send <i>Stop</i> or <i>Abort</i> command to stop current data transfer in progress.(CMD12, CMD52 for writing “I/O Abort” in SDIO CCCR)
13	R/W	0x0	WAIT_PRE_OVER Wait Data Transfer Over 0: Send command at once, do not care of data transferring 1: Wait for data transfer completion before sending current command
12	R/W	0x0	STOP_CMD_FLAG Send Stop CMD Automatically (CMD12) 0: Do not send stop command at end of data transfer 1: Send stop command automatically at end of data transfer If set, the <b>SMHC_RESP1</b> will record the response of auto CMD12.
11	R/W	0x0	TRANS_MODE

			Transfer Mode 0: Block data transfer command 1: Stream data transfer command
10	R/W	0x0	TRANS_DIR Transfer Direction 0: Read operation 1: Write operation
9	R/W	0x0	DATA_TRANS Data Transfer 0: Without data transfer 1: With data transfer
8	R/W	0x0	CHK_RESP_CRC Check Response CRC 0: Do not check response CRC 1: Check response CRC
7	R/W	0x0	LONG_RESP Response Type 0:Short Response (48 bits) 1:Long Response (136 bits)
6	R/W	0x0	RESP_RCV Response Receive 0: Command without response 1: Command with response
5:0	R/W	0x0	CMD_IDX CMD Index Command index value

#### 6.1.6.8. 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

#### 6.1.6.9. 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

#### 6.1.6.10. SMHC Response 1 Register(Default Value: 0x0000\_0000)

Offset: 0x0024			Register Name: SMHC_RESP1
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	CMD_RESP1 Response 1 Bit[63:31] of response

#### 6.1.6.11. 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

#### 6.1.6.12. 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

#### 6.1.6.13. SMHC Interrupt Mask Register(Default Value: 0x0000\_0000)

Offset: 0x0030			Register Name: SMHC_INTMASK
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	CARD_REMOVAL_INT_EN Card Removed Interrupt Enable
30	R/W	0x0	CARD_INSERT_INT_EN Card Inserted Interrupt Enable
29:17	/	/	/
16	R/W	0x0	SDIO_INT_EN SDIO Interrupt Enable
15	R/W	0x0	DEE_INT_EN Data End-bit Error Interrupt Enable
14	R/W	0x0	ACD_INT_EN Auto Command Done Interrupt Enable
13	R/W	0x0	DSE_BC_INT_EN Data Start Error Interrupt Enable

12	R/W	0x0	CB_IW_INT_EN Command Busy and Illegal Write Interrupt Enable
11	R/W	0x0	FU_FO_INT_EN FIFO Underrun/Overflow Interrupt Enable
10	R/W	0x0	DSTO_VSD_INT_EN Data Starvation Timeout/V1.8 Switch Done Interrupt Enable
9	R/W	0x0	DTO_BDS_INT_EN Data Timeout/Boot Data Start Interrupt Enable
8	R/W	0x0	RTO_BACK_INT_EN Response Timeout/Boot ACK Received Interrupt Enable
7	R/W	0x0	DCE_INT_EN Data CRC Error Interrupt Enable
6	R/W	0x0	RCE_INT_EN Response CRC Error Interrupt Enable
5	R/W	0x0	DRR_INT_EN Data Receive Request Interrupt Enable
4	R/W	0x0	DTR_INT_EN Data Transmit Request Interrupt Enable
3	R/W	0x0	DTC_INT_EN Data Transfer Complete Interrupt Enable
2	R/W	0x0	CC_INT_EN Command Complete Interrupt Enable
1	R/W	0x0	RE_INT_EN Response Error Interrupt Enable
0	/	/	/

#### 6.1.6.14. SMHC Masked Interrupt Status Register(Default Value: 0x0000\_0000)

Offset: 0x0034			Register Name: SMHC_MINTSTS
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	M_CARD_REMOVAL_INT Card Removed
30	R/W	0x0	M_CARD_INSERT Card Inserted
29:17	/	/	/
16	R/W	0x0	M_SDIO_INT SDIO Interrupt
15	R/W	0x0	M_DEE_INT Data End-bit Error When set during receiving data, it means that host controller does not receive valid data end bit. When set during transmitting data, it means that host controller does not receive CRC status taken or received CRC status taken is negative.

14	R/W	0x0	M_ACD_INT Auto Command Done When set, it means auto stop command(CMD12) completed.
13	R/W	0x0	M_DSE_BC_INT Data Start Error When set during receiving data, it means that host controller found an error start bit. When set during transmitting data, it means that busy signal is cleared.
12	R/W	0x0	M_CB_IW_INT Command Busy and Illegal Write
11	R/W	0x0	M_FU_FO_INT FIFO Underrun/Overflow
10	R/W	0x0	M_DSTO_VSD_INT Data Starvation Timeout/V1.8 Switch Done
9	R/W	0x0	M.DTO_BDS_INT Data Timeout/Boot Data Start
8	R/W	0x0	M.RTO_BACK_INT Response Timeout/Boot ACK Received
7	R/W	0x0	M.DCE_INT Data CRC Error When set during receiving data, it means that the received data have data CRC error. When set during transmitting data, it means that the received CRC status taken is negative.
6	R/W	0x0	M.RCE_INT Response CRC Error
5	R/W	0x0	M.DRR_INT Data Receive Request When set, it means that there are enough data in FIFO during receiving data.
4	R/W	0x0	M.DTR_INT Data Transmit Request When set, it means that there are enough space in FIFO during transmitting data.
3	R/W	0x0	M.DTC_INT Data Transfer Complete
2	R/W	0x0	M.CC_INT Command Complete
1	R/W	0x0	M.RE_INT Response Error When set, Transmit Bit error or End Bit error or CMD Index error may occurs.
0	/	/	/

#### 6.1.6.15. SMHC Raw Interrupt Status Register(Default Value: 0x0000\_0000)

Offset: 0x0038			Register Name: SMHC_RINTSTS
Bit	Read/Write	Default/Hex	Description
31	R/W1C	0x0	CARD_REMOVAL Card Removed This is write-1-to-clear bits.
30	R/W1C	0x0	CARD_INSERT Card Inserted This is write-1-to-clear bits.
29:17	/	/	/
16	R/W1C	0x0	SDIOI_INT SDIO Interrupt This is write-1-to-clear bits.
15	R/W1C	0x0	DEE Data End-bit Error When set during receiving data, it means that host controller does not receive valid data end bit. When set during transmitting data, it means that host controller does not receive CRC status taken. This is write-1-to-clear bits.
14	R/W1C	0x0	ACD Auto Command Done When set, it means auto stop command(CMD12) completed. This is write-1-to-clear bits.
13	R/W1C	0x0	DSE_BC Data Start Error When set during receiving data, it means that host controller found a error start bit. When set during transmitting data, it means that busy signal is cleared. This is write-1-to-clear bits.
12	R/W1C	0x0	CB_IW Command Busy and Illegal Write This is write-1-to-clear bits.
11	R/W1C	0x0	FU_FO FIFO Underrun/Overflow This is write-1-to-clear bits.
10	R/W1C	0x0	DSTO_VSD Data Starvation Timeout/V1.8 Switch Done This is write-1-to-clear bits.
9	R/W1C	0x0	DTO_BDS Data Timeout/Boot Data Start This is write-1-to-clear bits.
8	R/W1C	0x0	RTO_BACK Response Timeout/Boot ACK Received

			This is write-1-to-clear bits.
7	R/W1C	0x0	<p>DCE Data CRC Error When set during receiving data, it means that the received data have data CRC error. When set during transmitting data, it means that the received CRC status taken is negative. This is write-1-to-clear bits.</p>
6	R/W1C	0x0	<p>RCE Response CRC Error This is write-1-to-clear bits.</p>
5	R/W1C	0x0	<p>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.</p>
4	R/W1C	0x0	<p>DTR Data Transmit Request When set, it means that there are enough space in FIFO during transmitting data. This is write-1-to-clear bits.</p>
3	R/W1C	0x0	<p>DTC Data Transfer Complete This is write-1-to-clear bits.</p>
2	R/W1C	0x0	<p>CC Command Complete When set, it means that current command completes even through error occurs. This is write-1-to-clear bits.</p>
1	R/W1C	0x0	<p>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.</p>
0	/	/	/

#### 6.1.6.16. SMHC Status Register(Default Value: 0x0000\_0006)

Offset: 0x003C			Register Name: SMHC_STATUS
Bit	Read/Write	Default/Hex	Description
31	R	0x0	<p>DMA_REQ DMA Request DMA request signal state</p>
30:26	/	/	/

25:17	R	0x0	FIFO_LEVEL FIFO Level Number of filled locations in FIFO
16:11	R	0x0	RESP_IDX Response Index Index of previous response, including any auto-stop sent by controller
10	R	0x0	FSM_BUSY Data FSM Busy Data transmit or receive state-machine is busy
9	R	0x0	CARD_BUSY Card Data Busy Inverted version of DATA[0] 0: card data not busy 1: card data busy
8	R	0x0	CARD_PRESENT Data[3] Status Level of DATA[3], checks whether card is present 0: card not present 1: card present
7:4	R	0x0	FSM_STA Command FSM States 0000: Idle 0001: Send init sequence 0010: TX CMD start bit 0011: TX CMD TX bit 0100: TX CMD index + argument 0101: TX CMD CRC7 0110: TX CMD end bit 0111: RX response start bit 1000: RX response IRQ response 1001: RX response TX bit 1010: RX response CMD index 1011: RX response data 1100: RX response CRC7 1101: RX response end bit 1110: CMD path wait NCC 1111: Wait; CMD-to-response turnaround
3	R	0x0	FIFO_FULL FIFO Full 1: FIFO full 0: FIFO not full
2	R	0x1	FIFO_EMPTY FIFO Empty 1: FIFO Empty 0: FIFO not Empty

1	R	0x1	FIFO_TX_LEVEL FIFO TX Water Level Flag 0: FIFO didn't reach transmit trigger level 1: FIFO reached transmit trigger level
0	R	0x0	FIFO_RX_LEVEL FIFO RX Water Level Flag 0: FIFO didn't reach receive trigger level 1: FIFO reached receive trigger level

#### 6.1.6.17. SMHC FIFO Water Level Register(Default Value: 0x000F\_0000)

Offset: 0x0040			Register Name: SMHC_FIFOTH
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30:28	R/W	0x0	BSIZE_OF_TRANS Burst Size of Multiple Transaction 000: 1 transfers 001: 4 010: 8 011: 16 Others: Reserved It should be programmed same as DMA controller multiple transaction size. The units for transfers are the DWORD. A single transfer would be signaled based on this value. Value should be sub-multiple of (RX_TL + 1) and (FIFO_DEPTH - TX_TL)  Recommended: MSize = 8, TX_TL = 248, RX_TL = 7 FIFO_DEPTH = 256, FIFO_SIZE = 256 * 32 = 1K
27:24	/	/	/
23:16	R/W	0xF	RX_TL RX Trigger Level 0x0~0xFE: RX Trigger Level is 0~254 0xFF: Reserved FIFO threshold when FIFO request host to receive data from FIFO. When FIFO data level is greater than this value, DMA is request is raised if DMA enabled, or RX interrupt bit is set if interrupt enabled. At the end of packet, if the last transfer is less than this level, the value is ignored and relative request will be raised as usual.  Recommended: 7 (means greater than 7)
15:8	/	/	/
7:0	R/W	0x0	TX_TL

		<p><b>TX Trigger Level</b>      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.</p> <p><b>Recommended:</b>      248(means less than or equal to 248)</p>
--	--	--

#### 6.1.6.18. SMHC Function Select Register(Default Value: 0x0000\_0000)

<b>Offset: 0x0044</b>			<b>Register Name: SMHC_FUNS</b>
<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>
31:3	/	/	/
2	R/W	0x0	<p>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.</p>
1	R/W	0x0	<p>READ_WAIT          Read Wait          0: Clear SDIO read wait          1: Assert SDIO read wait</p>
0	R/W	0x0	<p>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.</p>

#### 6.1.6.19. SMHC Transferred Byte Count Register 0 (Default Value: 0x0000\_0000)

<b>Offset: 0x0048</b>			<b>Register Name: SMHC_TBC0</b>
<b>Bit</b>	<b>Read/Write</b>	<b>Default/Hex</b>	<b>Description</b>

31:0	R	0x0	TBC0 Transferred Count 0 Number of bytes transferred between card and internal FIFO. The register should be accessed in full to avoid read-coherency problems and read only after data transfer completes.
------	---	-----	---

#### 6.1.6.20. SMHC Transferred Byte Count Register 1 (Default Value: 0x0000\_0000)

Offset: 0x004C			Register Name: SMHC_TBC1
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	TBC1 Transferred Count 1 Number of bytes transferred between Host/DMA memory and internal FIFO. The register should be accessed in full to avoid read-coherency problems and read only after data transfer completes.

#### 6.1.6.21. SMHC Auto Command 12 Argument Register (Default Value: 0x0000\_FFFF)

Offset: 0x0058			Register Name: SMHC_A12A
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xffff	SD_A12A. SD_A12A set the argument of command 12 automatically send by controller.

#### 6.1.6.22. SMHC New Timing Set Register (Default Value: 0x8171\_0000)

Offset: 0x005C			Register Name: SMHC_NTSR
Bit	Read/Write	Default/Hex	Description
31	R/W	0x1	MODE_SELEC 0: Old mode of Sample/Output Timing 1: New mode of Sample/Output Timing
30:28	/	/	/
27	R/W	0x0	DAT0_BYPASS Select data0 input asyn or bypass sample logic, it is used to check card busy or not 0: Enable data0 bypass 1: Disable data0 bypass
26:25	/	/	/
24	R/W	0x1	CMD_DAT_RX_PHASE_CLR

			During update clock , command and data rx phase clear 0: Disable 1: Enable
23	/	/	/
22	R/W	0x1	DAT_CRC_STATUS_RX_PHASE_CLR Before receive CRC status, data rx phase clear 0: Disable 1: Enable
21	R/W	0x1	DAT_TRANS_RX_PHASE_CLR Before transfer data , data rx phase clear 0: Disable 1: Enable
20	R/W	0x1	DAT_RECV_RX_PHASE_CLR Before receive data , data rx phase clear 0: Disable 1: Enable
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_SAMPLE_TIMING_PHASE 00: Sample timing phase offset 90° 01: Sample timing phase offset 180° 10: Sample timing phase offset 270° 11: Ignore
7:6	/	/	/
5:4	R/W	0x0	CMD_SAMPLE_TIMING_PHASE 00: Sample timing phase offset 90° 01: Sample timing phase offset 180° 10: Sample timing phase offset 270° 11: Ignore
3:0	/	/	/

#### 6.1.6.23. SMHC Hardware Reset Register (Default Value: 0x0000\_0001)

Offset: 0x78			Register Name: SMHC_HWRST
Bit	Read/Write	Default/Hex	Description
31:1	/	/	/
0	R/W	0x1	HW_RST 1: Active mode 0: Reset

			These bits cause the cards to enter pre-idle state, which requires them to be re-initialized.
--	--	--	---

#### 6.1.6.24. SMHC IDMAC Control Register (Default Value: 0x0000\_0000)

Offset: 0x0080			Register Name: SMHC_IDMAC
Bit	Read/Write	Default/Hex	Description
31	W	0x0	DES_LOAD_CTRL When IDMAC fetches a descriptor, if the valid bit of a descriptor is not set, IDMAC FSM will go to the suspend state. Setting this bit will make IDMAC refetch descriptor again and do the transfer normally.
30:11	/	/	/
10:8	R	0x0	Reserved
7	R/W	0x0	IDMAC_ENB IDMAC Enable When set, the IDMAC is enabled.
6:2	R/W	0x0	Reserved
1	R/W	0x0	FIX_BUST_CTRL Fixed Burst Controls whether the AHB Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 during start of normal burst transfers. When reset, the AHB will use SINGLE and INCR burst transfer operations.
0	R/W	0x0	IDMAC_RST DMA Reset When set, the DMA Controller resets all its internal registers. SWR is read/write. It is automatically cleared after 1 clock cycle.

#### 6.1.6.25. SMHC Descriptor List Base Address Register (Default Value: 0x0000\_0000)

Offset: 0x0084			Register Name: SMHC_DLBA_REG
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	DES_BASE_ADDR Start of Descriptor List Contains the base address of the First Descriptor. The LSB bits [1:0] are ignored and taken as all-zero by the IDMAC internally. Hence these LSB bits are read-only.

#### 6.1.6.26. SMHC IDMAC Status Register (Default Value: 0x0000\_0000)

Offset: 0x0088	Register Name: SMHC_IDST_REG
----------------	------------------------------

Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16:13	R	0x0	Reserved
12:10	R	0x0	<p>IDMAC_ERR_STA Error Bits</p> <p>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.</p> <p>001: Host Abort received during transmission</p> <p>010: Host Abort received during reception</p> <p>Others: Reserved</p>
9	R/W1C	0x0	<p>ABN_INT_SUM Abnormal Interrupt Summary</p> <p>Logical OR of the following:</p> <p>SMHC_IDST[2]: Fatal Bus Interrupt</p> <p>SMHC_IDST[4]: Descriptor Unavailable bit Interrupt</p> <p>SMHC_IDST[5]: Card Error Summary Interrupt</p> <p>Only unmasked bits affect this bit.</p> <p>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.</p>
8	R/W1C	0x0	<p>NOR_INT_SUM Normal Interrupt Summary</p> <p>Logical OR of the following:</p> <p>SMHC_IDST[0]: Transmit Interrupt</p> <p>SMHC_IDST[1]: Receive Interrupt</p> <p>Only unmasked bits affect this bit.</p> <p>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.</p>
7:6	/	/	/
5	R/W1C	0x0	<p>ERR_FLAG_SUM Card Error Summary</p> <p>Indicates the status of the transaction to/from the card; also present in RINTSTS. Indicates the logical OR of the following bits:</p> <p>EBE: End Bit Error</p> <p>RTO: Response Timeout/Boot ACK Timeout</p> <p>RCRC: Response CRC</p> <p>SBE: Start Bit Error</p> <p>DRTO: Data Read Timeout/BDS timeout</p> <p>DCRC: Data CRC for Receive</p> <p>RE: Response Error</p> <p>Writing a 1 clears this bit.</p>
4	R/W1C	0x0	<p>DES_UNAVL_INT Descriptor Unavailable Interrupt</p> <p>This bit is set when the descriptor is unavailable due to OWN bit = 0 (DESO[31]=0). Writing a 1 clears this bit.</p>
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.

#### 6.1.6.27. SMHC IDM MAC Interrupt Enable Register (Default Value: 0x0000\_0000)

Offset: 0x008C			Register Name: SMHC_IDIE_REG
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9:8	R/W	0x0	Reserved
7:6	/	/	/
5	R/W	0x0	ERR_SUM_INT_ENB Card Error Summary Interrupt Enable. When set, it enables the Card Interrupt Summary.
4	R/W	0x0	DES_UNAVL_INT_ENB Descriptor Unavailable Interrupt. When set along with Abnormal Interrupt Summary Enable, the Descriptor Unavailable interrupt is enabled.
3	/	/	/
2	R/W	0x0	FERR_INT_ENB Fatal Bus Error Enable When set with Abnormal Interrupt Summary Enable, the Fatal Bus Error Interrupt is enabled. When reset, Fatal Bus Error Enable Interrupt is disabled.
1	R/W	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.

#### 6.1.6.28. SMHC Card Threshold Control Register (Default Value: 0x0000\_0000)

Offset: 0x0100			Register Name: SMHC_THLD
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0x0	CARD_RD_THLD Card Read Threshold Size
15:1	/	/	/
0	R/W	0x0	CARD_RD_THLD_ENB Card Read Threshold Enable 0: Card read threshold disabled 1: Card read threshold enabled Host controller initiates Read Transfer only if CARD_RD_THLD amount of space is available in receive FIFO.

#### 6.1.6.29. 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 mstorage based on duration of start bit. For eMMC 4.5, start bit can be: 0: Full cycle 1: Less than one full cycle Set HALF_START_BIT=1 for eMMC 4.5 and above; set to 0 for SD applications.

#### 6.1.6.30. SMHC Response CRC Register (Default Value: 0x0000\_0000)

Offset: 0x0110			Register Name: SMHC_RESP_CRC
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:0	R	0x0	RESP_CRC Response CRC Response CRC from device.

#### 6.1.6.31. SMHC Data7 CRC Register (Default Value: 0x0000\_0000)

Offset: 0x0114	Register Name: SMHC_DAT7_CRC
----------------	------------------------------

Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>DAT7_CRC Data[7] CRC CRC in data[7] from device.</p> <p>In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data.</p> <p>In 4 bits DDR mode,it is not used.</p> <p>In SDR mode, the higher 16 bits indicate the CRC of all data.</p>

#### 6.1.6.32. SMHC Data6 CRC Register (Default Value: 0x0000\_0000)

Offset: 0x0118			Register Name: SMHC_DAT6_CRC
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>DAT6_CRC Data[6] CRC CRC in data[6] from device.</p> <p>In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data.</p> <p>In 4 bits DDR mode,it is not used.</p> <p>In SDR mode, the higher 16 bits indicate the CRC of all data.</p>

#### 6.1.6.33. SMHC Data5 CRC Register (Default Value: 0x0000\_0000)

Offset: 0x011C			Register Name: SMHC_DAT5_CRC
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>DAT5_CRC Data[5] CRC CRC in data[5] from device.</p> <p>In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data.</p> <p>In 4 bits DDR mode,it is not used.</p> <p>In SDR mode, the higher 16 bits indicate the CRC of all data.</p>

#### 6.1.6.34. SMHC Data4 CRC Register (Default Value: 0x0000\_0000)

Offset: 0x0120			Register Name: SMHC_DAT4_CRC
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>DAT4_CRC Data[4] CRC CRC in data[4] from device.</p> <p>In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data.</p>

			In 4 bits DDR mode, the higher 16 bits indicate the CRC of odd data, and the lower 16 bits indicate the CRC of even data. In SDR mode, the higher 16 bits indicate the CRC of all data.
--	--	--	--

#### 6.1.6.35. SMHC Data3 CRC Register (Default Value: 0x0000\_0000)

Offset: 0x0124			Register Name: SMHC_DAT3_CRC
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	DAT3_CRC Data[3] CRC CRC in data[3] from device.  In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data.  In 4 bits DDR mode, the higher 16 bits indicate the CRC of odd data, and the lower 16 bits indicate the CRC of even data.  In SDR mode, the higher 16 bits indicate the CRC of all data.

#### 6.1.6.36. SMHC Data2 CRC Register (Default Value: 0x0000\_0000)

Offset: 0x0128			Register Name: SMHC_DAT2_CRC
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	DAT2_CRC Data[2] CRC CRC in data[2] from device.  In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data.  In 4 bits DDR mode, the higher 16 bits indicate the CRC of odd data, and the lower 16 bits indicate the CRC of even data.  In SDR mode, the higher 16 bits indicate the CRC of all data.

#### 6.1.6.37. SMHC Data1 CRC Register (Default Value: 0x0000\_0000)

Offset: 0x012C			Register Name: SMHC_DAT1_CRC
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	DAT1_CRC Data[1] CRC CRC in data[1] from device.  In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data.  In 4 bits DDR mode, the higher 16 bits indicate the CRC of odd data, and the lower 16 bits indicate the CRC of even data.  In SDR mode, the higher 16 bits indicate the CRC of all data.

#### 6.1.6.38. SMHC Data0 CRC Register (Default Value: 0x0000\_0000)

Offset: 0x0130			Register Name: SMHC_DAT0_CRC
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>DAT0_CRC Data[0] CRC CRC in data[0] from device.</p> <p>In 8 bits DDR mode, the higher 16 bits indicate the CRC of even data, and the lower 16 bits indicate the CRC of odd data.</p> <p>In 4 bits DDR mode, the higher 16 bits indicate the CRC of odd data, and the lower 16 bits indicate the CRC of even data.</p> <p>In SDR mode, the higher 16 bits indicate the CRC of all data.</p>

#### 6.1.6.39. SMHC CRC Status Register (Default Value: 0x0000\_0000)

Offset: 0x0134			Register Name: SMHC_CRC_STA
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2:0	R	0x0	<p>CRC_STA CRC Status CRC status from device in write operation Positive CRC status token: 3'b010 Negative CRC status token: 3'b101</p>

#### 6.1.6.40. 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	<p>DAT_DRV_PH_SEL Data Drive Phase Select 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</p>
16	R/W	0x1	<p>CMD_DRV_PH_SEL Command Drive Phase Select 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</p>
15:0	/	/	/

#### 6.1.6.41. SMHC Sample Delay Control Register (Default Value: 0x0000\_2000)

Offset: 0x0144			Register Name: SMHC_SAMP_DL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R/W	0x0	SAMP_DL_CAL_START Sample Delay Calibration Start When set, start sample delay chain calibration.
14	R	0x0	SAMP_DL_CAL_DONE Sample Delay Calibration Done When set, it means that sample delay chain calibration is done and the result of calibration is shown in SAMP_DL.
13:8	R	0x20	SAMP_DL Sample Delay It indicates the number of delay cells corresponding to current card clock. The delay time generated by these delay cells is equal to the cycle of card clock nearly. Generally, it is necessary to do drive delay calibration when card clock is changed. This bit is valid only when SAMP_DL_CAL_DONE is set.
7	R/W	0x0	SAMP_DL_SW_EN Sample Delay Software Enable When set, enable sample delay specified at SAMP_DL_SW
6	/	/	/
5:0	R/W	0x0	SAMP_DL_SW Sample Delay Software The relative delay between clock line and command line, data lines. It can be determined according to the value of SAMP_DL, the cycle of card clock and device's input timing requirement.

#### 6.1.6.42. SMHC Data Strobe Delay Control Register(Default Value: 0x0000\_2000)

Offset: 0x0148			Register Name: SMHC_DS_DL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R/W	0x0	DS_DL_CAL_START Data Strobe Delay Calibration Start When set, start sample delay chain calibration.
14	R	0x0	DS_DL_CAL_DONE Data Strobe Delay Calibration Done When set, it means that sample delay chain calibration is done and the result of calibration is shown in DS_DL.
13:8	R	0x20	DS_DL

			Data Strobe Delay It indicates the number of delay cells corresponding to current card clock. The delay time generated by these delay cells is equal to the cycle of SMHC's clock nearly. This bit is valid only when SAMP_DL_CAL_DONE is set.
7	R/W	0x0	DS_DL_SW_EN Sample Delay Software Enable
6	/	/	/
5:0	R/W	0x0	DS_DL_SW Data Strobe Delay Software

#### 6.1.6.43. 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

## 6.2. TWI

### 6.2.1. Overview

The TWI is designed as an interface between CPU host and the serial TWI bus. It can support all the standard TWI transfer, including slave and master. The communication of the 2-wire bus is carried out by a byte-wise mode based on interrupt or polled handshaking. The TWI can be operated in standard mode (100 kbit/s) or fast-mode (400 kbit/s). The 10-bit addressing mode is supported for this specified application. General call addressing is also supported in slave mode.

The TWI has the following features:

- Software-programmable for slave or master
- Supports repeated START signal
- Allows 10-bit addressing with TWI bus
- Performs arbitration and clock synchronization
- Own address and general call address detection
- Interrupt on address detection
- Supports speed up to 400 kbit/s ('fast mode')
- Allows operation from a wide range of input clock frequency
- TWI driver supports packet transmission and DMA when TWI works in master mode

### 6.2.2. Block Diagram

Figure 6-3 shows the block diagram of TWI.

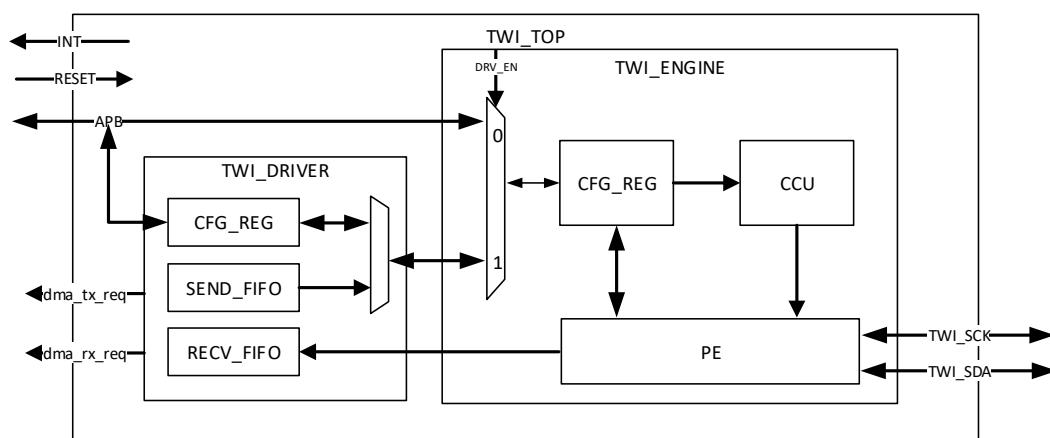


Figure 6- 3. TWI Block Diagram

RESET: Module reset signal

INT: Module output interrupt signal

CFG\_REG: Module configuration register in TWI

PE: Packet encoding/decoding

CCU: Module clock controller unit

### 6.2.3. Operations and Functional Descriptions

#### 6.2.3.1. External Signals

The TWI controller has 2 TWIs. Table 6-3 describes the external signals of TWI. TWI-SCK and TWI-SDA are bidirectional I/O, when TWI is configured as master device, TWI-SCK is output pin; when TWI is configurable as slave device, TWI-SCK is input pin. The unused TWI ports are used as General Purpose I/O ports. For information about General Purpose I/O ports, see **Port Controller** in chapter6.

Table 6- 3. TWI External Signals

Signal	Description	Type
TWI0-SCK	TWI0 Clock Signal	I/O,OD
TWI0-SDA	TWI0 Serial Data	I/O,OD
TWI1-SCK	TWI1 Clock Signal	I/O,OD
TWI1-SDA	TWI1 Serial Data	I/O,OD

#### 6.2.3.2. Clock Sources

Each TWI controller has a fixed clock source. Table 6-4 describes the clock source for TWI. Users can see **Clock Controller Unit(CCU)** in chapter3 for clock setting, configuration and gating information.

Table 6- 4. TWI Clock Sources

Clock Sources	Description
APB2_CLK	TWI clock source, for details on APB2 refer to CCU

#### 6.2.3.3. TWI Engine Master and Slave Mode

There are four operation modes on the TWI bus. They are Master Transmit, Master Receive, Slave Transmit and Slave Receive. In general, CPU host controls TWI engine by writing command and data to its registers. TWI engine transmits an interrupt to CPU when each time a byte transfer is done or a START/STOP command is detected. The CPU host can also poll the status register for current status if the interrupt mechanism is not disabled by the CPU host.

When the CPU host wants to start a bus transfer, it initiates a bus START to enter the master mode by setting IM\_STA bit of the TWI\_CNTR register to high (before it must be low). The TWI engine will assert INT line and INT\_FLAG to indicate a completion for the START command and each consequent byte transfer. At each interrupt, the micro-processor needs to check the TWI\_STAT register for current status. A transfer has to be concluded with STOP command by setting M\_STP bit to high.

In Slave mode, the TWI engine also constantly samples the bus and look for its own slave address during addressing

cycles. Once a match is found, it is addressed and interrupt the CPU host with the corresponding status. Upon request, the CPU host should read the status, read/write TWI\_DATA data register, and set the TWI\_CNTR control register. After each byte transfer, a slave device always stop the operation of remote master by holding the next low pulse on SCL line until the microprocessor responds to the status of previous byte transfer or START command.

#### 6.2.3.4. TWI Driver

TWI driver is only supported for 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 in the following figures . Reg address bytes and Write data bytes are buffered in SEND FIFO, Read data is buffered in RECV FIFO.

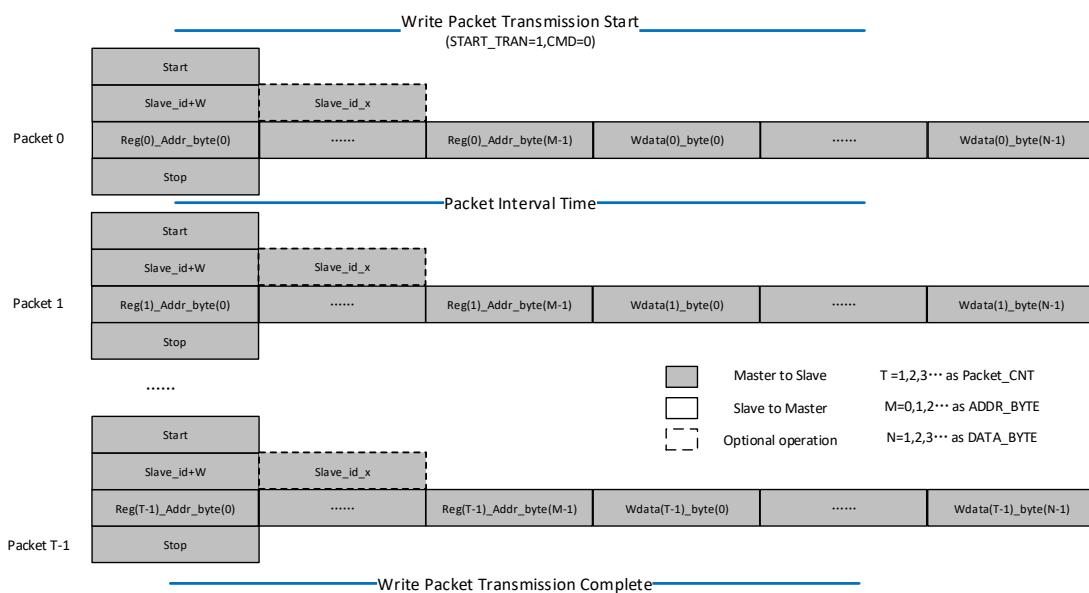
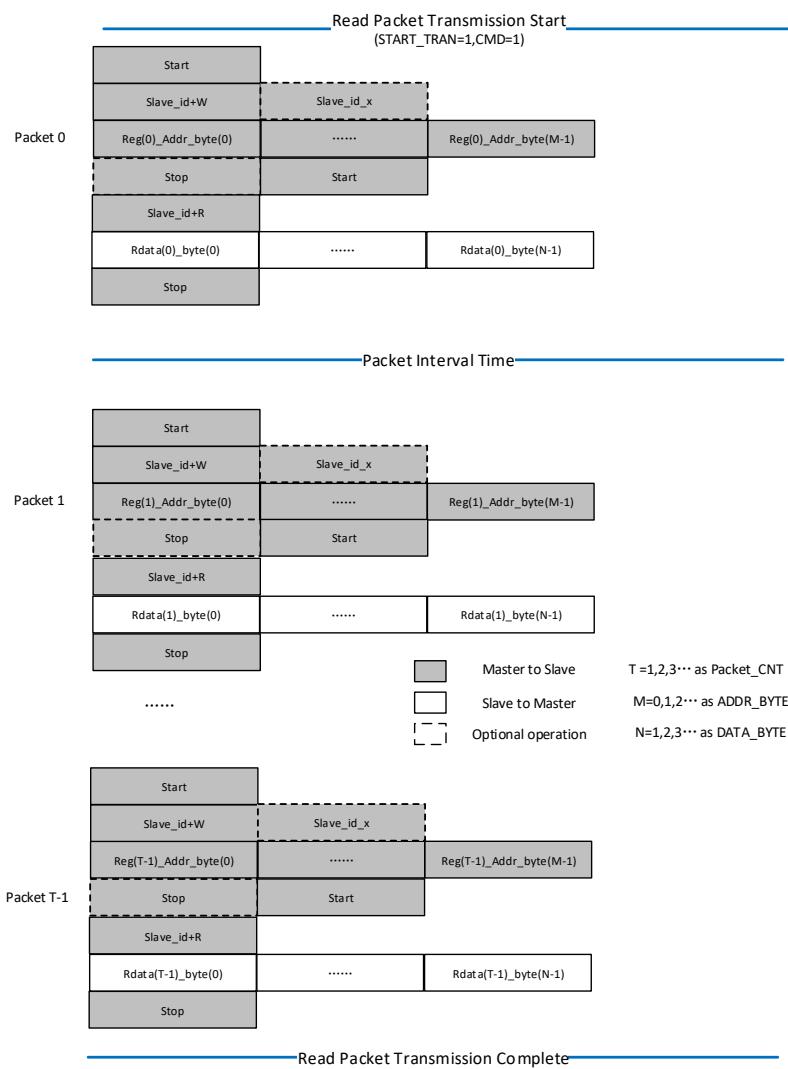


Figure 6- 4. TWI Driver Write Packet Transmission

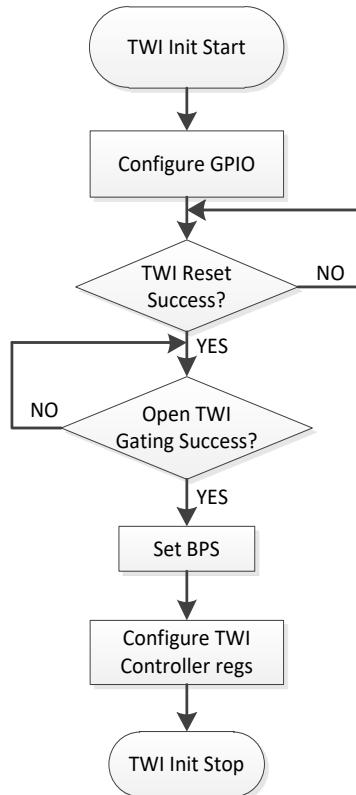


**Figure 6- 5. TWI Driver Read Packet Transmission**

#### 6.2.4. Programming Guidelines

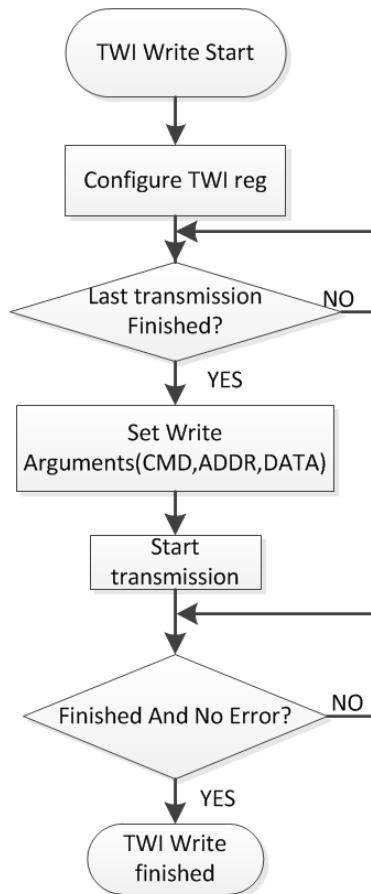
The TWI controller operates in 8-bit data format. The data on the TWI\_SDA line is always 8 bits long. At first, the TWI controller will send a start condition. When in the addressing formats of 7-bit, TWI sends out a 8 bits message which include 7 MSB slave address and 1 LSB read/write flag. The least significant of the slave 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 please refer to register description in Section 6.2.6.1 and 6.2.6.2.

Figure 6-6 shows a software operation flow of TWI Initialization.



**Figure 6- 6. TWI Initial Flow**

Figure 6-7 shows a software operation flow of TWI engine write to device.



**Figure 6- 7. TWI Engine Write Flow**

Figure 6-8 shows a software operation flow of TWI engine read from device.

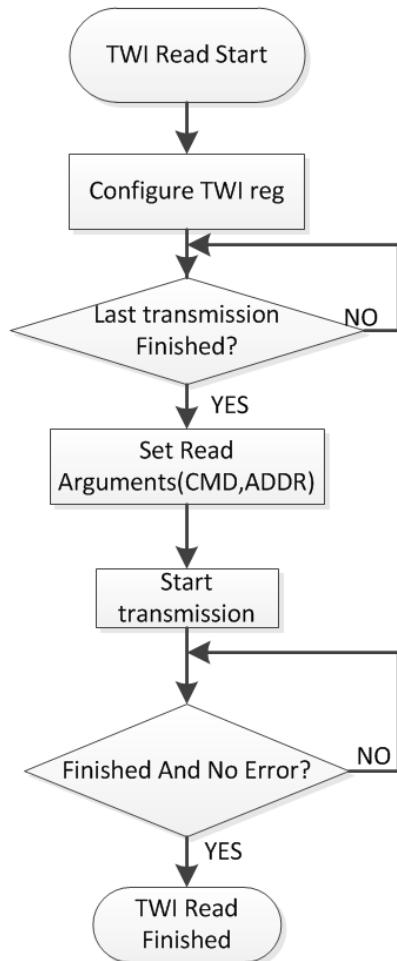
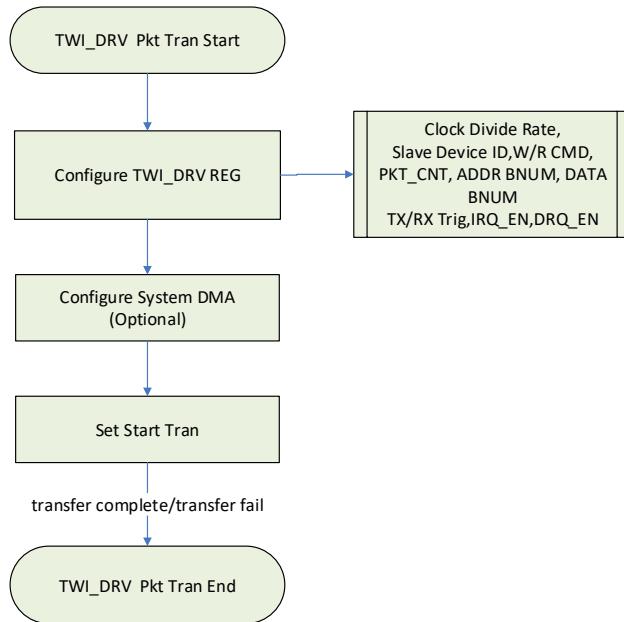


Figure 6- 8. TWI Engine Read Flow

Figure 6-9 shows a software operation flow for packet transmission by TWI driver.



**Figure 6- 9. TWI Driver Packet Transmission Flow**

### 6.2.5. Register List

Module Name	Base Address
TWI0	0x05002000
TWI1	0x05002400

Register Name	Offset	Description
TWI_ADDR	0x0000	TWI Slave Address
TWI_XADDR	0x0004	TWI Extended Slave Address
TWI_DATA	0x0008	TWI Data Byte
TWI_CNTR	0x000C	TWI Control Register
TWI_STAT	0x0010	TWI Status Register
TWI_CCR	0x0014	TWI Clock Control Register
TWI_SRST	0x0018	TWI Software Reset
TWI_EFR	0x001C	TWI Enhance Feature Register
TWI_LCR	0x0020	TWI Line Control Register
TWI_DRV_CTRL	0x0200	TWI_DRV Control Register
TWI_DRV_CFG	0x0204	TWI_DRV Transmission Configuration Register
TWI_DRV_SLV	0x0208	TWI_DRV Slave ID Register
TWI_DRV_FMT	0x020C	TWI_DRV Packet Format Register
TWI_DRV_BUS_CTRL	0x0210	TWI_DRV Bus Control Register
TWI_DRV_INT_CTRL	0x0214	TWI_DRV Interrupt Control Register

TWI_DRV_DMA_CFG	0x0218	TWI_DRV DMA Configure Register
TWI_DRV_FIFO_CON	0x021C	TWI_DRV FIFO Content Register
TWI_DRV_SEND_FIFO_ACC	0x0300	TWI_DRV Send Data FIFO Access Register
TWI_DRV_RECV_FIFO_ACC	0x0304	TWI_DRV Receive Data FIFO Access Register

## 6.2.6. Register Description

### 6.2.6.1. TWI Slave Address Register(Default Value:0x0000\_0000)

Offset: 0x0000			Register Name: TWI_ADDR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:1	R/W	0x0	SLA Slave Address 7-bit addressing: SLA6, SLA5, SLA4, SLA3, SLA2, SLA1, SLA0  10-bit addressing: 1, 1, 1, 1, 0, SLAX[9:8]
0	R/W	0x0	GCE General Call Address Enable 0: Disable 1: Enable



#### NOTE

For 7-bit addressing:

SLA6 – SLA0 is the 7-bit address of the TWI in slave mode. When the TWI receives this address after a START condition, it will generate an interrupt and enter slave mode. (SLA6 corresponds to the first bit received from the TWI bus.) If GCE is set to '1', the TWI will also recognize the general call address (00h).

For 10-bit addressing:

When the address received starts with 11110b, the TWI recognizes this as the first part of a 10-bit address and if the next two bits match ADDR[2:1] (i.e. SLAX9 and SLAX8 of the device's extended address), it sends an ACK. (The device does not generate an interrupt at this point.) If the next byte of the address matches the XADDR register (SLAX7 – SLAX0), the TWI generates an interrupt and goes into slave mode.

### 6.2.6.2. 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]
--	--	-----------

#### 6.2.6.3. TWI Data Register(Default Value:0x0000\_0000)

Offset: 0x0008			Register Name: TWI_DATA
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	TWI_DATA Data byte transmitted or received

#### 6.2.6.4. TWI Control Register(Default Value:0x0000\_0000)

Offset: 0x000C			Register Name: TWI_CNTR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	INT_EN Interrupt Enable 0: The interrupt line always low 1: The interrupt line will go high when INT_FLAG is set.
6	R/W	0x0	BUS_EN TWI Bus Enable 0: The TWI bus ISDA/ISCL is ignored and the TWI Controller will not respond to any address on the bus 1: The TWI will respond to call to its slave address – and to the general call address if the GCE bit in the ADDR register is set.   <b>NOTE</b> <b>In master operation mode, this bit should be set to '1'.</b>
5	R/WAC	0x0	M_STA Master Mode Start When M_STA is set to '1', TWI Controller enters master mode and will transmit a START condition on the bus when the bus is free. If the M_STA bit is set to '1' when the TWI Controller is already in master mode and one or more bytes have been transmitted, then a repeated START condition will be sent. If the M_STA bit is set to '1' when the TWI is being accessed in slave mode, the TWI will complete the data transfer in slave mode then enter master mode when the bus has been released.  The M_STA bit is cleared automatically after a START condition has been sent. Writing a '0' to this bit has no effect.
4	R/W1C	0x0	M_STP Master Mode Stop If M_STP is set to '1' in master mode, a STOP condition is transmitted on the

			TWI bus. If the M_STP bit is set to '1' in slave mode, the TWI will indicate if a STOP condition has been received, but no STOP condition will be transmitted on the TWI bus. If both M_STA and M_STP bits are set, the TWI will first transmit the STOP condition (if in master mode) then transmit the START condition.  The M_STP bit is cleared automatically: writing a '0' to this bit has no effect.
3	R/W1C	0x0	INT_FLAG  Interrupt Flag  INT_FLAG is automatically set to '1' when any of 28 (out of the possible 29) states is entered (see 'STAT Register' below). The only state that does not set INT_FLAG is state F8h. If the INT_EN bit is set, the interrupt line goes high when IFLG is set to '1'. If the TWI is operating in slave mode, data transfer is suspended when INT_FLAG is set and the low period of the TWI bus clock line (SCL) is stretched until '1' is written to INT_FLAG. The TWI clock line is then released and the interrupt line goes low.
2	R/W	0x0	A_ACK  Assert Acknowledge  When A_ACK is set to '1', an Acknowledge (low level on SDA) will be sent during the acknowledge clock pulse on the TWI bus if:  (1). Either the whole of a matching 7-bit slave address or the first or the second byte of a matching 10-bit slave address has been received.  (2). The general call address has been received and the GCE bit in the ADDR register is set to '1'.  (3). A data byte has been received in master or slave mode.  When A_ACK is '0', a Not Acknowledge (high level on SDA) will be sent when a data byte is received in master or slave mode.  If A_ACK is cleared to '0' in slave transmitter mode, the byte in the DATA register is assumed to be the 'last byte'. After this byte has been transmitted, the TWI will enter state C8h then return to the idle state (status code F8h) when INT_FLAG is cleared.  The TWI will not respond as a slave unless A_ACK is set.
1:0	/	/	/

#### 6.2.6.5. TWI Status Register(Default Value:0x0000\_00F8)

Offset: 0x0010			Register Name: TWI_STAT
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R	0xF8	STA  Status Information Byte  <b>Code Status</b>

		<p>0x00: Bus error</p> <p>0x08: START condition transmitted</p> <p>0x10: Repeated START condition transmitted</p> <p>0x18: Address + Write bit transmitted, ACK received</p> <p>0x20: Address + Write bit transmitted, ACK not received</p> <p>0x28: Data byte transmitted in master mode, ACK received</p> <p>0x30: Data byte transmitted in master mode, ACK not received</p> <p>0x38: Arbitration lost in address or data byte</p> <p>0x40: Address + Read bit transmitted, ACK received</p> <p>0x48: Address + Read bit transmitted, ACK not received</p> <p>0x50: Data byte received in master mode, ACK transmitted</p> <p>0x58: Data byte received in master mode, not ACK transmitted</p> <p>0x60: Slave address + Write bit received, ACK transmitted</p> <p>0x68: Arbitration lost in address as master, slave address + Write bit received, ACK transmitted</p> <p>0x70: General Call address received, ACK transmitted</p> <p>0x78: Arbitration lost in address as master, General Call address received, ACK transmitted</p> <p>0x80: Data byte received after slave address received, ACK transmitted</p> <p>0x88: Data byte received after slave address received, not ACK transmitted</p> <p>0x90: Data byte received after General Call received, ACK transmitted</p> <p>0x98: Data byte received after General Call received, not ACK transmitted</p> <p>0xA0: STOP or repeated START condition received in slave mode</p> <p>0xA8: Slave address + Read bit received, ACK transmitted</p> <p>0xB0: Arbitration lost in address as master, slave address + Read bit received, ACK transmitted</p> <p>0xB8: Data byte transmitted in slave mode, ACK received</p> <p>0xC0: Data byte transmitted in slave mode, ACK not received</p> <p>0xC8: Last byte transmitted in slave mode, ACK received</p> <p>0xD0: Second Address byte + Write bit transmitted, ACK received</p> <p>0xD8: Second Address byte + Write bit transmitted, ACK not received</p> <p>0xF8: No relevant status information, INT_FLAG=0</p> <p>Others: Reserved</p>
--	--	--

#### 6.2.6.6. TWI Clock Register(Default Value:0x0000\_0000)

Offset: 0x0014			Register Name: TWI_CCR
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
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: $F_{samp} = F_0 = F_{in} / 2^{CLK\_N}$

		<p>The TWI OSCL output frequency, in master mode, is F1 / 10:  <math>F1 = F0 / (\text{CLK\_M} + 1)</math>  <math>F_{\text{oscl}} = F1 / 10 = F_{\text{in}} / (2^{\text{CLK\_N}} * (\text{CLK\_M} + 1) * 10)</math></p> <p>For Example:</p> <p><math>F_{\text{in}} = 48 \text{ MHz}</math> (APB clock input)  For 400 kHz full speed 2Wire, CLK_N = 2, CLK_M=2  <math>F0 = 48 \text{ MHz} / 2^2 = 12 \text{ MHz}</math>, <math>F1 = F0 / (10 * (2+1)) = 0.4 \text{ MHz}</math></p> <p>For 100 kHz standard speed 2Wire, CLK_N=2, CLK_M=11  <math>F0 = 48 \text{ MHz} / 2^2 = 12 \text{ MHz}</math>, <math>F1 = F0 / (10 * (11+1)) = 0.1 \text{ MHz}</math></p>
--	--	---

#### 6.2.6.7. 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.

#### 6.2.6.8. TWI Enhance Feature Register(Default Value:0x0000\_0000)

Offset: 0x001C			Register Name: TWI_EFR
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
0:1	R/W	0x0	DBN Data Byte Number Follow Read Command Control 00 : No data byte can be written after read command 01 : Only 1 byte data can be written after read command 10 : 2 bytes data can be written after read command 11 : 3 bytes data can be written after read command

#### 6.2.6.9. TWI Line Control Register(Default Value:0x0000\_003A)

Offset: 0x0020			Register Name: TWI_LCR
Bit	Read/Write	Default/Hex	Description
31:6	/	/	/
5	R	0x1	SCL_STATE Current State of TWI_SCL

			0 : Low 1 : High
4	R	0x1	SDA_STATE Current State of TWI_SDA 0 : Low 1 : High
3	R/W	0x1	SCL_CTL TWI_SCL Line State Control Bit When line control mode is enabled (bit[2] set), this bit decides the output level of TWI_SCL. 0 : Output low level 1 : Output high level
2	R/W	0x0	SCL_CTL_EN TWI_SCL Line State Control Enable When this bit is set, the state of TWI_SCL is controlled by the value of bit[3]. 0 : Disable TWI_SCL line control mode 1 : Enable TWI_SCL line control mode
1	R/W	0x1	SDA_CTL TWI_SDA Line State Control Bit When line control mode is enabled (bit[0] set), this bit decides the output level of TWI_SDA. 0 : Output low level 1 : Output high level
0	R/W	0x0	SDA_CTL_EN TWI_SDA Line State Control Enable When this bit is set, the state of TWI_SDA is controlled by the value of bit[1]. 0 : Disable TWI_SDA line control mode 1 : Enable TWI_SDA line control mode

#### 6.2.6.10. TWI\_DRV Control Register(Default Value:0x00F8\_0000)

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 <b>Setting this bit to 1 if reading from a slave which register width is equal to 0.</b>
27:24	R	0x0	TRAN_RESULT 000: OK 001: FAIL Other: Reserved
23:16	R	0xf8	TWI_STA 0x00: bus error 0x08: START condition transmitted 0x10: Repeated START condition transmitted 0x18: Address + Write bit transmitted, ACK received 0x20: Address + Write bit transmitted, ACK not received 0x28: Data byte transmitted in master mode, ACK received 0x30: Data byte transmitted in master mode, ACK not received 0x38: Arbitration lost in address or data byte 0x40: Address + Read bit transmitted, ACK received 0x48: Address + Read bit transmitted, ACK not received 0x50: Data byte received in master mode, ACK received 0x58: Data byte received in master mode, ACK not received 0x01: Timeout when sending 9 <sup>th</sup> SCL clk Other: Reserved
15:2	/	/	/
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)

#### 6.2.6.11. TWI\_DRV Transmission Configuration Register(Default Value:0x1000\_0001)

Offset: 0x0204			Register Name: TWI_DRV_CFG
Bit	Read/Write	Default/Hex	Description
31:24	R/W	0x10	TIMEOUT_N When sending the 9 <sup>th</sup> clock, assert fail signal when slave device did not response after $N \cdot F_{SCL}$ cycles. And software must do a reset to TWI_DRV module and send a stop condition to slave.
23:16	R/W	0x0	PKT_INTERVAL Define the interval between each packet in $32 \cdot F_{SCL}$ cycles. 0~255

15:0	R/W	0x1	PACKET_CNT FIFO data be transmitted as PACKET_CNT packets in current format.
------	-----	-----	---

#### 6.2.6.12. TWI\_DRV Slave ID Register(Default Value:0x0000\_0000)

Offset: 0x0208			Register Name: TWI_DRV_SLV
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:9	R/W	0x0	SLV_ID Slave device ID <ul style="list-style-type: none"> <li>• 7-bit addressing SLA6, SLA5, SLA4, SLA3, SLA2, SLA1, SLA0</li> <li>• 10-bit addressing 1, 1, 1, 1, 0, SLAX[9:8]</li> </ul>
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

#### 6.2.6.13. 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_BYT How many bytes be sent as slave device reg address 0~255
15:0	R/W	0x1	DATA_BYT How many bytes be sent/received as data 1~65535

#### 6.2.6.14. TWI\_DRV Bus Control Register(Default Value:0x0000\_00C0)

Offset: 0x0210			Register Name: TWI_DRV_BUS_CTRL
Bit	Read/Write	Default/Hex	Description
31:15	/	/	/
14:12	R/W	0x0	CLK_N TWI_DRV bus sampling clock F0=24MHz/2^CLK_N
11:8	R/W	0x0	CLK_M

			TWI_DRV output SCL frequency is $F_{SCL}=F1/10=(F0/(CLK\_M+1))/10$
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

#### 6.2.6.15. 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

#### 6.2.6.16. TWI\_DRV DMA Configure Register(Default Value:0x0010\_0010)

Offset: 0x0218			Register Name: TWI_DRV_DMA_CFG
Bit	Read/Write	Default/Hex	Description
31:25	/	/	/
24	R/W	0x0	DMA_RX_EN
23:22	/	/	/

21:16	R/W	0x10	RX_TRIG When DMA_RX_EN set, send DMA RX Req when the data byte number in RECV_FIFO reaches RX_TRIG or Read Packet Transmission completed with RECV_FIFO not empty
15:9	/	/	/
8	R/W	0x0	DMA_TX_EN
7:6	/	/	/
5:0	R/W	0x10	TX_TRIG When DMA_TX_EN set, send DMA TX Req when there is no less than DMA_TX_TRIG empty byte number in SEND_FIFO

#### 6.2.6.17. 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

#### 6.2.6.18. 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

#### 6.2.6.19. 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
--	--	---

## 6.3. UART

### 6.3.1. Overview

The UART is used for serial communication with a peripheral, modem (data carrier equipment, DCE) or data set. Data is written from a master (CPU) over the APB bus to the UART and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back.

The UART contains registers to control the character length, baud rate, parity generation/checking, and interrupt generation. Although there is only one interrupt output signal from the UART, there are several prioritized interrupt types that can be responsible for its assertion. Each of the interrupt types can be separately enabled/disabled with the control registers.

The UART has 16450 and 16550 modes of operation, which are compatible with a range of standard software drivers. In 16550 mode, transmit and receive operations are both buffered by FIFOs. In 16450 mode, these FIFOs are disabled.

The UART supports word lengths from five to eight bits, an optional parity bit and 1, 1 ½ or 2 stop bits, and is fully programmable by an AMBA APB CPU interface. A 16-bit programmable baud rate generator and an 8-bit scratch register are included, together with separate transmit and receive FIFOs. Eight modem control lines and a diagnostic loop-back mode are provided.

Interrupts can be generated for a range of TX Buffer/FIFO, RX Buffer/FIFO, Modem Status and Line Status conditions.

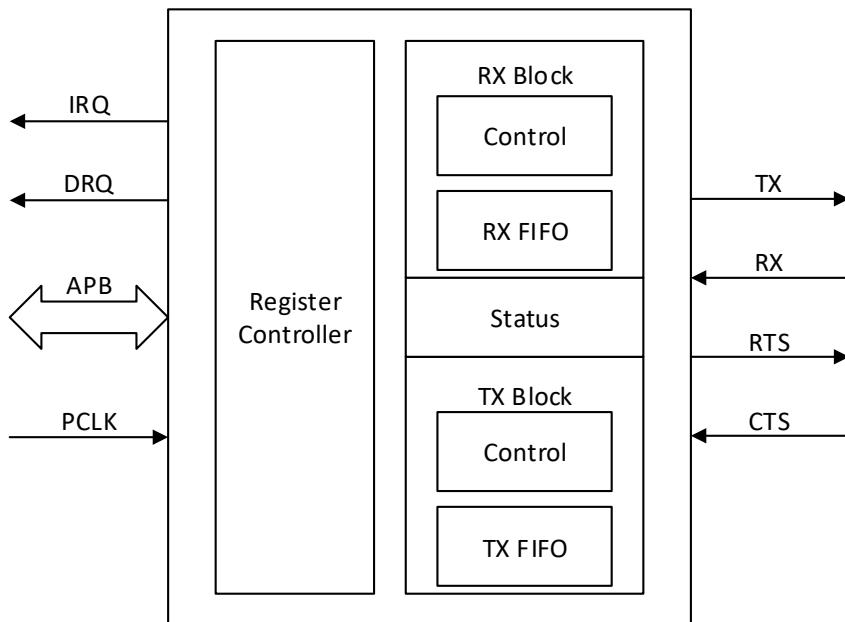
For integration in system where Infrared SIR serial data format is required, the UART can be configured to have a software-programmable IrDA SIR mode. If this mode is not selected, only the UART (RS232 standard) serial data format is available.

The UART has the following features:

- Compatible with industry-standard 16550 UARTs
- 256 bytes transmit and receive data FIFOs
- Capable of speed up to 4 Mbit/s
- Supports 5 to 8 data bits and 1/1.5/2 stop bits
- Supports even, odd or no parity
- Supports DMA controller interface
- Supports software/hardware flow control
- Supports IrDA 1.0 SIR
- Supports RS-485/9-bit mode

### 6.3.2. Block Diagram

Figure 6-10 shows a block diagram of the UART.



**Figure 6- 10. UART Block Diagram**

### 6.3.3. Operations and Functional Descriptions

#### 6.3.3.1. External Signals

Table 6-5 describes the external signals of UART.

**Table 6- 5. UART External Signals**

Signal	Type	Description
UART0-TX	O	UART0 Data Transmit
UART0-RX	I	UART0 Data Receive
UART1-TX	O	UART1 Data Transmit
UART1-RX	I	UART1 Data Receive
UART1-CTS	I	UART1 Data Clear to Send
UART1-RTS	O	UART1 Data Request to Send
UART2-TX	O	UART2 Data Transmit
UART2-RX	I	UART2 Data Receive
UART2-CTS	I	UART2 Data Clear to Send
UART2-RTS	O	UART2 Data Request to Send
UART3-TX	O	UART3 Data Transmit
UART3-RX	I	UART3 Data Receive
UART3-CTS	I	UART3 Data Clear to Send
UART3-RTS	O	UART3 Data Request to Send

### 6.3.3.2. Clock Sources

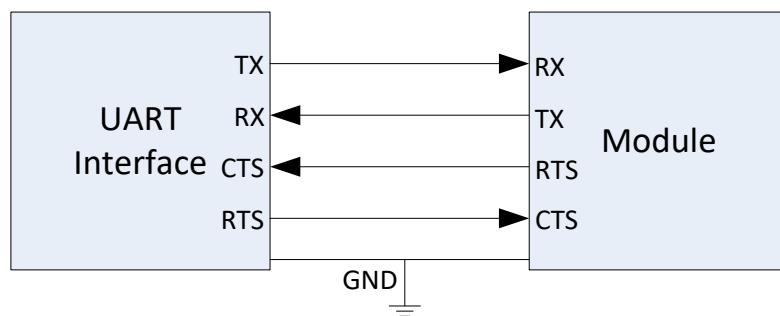
Table 6-6 describes the clock sources of UART.

**Table 6- 6. UART Clock Sources**

Clock Sources	Description
APB2_CLK	Clock of APB2

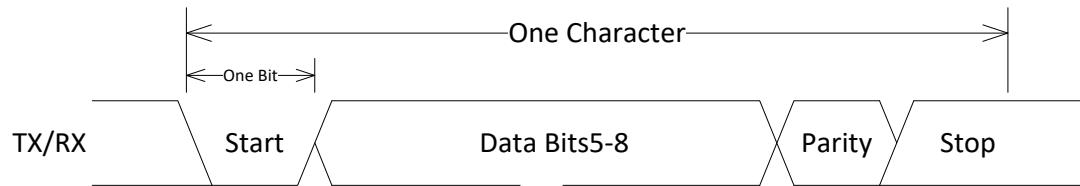
### 6.3.3.3. Typical Application

Figure 6-11 shows the application block diagram of UART.

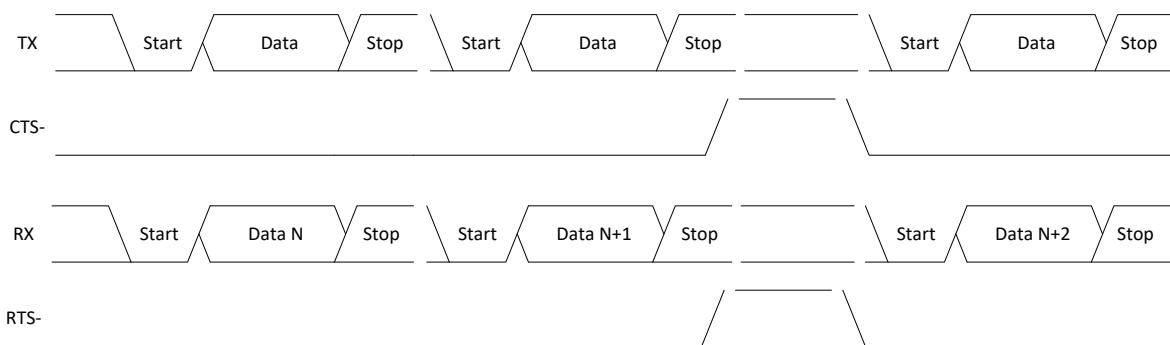


**Figure 6- 11. UART Application Diagram**

### 6.3.3.4. UART Timing Diagram



**Figure 6- 12. UART Serial Data Format**



**Figure 6- 13. RTS/CTS Autoflow Control Timing**

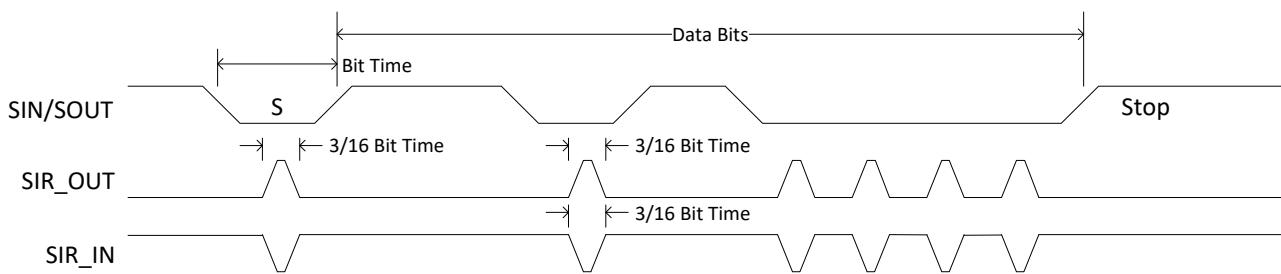


Figure 6-14. Serial IrDA Data Format

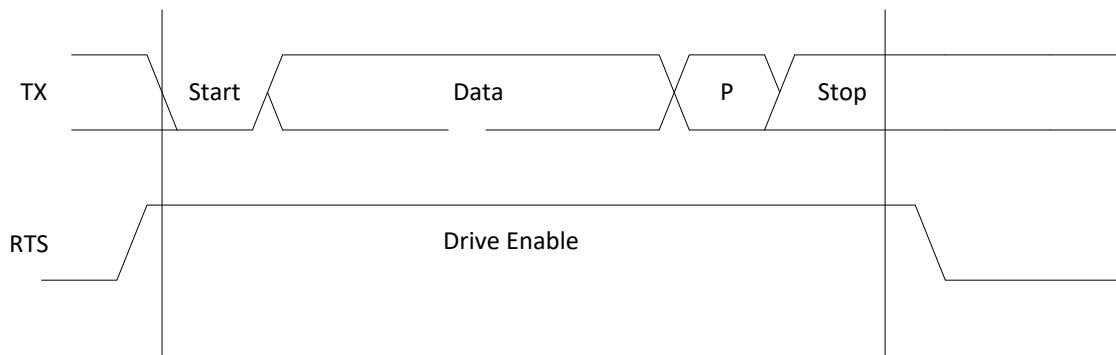


Figure 6-15. RS-485 Timing

### 6.3.3.5. UART Operating Mode

#### 6.3.3.5.1. Basic Mode Setting

The **UART\_LCR** register can set basic parameter of a data frame: data width(5 to 8 bits), stop bit number(1/1.5/2), parity type.

A frame transfer of the UART includes the start signal, data signal, parity bit and stop signal. The LSB is transmitted first.

- Start signal(start bit): It is the start flag of a data frame. According to UART protocol, the low level of TXD signal indicates the start of a data frame. When the UART transmits data, the level need hold high.
- Data signal(data bit): The data bit width can be configured as 5-bit,6-bit,7-bit,8-bit through different applications.
- Parity bit: It is 1-bit error correction signal. Parity bit includes odd parity, even parity. The UART can enable and disable the parity bit by setting the **UART\_LCR** register.
- Stop Signal(stop bit): It is the stop bit of a data frame. The stop bit can be set to 1-bit,1.5-bit and 2-bit by the **UART\_LCR** register. The high level of TXD signal indicates the end of a data frame.

#### 6.3.3.5.2. Baud Rate Setting

The baud rate is calculated as follows:  $\text{Baud rate} = \text{SCLK} / (16 * \text{divisor})$ . SCLK is usually APB2 and can be set in CCU.

Divisor is frequency divider of UART. The frequency divider has 16-bit, the low 8-bit is in the **UART\_DLL** register, the

high 8-bit is in the UART\_DLH register.

The relationship between different UART mode and error rate is as follows.

**Table 6- 7. UART Mode Baud and Error Rates**

Clock source	Divisor	Baud rate	Over sampling	Error(%)
24000000	5000	300	16	0
24000000	2500	600	16	0
24000000	1250	1200	16	0
24000000	625	2400	16	0
24000000	313	4800	16	-0.16
24000000	156	9600	16	0.16
24000000	78	19200	16	0.16
24000000	39	38400	16	0.16
24000000	26	57600	16	0.16
24000000	13	115200	16	0.16
48000000	13	230400	16	0.16
64000000	7	576000	16	-0.794
75000000	5	921600	16	1.725
48000000	3	1000000	16	0
24000000	1	1500000	16	0
48000000	1	3000000	16	0
64000000	1	4000000	16	0

**Table 6- 8. 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 6- 9. 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

### 6.3.3.5.3. DLAB Setting

DLAB control bit (**UART\_LCR[7]**) is the access control bit of divisor Latch register.

If DLAB is 0, then 0x00 offset address is **TX/RX FIFO** register, 0x04 offset address is **IER** register.

If DLAB is 1, then 0x00 offset address is **DLL** register, 0x04 offset address is **DLH** register.

When UART initial, divisor need be set. That is, writing 1 to DLAB can access the **DLL** and **DLH** register, after finished setting, writing 0 to DLAB can access the **TX/RX FIFO** register.

### 6.3.3.5.4. CHCFG\_AT\_BUSY Setting

The function of **CHCFG\_AT\_BUSY**(UART\_HALT[1]) and **CHANGE\_UPDATE**(UART\_HALT[2]) are as follows.

**CHCFG\_AT\_BUSY**(configure at busy): Enable the bit, software can also set UART controller when UART is busy, such as the LCR,DLH,DLL register.

**CHANGE\_UPDATE**(change update): If **CHCFG\_AT\_BUSY** is enabled, and **CHANGE\_UPDATE** is written to 1, the configuration of UART controller can be updated. After completed update, the bit is cleared to 0 automatically.

Setting divisor, performs the following steps:

Step1 Write 1 to **CHCFG\_AT\_BUSY** to enable “configure at busy”.

Step2 Write 1 to **DLAB** , and set **DLH** and **DLL**.

Step3 Write 1 to **CHANGE\_UPDATE** to update configuration. The bit is cleared to 0 automatically after completed update.

### 6.3.3.5.5. UART Busy

**UART\_USR[0]** is a busy flag of UART controller or not.

When TX transmits data, or RX receives data,or TX FIFO is not empty, or RX FIFO is not empty, then the BUSY flag bit can be set to 1 by hardware, which indicates the UART controller is busy.

#### 6.3.4. Programming Guidelines

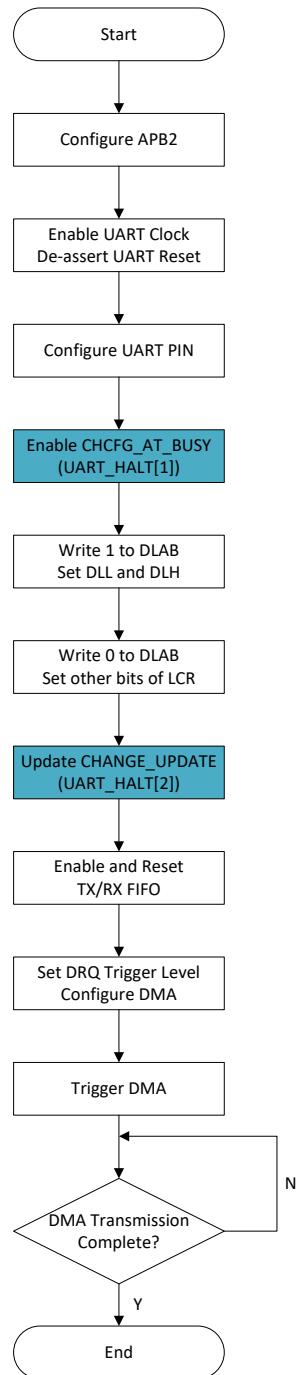


Figure 6- 16. UART DRQ Flow Chart

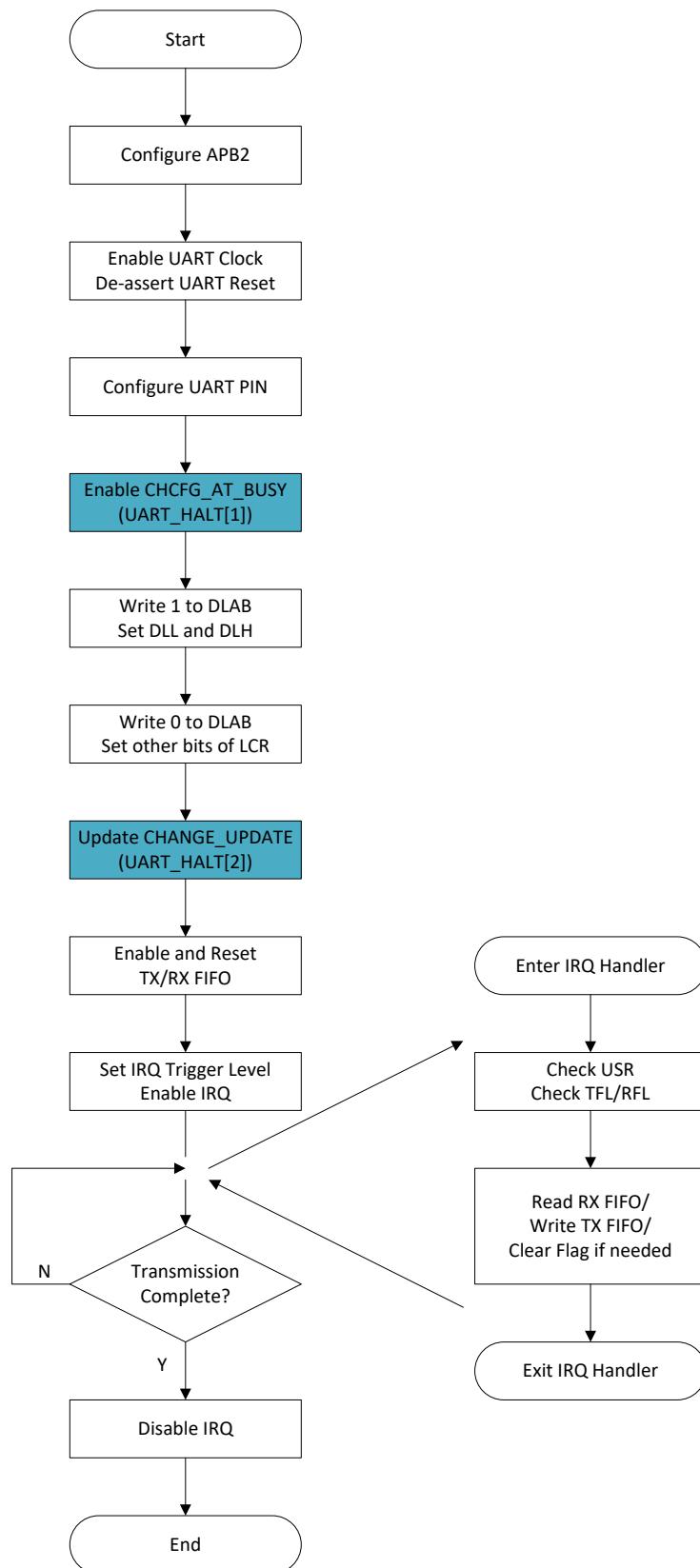


Figure 6- 17. UART IRQ Flow Chart

### 6.3.5. Register List

Module Name	Base Address
UART0	0x05000000
UART1	0x05000400
UART2	0x05000800
UART3	0x05000C00

Register Name	Offset	Description
UART_RBR	0x0000	UART Receive Buffer Register
UART_THR	0x0000	UART Transmit Holding Register
UART_DLL	0x0000	UART Divisor Latch Low Register
UART_DLH	0x0004	UART Divisor Latch High Register
UART_IER	0x0004	UART Interrupt Enable Register
UART_IIR	0x0008	UART Interrupt Identity Register
UART_FCR	0x0008	UART FIFO Control Register
UART_LCR	0x000C	UART Line Control Register
UART_MCR	0x0010	UART Modem Control Register
UART_LSR	0x0014	UART Line Status Register
UART_MSR	0x0018	UART Modem Status Register
UART_SCH	0x001C	UART Scratch Register
UART_USR	0x007C	UART Status Register
UART_TFL	0x0080	UART Transmit FIFO Level Register
UART_RFL	0x0084	UART Receive FIFO Level Register
UART_HSK	0x0088	UART DMA Handshake Configuration Register
UART_HALT	0x00A4	UART Halt TX Register
UART_DBG_DLL	0x00B0	UART Debug DLL Register
UART_DBG_DLH	0x00B4	UART Debug DLH Register
UART_485_CTL	0x00C0	UART RS485 Control and Status Register
RS485_ADDR_MATCH	0x00C4	UART RS485 Address Match Register
BUS_IDLE_CHK	0x00C8	UART RS485 Bus Idle Check Register
TX_DLY	0x00CC	UART TX Delay Register

### 6.3.6. Register Description

#### 6.3.6.1. UART Receiver Buffer Register(Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: UART_RBR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R	0x0	RBR Receiver Buffer Register

			<p>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.</p> <p>If in FIFO mode and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register can not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an overrun error occurs.</p>
--	--	--	---

#### 6.3.6.2. 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	<p>THR Transmit Holding Register Data be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set.</p> <p>If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, 16 number of characters data may be written to the THR before the FIFO is full. When the FIFO is full, any write data results in the write data being lost.</p>

#### 6.3.6.3. UART Divisor Latch Low Register(Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: UART_DLL
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	<p>DLL Divisor Latch Low Lower 8 bits of a 16 bits, read/write, Divisor Latch Register contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero).</p> <p>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).</p> <p>Note that when the Divisor Latch Registers (DLL and DLH) are set to zero, the baud clock is disabled and no serial communications occur. Also, once</p>

			the DLL is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.
--	--	--	--

#### 6.3.6.4. 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	<p>DLH Divisor Latch High Upper 8 bits of a 16 bits, read/write, Divisor Latch Register contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero).</p> <p>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).</p> <p>Note that when the Divisor Latch Registers (DLL and DLH) is set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.</p>

#### 6.3.6.5. UART Interrupt Enable Register(Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: UART_IER
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	<p>PTIME Programmable THRE Interrupt Mode Enable This is used to enable/disable the generation of THRE Interrupt. 0: Disable 1: Enable</p>
6:5	/	/	/
4	R/W	0x0	<p>RS485_INT_EN RS485 Interrupt Enable 0:Disable 1:Enable</p>
3	R/W	0x0	<p>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</p>

			1: Enable
2	R/W	0x0	<p>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.</p> <p>0: Disable 1: Enable</p>
1	R/W	0x0	<p>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.</p> <p>0: Disable 1: Enable</p>
0	R/W	0x0	<p>ERBFI Enable Received Data Available Interrupt This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupt.</p> <p>0: Disable 1: Enable</p>

#### 6.3.6.6. 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	<p>FEFLAG FIFOs Enable Flag This is used to indicate whether the FIFOs are enabled or disabled.</p> <p>00: Disable 11: Enable</p>
5:4	/	/	/
3:0	R	0x1	<p>IID Interrupt ID This indicates the highest priority pending interrupt which can be one of the following types:</p> <p>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</p>

			The bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt.
--	--	--	---

Interrupt ID	Priority 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, receives address data and match setting address	Writes 1 to addr flag to reset
0100	Third	Received data available	Receiver data available (non-FIFO mode or FIFOs disabled) or RCVR FIFO trigger level reached (FIFO mode and FIFOs enabled)	Reading the receiver buffer register (non-FIFO mode or FIFOs disabled) or the FIFO drops below the trigger level (FIFO mode and FIFOs enabled)
1100	Fourth	Character timeout indication	No characters in or out of the RCVR FIFO during the last 4 character times and there is at least 1character in it during This time	Reading the receiver buffer register
0010	Fifth	Transmit holding register empty	Transmitter holding register empty (Program THRE mode disabled) or XMIT FIFO at or below threshold (Program THRE mode enabled)	Reading the IIR register (if source of interrupt); or, writing into THR (FIFOs or THRE mode not selected or disabled) or XMIT FIFO above threshold (FIFOs and THRE mode selected and enabled).
0000	Sixth	Modem status	Clear to send or data set ready or ring indicator or data carrier detect. Note that if auto flow control mode is enabled, a change in CTS (that is, DCTS set) does not cause an interrupt.	Reading the Modem status register
0111	Seventh	Busy detect indication	UART_16550_COMPATIBLE = NO and master has tried to write to the Line Control Register while the UART is busy (USR[0] is set to one).	Reading the UART status register

### 6.3.6.7. UART FIFO Control Register(Default Value: 0x0000\_0000)

Offset: 0x0008			Register Name: UART_FCR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:6	W	0x0	RT RCVR Trigger This is used to select the trigger level in the receiver FIFO when the Received Data Available Interrupt is generated. In auto flow control mode it

			<p>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.</p> <p>00: 1 character in the FIFO 01: FIFO ¼ full 10: FIFO ½ full 11: FIFO-2 less than full</p>
5:4	W	0x0	<p>TFT TX Empty Trigger</p> <p>This is used to select the empty threshold level when the THRE Interrupts are generated and the mode is active. It also determines when the dma_tx_req_n signal is asserted in certain modes of operation.</p> <p>00: FIFO empty 01: 2 characters in the FIFO 10: FIFO ¼ full 11: FIFO ½ full</p>
3	W	0x0	<p>DMAM DMA Mode 0: Mode 0</p> <p>In this mode, if PTE is high and TX FIFO is enabled, the TX DMA request will send when TFL is less than or equal to FIFO Trigger Level. If PTE is high and TX FIFO is disabled, the TX DMA request will send when THRE is empty. If PTE is low, the TX DMA request will send when the TX FIFO is empty.</p> <p>If dma_pte_rx is high and RX FIFO is enabled, the rx drq will send when RFL is equal to or more than FIFO Trigger Level.</p> <p>1: Mode 1</p> <p>In this mode, if TX FIFO is enabled and the PTE is high, the TX DMA request will send when TFL is less than or equal to FIFO Trigger Level. If PTE is low, the TX DMA request will send when TX FIFO is empty and the request stops only when TX FIFO is full.</p> <p>If RFL is equal to or more than FIFO Trigger Level, the rx drq will be set to 1, in otherwise, it will be set to 0.</p>
2	W	0x0	<p>XFIFOR XMIT FIFO Reset</p> <p>The bit resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request.</p> <p>It is 'self-clearing'. It is not necessary to clear this bit.</p>
1	W	0x0	<p>RFIFOR RCVR FIFO Reset</p> <p>The bit resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request.</p> <p>It is 'self-clearing'. It is not necessary to clear this bit.</p>
0	W	0x0	<p>FIFOE Enable FIFOs</p>

			The bit enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed, both the XMIT and RCVR controller portion of FIFOs is reset.
--	--	--	--

#### 6.3.6.8. 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	<p>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)</p>
6	R/W	0x0	<p>BC Break Control Bit This is used to cause a break condition to be transmitted to the receiving device. If setting to 0, the serial output is forced to the spacing (logic 0) state. When not in Loopback mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE is enabled and active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.</p>
5:4	R/W	0x0	<p>EPS Even Parity Select It is writeable only when UART is not busy (USR[0] is zero) and always writable/readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). Setting the LCR[5] is unset to reverse the LCR[4]. 00: Odd Parity 01: Even Parity 1X: Reverse LCR[4] In RS485 mode, it is the 9th bit--address bit. 11:9th bit = 0, indicates that this is a data byte. 10:9th bit = 1, indicates that this is an address byte.</p> <p> <b>NOTE</b></p> <p><b>When using this function, PEN(LCR[3]) must set to 1.</b></p>
3	R/W	0x0	PEN Parity Enable

			<p>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.</p> <p>0: Parity disabled 1: Parity enabled</p>
2	R/W	0x0	<p>STOP Number of stop bits</p> <p>It is writeable only when UART is not busy (USR[0] is zero) and always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If setting to 0, one stop bit is transmitted in the serial data. If setting to 1 and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit.</p> <p>0: 1 stop bit 1: 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit</p>
1:0	R/W	0x0	<p>DLS Data Length Select</p> <p>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:</p> <p>00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits</p>

#### 6.3.6.9. 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	<p>UART_FUNCTION Select IrDA or RS485</p> <p>00:UART Mode 01:IrDA SIR Mode 10:RS485 Mode 11:Reserved</p>
5	R/W	0x0	<p>AFCE Auto Flow Control Enable</p> <p>When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled.</p> <p>0: Auto Flow Control mode disabled 1: Auto Flow Control mode enabled</p>

4	R/W	0x0	<p><b>LOOP</b></p> <p>Loop Back Mode</p> <p>0: Normal Mode</p> <p>1: Loop Back Mode</p> <p>This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE != Enabled or not active, MCR[6] is set to zero), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE == Enabled AND active, MCR[6] is set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.</p>
3:2	/	/	/
1	R/W	0x0	<p><b>RTS</b></p> <p>Request to Send</p> <p>This is used to directly control the Request to Send (rts_n) output. The RTS (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] is set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] is set to one) and FIFOs enable (FCR[0] is set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low.</p> <p>0: rts_n de-asserted (logic 1) 1: rts_n asserted (logic 0)</p> <p>Note that in Loopback mode (MCR[4] is set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input.</p>
0	R/W	0x0	<p><b>DTR</b></p> <p>Data Terminal Ready</p> <p>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.</p> <p>0: dtr_n de-asserted (logic 1) 1: dtr_n asserted (logic 0)</p> <p>The Data Terminal Ready output is used to inform the modem or data set that the UART is ready to establish communications.</p> <p>Note that in Loopback mode (MCR[4] is set to one), the dtr_n output is held inactive high while the value of this location is internally looped back to an input.</p>

## 6.3.6.10. UART Line Status Register(Default Value: 0x0000\_0060)

Offset:0x0014			Register Name: UART_LSR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R	0x0	FIFOERR RX Data Error in FIFO  When FIFOs are disabled, this bit is always 0. When FIFOs are enabled, this bit is set to "1" when there is at least one PE, FE, or BI in the RX FIFO. It is cleared by a read from the LSR register provided, there are no subsequent errors in the FIFO.
6	R	0x1	TEMT Transmitter Empty  If the FIFOs are disabled, this bit is set to "1" whenever the TX Holding Register and the TX Shift Register are empty. If the FIFOs are enabled, this bit is set whenever the TX FIFO and the TX Shift Register are empty. In both cases, this bit is cleared when a byte is written to the TX data channel.
5	R	0x1	THRE TX Holding Register Empty  If the FIFOs are disabled, this bit is set to "1" whenever the TX Holding Register is empty and ready to accept new data and it is cleared when the CPU writes to the TX Holding Register.  If the FIFOs are enabled, this bit is set to "1" whenever the TX FIFO is empty and it is cleared when at least one byte is written to the TX FIFO.
4	R	0x0	BI Break Interrupt  This is used to indicate the detection of a break sequence on the serial input data.  If in UART mode (SIR_MODE == Disabled), it is set whenever the serial input, sir_in, is held in a logic '0' state for longer than the sum of <i>start time + data bits + parity + stop bits</i> .  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 <i>start time + data bits + parity + stop bits</i> . A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART.  In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read.
3	RC	0x0	FE Framing Error

			<p>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.</p> <p>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]).</p> <p>0: no framing error 1:framing error Reading the LSR clears the FE bit.</p>
2	RC	0x0	<p>PE Parity Error</p> <p>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]).</p> <p>0: no parity error 1: parity error Reading the LSR clears the PE bit.</p>
1	RC	0x0	<p>OE Overrun Error</p> <p>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.</p> <p>0: no overrun error 1: overrun error Reading the LSR clears the OE bit.</p>
0	R	0x0	<p>DR Data Ready</p> <p>This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO.</p> <p>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.</p>

### 6.3.6.11. UART Modem Status Register(Default Value: 0x0000\_0000)

Offset: 0x0018			Register Name: UART_MSR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R	0x0	<p>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)</p>
6	R	0x0	<p>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)</p>
5	R	0x0	<p>DSR Line State of Data Set Ready This is used to indicate the current state of the modem control line dsr_n. This bit is the complement of dsr_n. When the Data Set Ready input (dsr_n) is asserted it is an indication that the modem or data set is ready to establish communications with UART. 0: dsr_n input is de-asserted (logic 1) 1: dsr_n input is asserted (logic 0) In Loopback Mode (MCR[4] is set to 1), DSR is the same as MCR[0] (DTR).</p>
4	R	0x0	<p>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).</p>
3	RC	0x0	<p>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.</p>

			<p>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.</p> <p> <b>NOTE</b></p> <p>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.</p>
2	RC	0x0	<p>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.</p>
1	RC	0x0	<p>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).</p> <p> <b>NOTE</b></p> <p>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.</p>
0	RC	0x0	<p>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).</p> <p> <b>NOTE</b></p> <p>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.</p>

### 6.3.6.12. 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.

### 6.3.6.13. UART Status Register(Default Value: 0x0000\_0006)

Offset: 0x007C			Register Name: UART_USR
Bit	Read/Write	Default/Hex	Description
31:5	/	/	/
4	R	0x0	RFF Receive FIFO Full This is used to indicate that the receive FIFO is completely full. 0: Receive FIFO not full 1: Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.
3	R	0x0	RFNE Receive FIFO Not Empty This is used to indicate that the receive FIFO contains one or more entries. 0: Receive FIFO is empty 1: Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.
2	R	0x1	TFE Transmit FIFO Empty This is used to indicate that the transmit FIFO is completely empty. 0: Transmit FIFO is not empty 1: Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty.
1	R	0x1	TFNF Transmit FIFO Not Full This is used to indicate that the transmit FIFO is not full. 0: Transmit FIFO is full 1: Transmit FIFO is not full This bit is cleared when the TX FIFO is full.
0	R	0x0	BUSY UART Busy Bit 0: Idle or inactive 1: Busy

#### 6.3.6.14. UART Transmit FIFO Level Register(Default Value: 0x0000\_0000)

Offset: 0x0080			Register Name: UART_TFL
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8:0	R	0x0	TFL Transmit FIFO Level The bit indicates the number of data entries in the transmit FIFO.

#### 6.3.6.15. UART Receive FIFO Level Register(Default Value: 0x0000\_0000)

Offset: 0x0084			Register Name: UART_RFL
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8:0	R	0x0	RFL Receive FIFO Level The bit indicates the number of data entries in the receive FIFO.

#### 6.3.6.16. UART DMA Handshake Configuration Register(Default Value: 0x0000\_00E5)

Offset: 0x0088			Register Name: UART_HSK
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0xE5	Handshake configuration 0xA5: DMA wait cycle mode 0xE5: DMA handshake mode

#### 6.3.6.17. UART Halt TX Register(Default Value: 0x0000\_0000)

Offset: 0x00A4			Register Name: UART_HALTI
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	PTE The sending of TX_REQ. In DMA1 mode (FIFO on), if PTE is set to 1, when TFL is less than trig, send the DMA request. If PTE is set to 0, when FIFO is empty, send the DMA request. The DMA request will stop when FIFO is full.  In DMA0 mode, if PTE is set to 1 and FIFO is on, when TFL is less than trig, send DMA request. If PTE is set to 1 and FIFO is off, when THRE is empty, send DMA request. If PTE is set to 0, when FIFO is empty, send DMA request.

6	R/W	0x0	DMA_PTE_RX The sending of RX_DRQ. In DMA1 mode, when RFL is more than or equal to trig or receive timeout, send DRQ.  In DMA0 mode, if DMA_PTE_RX is 1 and FIFO is on, when RFL is more than trig, send DRQ. In other cases, once the receive data is valid, send DRQ.
5	R/W	0x0	SIR_RX_INVERT SIR Receiver Pulse Polarity Invert 0: Not invert receiver signal 1: Invert receiver signal
4	R/W	0x0	SIR_TX_INVERT SIR Transmit Pulse Polarity Invert 0: Not invert transmit pulse 1: Invert transmit pulse
3	/	/	/
2	R/WAC	0x0	CHANGE_UPDATE After the user uses HALT[1] to change the baud rate or LCR configuration, write 1 to update the configuration and wait this bit to self-clear to 0 to finish update process. Writing 0 to this bit has no effect. 1: Update trigger, Self clear to 0 when finish update.
1	R/W	0x0	CHCFG_AT_BUSY This is an enable bit for the user to change LCR register configuration and baud rate register (DLH and DLL) when the UART is busy. 1: Enable change when busy
0	R/W	0x0	HALT_TX Halt TX This register is used to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 0 : Halt TX disabled 1 : Halt TX enabled   <b>NOTE</b> <b>If FIFOs are not enabled, the setting has no effect on operation.</b>

#### 6.3.6.18. UART DBG DLL Register(Default Value: 0x0000\_0000)

Offset: 0x00B0			Register Name: UART_DBG_DLL
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	DEBUG DLL

#### 6.3.6.19. UART DBG DLH Register(Default Value: 0x0000\_0000)

Offset: 0x00B4			Register Name: UART_DBG_DLH
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	DEBUG DLH

#### 6.3.6.20. UART RS485 Control and Status Register(Default Value: 0x0000\_0000)

Offset: 0x00C0			Register Name: UART_485_CTL
Bit	Read/Write	Default/Hex	Description
31:7	/	/	Reserved
6	R/W1C	0x0	<p>AAD_ADDR_F</p> <p>In AAD mode, when UART receives an address byte and the byte is the same as RS485_ADDR_MATCH, this bit will be set to 1. If RS485 interrupt is enabled, the RS485 interrupt will arrive.</p> <p>Write 1 to clear this bit and reset the RS485 interrupt.</p>
5	R/W1C	0x0	<p>RS485_ADDR_DET_F</p> <p>This is a flag of the detecting of address bytes. When UART receives an address byte, this bit will be set to 1. If the RS485 Interrupt is enabled, the RS485 interrupt will arrive.</p> <p>1:An address byte is detected 0:No address byte is detected</p> <p>Write 1 to clear this bit and reset the RS485 interrupt.</p>
4	/	/	/
3	R/W	0x0	<p>RX_BF_ADDR</p> <p>In NMM mode, If setting this bit to 1, UART will receive all the bytes into FIFO before receiving an address byte. If setting to 0, it will not.</p> <p>1:Receive 0:Not Receive</p>
2	R/W	0x0	<p>RX_AF_ADDR</p> <p>In NMM mode, if setting this bit to 1, UART will receive all the bytes into FIFO after receiving an address byte. If setting to 0, it will not.</p> <p>1:Receive 0:Not Receive</p>
1:0	R/W	0x0	<p>RS485_SLAVE_MODE_SEL</p> <p>RS485 Slave Mode</p> <p>00: Normal Multidrop Operation(NMM) 01: Auto Address Detection Operation(AAD) 10: Reserved 11: Reserved</p>

### 6.3.6.21. UART RS485 Address Match Register(Default Value: 0x0000\_0000)

Offset: 0x00C4			Register Name: RS485_ADDR_MATCH
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	<p>ADDR_MATCH The matching address uses in AAD mode.</p> <p> <b>NOTE</b> It is only available for AAD.</p>

### 6.3.6.22. UART RS485 Bus Idle Check Register(Default Value: 0x0000\_0000)

Offset: 0x00C8			Register Name: BUS_IDLE_CHK
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	<p>BUS_IDLE_CHK_EN 0: Disable bus idle check function 1: Enable bus idle check function</p>
6	R	0x0	<p>BUS_STATUS The Flag of Bus Status 0:Idle 1:Busy</p>
5:0	R	0x0	<p>ADJ_TIME Bus Idle Time The unit is 8*16*Tclk.</p>

### 6.3.6.23. UART TX Delay Register(Default Value: 0x0000\_0000)

Offset: 0x00CC			Register Name: TX_DLY
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x0	<p>DLY The delay time between the last stop bit and the next start bit. The unit is 16*Tclk. It is used to control the space between two bytes in TX.</p>

## 6.4. SPI

### 6.4.1. Overview

The SPI is a full-duplex, synchronous, serial communication interface which allows rapid data communication with fewer software interrupts. The SPI controller contains one 64x8 bits receiver buffer (RXFIFO) and one 64x8 bits transmit buffer (TXFIFO). It can work at master mode and slave mode.

The SPI has the following features:

- Full-duplex synchronous serial interface
- Master/slave configurable
- 8-bit wide by 64-entry FIFO for both transmit and receive data
- Polarity and phase of the Chip Select (SPI\_SS) and SPI Clock (SPI\_SCLK) are configurable
- Interrupt or DMA support
- Supports mode0, mode1, mode2 and mode3
- Supports 3-wire/4-wire SPI
- Supports programmable serial data frame length: 1 bit to 32 bits
- Supports the SPI NAND flash and SPI NOR flash
- Supports standard SPI, dual-output/dual-input SPI, dual I/O SPI, quad-output/quad-input SPI
- Supports maximum IO rate: 100MHz

### 6.4.2. Block Diagram

Figure 6-18 shows a block diagram of the SPI.

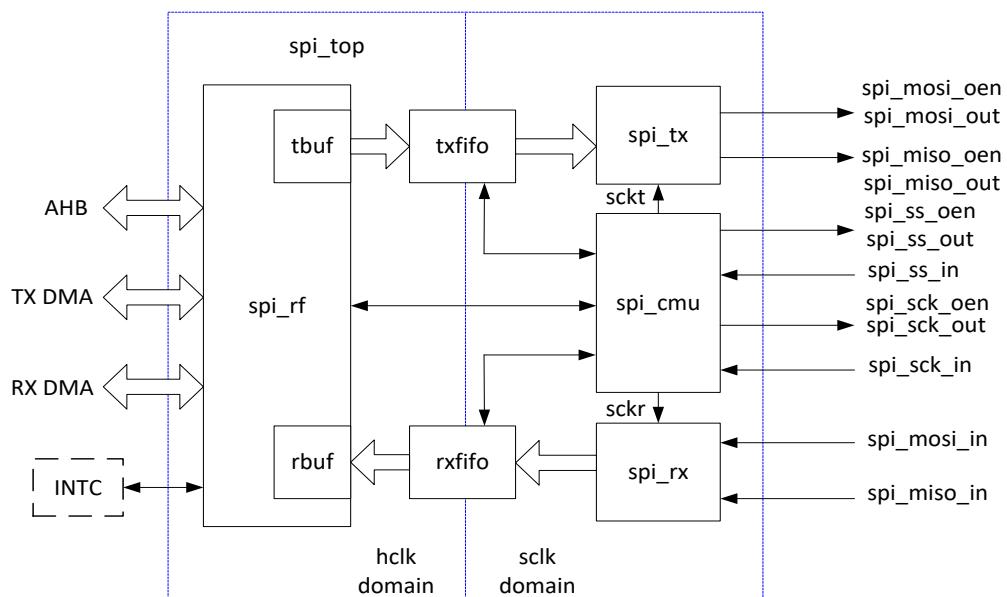


Figure 6- 18. SPI Block Diagram

The SPI comprises with:

spi\_rf: Responsible for implementing the internal register, interrupt and DMA Request.

spi\_tbuf: The data length transmitted from AHB to txfifo is converted into 8bits,then the data is written into the rxfifo.

spi\_rbuf: The block is used to convert the rxfifo data into read data length of AHB.

txfifo, rxfifo: For transmit and receive transfers, data transmitted from the SPI to the external serial device is written into the txfifo; data received from the external serial device into SPI is pushed into the rxfifo.

spi\_cmu: Responsible for implementing SPI bus clock, chip select, internal sample and the generation of transfer clock.

spi\_tx: Responsible for implementing SPI data transfer ,the interface of the internal txfifo and status register.

spi\_rx: Responsible for implementing SPI data receive, the interface of the internal rxfifo and status register.

### 6.4.3. Operations and Functional Descriptions

#### 6.4.3.1. External Signals

Table 6-10 describes the external signals of SPI. MOSI and MISO are bidirectional I/O, when SPI is configured as master device, CLK and CS is output pin; when SPI is configurable as slave device, CLK and CS is input pin. The unused SPI ports are used as General Purpose I/O ports.

Table 6- 10. SPI External Signals

Signal	Description	Type
SPI0-CS	SPI0 Chip Select Signal, Low Active	I/O
SPI0-CLK	SPI0 Clock Signal	I/O
SPI0-MOSI	SPI0 Master Data Out, Slave Data In	I/O
SPI0-MISO	SPI0 Master Data In, Slave Data Out	I/O
SPI0-WP	Write protection and active low or Serial Data Input and Output for Quad Input or Quad Output	I/O
SPI0-HOLD	The HOLD pin is used to temporarily pause serial communication without deselecting or resetting the device. While the HOLD pin is asserted, transitions on the SCK pin and data on the SI pin will be ignored, or Serial Data Input and Output for Quad Input or Quad Output	I/O
SPI1-CS	SPI1 Chip Select Signal, Low Active	I/O
SPI1-CLK	SPI1 Clock Signal	I/O
SPI1-MOSI	SPI1 Master Data Out, Slave Data In	I/O
SPI1-MISO	SPI1 Master Data In, Slave Data Out	I/O

#### 6.4.3.2. Clock Sources

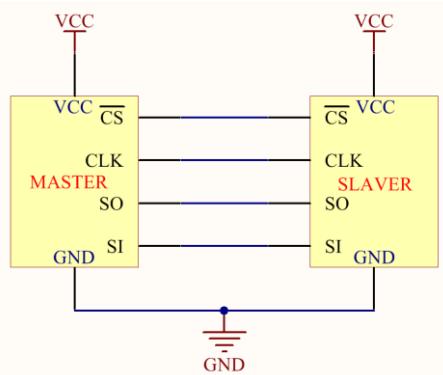
The SPI0 and SPI1 controller get 5 different clock sources, users can select one of them to make SPI clock source. Table 6-11 describes the clock sources for SPI.

**Table 6- 11. SPI Clock Sources**

Clock Sources	Description
OSC24M	24MHz Crystal
PLL_PERIPH0(1X)	Peripheral Clock, default value is 600MHz
PLL_PERIPH1(1X)	Peripheral Clock, default value is 600MHz
PLL_PERIPH0(2X)	Peripheral Clock, default value is 1200MHz
PLL_PERIPH1(2X)	Peripheral Clock, default value is 1200MHz

#### 6.4.3.3. Typical Application

Figure 6-19 shows the application block diagram when the SPI master device is connected to a slave device.

**Figure 6- 19. SPI Application Block Diagram**

#### 6.4.3.4. SPI Transmit Format

The SPI supports 4 different formats for data transfer. Software can select one of the four modes in which the SPI works by setting the bit1(Polarity) and bit0(Phase) of **SPI Transfer Control Register**. The SPI controller master uses the SPI\_SCLK signal to transfer data in and out of the shift register. Data is clocked using any one of four programmable clock phase and polarity combinations.

During Phase 0, Polarity 0 and Phase 1, Polarity 1 operations, output data changes on the falling clock edge and input data is shifted in on the rising edge.

During Phase 1, Polarity 0 and Phase 0, Polarity 1 operations, output data changes on the rising edges of the clock and is shifted in on falling edges.

The POL defines the signal polarity when SPI\_SCLK is in idle state. The SPI\_SCLK is high level when POL is '1' and it is low level when POL is '0'. The PHA decides whether the leading edge of SPI\_SCLK is used for setup or sample data. The leading edge is used for setup data when PHA is '1' and for sample data when PHA is '0'. The four modes are listed in Table 6-12.

Table 6- 12. SPI Transmit Format

SPI Mode	POL	PHA	Leading Edge	Trailing Edge
0	0	0	Rising, Sample	Falling, Setup
1	0	1	Rising, Setup	Falling, Sample
2	1	0	Falling, Sample	Rising, Setup
3	1	1	Falling, Setup	Rising, Sample

Figure 6-20 and Figure 6-21 describe four waveforms for SPI\_SCLK.

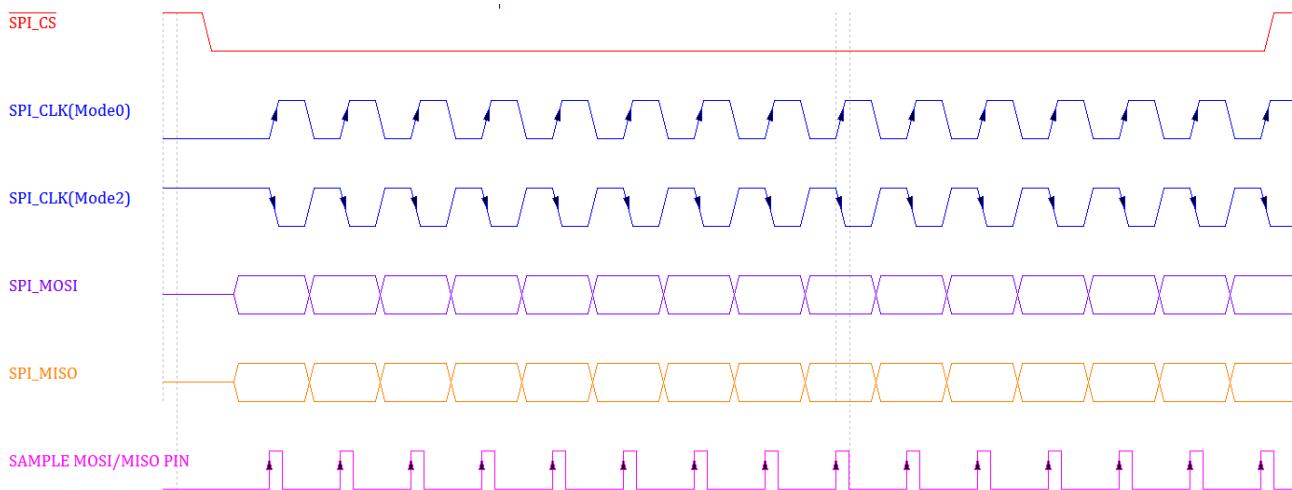


Figure 6- 20. SPI Phase 0 Timing Diagram

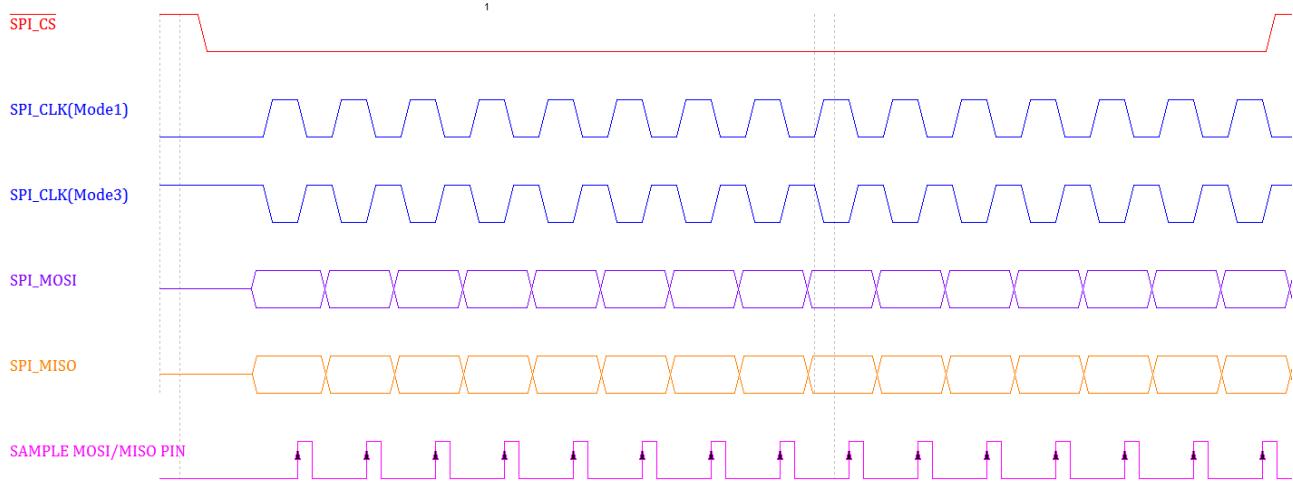


Figure 6- 21. SPI Phase 1 Timing Diagram

#### 6.4.3.5. SPI Master and Slave Mode

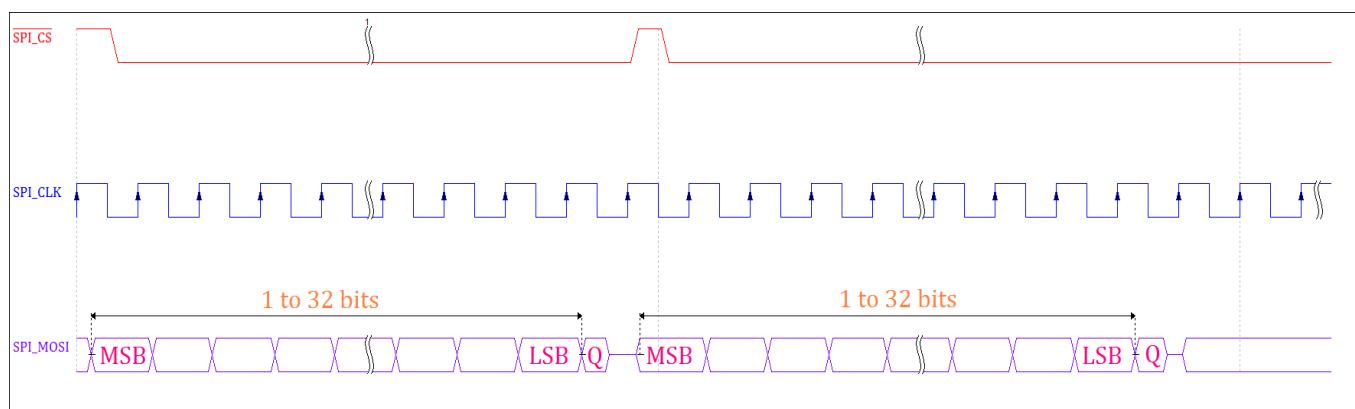
The SPI controller can be configured to a master or slave device. Master mode is selected by setting the **MODE** bit in the **SPI Global Control Register**; slave mode is selected by clearing the the **MODE** bit in the **SPI Global Control Register**.

In master mode, SPI\_CLK is generated and transmitted to external device, and data from the TX FIFO is transmitted on the MOSI pin, the data from slave is received on the MISO pin and sent to RX FIFO. Chip Select(SPI\_SS) is active low signal. SPI\_SS must be set low before data are transmitted or received. SPI\_SS can be selected SPI auto control or software manual control. When using auto control, **SS\_OWNER**(the bit 6 in the **SPI Transfer Control Register**) must be cleared(default value is 0);when using manual control, **SS\_OWNER** must be set, Chip Select level is controlled by **SS\_LEVEL** bit(the bit 7 in the **SPI Transfer Control Register**).

In slave mode, after software selects the **MODE** bit to '0',it waits for master initiate a transaction. When the master assertes 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.

#### 6.4.3.6. 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[1:0])** is equal to 0x2 in the **SPI Bit-Aligned Transfer Configure Register**. And in the 3-wire mode, the input data and the output data use the same single data line. The following figure describes this mode.



**Figure 6- 22. SPI 3-Wire Mode**

#### 6.4.3.7. SPI Dual-Input/Dual-Output and Dual I/O Mode

The dual read mode(SPI x2) is selected when the **DRM**(bit28) is set in the **SPI Master Burst Control Counter Register**. Using the dual mode allows data to be transferred to or from the device at double the rate of standard single mode SPI devices, data can be read at fast speed using two data bits(MOSI and MISO) at a time. The following figure describes the dual-input/dual-output SPI(figure 6-23) and the dual I/O SPI(figure 6-24).

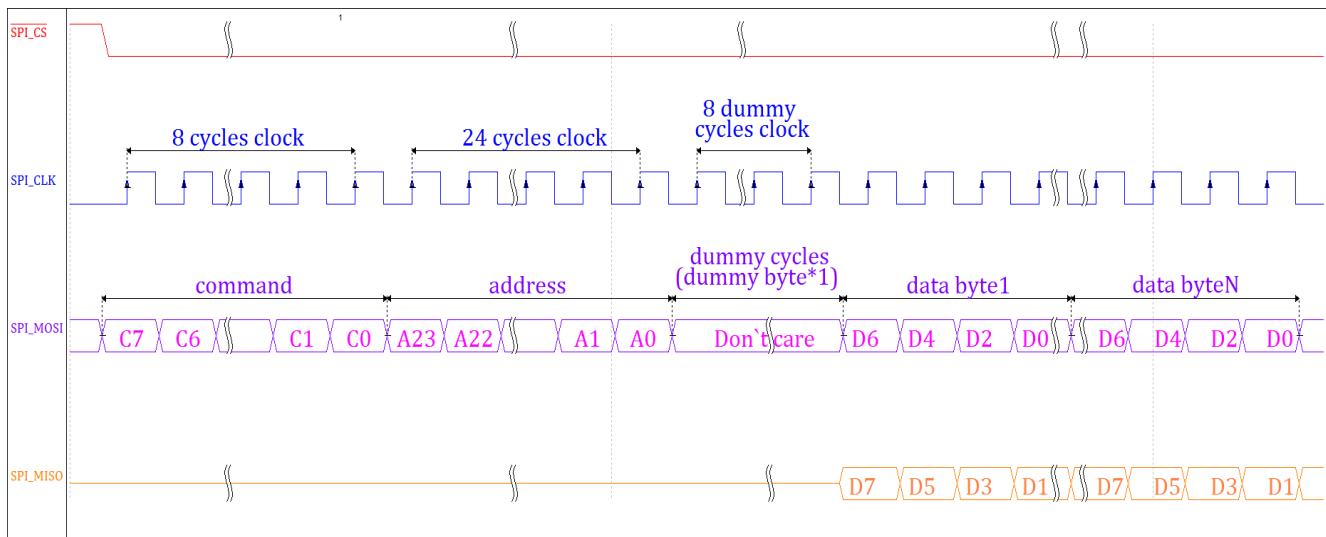


Figure 6- 23. SPI Dual-Input/Dual-Output Mode

In the dual-input/dual-output SPI, the command, address, and the dummy bytes output in unit of a single bit in serial mode through SPI\_MOSI line, only the data bytes are output(write) and input(read) in unit of dual bits through the SPI\_MOSI and SPI\_MISO.

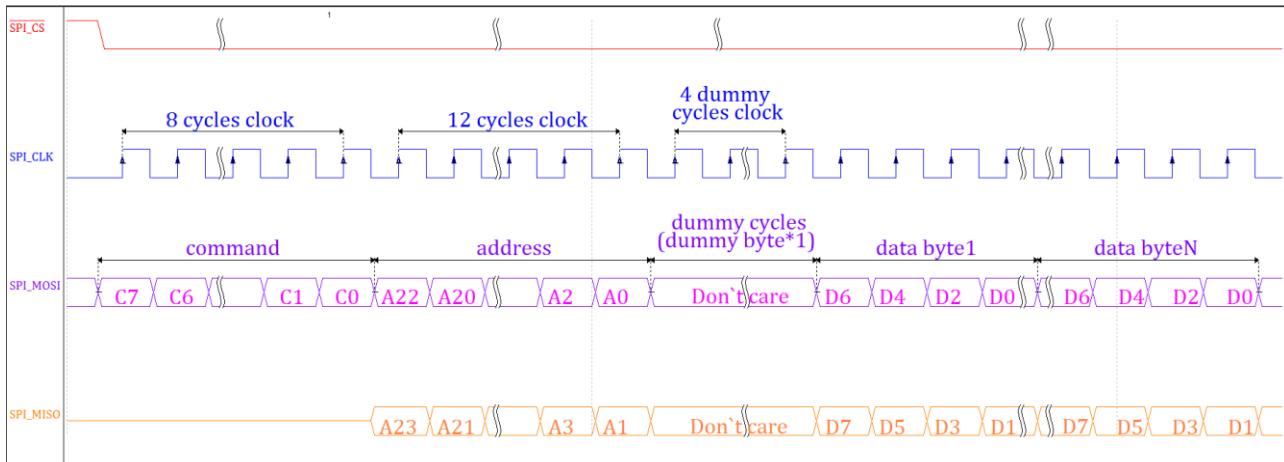
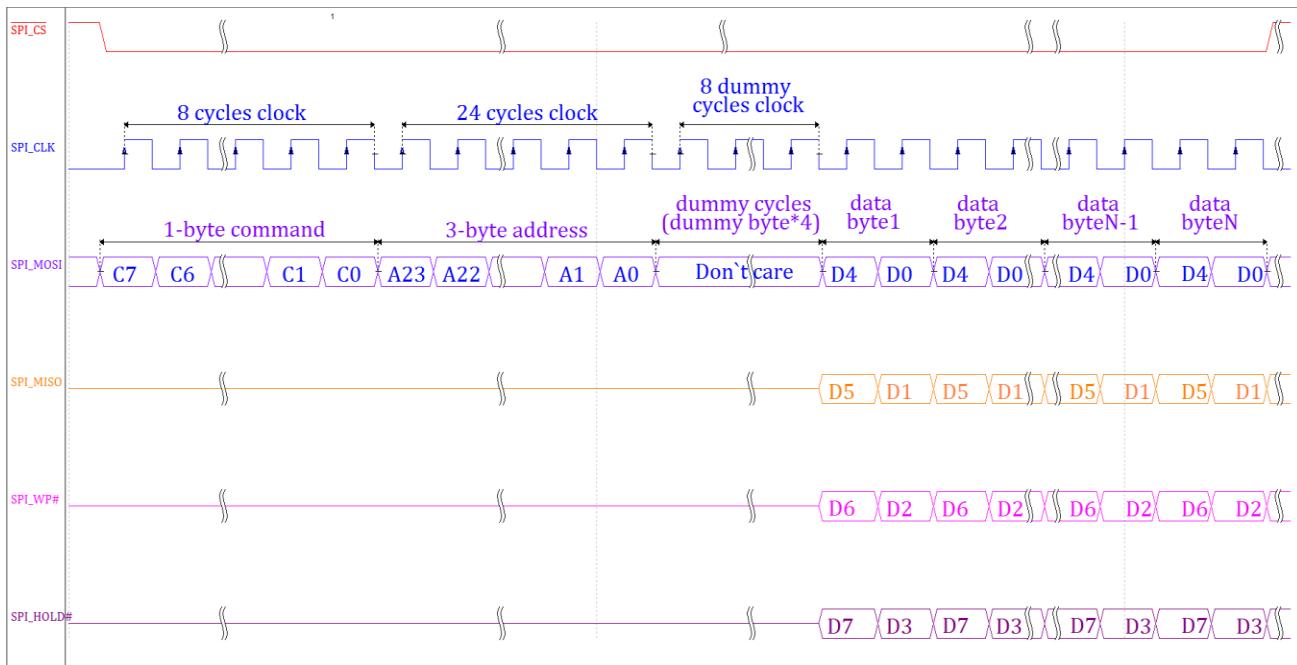


Figure 6- 24. SPI Dual I/O Mode

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.

#### 6.4.3.8. SPI Quad-Input/Quad-Output Mode

The quad read mode(SPI x4) is selected when the **Quad\_EN**(bit29) is set in the **SPI Master Burst Control Counter Register**. Using the quad mode allows data to be transferred to or from the device at 4 times the rate of standard single mode SPI devices, data can be read at fast speed using four data bits(MOSI, MISO, IO2(WP#)and IO3(HOLD#)) at the same time. The following figure describes the quad-input/quad-output SPI.



**Figure 6- 25. 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#.

#### 6.4.4. Programming Guidelines

##### 6.4.4.1. CPU or DMA Operation

The SPI transfers serial data between the processor and external device. CPU and DMA are the two main operational modes for SPI. For each SPI, data is simultaneously transmitted(shifted out serially) and received (shifted in serially). SPI has 2 channels, TX channel and RX channel. TX channel has the path from TX FIFO to external device. RX channel has the path from external device to RX FIFO.

**Write Data:** CPU or DMA must write data on the register SPI\_TXD, data on the register are automatically moved to TX FIFO.

**Read Data:** To read data from RX FIFO, CPU or DMA must access the register SPI\_RXD and data are automatically sent to the register SPI\_RXD.

In CPU or DMA mode, the SPI sends an completed interrupt(the TC bit in SPI Interrupt Status Register) to the processor at the end of each transfer.

##### (1).CPU Mode

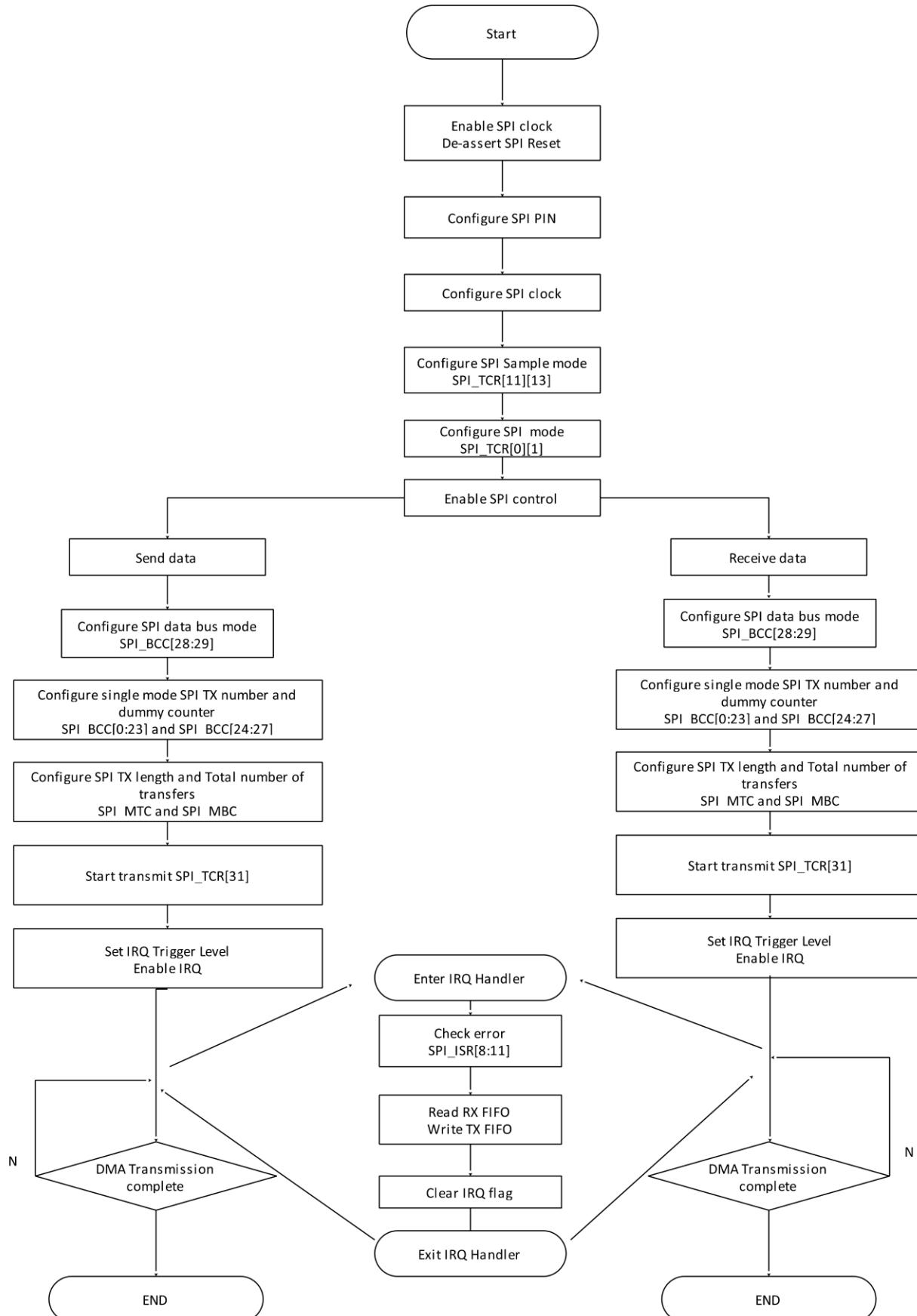


Figure 6- 26. SPI Write/Read Data in CPU Mode

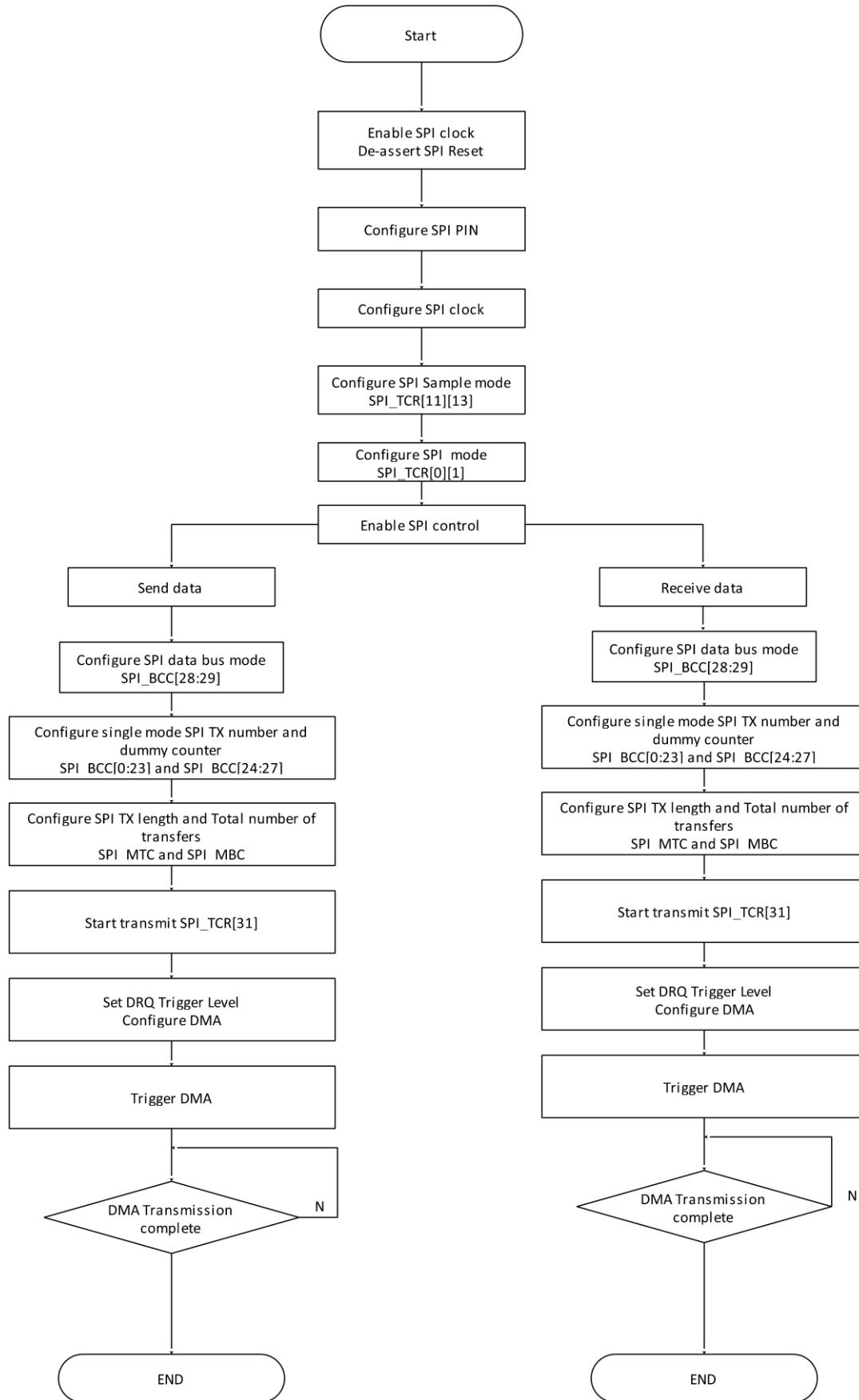


Figure 6- 27. SPI Write/Read Data in DMA Mode

#### 6.4.4.2. Transmit/Receive Burst in Master Mode

In SPI master mode, the transmit and receive burst(byte in unit) are configured before the SPI transfers serial data between the processor and external device. The transmit bursts are written in MWTC(bit[23:0]) of **SPI Master Transmit Counter Register**. The transmit bursts in single mode before automatically sending dummy burst are written in STC(bit[23:0]) of **SPI Master Burst Control Counter Register**. For dummy data, SPI controller can automatically sent before receiving by writing DBC(bit[27:24]) in **SPI Master Burst Control Counter Register**. If users donot use SPI controller to sent dummy data automatically, then the dummy bursts are used as the transmit counters to write together in MWTC(bit[23:0]) of **SPI Master Transmit Counter Register**. In master mode, the total burst numbers are written in MBC(bit[23:0]) of **SPI Master Burst Counter Register**. When all transmit burst and receive burst are transferred, SPI controller will send an completed interrupt, at the same time, SPI controller will clear DBC,MWTC and MBC.

#### 6.4.4.3. SPI Sample Mode and Run Clock Configuration

The SPI controller runs at 3 kHz~100 MHz at its interface to external SPI devices. The internal SPI clock should run at the same frequency as the outgoing clock in master mode. The SPI clock is selected different clock sources, SPI must configure different work mode. There are three work 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 40 MHz or below 40 MHz, SPI can work at normal sample mode or delay half cycle sample mode. When SPI runs over 60 MHz,setting the **SDC** bit in **SPI Transfer Control Register** to ‘1’ makes the internal read sample point with a half cycle delay of SPI\_CLK, which is used in high speed read operation to reduce the error caused by the time delay of SPI\_CLK propagating between master and slave. The different configuration of SPI sample mode shows in Table 6-13.

**Table 6- 13. SPI Sample Mode and Run Clock**

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

#### 6.4.4.4. SPI Error Conditions

If any error conditions occur, hardware will set the corresponding status bits in the **SPI Interrupt Status Register** and stop the transfer. For the SPI controller, the following error scenarios can happen.

##### (1) TX\_FIFO Underrun

TX\_FIFO underrun happens when the CPU/DMA reads from TX FIFO when it is empty. In the case, the SPI controller will end the transaction and flag the error bit along with the TF\_UDF bit in the **SPI Interrupt Status Register**. The SPI controller will generate an interrupt if interrupts are enabled. Software has to clear the error bit and the TF\_UDF bit. To start a new transaction, software has to reset the fifo by writing to the SRST(soft reset) bit in the **SPI Global Controller Register**.

## (2) TX\_FIFO Overflow

TX\_FIFO overflow happens when the CPU/DMA writes into the TX FIFO when it is full. In the case, the SPI controller will end the transaction and flag the error bit along with the TF\_OVF bit in the **SPI Interrupt Status Register**. The SPI controller will generate an interrupt if interrupts are enabled. Software has to clear the error bit and the TF\_OVF bit. To start a new transaction, software has to reset the fifo by writing to the SRST(soft reset) bit in the **SPI Global Controller Register**.

## (3) RX\_FIFO Underrun

RX\_FIFO underrun happens when the CPU/DMA reads from RX FIFO when it is empty. In the case, the SPI controller will end the transaction and flag the error bit along with the RF\_UDF bit in the **SPI Interrupt Status Register**. The SPI controller will generate an interrupt if interrupts are enabled. Software has to clear the error bit and the RF\_UDF bit. To start a new transaction, software has to reset the fifo by writing to the SRST(soft reset) bit in the **SPI Global Controller Register**.

## (4) RX\_FIFO Overflow

RX\_FIFO overflow happens when the CPU/DMA writes into the RX FIFO when it is full. In the case, the SPI controller will end the transaction and flag the error bit along with the RF\_OVF bit in the **SPI Interrupt Status Register**. The SPI controller will generate an interrupt if interrupts are enabled. Software has to clear the error bit and the RF\_OVF bit. To start a new transaction, software has to reset the FIFO by writing to the SRST(soft reset) bit in the **SPI Global Controller Register**.

### 6.4.5. Register List

Module Name	Base Address
SPI0	0x05010000
SPI1	0x05011000

Register Name	Offset	Description
SPI_GCR	0x0004	SPI Global Control Register
SPI_TCR	0x0008	SPI Transfer Control Register
SPI_IER	0x0010	SPI Interrupt Control Register
SPI_ISR	0x0014	SPI Interrupt Status Register
SPI_FCR	0x0018	SPI FIFO Control Register
SPI_FSR	0x001C	SPI FIFO Status Register
SPI_WCR	0x0020	SPI Wait Clock Counter Register
SPI_CCR	0x0024	SPI Clock Rate Control Register
SPI_MBC	0x0030	SPI Burst Counter Register
SPI_MTC	0x0034	SPI Transmit Counter Register
SPI_BCC	0x0038	SPI Burst Control Register
SPI_BATCR	0x003C	SPI Bit-Aligned Transfer Configure Register
SPI_3W_CCR	0x0040	SPI 3Wire Clock Configuration Register

SPI_TBR	0x0044	SPI TX Bit Register
SPI_RBR	0x0048	SPI RX Bit Register
SPI_NDMA_MODE_CTL	0x0088	SPI Normal DMA Mode Control Register
SPI_TXD	0x0200	SPI TX Data Register
SPI_RXD	0x0300	SPI RX Data Register

#### 6.4.6. Register Description

##### 6.4.6.1. SPI Global Control Register(Default Value: 0x0000\_0080)

Offset:0x0004			Register Name: SPI_GCR
Bit	Read/Write	Default/Hex	Description
31	R/WAC	0x0	<p>SRST Soft reset Writing ‘1’ to this bit will clear the SPI controller, and auto clear to ‘0’ when reset operation completes. Writing ‘0’ has no effect.</p>
30:8	/	/	/
7	R/W	0x1	<p>TP_EN Transmit Pause Enable In master mode, it is used to control transmit state machine to stop smart burst sending when RX FIFO is full. 0: Normal operation, ignore RXFIFO status 1: Stop transmit data when RXFIFO full Cannot be written when XCH=1</p>
6:2	/	/	/
1	R/W	0x0	<p>MODE SPI Function Mode Select 0: Slave mode 1: Master mode Cannot be written when XCH=1</p>
0	R/W	0x0	<p>EN SPI Module Enable Control 0: Disable 1: Enable After transforming from bit_mode to byte_mode, it must enable the SPI module again.</p>

##### 6.4.6.2. SPI Transfer Control Register(Default Value: 0x0000\_0087)

Offset: 0x0008			Register Name: SPI_TCR
Bit	Read/Write	Default/Hex	Description

31	R/WAC	0x0	XCH Exchange Burst In master mode it is used to start SPI burst 0: Idle 1: Initiates exchange. Writing "1" to this bit will start the SPI burst, and will auto clear after finishing the bursts transfer specified by BC. Writing "1" to SRST will also clear this bit. Writing '0' to this bit has no effect. Cannot be written when XCH=1.
30:15	/	/	/
14	R/W	0x0	SDDM Sending Data Delay Mode 0:Normal sending 1:Delay sending Set the bit to "1" to make the data that should be sent with a delay of half cycle of SPI_CLK in dual IO mode for SPI mode 0.
13	R/W	0x0	SDM Master Sample Data Mode 0: Delay sample mode 1: Normal sample mode In normal sample mode, SPI master samples the data at the correct edge for each SPI mode; In delay sample mode, SPI master samples data at the edge that is half cycle delayed by the correct edge defined in respective SPI mode.
12	R/W	0x0	FBS First Transmit Bit Select 0: MSB first 1: LSB first Cannot be written when XCH=1.
11	R/W	0x0	SDC Master Sample Data Control Set this bit to '1' to make the internal read sample point with a delay of half cycle of SPI_CLK. It is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK propagating between master and slave. 0: Normal operation, do not delay internal read sample point 1: Delay internal read sample point Cannot be written when XCH=1.
10	R/W	0x0	RPSM Rapids Mode Select Select rapid mode for high speed write. 0: Normal write mode 1: Rapid write mode Cannot be written when XCH=1.
9	R/W	0x0	DDB Dummy Burst Type

			0: The bit value of dummy SPI burst is zero 1: The bit value of dummy SPI burst is one Cannot be written when XCH=1.
8	R/W	0x0	DHB Discard Hash Burst In master mode it controls whether discarding unused SPI bursts 0: Receiving all SPI bursts in BC period 1: Discard unused SPI bursts, only fetching the SPI bursts during dummy burst period. The bursts number is specified by TC. Cannot be written when XCH=1.
7	R/W	0x1	SS_LEVEL When control SS signal manually (SPI_CTRL_REG.SS_CTRL==1), set this bit to '1' or '0' to control the level of SS signal. 0: set SS to low 1: set SS to high Cannot be written when XCH=1.
6	R/W	0x0	SS_OWNER SS Output Owner Select Usually, controller sends SS signal automatically with data together. When this bit is set to 1, software must manually write SPI_CTL_REG.SS_LEVEL to 1 or 0 to control the level of SS signal. 0: SPI controller 1: Software Cannot be written when XCH=1.
5:4	R/W	0x0	SS_SEL SPI Chip Select Select one of four external SPI Master/Slave Devices 00: SPI_SS0 will be asserted 01: SPI_SS1 will be asserted 10: SPI_SS2 will be asserted 11: SPI_SS3 will be asserted Cannot be written when XCH=1.
3	R/W	0x0	SSCTL In master mode, this bit selects the output wave form for the SPI_SSx signal. Only valid when SS_OWNER = 0. 0: SPI_SSx remains asserted between SPI bursts 1: Negate SPI_SSx between SPI bursts Cannot be written when XCH=1.
2	R/W	0x1	SPOL SPI Chip Select Signal Polarity Control 0: Active high polarity (0 = Idle) 1: Active low polarity (1 = Idle) Cannot be written when XCH=1.
1	R/W	0x1	CPOL SPI Clock Polarity Control

			0: Active high polarity (0 = Idle) 1: Active low polarity (1 = Idle) Cannot be written when XCH=1.
0	R/W	0x1	CPHA SPI Clock/Data Phase Control 0: Phase 0 (Leading edge for sample data) 1: Phase 1 (Leading edge for setup data) Cannot be written when XCH=1.

#### 6.4.6.3. SPI Interrupt Control Register(Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: SPI_IER
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13	R/W	0x0	SS_INT_EN SSI Interrupt Enable Chip select signal (SSx) from valid state to invalid state 0: Disable 1: Enable
12	R/W	0x0	TC_INT_EN Transfer Completed Interrupt Enable 0: Disable 1: Enable
11	R/W	0x0	TF_UDR_INT_EN TXFIFO Underrun Interrupt Enable 0: Disable 1: Enable
10	R/W	0x0	TF_OVF_INT_EN TX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
9	R/W	0x0	RF_UDR_INT_EN RXFIFO Underrun Interrupt Enable 0: Disable 1: Enable
8	R/W	0x0	RF_OVF_INT_EN RX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
7	/	/	/
6	R/W	0x0	TF_FUL_INT_EN TX FIFO Full Interrupt Enable 0: Disable 1: Enable

5	R/W	0x0	TX_EMP_INT_EN TX FIFO Empty Interrupt Enable 0: Disable 1: Enable
4	R/W	0x0	TX_ERQ_INT_EN TX FIFO Empty Request Interrupt Enable 0: Disable 1: Enable
3	/	/	/
2	R/W	0x0	RF_FUL_INT_EN RX FIFO Full Interrupt Enable 0: Disable 1: Enable
1	R/W	0x0	RX_EMP_INT_EN RX FIFO Empty Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	RF_RDY_INT_EN RX FIFO Ready Request Interrupt Enable 0: Disable 1: Enable

#### 6.4.6.4. 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 Underrun This bit is set when if the TXFIFO is underrun. Writing 1 to this bit clears it. 0: TXFIFO is not underrun 1: TXFIFO is underrun

10	R/W1C	0x0	<b>TF_OVF</b> 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	<b>RX_UDF</b> RXFIFO Underrun When set, this bit indicates that RXFIFO has underrun. Writing 1 to this bit clears it.
8	R/W1C	0x0	<b>RX_OVF</b> RXFIFO Overflow When set, this bit indicates that RXFIFO has overflowed. Writing 1 to this bit clears it. 0: RXFIFO is available 1: RXFIFO is overflowed
7	/	/	/
6	R/W1C	0x0	<b>TX_FULL</b> 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	<b>TX_EMP</b> 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	<b>TX_READY</b> TXFIFO Ready 0: TX_WL > TX_TRIG_LEVEL 1: TX_WL <= TX_TRIG_LEVEL This bit is set any time if TX_WL <= TX_TRIG_LEVEL. Writing "1" to this bit clears it. TX_WL is the water level of TXFIFO.
3	/	/	/
2	R/W1C	0x0	<b>RX_FULL</b> 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	<b>RX_EMP</b> 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	<b>RX_RDY</b> RXFIFO Ready

			<p>0: RX_WL &lt; RX_TRIG_LEVEL 1: RX_WL &gt;= RX_TRIG_LEVEL</p> <p>This bit is set any time if RX_WL &gt;= RX_TRIG_LEVEL. Writing "1" to this bit clears it. RX_WL is the water level of RXFIFO.</p>
--	--	--	--

#### 6.4.6.5. SPI FIFO Control Register(Default Value: 0x0040\_0001)

Offset: 0x0018			Register Name: SPI_FCR
Bit	Read/Write	Default/Hex	Description
31	R/WAC	0x0	<p>TX_FIFO_RST TX FIFO Reset Writing '1' to this bit will reset the control portion of the TX FIFO and auto clear to '0' when completing reset operation, writing to '0' has no effect.</p>
30	R/W	0x0	<p>TF_TEST_ENB TX Test Mode Enable 0: Disable 1: Enable</p> <p> <b>NOTE</b> <b>In normal mode, TX FIFO can only be read by SPI controller, writing '1' to this bit will switch TX FIFO read and write function to AHB bus. This bit is used to test the TX FIFO, do not set in normal operation and do not set RF_TEST and TF_TEST at the same time.</b></p>
29:25	/	/	/
24	R/W	0x0	<p>TF_DRQ_EN TX FIFO DMA Request Enable 0: Disable 1: Enable</p>
23:16	R/W	0x40	<p>TX_TRIG_LEVEL TX FIFO Empty Request Trigger Level</p>
15	R/WAC	0x0	<p>RF_RST RXFIFO Reset Writing '1' to this bit will reset the control portion of the receiver FIFO, and auto clear to '0' when completing reset operation, writing '0' to this bit has no effect.</p>
14	R/W	0x0	<p>RF_TEST RX Test Mode Enable 0: Disable 1: Enable</p> <p> <b>NOTE</b> <b>In normal mode, RX FIFO can only be written by SPI controller, writing '1' to this bit will switch RX FIFO read and write function to AHB bus. This bit is used to test the RX FIFO, do not set in normal operation and do not set</b></p>

			<b>RF_TEST and TF_TEST at the same time.</b>
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

#### 6.4.6.6. SPI FIFO Status Register(Default Value: 0x0000\_0000)

Offset: 0x001C			Register Name: SPI_FSR
Bit	Read/Write	Default/Hex	Description
31	R	0x0	TB_WR TX FIFO Write Buffer Write Enable
30:28	R	0x0	TB_CNT TX FIFO Write Buffer Counter These bits indicate the number of words in TX FIFO Write Buffer
27:24	/	/	/
23:16	R	0x0	TF_CNT TX FIFO Counter These bits indicate the number of words in TX FIFO 0: 0 byte in TX FIFO 1: 1 byte in TX FIFO ... 64: 64 bytes in TX FIFO other: Reserved
15	R	0x0	RB_WR RX FIFO Read Buffer Write Enable
14:12	R	0x0	RB_CNT RX FIFO Read Buffer Counter These bits indicate the number of words in RX FIFO Read Buffer
11:8	/	/	/
7:0	R	0x0	RF_CNT RX FIFO Counter These bits indicate the number of words in RX FIFO 0: 0 byte in RX FIFO 1: 1 byte in RX FIFO ... 64: 64 bytes in RX FIFO other: Reserved

#### 6.4.6.7. SPI Wait Clock Register(Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: SPI_WCR
Bit	Read/Write	Default/Hex	Description
31:20	/	/	/
19:16	R/W	0x0	<p>SWC Dual mode direction switch wait clock counter (for master mode only). Cannot be written when XCH=1. 0: No wait states inserted n: n SPI_SCLK wait states inserted</p> <p> <b>NOTE</b> These bits control the number of wait states to be inserted before start dual data transfer in dual SPI mode. The SPI module counts SPI_SCLK by SWC for delaying next word data transfer.</p>
15:0	R/W	0x0	<p>WCC Wait Clock Counter (In master mode) These bits control the number of wait states to be inserted in data transfers. The SPI module counts SPI_SCLK by WCC for delaying next word data transfer. 0: No wait states inserted N: N SPI_SCLK wait states inserted</p>

#### 6.4.6.8. SPI Clock Control Register(Default Value: 0x0000\_0002)

Offset: 0x0024			Register Name: SPI_CCR
Bit	Read/Write	Default/Hex	Description
31:13	/	/	/
12	R/W	0x0	<p>DRS Divide Rate Select (Master Mode Only) 0: Select Clock Divide Rate 1 1: Select Clock Divide Rate 2 Can't be written when XCH=1.</p>
11:8	R/W	0x0	<p>CDR1_M Clock Divide Rate 1 (Master Mode Only) The SPI_SCLK is determined according to the following equation: SPI_CLK = Source_CLK / (2^CDR1_M). Can't be written when XCH=1.</p>
7:0	R/W	0x2	<p>CDR2_N Clock Divide Rate 2 (Master Mode Only) The SPI_SCLK is determined according to the following equation: SPI_CLK = Source_CLK / (2*(CDR2_N + 1)). Can't be written when XCH=1.</p>

#### 6.4.6.9. 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	<p>MBC Master Burst Counter In master mode, this field specifies the total burst number. 0: 0 burst 1: 1 burst ... N: N bursts</p> <p> <b>NOTE</b></p> <ul style="list-style-type: none"> <li>• Total transfer data, include the TXD, RXD and dummy burst.</li> <li>• Can't be written when XCH=1.</li> </ul>

#### 6.4.6.10. 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	<p>MWTC Master Write Transmit Counter In master mode, this field specifies the burst number that should be sent to TXFIFO before automatically sending dummy burst. For saving bus bandwidth, the dummy burst (all zero bits or all one bits) is sent by SPI Controller automatically. 0: 0 burst 1: 1 burst ... N: N bursts Can't be written when XCH=1.</p>

#### 6.4.6.11. 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	<p>Quad_EN Quad_Mode_EN 0: Quad mode disable 1: Quad mode enable</p>

			 <b>NOTE</b> Quad mode includes Quad-Input and Quad-Output.
28	R/W	0x0	DRM Master Dual Mode RX Enable 0: RX use single-bit mode 1: RX use dual mode Cannot be written when XCH=1; It is only valid when Quad_Mode_EN=0.
27:24	R/W	0x0	DBC Master Dummy Burst Counter In master mode, this field specifies the burst number that should be sent before receive in dual SPI mode. The data does not care by the device. 0: 0 burst 1: 1 burst ... N: N bursts Cannot be written when XCH=1
23:0	R/W	0x0	STC Master Single Mode Transmit Counter In master mode, this field specifies the burst number that should be sent in single mode before automatically sending dummy burst. This is the first transmit counter in all bursts. 0: 0 burst 1: 1 burst ... N: N bursts Cannot be written when XCH=1

#### 6.4.6.12. SPI Bit-Aligned Transfer Configure Register(Default Value: 0x0000\_00A0)

Offset: 0x0040			Register Name: SPI_BATC
Bit	Read/Write	Default/Hex	Description
31	R/WAC	0x0	TCE Transfer Control Enable In master mode, it is used to start to transfer the serial bits frame, it is only valid when <b>Work Mode Select==0x10/0x11</b> . 0: Idle 1: Initiates transfer Writing “1” to this bit will start to transfer serial bits frame(the value comes from the <b>SPI TX Bit Register</b> or <b>SPI RX Bit Register</b> ), and will auto clear after the bursts transfer completely. Writing ‘0’ to this bit has no effect.
30	R/W	0x0	MSMS Master Sample Standard 0: Delay Sample Mode

			1: Standard Sample Mode In Standard Sample Mode, SPI master samples the data at the standard rising edge of SCLK for each SPI mode; In Delay Sample Mode, SPI master samples data at the edge that is half cycle delayed by the standard rising edge of SCLK defined in respective SPI mode.
29:26	/	/	/
25	R/W1C	0x0	TBC Transfer Bits Completed When set, this bit indicates that the last bit of the serial data frame in <b>SPI TX Bit Register</b> (or <b>SPI RX Bit Register</b> ) has been transferred completely. Writing 1 to this bit clears it. 0: Busy 1: Transfer Completed It is only valid when <b>Work Mode Select==0x10/0x11</b> .
24	R/W	0x0	TBC_INT_EN Transfer Bits Completed Interrupt Enable 0: Disable 1: Enable It is only valid when <b>Work Mode Select==0x10/0x11</b> .
23:22	/	/	/
21:16	R/W	0x00	Configure the length of serial data frame(burst) of RX 000000: 0bit 000001: 1bit ... 100000: 32bits Other values: reserved It is only valid when <b>Work Mode Select==0x10/0x11</b> , and cannot be written when TCE=1.
15:14	/	/	/
13:8	R/W	0x00	Configure the length of serial data frame(burst) of TX 000000: 0bit 000001: 1bit ... 100000: 32bits Other values: reserved It is only valid when <b>Work Mode Select==0x10/0x11</b> , and cannot be written when TCE=1.
7	R/W	0x1	SS_LEVEL When control SS signal manually , set this bit to '1' or '0' to control the level of SS signal. 0: Set SS to low 1: Set SS to high It is only valid when <b>Work Mode Select==0x10/0x11</b> , and only <b>work in Mode0</b> , cannot be written when TCE=1.
6	R/W	0x0	SS_OWNER

			SS Output Owner Select Usually, controller sends SS signal automatically with data together. When this bit is set to 1, software must manually write SPI_CTL_REG.SS_LEVEL to 1 or 0 to control the level of SS signal. 0: SPI controller 1: Software It is only valid when <b>Work Mode Select==0x10/0x11</b> , and only <b>work in Mode0</b> , cannot be written when TCE=1.
5	R/W	0x1	SPOL SPI Chip Select Signal Polarity Control 0: Active high polarity (0 = Idle) 1: Active low polarity (1 = Idle) It is only valid when <b>Work Mode Select==0x10/0x11</b> , and only <b>work in Mode0</b> , cannot be written when TCE=1.
4	/	/	/
3:2	R/W	0x0	SS_SEL SPI Chip Select Select one of four external SPI Master/Slave Devices 00: SPI_SS0 will be asserted 01: SPI_SS1 will be asserted 10: SPI_SS2 will be asserted 11: SPI_SS3 will be asserted It is only valid when <b>Work Mode Select= =0x10/0x11</b> , and only <b>work in Mode0</b> , cannot be written when TCE=1.
1:0	R/W	0x0	Work Mode Select 00: Data frame is byte aligned in standard SPI, dual-output/dual input SPI, dual IO SPI and quad-output/quad-input SPI. 01: Reserved 10: Data frame is bit aligned in 3-wire SPI 11: Data frame is bit aligned in standard SPI

#### 6.4.6.13. 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**.

#### 6.4.6.14. SPI TX Bit Register(Default Value: 0x0000\_0000)

Offset: 0x0048			Register Name: SPI_TBR
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	VTB The Value of the Transmit Bits This register is used to store the value of the transmitted serial data frame. In the process of transmission, the LSB is transmitted first.


**NOTE**

This register is only valid when **Work Mode Select==0x10/0x11**.

#### 6.4.6.15. SPI RX Bit Register(Default Value: 0x0000\_0000)

Offset: 0x004C			Register Name: SPI_RBR
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	VRB The Value of the Receive Bits This register is used to store the value of the received serial data frame. In the process of transmission, the LSB is transmitted first.


**NOTE**

This register is only valid when **Work Mode Select==0x10/0x11**.

#### 6.4.6.16. SPI Normal DMA Mode Control Register(Default Value: 0x0000\_00E5)

Offset: 0x0088			Register Name: NDFC_NDMA_MODE_CTL
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:6	R/W	0x11	00:dma_active is low 01:dma_active is high 10:dma_active is controlled by dma_request(DRQ) 11:dma_active is controlled by controller
5	R/W	0x1	0: active fall do not care ack 1: active fall must after detect ack is high
4:0	R/W	0x05	Delay Cycles The counts of hold cycles from DMA last signal high to dma_active high

#### 6.4.6.17. 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	<p>TDATA Transmit Data</p> <p>This register can be accessed in byte, half-word or word unit by AHB. In byte accessing method, if there are rooms in TXFIFO, one burst data is written to TXFIFO and the depth is increased by 1. In half-word accessing method, two SPI burst data are written and the TXFIFO depth is increased by 2. In word accessing method, four SPI burst data are written and the TXFIFO depth is increased by 4.</p> <p> <b>NOTE</b></p> <p><b>This address is writable-only if TF_TEST is '0', and if TF_TEST is set to '1', this address is readable and writable to test the TX FIFO through the AHB bus.</b></p>

#### 6.4.6.18. SPI RX Data Register(Default Value: 0x0000\_0000)

Offset: 0x0300			Register Name: SPI_RXD
Bit	Read/Write	Default/Hex	Description
31:0	R	0x0	<p>RDATA Receive Data</p> <p>This register can be accessed in byte, half-word or word unit by AHB. In byte accessing method, if there are data in RXFIFO, the top word is returned and the RXFIFO depth is decreased by 1. In half-word accessing method, two SPI bursts are returned and the RXFIFO depth is decreased by 2. In word accessing method, the four SPI bursts are returned and the RXFIFO depth is decreased by 4.</p> <p> <b>NOTE</b></p> <p><b>This address is readable-only if RF_TEST is '0', and if RF_TEST is set to '1', this address is readable and writable to test the RX FIFO through the AHB bus.</b></p>

## 6.5. USB2.0 OTG

### 6.5.1. Overview

The USB2.0 OTG is a dual-role device controller, which supports both device and host functions which can also be configured as a Host-only or Device-only controller, fully compliant with the USB2.0 Specification. It can support high-speed (HS, 480 Mbit/s), full-speed (FS, 12 Mbit/s), and low-speed (LS, 1.5 Mbit/s) transfers in Host mode. It can support high-speed (HS, 480 Mbit/s), and full-speed (FS, 12 Mbit/s) in Device mode. Standard USB transceiver can be used through its UTMI+PHY Level3 interface. The UTMI+PHY interface is bidirectional with 8-bit data bus. For saving CPU bandwidth, USB-OTG DMA interface can support external DMA controller to take care of the data transfer between the memory and USB-OTG FIFO. The USB-OTG core also supports USB power saving functions.

The USB2.0 OTG has the following features:

- Complies with USB2.0 Specification
- Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s), and Low-Speed (LS, 1.5 Mbit/s) in Host mode
- Supports High-Speed (HS, 480 Mbit/s), Full-Speed (FS, 12 Mbit/s) in Device mode
- Supports the UTMI+ Level 3 interface. The 8-bit bidirectional data buses are used
- Supports bi-directional endpoint0 for Control transfer
- Supports up to 8 User-Configurable Endpoints for Bulk, Isochronous and Interrupt bi-directional transfer
- Supports up to (4KB+64Bytes) FIFO for all EPs (including EP0)
- Supports High-Bandwidth Isochronous & Interrupt transfers
- Automated splitting/combining of packets for Bulk transfers
- Supports point-to-point and point-to-multipoint transfer in both Host and Peripheral mode
- Includes automatic ping capabilities
- Soft connect/disconnect function
- Performs all transaction scheduling in hardware
- Power Optimization and power management capabilities
- Includes interface to an external Normal DMA controller for every EPs
- Device and host controller share a 4K sram and a physical PHY

### 6.5.2. Block Diagram

Figure 6-28 shows the block diagram of USB2.0 OTG Controller.

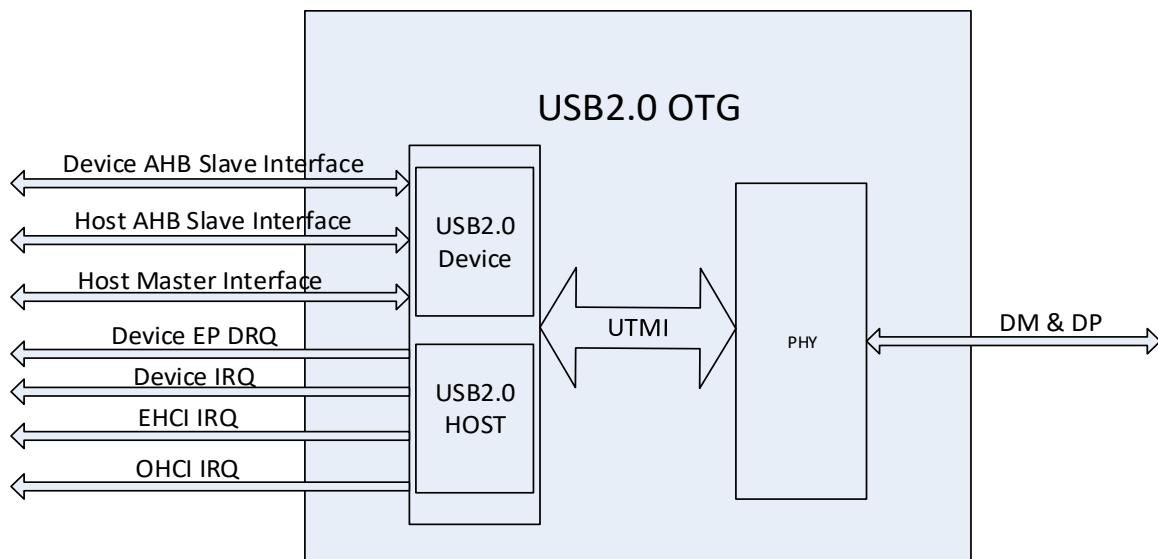


Figure 6- 28. USB2.0 OTG Controller Block Diagram

### 6.5.3. Operations and Functional Descriptions

#### 6.5.3.1. External Signals

Table 6- 14. USB2.0 OTG External Signals

Signal	Description	Type
USBO-DP	USB2.0 OTG differential signal positive	AI/O
USBO-DM	USB2.0 OTG differential signal negative	AI/O

#### 6.5.3.2. Clock and Reset

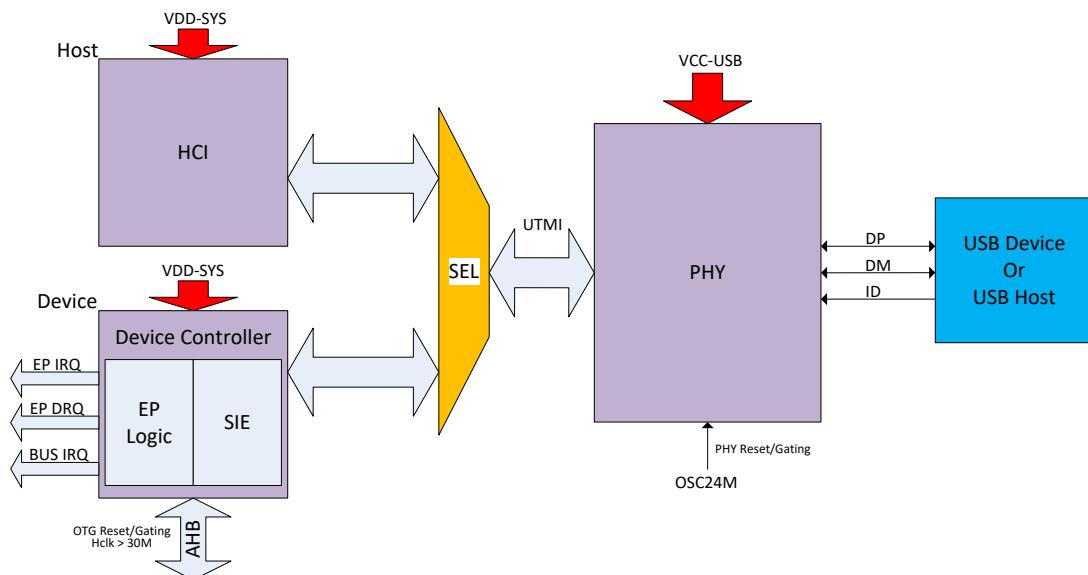


Figure 6- 29. USB2.0 OTG Clock and Reset Description

## 6.6. Port Controller

### 6.6.1. Overview

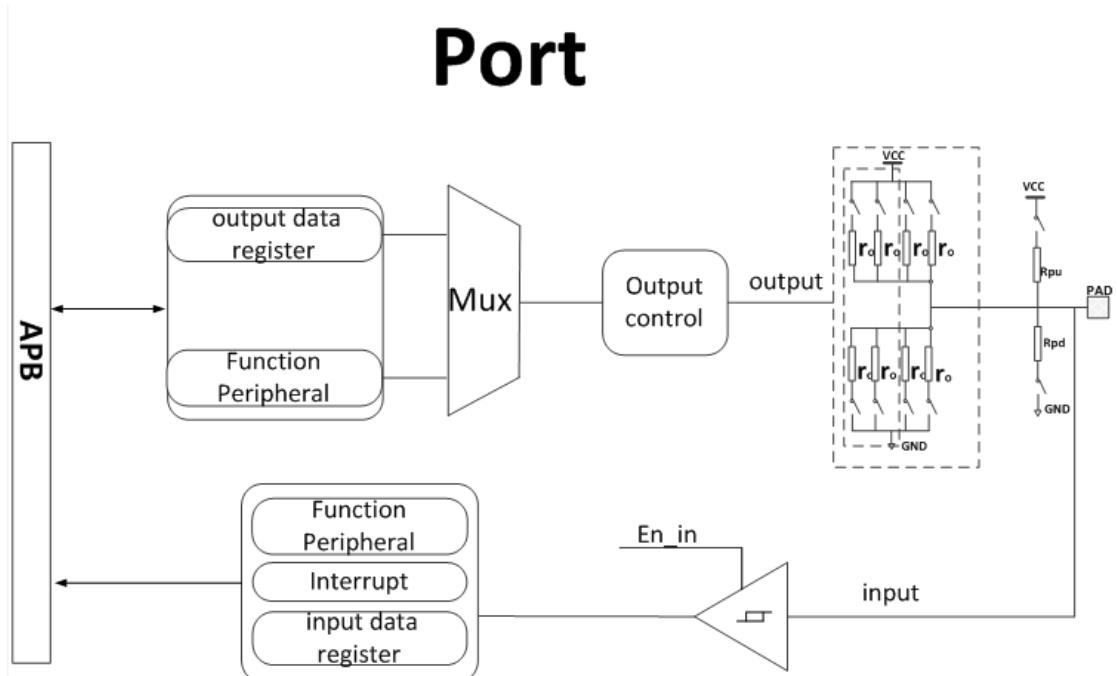
The Port Controller can be configured with multi-functional input/output pins. All these ports can be configured as GPIO only if multiplexed functions not used. The total 4 group external PIO interrupt sources are supported and interrupt mode can be configured by software.

The Port Controller has the following features:

- 5 ports(PB,PC,PE,PG,PH)
- Software control for each signal pin
- GPIO peripheral can produce interrupt
- Pull-up/Pull-down/no-Pull register control
- Control the direction of every signal
- 4 drive strengths in each operating mode
- Up to 46 interrupts
- Configurable interrupt edges

### 6.6.2. Block Diagram

The block diagram of port controller is shown in Figure 6-30.



**Figure 6- 30. Port Controller Block Diagram**

Port controller consists of digital part(GPIO, external interface) and IO analog part(output buffer, dual pull down, pad, etc). Digital part can select output interface by MUX switch; analog part can configure pull up/down, buffer strength.

When executing GPIO read state, the port controller reads the current level of pin into internal register bus. When not executing GPIO read state, external pin and internal register bus is off-status, that is high-impedance.

### 6.6.3. Operations and Functional Descriptions

#### 6.6.3.1. Multi-function Port Table

The R328-S3 includes 53 multi-functional input/output port pins. There are 5 ports as listed below:

**Table 6- 15. Multi-function Port Table**

Port Name	Number of Pins	Input Driver	Output Driver	Multiplex Pins	Power
PB	14	Schmitt	CMOS	DMIC/UART/JTAG/I2S/PWM/TWI/LEDC/ PB-EINT	3.3V
PC	7	Schmitt	CMOS	SPI	1.8V/3.3V
PE	7	Schmitt	CMOS	I2S/OWA/LEDC/UART/ PLL-LOCK-DBG/PE-EINT	1.8V/3.3V
PG	15	Schmitt	CMOS	SDC/UART/BIST/I2S/PG-EINT	1.8V/3.3V
PH	10	Schmitt	CMOS	TWI/UART/SPI/CPU-CUR-W/LEDC/OWA/ PH-EINT	3.3V

#### 6.6.3.2. Port Function

Port Controller supports 5 GPIOs, every GPIO can configure as Input, Output, Function Peripheral, IO disable or Interrupt function. The configuration instruction of every function is as follows.

**Table 6- 16. Port Function**

	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	Trigger	/	X	X

/: non-configure, configuration is invalid

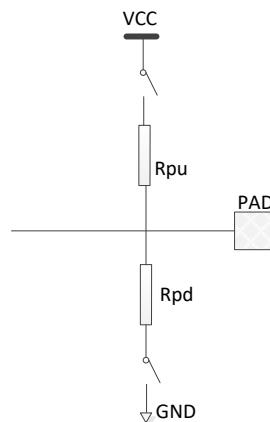
Y: configure

X: Select configuration according to actual situation

N: Forbid to configure

#### 6.6.3.3. Pull Up/Down and High-Impedance Logic

Each IO pin can configure the internal pull-up/down function or high-impedance.



**Figure 6- 31. Pull up/down Logic**

High-impedance, the output is float state, all buffer is off, the level is decided by external high/low level. When high-impedance, software configures the switch on Rpu and Rpd as off ,and the multiplexing function of IO is set as IO disable or input by software.

Pull-up, an uncertain signal is pulled high by a resistance, the resistance has current-limiting function. When pulling up, the switch on Rpu is breakover by software configuration, IO is pulled up to VCC by Rpu.

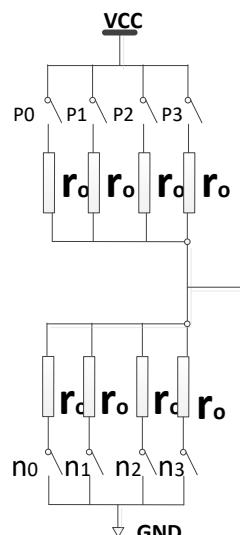
Pull-down, an uncertain signal is pulled low by a resistance. When pulling down, the switch on Rpd is breakover by software configuration, IO is pulled down to GND by Rpd.

The pull-up/down of each IO is weak pull-up/down, the pull-up/down resistance of PC3 is  $20k\Omega \pm 50\%$ , other pull-up/down resistances of GPIO are  $100k\Omega \pm 50\%$ .

The setting of pull-down,pull-up,high-impedance is decided by external circuit.

#### 6.6.3.4. Buffer Strength

Each IO can be set as different buffer strength.The IO buffer diagram is as follows.



**Figure 6- 32. IO Buffer Strength Diagram**

When output high level, the n0,n1,n2,n3 of NMOS is off, the p0,p1,p2,p3 of PMOS is on. When buffer strength is set to 0(buffer strength is weakest), only p0 is on, the output impedance is maximum ,the impedance value is r0. When buffer strength is set to 1, only p0 and p1 is on, the output impedance is equivalent to two r0 in parallel, the impedance value is r0/2. When buffer strength is 2, only p0,p1 and p2 is on, the output impedance is equivalent to three r0 in parallel, the impedance value is r0/3. When buffer strength is 3, p0,p1,p2 and p3 is on, the output impedance is equivalent to four r0 in parallel, the impedance value is r0/4.

When output low level, the p0,p1,p2,p3 of PMOS is off, the n0,n1,n2,n3 of NMOS is on. When buffer strength is set to 0(buffer strength is weakest), only n0 is on, the output impedance is maximum ,the impedance value is r0. When buffer strength is set to 1, only n0 and n1 is on, the output impedance is equivalent to two r0 in parallel, the impedance value is r0/2. When buffer strength is 2, only n0,n1 and n2 is on, the output impedance is equivalent to three r0 in parallel, the impedance value is r0/3. When buffer strength is 3, n0,n1,n2 and n3 is on, the output impedance is equivalent to four r0 in parallel, the impedance value is r0/4.

When GPIO is set to input or interrupt function, between output driver circuit and port is unconnected, driver configuration is invalid.



#### NOTE

The typical value of r0 is 180Ω.

#### 6.6.3.5. Interrupt

Each group IO has independent interrupt number.IO within group uses one interrupt number, when one IO generates interrupt, Port Controller sent interrupt request to GIC. External Interrupt Status Register is used to query which IO generates interrupt.

Interrupt trigger of GPIO supports the following trigger types.

- Positive Edge : When low level changes to high level, the interrupt will generate. No matter how long high level keeps, the interrupt generates only once.
- Negative Edge: When high level changes to low level, the interrupt will generate. No matter how long low level keeps, the interrupt generates only once.
- High Level : Just keep high level and the interrupt will always generate.
- Low Level : Just keep low level and the interrupt will always generate.
- Double Edge : Positive and negative edge.

External Interrupt Configure Register is used to configure trigger type.

GPIO interrupt supports hardware debounce function by setting External Interrupt Debounce Register. Sample trigger signal using lower sample clock, to reach the debounce effect because of the dither frequency of signal is higher than sample frequency.

Set sample clock source by PIO\_INT\_CLK\_SELECT and prescale factor by DEB\_CLK\_PRE\_SCALE.

#### 6.6.4. Register List

Module Name	Base Address
GPIO	0x0300B000

Register Name	Offset	Description
Pn_CFG0	n*0x0024+0x00	Port n Configure Register 0(n =1,2,4,6,7)
Pn_CFG1	n*0x0024+0x04	Port n Configure Register 1(n =1,2,4,6,7)
Pn_CFG2	n*0x0024+0x08	Port n Configure Register 2(n =1,2,4,6,7)
Pn_CFG3	n*0x0024+0x0C	Port n Configure Register 3(n =1,2,4,6,7)
Pn_DAT	n*0x0024+0x10	Port n Data Register(n =1,2,4,6,7)
Pn_DRV0	n*0x0024+0x14	Port n Multi-Driving Register 0(n =1,2,4,6,7)
Pn_DRV1	n*0x0024+0x18	Port n Multi-Driving Register 1(n =1,2,4,6,7)
Pn_PUL0	n*0x0024+0x1C	Port n Pull Register 0(n =1,2,4,6,7)
Pn_PUL1	n*0x0024+0x20	Port n Pull Register 1(n =1,2,4,6,7)
Pn_INT_CFG0	0x0200+n*0x20+0x00	PIO Interrupt Configure Register 0(n =1,2,6,7)
Pn_INT_CFG1	0x0200+n*0x20+0x04	PIO Interrupt Configure Register 1(n =1,2,6,7)
Pn_INT_CFG2	0x0200+n*0x20+0x08	PIO Interrupt Configure Register 2(n =1,2,6,7)
Pn_INT_CFG3	0x0200+n*0x20+0x0C	PIO Interrupt Configure Register 3(n =1,2,6,7)
Pn_INT_CTL	0x0200+n*0x20+0x10	PIO Interrupt Control Register(n =1,2,6,7)
Pn_INT_STA	0x0200+n*0x20+0x14	PIO Interrupt Status Register(n =1,2,6,7)
Pn_INT_DEB	0x0200+n*0x20+0x18	PIO Interrupt Debounce Register(n =1,2,6,7)

#### 6.6.5. Register Description

##### 6.6.5.1. PB Configure Register 0 (Default Value: 0x7777\_4444)

Offset: 0x0024			Register Name: PB_CFG0
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30:28	R/W	0x7	PB7_SELECT 000:Input 001:Output 010:reserved 011:PWM7 100:I2S0-DOUT3 101:I2S0-DIN2 110:PB-EINT7 111:IO Disable
27	/	/	/
26:24	R/W	0x7	PB6_SELECT 000:Input 001:Output 010:reserved 011:PWM6 100: I2S0-DOUT2 101:I2S0-DIN3

			110:PB-EINT6	111:IO Disable
23	/	/	/	
22:20	R/W	0x7	PB5_SELECT 000:Input 010:reserved 100:I2S0-DOUT1 110:PB-EINT5	001:Output 011:PWM5 101: I2S0-DIN0 111:IO Disable
19	/	/	/	
18:16	R/W	0x7	PB4_SELECT 000:Input 010:reserved 100:I2S0-DOUT0 110:PB_EINT4	001:Output 011:PWM4 101: I2S0-DIN1 111:IO Disable
15	/	/	/	
14:12	R/W	0x4	PB3_SELECT 000:Input 010:UART2-CTS 100:JTAG-DI 110:PB-EINT3	001:Output 011:PWM3 101:I2S0-BCLK 111:IO Disable
11	/	/	/	
10:8	R/W	0x4	PB2_SELECT 000:Input 010:UART2-RTS 100:JTAG-DO 110:PB_EINT2	001:Output 011:PWM2 101:I2S0-LRCK 111:IO Disable
7	/	/	/	
6:4	R/W	0x4	PB1_SELECT 000:Input 010:UART2-RX 100:JTAG-CK 110:PB_EINT1	001:Output 011:PWM1 101:reserved 111:IO Disable
3	/	/	/	
2:0	R/W	0x4	PB0_SELECT 000:Input 010:UART2-TX 100:JTAG-MS 110:PB-EINT0	001:Output 011:PWM0 101:LEDC-DO 111:IO Disable

#### 6.6.5.2. PB Configure Register 1 (Default Value: 0x0077\_7777)

Offset: 0x0028			Register Name: PB_CFG1
Bit	Read/Write	Default/Hex	Description
31:23	/	/	/

			PB13_SELECT 000:Input 010:Reserved 100:Reserved 110:PB-EINT13
22:20	R/W	0x7	001:Output 011:Reserved 101:I2S0-MCLK 111:IO Disable
19	/	/	/
18:16	R/W	0x7	PB12_SELECT 000:Input 010:I2S1-MCLK 100:Reserved 110:PB-EINT12
15	/	/	/
14:12	R/W	0x7	PB11_SELECT 000:Input 010:I2S1-DOUT1 100:I2S1-DINO 110:PB_EINT11
11	/	/	/
10:8	R/W	0x7	PB10_SELECT 000:Input 010:I2S1-DOUT0 100:I2S1-DIN1 110:PB_EINT10
7	/	/	/
6:4	R/W	0x7	PB9_SELECT 000:Input 010:I2S1-BCLK 100:TWI1-SDA 110:PB_EINT9
3	/	/	/
2:0	R/W	0x7	PB8_SELECT 000:Input 010:I2S1-LRCK 100:TWI1-SCK 110:PB_EINT8

#### 6.6.5.3. PB Configure Register 2 (Default Value: 0x0000\_0000)

Offset: 0x002C			Register Name: PB_CFG2
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.4. PB Configure Register 3 (Default Value: 0x0000\_0000)

Offset: 0x0030			Register Name: PB_CFG3
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.5. PB Data Register (Default Value: 0x0000\_0000)

Offset: 0x0034			Register Name: PB_DAT
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13:0	R/W	0x0	PB_DAT If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read.

#### 6.6.5.6. PB Multi-Driving Register 0 (Default Value: 0x0555\_5555)

Offset: 0x0038			Register Name: PB_DRV0
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:26	R/W	0x1	PB13_DRV PB13 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
25:24	R/W	0x1	PB12_DRV PB12 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
23:22	R/W	0x1	PB11_DRV PB11 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
21:20	R/W	0x1	PB10_DRV PB10 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
19:18	R/W	0x1	PB9_DRV PB9 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3

17:16	R/W	0x1	PB8_DRV PB8 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
15:14	R/W	0x1	PB7_DRV PB7 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
13:12	R/W	0x1	PB6_DRV PB6 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
11:10	R/W	0x1	PB5_DRV PB5 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
9:8	R/W	0x1	PB4_DRV PB4 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
7:6	R/W	0x1	PB3_DRV PB3 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
5:4	R/W	0x1	PB2_DRV PB2 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
3:2	R/W	0x1	PB1_DRV PB1 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
1:0	R/W	0x1	PB0_DRV PB0 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3

#### 6.6.5.7. PB Multi-Driving Register 1 (Default Value: 0x0000\_0000)

Offset: 0x003C			Register Name: PB_DRV1
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.8. PB Pull Register 0 (Default Value: 0x0000\_0000)

Offset: 0x0040			Register Name: PB_PULL0
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:26	R/W	0x0	PB13_PULL PB13 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
25:24	R/W	0x0	PB12_PULL PB12 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
23:22	R/W	0x0	PB11_PULL PB11 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
21:20	R/W	0x0	PB10_PULL PB10 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
19:18	R/W	0x0	PB9_PULL PB9 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
17:16	R/W	0x0	PB8_PULL PB8 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
15:14	R/W	0x0	PB7_PULL PB7 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
13:12	R/W	0x0	PB6_PULL PB6 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
11:10	R/W	0x0	PB5_PULL PB5 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
9:8	R/W	0x0	PB4_PULL PB4 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved

7:6	R/W	0x0	PB3_PULL PB3 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
5:4	R/W	0x0	PB2_PULL PB2 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
3:2	R/W	0x0	PB1_PULL PB1 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
1:0	R/W	0x0	PB0_PULL PB0 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved

#### 6.6.5.9. PB Pull Register 1 (Default Value: 0x0000\_0000)

Offset: 0x0044			Register Name: PB_PULL1
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.10. PC Configure Register 0 (Default Value: 0x7777\_7777)

Offset: 0x0048			Register Name: PC_CFG0
Bit	Read/Write	Default/Hex	Description
31:19	R/W	0x777	Reserved
18:16	R/W	0x7	PC4_SELECT 000:Input                          001:Output 010:Reserved                          011:Reserved 100:SPI0-MISO                        101:Reserved 110:Reserved                           111:IO Disable
15	/	/	/
14:12	R/W	0x7	PC3_SELECT 000:Input                                  001:Output 010:Reserved                                011:Reserved 100:SPI0-CS0                                101:Reserved 110:Reserved                                   111:IO Disable
11	/	/	/
10:8	R/W	0x7	PC2_SELECT 000:Input                                    001:Output

			010:Reserved 100:SPI0-MOSI 110:Reserved	011:Reserved 101:Reserved 111:IO Disable
7	/	/	/	
6:4	R/W	0x7	PC1_SELECT 000:Input 010:Reserved 100:Reserved 110:Reserved	001:Output 011:Reserved 101:Reserved 111:IO Disable
3	/	/	/	
2:0	R/W	0x7	PC0_SELECT 000:Input 010:Reserved 100:SPI0-CLK 110:Reserved	001:Output 011:Reserved 101:Reserved 111:IO Disable

#### 6.6.5.11. PC Configure Register 1 (Default Value: 0x7777\_7777)

Offset: 0x004C			Register Name: PC_CFG1
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30:28	R/W	0x7	PC15_SELECT 000:Input 010:Reserved 100:SPI0-WP 110:Reserved
27:0	R/W	0x77777777	Reserved

#### 6.6.5.12. PC Configure Register 2 (Default Value: 0x0000\_0007)

Offset: 0x0050			Register Name: PC_CFG2
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2:0	R/W	0x7	PC16_SELECT 000:Input 010:Reserved 100:SPI0-HOLD 110:Reserved

#### 6.6.5.13. PC Configure Register 3 (Default Value: 0x0000\_0000)

Offset: 0x0054			Register Name: PC_CFG3
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.14. PC Data Register (Default Value: 0x0000\_0000)

Offset: 0x0058			Register Name: PC_DAT
Bit	Read/Write	Default/Hex	Description
31:17	/	/	/
16:0	R/W	0x0	PC_DAT If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read.

#### 6.6.5.15. PC Multi-Driving Register 0 (Default Value: 0x5555\_5555)

Offset: 0x005C			Register Name: PC_DRV0
Bit	Read/Write	Default/Hex	Description
31:30	R/W	0x1	PC15_DRV PC15 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
29:28	R/W	0x1	Reserved
27:26	R/W	0x1	Reserved
25:24	R/W	0x1	Reserved
23:22	R/W	0x1	Reserved
21:20	R/W	0x1	Reserved
19:18	R/W	0x1	Reserved
17:16	R/W	0x1	Reserved
15:14	R/W	0x1	Reserved
13:12	R/W	0x1	Reserved
11:10	R/W	0x1	Reserved
9:8	R/W	0x1	PC4_DRV PC4 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
7:6	R/W	0x1	PC3_DRV PC3 Multi-Driving Select 00: Level 0 01: Level 1

			10: Level 2	11: Level 3
5:4	R/W	0x1	PC2_DRV PC2 Multi-Driving Select 00: Level 0 10: Level 2	01: Level 1 11: Level 3
3:2	R/W	0x1	PC1_DRV PC1 Multi-Driving Select 00: Level 0 10: Level 2	01: Level 1 11: Level 3
1:0	R/W	0x1	PC0_DRV PC0 Multi-Driving Select 00: Level 0 10: Level 2	01: Level 1 11: Level 3

#### 6.6.5.16. PC Multi-Driving Register 1 (Default Value: 0x0000\_0001)

Offset: 0x0060			Register Name: PC_DRV1
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1:0	R/W	0x1	PC16_DRV PC16 Multi-Driving Select 00: Level 0 10: Level 2

#### 6.6.5.17. PC Pull Register 0 (Default Value: 0x0000\_5140)

Offset: 0x0064			Register Name: PC_PULL0
Bit	Read/Write	Default/Hex	Description
31:30	R/W	0x1	PC15_PULL PC15 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down
29:28	R/W	0x0	Reserved
27:26	R/W	0x0	Reserved
25:24	R/W	0x0	Reserved
23:22	R/W	0x0	Reserved
21:20	R/W	0x0	Reserved
19:18	R/W	0x0	Reserved
17:16	R/W	0x0	Reserved
15:14	R/W	0x1	Reserved
13:12	R/W	0x1	Reserved
11:10	R/W	0x0	Reserved

9:8	R/W	0x1	PC4_PULL PC4 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
7:6	R/W	0x1	PC3_PULL PC3 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
5:4	R/W	0x0	PC2_PULL PC2 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
3:2	R/W	0x0	PC1_PULL PC1 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
1:0	R/W	0x0	PC0_PULL PC0 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved

#### 6.6.5.18. PC Pull Register 1 (Default Value: 0x0000\_0000)

Offset: 0x0068			Register Name: PC_PULL1
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1:0	R/W	0x0	PC16_PULL PC16 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved

#### 6.6.5.19. PE Configure Register 0 (Default Value: 0x0777\_7777)

Offset: 0x0090			Register Name: PE_CFG0
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26:24	R/W	0x7	PE6_SELECT 000:Input      001:Output 010:I2S1-DOUT1      011:I2S1-DIN0 100:UART2-CTS      101:Reserved 110:PE-EINT6      111:IO Disable
23	/	/	/

			PE5_SELECT 000:Input 010:I2S1-DOUT0 100:PLL-LOCK-DBG 110:PE-EINT5	001:Output 011: I2S1-DIN1 101:Reserved 111:IO Disable
22:20	R/W	0x7	/	
19	/	/	/	
18:16	R/W	0x7	PE4_SELECT 000:Input 010:I2S1-BCLK 100:UART2-RX 110:PE-EINT4	001:Output 011:Reserved 101:Reserved 111:IO Disable
15	/	/	/	
14:12	R/W	0x7	PE3_SELECT 000:Input 010:I2S1-LRCK 100:UART2-TX 110:PE-EINT3	001:Output 011:Reserved 101:Reserved 111:IO Disable
11	/	/	/	
10:8	R/W	0x7	PE2_SELECT 000:Input 010:LEDC-DO 100:Reserved 110:PE-EINT2	001:Output 011:Reserved 101:Reserved 111:IO Disable
7	/	/	/	
6:4	R/W	0x7	PE1_SELECT 000:Input 010:OWA-OUT 100:OWA-IN 110:PE-EINT1	001:Output 011:Reserved 101:Reserved 111:IO Disable
3	/	/	/	
2:0	R/W	0x7	PE0_SELECT 000:Input 010:I2S1-MCLK 100:UART2-RTS 110:PE-EINT0	001:Output 011:Reserved 101:Reserved 111:IO Disable

#### 6.6.5.20. PE Configure Register 1 (Default Value: 0x0000\_0000)

Offset: 0x0094			Register Name: PE_CFG1
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.21. PE Configure Register 2 (Default Value: 0x0000\_0000)

Offset: 0x0098			Register Name: PE_CFG2
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.22. PE Configure Register 3 (Default Value: 0x0000\_0000)

Offset: 0x009C			Register Name: PE_CFG3
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.23. PE Data Register (Default Value: 0x0000\_0000)

Offset: 0x00A0			Register Name: PE_DAT
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:0	R/W	0x0	<p>PE_DAT</p> <p>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.</p>

#### 6.6.5.24. PE Multi-Driving Register 0 (Default Value: 0x0000\_1555)

Offset: 0x00A4			Register Name: PE_DRV0				
Bit	Read/Write	Default/Hex	Description				
31:14	/	/	/				
13:12	R/W	0x1	<p>PE6_DRV</p> <p>PE6 Multi-Driving Select</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						
11:10	R/W	0x1	<p>PE5_DRV</p> <p>PE5 Multi-Driving Select</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						
9:8	R/W	0x1	<p>PE4_DRV</p> <p>PE4 Multi-Driving Select</p> <table> <tr> <td>00: Level 0</td> <td>01: Level 1</td> </tr> <tr> <td>10: Level 2</td> <td>11: Level 3</td> </tr> </table>	00: Level 0	01: Level 1	10: Level 2	11: Level 3
00: Level 0	01: Level 1						
10: Level 2	11: Level 3						
7:6	R/W	0x1	PE3_DRV				

			PE3 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
5:4	R/W	0x1	PE2_DRV PE2 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
3:2	R/W	0x1	PE1_DRV PE1 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
1:0	R/W	0x1	PE0_DRV PE0 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3

#### 6.6.5.25. PE Multi-Driving Register 1 (Default Value: 0x0000\_0000)

Offset: 0x00A8			Register Name: PE_DRV1
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.26. PE Pull Register 0 (Default Value: 0x0000\_0000)

Offset: 0x00AC			Register Name: PE_PULL0
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13:12	R/W	0x0	PE6_PULL PE6 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
11:10	R/W	0x0	PE5_PULL PE5 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
9:8	R/W	0x0	PE4_PULL PE4 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up 10: Pull-down 11: Reserved
7:6	R/W	0x0	PE3_PULL PE3 Pull-up/down Select 00: Pull-up/down disable 01: Pull-up

			10: Pull-down	11: Reserved
5:4	R/W	0x0	PE2_PULL PE2 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	01: Pull-up 11: Reserved
3:2	R/W	0x0	PE1_PULL PE1 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	01: Pull-up 11: Reserved
1:0	R/W	0x0	PE0_PULL PE0 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	01: Pull-up 11: Reserved

#### 6.6.5.27. PE Pull Register 1 (Default Value: 0x0000\_0000)

Offset: 0x00B0			Register Name: PE_PULL1
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.28. PG Configure Register 0 (Default Value: 0x7777\_7777)

Offset: 0x00D8			Register Name: PG_CFG0
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30:28	R/W	0x7	PG7_SELECT 000:Input 010:UART1-RX 100:Reserved 110:PG-EINT7 001:Output 011:Reserved 101:Reserved 111:IO Disable
27	/	/	/
26:24	R/W	0x7	PG6_SELECT 000:Input 010:UART1-TX 100:Reserved 110:PG-EINT6 001:Output 011:Reserved 101:Reserved 111:IO Disable
23	/	/	/
22:20	R/W	0x7	PG5_SELECT 000:Input 010:SDC1-D3 100:Reserved 110:PG-EINT5 001:Output 011:Reserved 101:Reserved 111:IO Disable

19	/	/	/
18:16	R/W	0x7	PG4_SELECT 000:Input 010:SDC1-D2 100:Reserved 110:PG-EINT4 001:Output 011:Reserved 101:Reserved 111:IO Disable
15	/	/	/
14:12	R/W	0x7	PG3_SELECT 000:Input 010:SDC1-D1 100:Reserved 110:PG-EINT3 001:Output 011:Reserved 101:Reserved 111:IO Disable
11	/	/	/
10:8	R/W	0x7	PG2_SELECT 000:Input 010:SDC1-D0 100:Reserved 110:PG-EINT2 001:Output 011:Reserved 101:Reserved 111:IO Disable
7	/	/	/
6:4	R/W	0x7	PG1_SELECT 000:Input 010:SDC1-CMD 100:Reserved 110:PG-EINT1 001:Output 011:Reserved 101:Reserved 111:IO Disable
3	/	/	/
2:0	R/W	0x7	PG0_SELECT 000:Input 010:SDC1-CLK 100:Reserved 110:PG-EINT0 001:Output 011:Reserved 101:Reserved 111:IO Disable

#### 6.6.5.29. PG Configure Register 1 (Default Value: 0x0777\_7777)

Offset: 0x00DC			Register Name: PG_CFG1
Bit	Read/Write	Default/Hex	Description
31:27	/	/	/
26:24	R/W	0x7	PG14_SELECT 000:Input 010:Reserved 100:I2S2-DIN0 110:PG-EINT14 001:Output 011:I2S2-DOUT1 101:Reserved 111:IO Disable
23	/	/	/
22:20	R/W	0x7	PG13_SELECT

			000:Input 010:Reserved 100:I2S2-DIN1 110:PG-EINT13	001:Output 011:I2S2-DOUT0 101:Reserved 111:IO Disable
19	/	/	/	
18:16	R/W	0x7	PG12_SELECT 000:Input 010:Reserved 100:Reserved 110:PG-EINT12	001:Output 011:I2S2-BCLK 101:BIST-RESULT1 111:IO Disable
15	/	/	/	
14:12	R/W	0x7	PG11_SELECT 000:Input 010:Reserved 100:Reserved 110:PG-EINT11	001:Output 011:I2S2-LRCK 101:BIST-RESULT0 111:IO Disable
11	/	/	/	
10:8	R/W	0x7	PG10_SELECT 000:Input 010:Reserved 100:Reserved 110:PG-EINT10	001:Output 011:I2S2-MCLK 101:Reserved 111:IO Disable
7	/	/	/	
6:4	R/W	0x7	PG9_SELECT 000:Input 010:UART1-CTS 100:Reserved 110:PG-EINT9	001:Output 011:Reserved 101:Reserved 111:IO Disable
3	/	/	/	
2:0	R/W	0x7	PG8_SELECT 000:Input 010:UART1-RTS 100:Reserved 110:PG-EINT8	001:Output 011:Reserved 101:Reserved 111:IO Disable

#### 6.6.5.30. PG Configure Register 2 (Default Value: 0x0000\_0000)

Offset: 0x00E0			Register Name: PG_CFG2
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.31. PG Configure Register 3 (Default Value: 0x0000\_0000)

Offset: 0x00E4			Register Name: PG_CFG3
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.32. PG Data Register (Default Value: 0x0000\_0000)

Offset: 0x00E8			Register Name: PG_DAT
Bit	Read/Write	Default/Hex	Description
31:15	/	/	/
14:0	R/W	0x0	PG_DAT If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read.

#### 6.6.5.33. PG Multi-Driving Register 0 (Default Value: 0x1555\_5555)

Offset: 0x00EC			Register Name: PG_DRV0
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:28	R/W	0x1	PG14_DRV PG14 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
27:26	R/W	0x1	PG13_DRV PG13 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
25:24	R/W	0x1	PG12_DRV PG12 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
23:22	R/W	0x1	PG11_DRV PG11 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
21:20	R/W	0x1	PG10_DRV PG10 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3

19:18	R/W	0x1	<b>PG9_DRV</b> <b>PG9 Multi-Driving Select</b> 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
17:16	R/W	0x1	<b>PG8_DRV</b> <b>PG8 Multi-Driving Select</b> 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
15:14	R/W	0x1	<b>PG7_DRV</b> <b>PG7 Multi-Driving Select</b> 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
13:12	R/W	0x1	<b>PG6_DRV</b> <b>PG6 Multi-Driving Select</b> 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
11:10	R/W	0x1	<b>PG5_DRV</b> <b>PG5 Multi-Driving Select</b> 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
9:8	R/W	0x1	<b>PG4_DRV</b> <b>PG4 Multi-Driving Select</b> 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
7:6	R/W	0x1	<b>PG3_DRV</b> <b>PG3 Multi-Driving Select</b> 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
5:4	R/W	0x1	<b>PG2_DRV</b> <b>PG2 Multi-Driving Select</b> 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
3:2	R/W	0x1	<b>PG1_DRV</b> <b>PG1 Multi-Driving Select</b> 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3
1:0	R/W	0x1	<b>PG0_DRV</b> <b>PG0 Multi-Driving Select</b> 00: Level 0                    01: Level 1 10: Level 2                    11: Level 3

#### 6.6.5.34. PG Multi-Driving Register 1 (Default Value: 0x0000\_0000)

Offset: 0x00F0	Register Name: PG_DRV1
----------------	------------------------

Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.35. PG Pull Register 0 (Default Value: 0x0000\_0000)

Offset: 0x00F4			Register Name: PG_PULL0
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:28	R/W	0x0	PG14_PULL PG14 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
27:26	R/W	0x0	PG13_PULL PG13 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
25:24	R/W	0x0	PG12_PULL PG12 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
23:22	R/W	0x0	PG11_PULL PG11 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
21:20	R/W	0x0	PG10_PULL PG10 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
19:18	R/W	0x0	PG9_PULL PG9 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
17:16	R/W	0x0	PG8_PULL PG8 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
15:14	R/W	0x0	PG7_PULL PG7 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down      11: Reserved
13:12	R/W	0x0	PG6_PULL PG6 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up

			10: Pull-down	11: Reserved
11:10	R/W	0x0	PG5_PULL PG5 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	01: Pull-up 11: Reserved
9:8	R/W	0x0	PG4_PULL PG4 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	01: Pull-up 11: Reserved
7:6	R/W	0x0	PG3_PULL PG3 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	01: Pull-up 11: Reserved
5:4	R/W	0x0	PG2_PULL PG2 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	01: Pull-up 11: Reserved
3:2	R/W	0x0	PG1_PULL PG1 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	01: Pull-up 11: Reserved
1:0	R/W	0x0	PG0_PULL PG0 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	01: Pull-up 11: Reserved

#### 6.6.5.36. PG Pull Register 1 (Default Value: 0x0000\_0000)

Offset: 0x00F8			Register Name: PG_PULL1
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.37. PH Configure Register 0 (Default Value: 0x7777\_7777)

Offset: 0x00FC			Register Name: PH_CFG0
Bit	Read/Write	Default/Hex	Description
31	/	/	/
30:28	R/W	0x7	PH7_SELECT 000:Input 010:UART3-CTS 100:Reserved 110:PH-EINT7 001:Output 011:SPI1-MISO 101:Reserved 111:IO Disable

27	/	/	/
			PH6_SELECT 000:Input 010:UART3-RTS 100:Reserved 110:PH-EINT6
26:24	R/W	0x7	001:Output 011:SPI1-MOSI 101:Reserved 111:IO Disable
23	/	/	/
			PH5_SELECT 000:Input 010:UART3-RX 100:Reserved 110:PH-EINT5
22:20	R/W	0x7	001:Output 011:SPI1-CLK 101:Reserved 111:IO Disable
19	/	/	/
			PH4_SELECT 000:Input 010:UART3-TX 100:Reserved 110:PH-EINT4
18:16	R/W	0x7	001:Output 011:SPI1-CS 101:Reserved 111:IO Disable
15	/	/	/
			PH3_SELECT 000:Input 010:TWI1-SDA 100:SPI1-MISO 110:PH-EINT3
14:12	R/W	0x7	001:Output 011:OWA-OUT 101:Reserved 111:IO Disable
11	/	/	/
			PH2_SELECT 000:Input 010:TWI1-SCK 100:SPI1-CS 110:PH-EINT2
10:8	R/W	0x7	001:Output 011:LEDC-DO 101:Reserved 111:IO Disable
7	/	/	/
			PH1_SELECT 000:Input 010:TWI0-SDA 100:SPI1-CLK 110:PH-EINT1
6:4	R/W	0x7	001:Output 011:UART0-RX 101:Reserved 111:IO Disable
3	/	/	/
			PH0_SELECT 000:Input 010:TWI0-SCK 100:SPI1-MOSI 110:PH-EINT0
2:0	R/W	0x7	001:Output 011:UART0-TX 101:Reserved 111:IO Disable

#### 6.6.5.38. PH Configure Register 1 (Default Value: 0x0000\_0077)

Offset: 0x0100			Register Name: PH_CFG1
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:4	R/W	0x7	PH9_SELECT 000:Input 001:Output 010:Reserved 011:Reserved 100:CPU-CUR-W 101:Reserved 110:PH-EINT9 111:IO Disable
3	/	/	/
2:0	R/W	0x7	PH8_SELECT 000:Input 001:Output 010:LEDC-DO 011:OWA-IN 100:Reserved 101:Reserved 110:PH-EINT8 111:IO Disable

#### 6.6.5.39. PH Configure Register 2 (Default Value: 0x0000\_0000)

Offset: 0x0104			Register Name: PH_CFG2
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.40. PH Configure Register 3 (Default Value: 0x0000\_0000)

Offset: 0x0108			Register Name: PH_CFG3
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.41. PH Data Register (Default Value: 0x0000\_0000)

Offset: 0x010C			Register Name: PH_DAT
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9:0	R/W	0x0	PH_DAT If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read.

**6.6.5.42. PH Multi-Driving Register 0 (Default Value: 0x0005\_5555)**

Offset: 0x0110			Register Name: PH_DRV0
Bit	Read/Write	Default/Hex	Description
31:20	/	/	/
19:18	R/W	0x1	PH9_DRV PH9 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
17:16	R/W	0x1	PH8_DRV PH8 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
15:14	R/W	0x1	PH7_DRV PH7 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
13:12	R/W	0x1	PH6_DRV PH6 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
11:10	R/W	0x1	PH5_DRV PH5 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
9:8	R/W	0x1	PH4_DRV PH4 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
7:6	R/W	0x1	PH3_DRV PH3 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
5:4	R/W	0x1	PH2_DRV PH2 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
3:2	R/W	0x1	PH1_DRV PH1 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3
1:0	R/W	0x1	PH0_DRV PH0 Multi-Driving Select 00: Level 0 01: Level 1 10: Level 2 11: Level 3

#### 6.6.5.43. PH Multi-Driving Register 1 (Default Value: 0x0000\_0000)

Offset: 0x0114			Register Name: PH_DRV1
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.44. PH Pull Register 0 (Default Value: 0x0000\_0000)

Offset: 0x0118			Register Name: PH_PULL0
Bit	Read/Write	Default/Hex	Description
31:20	/	/	/
19:18	R/W	0x0	PH9_PULL PH9 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down                  11: Reserved
17:16	R/W	0x0	PH8_PULL PH8 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down                  11: Reserved
15:14	R/W	0x0	PH7_PULL PH7 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down                  11: Reserved
13:12	R/W	0x0	PH6_PULL PH6 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down                  11: Reserved
11:10	R/W	0x0	PH5_PULL PH5 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down                  11: Reserved
9:8	R/W	0x0	PH4_PULL PH4 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down                  11: Reserved
7:6	R/W	0x0	PH3_PULL PH3 Pull-up/down Select 00: Pull-up/down disable      01: Pull-up 10: Pull-down                  11: Reserved
5:4	R/W	0x0	PH2_PULL PH2 Pull-up/down Select

			00: Pull-up/down disable 10: Pull-down	01: Pull-up 11: Reserved
3:2	R/W	0x0	PH1_PULL PH1 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	01: Pull-up 11: Reserved
1:0	R/W	0x0	PH0_PULL PH0 Pull-up/down Select 00: Pull-up/down disable 10: Pull-down	01: Pull-up 11: Reserved

#### 6.6.5.45. PH Pull Register 1 (Default Value: 0x0000\_0000)

Offset: 0x011C			Register Name: PH_PULL1
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.46. PB External Interrupt Configure Register 0 (Default Value: 0x0000\_0000)

Offset: 0x0220			Register Name: PB_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	EINT7_CFG External INT7 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/ Negative) Others: Reserved
27:24	R/W	0x0	EINT6_CFG External INT6 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/ Negative) Others: Reserved
23:20	R/W	0x0	EINT5_CFG External INT5 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level

			0100: Double Edge (Positive/ Negative) Others: Reserved
19:16	R/W	0x0	EINT4_CFG External INT4 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/ Negative) Others: Reserved
15:12	R/W	0x0	EINT3_CFG External INT3 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/ Negative) Others: Reserved
11:8	R/W	0x0	EINT2_CFG External INT2 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/ Negative) Others: Reserved
7:4	R/W	0x0	EINT1_CFG External INT1 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/ Negative) Others: Reserved
3:0	R/W	0x0	EINT0_CFG External INT0 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/ Negative) Others: Reserved

**6.6.5.47. PB External Interrupt Configure Register 1 (Default Value: 0x0000\_0000)**

Offset: 0x0224			Register Name: PB_EINT_CFG1
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:20	R/W	0x0	EINT13_CFG External INT13 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
19:16	R/W	0x0	EINT12_CFG External INT12 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
15:12	R/W	0x0	EINT11_CFG External INT11 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
11:8	R/W	0x0	EINT10_CFG External INT10 Mode 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved
7:4	R/W	0x0	EINT9_CFG External INT9 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/ Negative) Others: Reserved

3:0	R/W	0x0	EINT8_CFG External INT8 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/ Negative) Others: Reserved
-----	-----	-----	--

#### 6.6.5.48. PB External Interrupt Configure Register 2 (Default Value: 0x0000\_0000)

Offset: 0x0228			Register Name: PB_EINT_CFG2
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.49. PB External Interrupt Configure Register 3 (Default Value: 0x0000\_0000)

Offset: 0x022C			Register Name: PB_EINT_CFG3
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.50. PB External Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x0230			Register Name: PB_EINT_CTL
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13	R/W	0x0	EINT13_CTL External INT13 Enable 0: Disable 1: Enable
12	R/W	0x0	EINT12_CTL External INT12 Enable 0: Disable 1: Enable
11	R/W	0x0	EINT11_CTL External INT11 Enable 0: Disable 1: Enable
10	R/W	0x0	EINT10_CTL External INT10 Enable 0: Disable

			1: Enable
9	R/W	0x0	EINT9_CTL External INT9 Enable 0: Disable 1: Enable
8	R/W	0x0	EINT8_CTL External INT8 Enable 0: Disable 1: Enable
7	R/W	0x0	EINT7_CTL External INT7 Enable 0: Disable 1: Enable
6	R/W	0x0	EINT6_CTL External INT6 Enable 0: Disable 1: Enable
5	R/W	0x0	EINT5_CTL External INT5 Enable 0: Disable 1: Enable
4	R/W	0x0	EINT4_CTL External INT4 Enable 0: Disable 1: Enable
3	R/W	0x0	EINT3_CTL External INT3 Enable 0: Disable 1: Enable
2	R/W	0x0	EINT2_CTL External INT2 Enable 0: Disable 1: Enable
1	R/W	0x0	EINT1_CTL External INT1 Enable 0: Disable 1: Enable
0	R/W	0x0	EINT0_CTL External INT0 Enable 0: Disable 1: Enable

#### 6.6.5.51. PB External Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0234			Register Name: PB_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13	R/W1C	0x0	EINT13_STATUS External INT13 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
12	R/W1C	0x0	EINT12_STATUS External INT12 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
11	R/W1C	0x0	EINT11_STATUS External INT11 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
10	R/W1C	0x0	EINT10_STATUS External INT10 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
9	R/W1C	0x0	EINT9_STATUS External INT9 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
8	R/W1C	0x0	EINT8_STATUS External INT8 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
7	R/W1C	0x0	EINT7_STATUS External INT7 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
6	R/W1C	0x0	EINT6_STATUS External INT6 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

5	R/W1C	0x0	EINT5_STATUS External INT5 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
4	R/W1C	0x0	EINT4_STATUS External INT4 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
3	R/W1C	0x0	EINT3_STATUS External INT3 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
2	R/W1C	0x0	EINT2_STATUS External INT2 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
1	R/W1C	0x0	EINT1_STATUS External INT1 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
0	R/W1C	0x0	EINT0_STATUS External INT0 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

#### 6.6.5.52. PB External Interrupt Debounce Register (Default Value: 0x0000\_0000)

Offset: 0x0238			Register Name: PB_EINT_DEB
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:4	R/W	0x0	DEB_CLK_PRE_SCALE Debounce Clock Pre-scale n The selected clock source is prescaled by $2^n$ .
3:1	/	/	/
0	R/W	0x0	PIO_INT_CLK_SELECT PIO Interrupt Clock Select 0: LOSC 32KHz 1: HOSC 24MHz

#### 6.6.5.53. PE External Interrupt Configure Register 0 (Default Value: 0x0000\_0000)

Offset: 0x0280			Register Name: PE_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:24	R/W	0x0	EINT6_CFG External INT6 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
23:20	R/W	0x0	EINT5_CFG External INT5 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
19:16	R/W	0x0	EINT4_CFG External INT4 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
15:12	R/W	0x0	EINT3_CFG External INT3 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
11:8	R/W	0x0	EINT2_CFG External INT2 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level

			0100: Double Edge (Positive/Negative) Others: Reserved
7:4	R/W	0x0	EINT1_CFG External INT1 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
3:0	R/W	0x0	EINT0_CFG External INT0 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved

#### 6.6.5.54. PE External Interrupt Configure Register 1 (Default Value: 0x0000\_0000)

Offset: 0x0284			Register Name: PE_EINT_CFG1
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.55. PE External Interrupt Configure Register 2 (Default Value: 0x0000\_0000)

Offset: 0x0288			Register Name: PE_EINT_CFG2
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.56. PE External Interrupt Configure Register 3 (Default Value: 0x0000\_0000)

Offset: 0x028C			Register Name: PE_EINT_CFG3
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.57. PE External Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x0290	Register Name: PE_EINT_CTL
----------------	----------------------------

Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6	R/W	0x0	EINT6_CTL External INT6 Enable 0: Disable 1: Enable
5	R/W	0x0	EINT5_CTL External INT5 Enable 0: Disable 1: Enable
4	R/W	0x0	EINT4_CTL External INT4 Enable 0: Disable 1: Enable
3	R/W	0x0	EINT3_CTL External INT3 Enable 0: Disable 1: Enable
2	R/W	0x0	EINT2_CTL External INT2 Enable 0: Disable 1: Enable
1	R/W	0x0	EINT1_CTL External INT1 Enable 0: Disable 1: Enable
0	R/W	0x0	EINT0_CTL External INT0 Enable 0: Disable 1: Enable

#### 6.6.5.58. PE External Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0294			Register Name: PE_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6	R/W1C	0x0	EINT6_STATUS External INT6 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
5	R/W1C	0x0	EINT5_STATUS External INT5 Pending Bit

			0: No IRQ pending 1: IRQ pending Write '1' to clear
4	R/W1C	0x0	EINT4_STATUS External INT4 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
3	R/W1C	0x0	EINT3_STATUS External INT3 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
2	R/W1C	0x0	EINT2_STATUS External INT2 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
1	R/W1C	0x0	EINT1_STATUS External INT1 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
0	R/W1C	0x0	EINT0_STATUS External INTO Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

#### 6.6.5.59. PE External Interrupt Debounce Register (Default Value: 0x0000\_0000)

Offset: 0x0298			Register Name: PE_EINT_DEB
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:4	R/W	0x0	DEB_CLK_PRE_SCALE Debounce Clock Pre-scale n The selected clock source is prescaled by $2^n$ .
3:1	/	/	/
0	R/W	0x0	PIO_INT_CLK_SELECT PIO Interrupt Clock Select 0: LOSC 32KHz 1: HOSC 24MHz

#### 6.6.5.60. PG External Interrupt Configure Register 0 (Default Value: 0x0000\_0000)

Offset: 0x02C0			Register Name:PG_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	EINT7_CFG External INT7 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
27:24	R/W	0x0	EINT6_CFG External INT6 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
23:20	R/W	0x0	EINT5_CFG External INT5 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
19:16	R/W	0x0	EINT4_CFG External INT4 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
15:12	R/W	0x0	EINT3_CFG External INT3 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative)

			Others: Reserved
11:8	R/W	0x0	EINT2_CFG External INT2 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
7:4	R/W	0x0	EINT1_CFG External INT1 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
3:0	R/W	0x0	EINT0_CFG External INT0 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved

#### 6.6.5.61. PG External Interrupt Configure Register 1 (Default Value: 0x0000\_0000)

Offset: 0x02C4			Register Name: PG_EINT_CFG1
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:24	R/W	0x0	EINT14_CFG External INT14 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
23:20	R/W	0x0	EINT13_CFG External INT13 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level

			0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
19:16	R/W	0x0	EINT12_CFG External INT12 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
15:12	R/W	0x0	EINT11_CFG External INT11 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
11:8	R/W	0x0	EINT10_CFG External INT10 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
7:4	R/W	0x0	EINT9_CFG External INT9 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
3:0	R/W	0x0	EINT8_CFG External INT8 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved

#### 6.6.5.62. PG External Interrupt Configure Register 2 (Default Value: 0x0000\_0000)

Offset: 0x02C8			Register Name: PG_EINT_CFG2
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.63. PG External Interrupt Configure Register 3 (Default Value: 0x0000\_0000)

Offset: 0x02CC			Register Name: PG_EINT_CFG3
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.64. PG External Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x02D0			Register Name: PG_EINT_CTL
Bit	Read/Write	Default/Hex	Description
31:15	/	/	/
14	R/W	0x0	EINT14_CTL External INT14 Enable 0: Disable 1: Enable
13	R/W	0x0	EINT13_CTL External INT13 Enable 0: Disable 1: Enable
12	R/W	0x0	EINT12_CTL External INT12 Enable 0: Disable 1: Enable
11	R/W	0x0	EINT11_CTL External INT11 Enable 0: Disable 1: Enable
10	R/W	0x0	EINT10_CTL External INT10 Enable 0: Disable 1: Enable
9	R/W	0x0	EINT9_CTL External INT9 Enable 0: Disable 1: Enable
8	R/W	0x0	EINT8_CTL

			External INT8 Enable 0: Disable 1: Enable
7	R/W	0x0	EINT7_CTL External INT7 Enable 0: Disable 1: Enable
6	R/W	0x0	EINT6_CTL External INT6 Enable 0: Disable 1: Enable
5	R/W	0x0	EINT5_CTL External INT5 Enable 0: Disable 1: Enable
4	R/W	0x0	EINT4_CTL External INT4 Enable 0: Disable 1: Enable
3	R/W	0x0	EINT3_CTL External INT3 Enable 0: Disable 1: Enable
2	R/W	0x0	EINT2_CTL External INT2 Enable 0: Disable 1: Enable
1	R/W	0x0	EINT1_CTL External INT1 Enable 0: Disable 1: Enable
0	R/W	0x0	EINT0_CTL External INT0 Enable 0: Disable 1: Enable

#### 6.6.5.65. PG External Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x02D4			Register Name: PG_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
31:15	/	/	/
14	R/W1C	0x0	EINT14_STATUS External INT14 Pending Bit 0: No IRQ pending

			1: IRQ pending Write '1' to clear
13	R/W1C	0x0	EINT13_STATUS External INT13 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
12	R/W1C	0x0	EINT12_STATUS External INT12 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
11	R/W1C	0x0	EINT11_STATUS External INT11 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
10	R/W1C	0x0	EINT10_STATUS External INT10 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
9	R/W1C	0x0	EINT9_STATUS External INT9 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
8	R/W1C	0x0	EINT8_STATUS External INT8 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
7	R/W1C	0x0	EINT7_STATUS External INT7 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
6	R/W1C	0x0	EINT6_STATUS External INT6 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
5	R/W1C	0x0	EINT5_STATUS External INT5 Pending Bit 0: No IRQ pending

			1: IRQ pending Write '1' to clear
4	R/W1C	0x0	EINT4_STATUS External INT4 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
3	R/W1C	0x0	EINT3_STATUS External INT3 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
2	R/W1C	0x0	EINT2_STATUS External INT2 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
1	R/W1C	0x0	EINT1_STATUS External INT1 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
0	R/W1C	0x0	EINT0_STATUS External INT0 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

#### 6.6.5.66. PG External Interrupt Debounce Register (Default Value: 0x0000\_0000)

Offset: 0x02D8			Register Name: PG_EINT_DEB
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:4	R/W	0x0	DEB_CLK_PRE_SCALE Debounce Clock Pre-scale n The selected clock source is prescaled by $2^n$ .
3:1	/	/	/
0	R/W	0x0	PIO_INT_CLK_SELECT PIO Interrupt Clock Select 0: LOSC 32KHz 1: HOSC 24MHz

**6.6.5.67. PH External Interrupt Configure Register 0 (Default Value: 0x0000\_0000)**

Offset: 0x02E0			Register Name:PH_EINT_CFG0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	EINT7_CFG External INT7 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
27:24	R/W	0x0	EINT6_CFG External INT6 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
23:20	R/W	0x0	EINT5_CFG External INT5 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
19:16	R/W	0x0	EINT4_CFG External INT4 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
15:12	R/W	0x0	EINT3_CFG External INT3 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
11:8	R/W	0x0	EINT2_CFG

			External INT2 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
7:4	R/W	0x0	EINT1_CFG External INT1 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
3:0	R/W	0x0	EINT0_CFG External INT0 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved

#### 6.6.5.68. PH External Interrupt Configure Register 1 (Default Value: 0x0000\_0000)

Offset: 0x02E4			Register Name: PH_EINT_CFG1
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:4	R/W	0x0	EINT9_CFG External INT9 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative) Others: Reserved
3:0	R/W	0x0	EINT8_CFG External INT8 Mode 0000: Positive Edge 0001: Negative Edge 0010: High Level 0011: Low Level 0100: Double Edge (Positive/Negative)

			Others: Reserved
--	--	--	------------------

#### 6.6.5.69. PH External Interrupt Configure Register 2 (Default Value: 0x0000\_0000)

Offset: 0x02E8			Register Name: PH_EINT_CFG2
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.70. PH External Interrupt Configure Register 3 (Default Value: 0x0000\_0000)

Offset: 0x02EC			Register Name: PH_EINT_CFG3
Bit	Read/Write	Default/Hex	Description
31:0	/	/	/

#### 6.6.5.71. PH External Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x02F0			Register Name: PH_EINT_CTL
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9	R/W	0x0	EINT9_CTL External INT9 Enable 0: Disable 1: Enable
8	R/W	0x0	EINT8_CTL External INT8 Enable 0: Disable 1: Enable
7	R/W	0x0	EINT7_CTL External INT7 Enable 0: Disable 1: Enable
6	R/W	0x0	EINT6_CTL External INT6 Enable 0: Disable 1: Enable
5	R/W	0x0	EINT5_CTL External INT5 Enable 0: Disable 1: Enable
4	R/W	0x0	EINT4_CTL External INT4 Enable

			0: Disable 1: Enable
3	R/W	0x0	EINT3_CTL External INT3 Enable 0: Disable 1: Enable
2	R/W	0x0	EINT2_CTL External INT2 Enable 0: Disable 1: Enable
1	R/W	0x0	EINT1_CTL External INT1 Enable 0: Disable 1: Enable
0	R/W	0x0	EINT0_CTL External INT0 Enable 0: Disable 1: Enable

#### 6.6.5.72. PH External Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x02F4			Register Name: PH_EINT_STATUS
Bit	Read/Write	Default/Hex	Description
31:10	/	/	/
9	R/W1C	0x0	EINT9_STATUS External INT9 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
8	R/W1C	0x0	EINT8_STATUS External INT8 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
7	R/W1C	0x0	EINT7_STATUS External INT7 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
6	R/W1C	0x0	EINT6_STATUS External INT6 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

5	R/W1C	0x0	EINT5_STATUS External INT5 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
4	R/W1C	0x0	EINT4_STATUS External INT4 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
3	R/W1C	0x0	EINT3_STATUS External INT3 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
2	R/W1C	0x0	EINT2_STATUS External INT2 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
1	R/W1C	0x0	EINT1_STATUS External INT1 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear
0	R/W1C	0x0	EINT0_STATUS External INT0 Pending Bit 0: No IRQ pending 1: IRQ pending Write '1' to clear

#### 6.6.5.73. PH External Interrupt Debounce Register (Default Value: 0x0000\_0000)

Offset: 0x02F8			Register Name: PH_EINT_DEB
Bit	Read/Write	Default/Hex	Description
31:7	/	/	/
6:4	R/W	0x0	DEB_CLK_PRE_SCALE Debounce Clock Pre-scale n The selected clock source is prescaled by $2^n$ .
3:1	/	/	/
0	R/W	0x0	PIO_INT_CLK_SELECT PIO Interrupt Clock Select 0: LOSC 32KHz 1: HOSC 24MHz

## 6.7. GPADC

### 6.7.1. Overview

The General Purpose ADC(GPADC) is one analog to digital converter with 12-bit sampling resolution. This ADC is a type of successive approximation register (SAR) converter.

The GPADC has the following features:

- 12-bit resolution
- 8-bit effective SAR type A/D converter
- 64 FIFO depth of data register
- Power reference voltage: AVCC, analog input voltage range: 0 to AVCC
- Maximum sampling frequency: 1 MHz
- Supports data compare and interrupt
- Supports DMA transport
- Supports three operation modes
  - Single conversion mode
  - Continuous conversion mode
  - Burst conversion mode

### 6.7.2. Block Diagram

Figure 6-33 shows the block diagram of the GPADC.

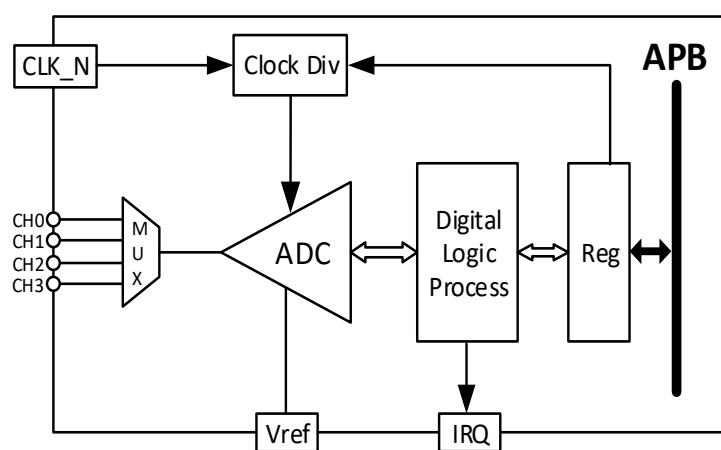


Figure 6- 33. GPADC Block Diagram

### 6.7.3. Operations and Functional Descriptions

#### 6.7.3.1. External Signals

Table 6-17 describes the external signals of GPADC.

**Table 6- 17. GPADC External Signals**

Signal	Description	Type
GPADC0	ADC Input Channel0	AI
GPADC1	ADC Input Channel1	AI
GPADC2	ADC Input Channel2	AI
GPADC3	ADC Input Channel3	AI

#### 6.7.3.2. Clock Sources

GPADC has one clock source. Table 6-18 describes the clock source for GPADC. Users can see **Clock Controller Unit(CCU)** for clock setting, configuration and gating information.

**Table 6- 18. GPADC Clock Sources**

Clock Sources	Description
OSC24M	24MHz

#### 6.7.3.3. GPADC Work Mode

##### (1).Single conversion mode

GPADC completes one conversion in specified channel, the converted data is updated at the data register of corresponding channel.

##### (2).Continuous conversion mode

GPADC has continuous conversion in specified channel until the software stops, the converted data is updated at the data register of corresponding channel.

##### (3).Burst conversion mode

GPADC samples and converts in the specified channel, and sequentially stores the results in FIFO.

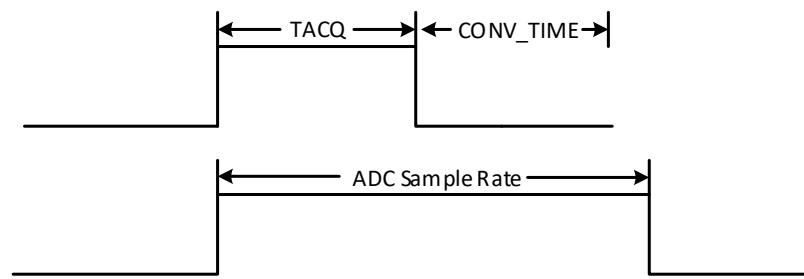
#### 6.7.3.4. Clock and Timing Requirements

CLK\_IN = 24MHz

CONV\_TIME(Conversion Time) = 1/(24MHz/14Cycles) =0.583 (us)

TACQ> 10RC (R is output impedance of ADC sample circuit, C= 6.4pF)

ADC Sample Frequency > TACQ+CONV\_TIME



**Figure 6- 34. GPADC Clock and Timing Requirement**

#### 6.7.3.5. GPADC Calculate Formula

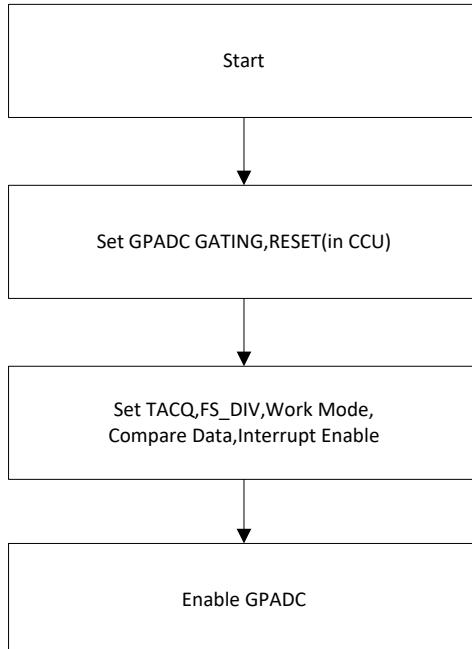
GPADC calculate formula:  $GPADC\_DATA = Vin/V_{REF} * 4096$

Where:

$V_{REF}=1.8V$

#### 6.7.4. Programming Guidelines

GPADC initial process is as follows.



**Figure 6- 35. GPADC Initial Process**

#### 6.7.5. Register List

Module Name	Base Address
GPADC	0x05070000

Register Name	Offset	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_CH0_CMP_DATA	0x0040	GPADC CH0 Compare Data Register
GP_CH1_CMP_DATA	0x0044	GPADC CH1 Compare Data Register
GP_CH2_CMP_DATA	0x0048	GPADC CH2 Compare Data Register
GP_CH3_CMP_DATA	0x004C	GPADC CH3 Compare Data Register
GP_CH0_DATA	0x0080	GPADC CH0 Data Register
GP_CH1_DATA	0x0084	GPADC CH1 Data Register
GP_CH2_DATA	0x0088	GPADC CH2 Data Register
GP_CH3_DATA	0x008C	GPADC CH3 Data Register

## 6.7.6. Register Description

### 6.7.6.1. GPADC Sample Rate Configure Register (Default Value: 0x01DF\_002F)

Offset: 0x0000			Register Name: GP_SR_CON
Bit	Read/Write	Default/Hex	Description
31: 16	R/W	0x1DF	FS_DIV ADC sample frequency divider CLK_IN/(n+1) Default value: 50K
15:0	R/W	0x2F	TACQ ADC acquire time CLK_IN/(N+1) Default value: 2us

### 6.7.6.2. GPADC Control Register (Default Value: 0x0000\_0000)

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	R/ W	0x1	ADC_AUTOCALI_EN ADC Auto Calibration
22	/	/	/
21:20	R/W	0x0	ADC_OP_BIAS ADC OP Bias Adjust the bandwidth of the ADC amplifier
19:18	R/W	0x0	GPADC Work Mode 00: Single conversion mode 01: Reserved 10: Continuous conversion mode 11: Burst conversion mode
17	R/W	0x0	ADC_CALI_EN ADC Calibration 1: Start Calibration, it is cleared to 0 after calibration
16	R/W	0x0	ADC_EN ADC Function Enable Before the bit is enabled, configure ADC parameters including the work mode and channel number,etc. 0: Disable 1: Enable
15:0	/	/	/

#### 6.7.6.3. GPADC Compare and Select Enable Register (Default Value: 0x0000\_0000)

Offset: 0x0008			Register Name: GP_CS_EN
Bit	Read/Write	Default/Hex	Description
31:20	/	/	/
19	R/W	0x0	ADC_CH3_CMP_EN Channel 3 Compare Enable 0: Disable 1: Enable
18	R/W	0x0	ADC_CH2_CMP_EN Channel 2 Compare Enable 0: Disable 1: Enable
17	R/W	0x0	ADC_CH1_CMP_EN Channel 1 Compare Enable 0: Disable 1: Enable

16	R/W	0x0	ADC_CH0_CMP_EN Channel 0 Compare Enable 0: Disable 1: Enable
15:4	/	/	/
3	R/W	0x0	ADC_CH3_SELECT Analog input channel 3 Select 0: Disable 1: Enable
2	R/W	0x0	ADC_CH2_SELECT Analog input channel 2 Select 0: Disable 1: Enable
1	R/W	0x0	ADC_CH1_SELECT Analog input channel 1 Select 0: Disable 1: Enable
0	R/W	0x0	ADC_CH0_SELECT Analog input channel 0 Select 0: Disable 1: Enable

#### 6.7.6.4. GPADC FIFO Interrupt Control Register (Default Value: 0x0000\_1F00)

Offset: 0x000C			Register Name: GP_FIFO_INTC
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R/W	0x0	FIFO_OVERRUN_IRQ_EN ADC FIFO Overrun 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_TRIGGER_LEVEL Interrupt trigger level for ADC Trigger Level = TXTL + 1
7:5	/	/	/
4	R/WAC	0x0	FIFO_FLUSH ADC FIFO Flush Write '1' to flush TX FIFO, clear automatically to '0'.

3:0	/	/	/
-----	---	---	---

#### 6.7.6.5. GPADC FIFO Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0010			Register Name: GP_FIFO_INTS
Bit	Read/Write	Default/Hex	Description
31:18	/	/	/
17	R/W1C	0x0	FIFO_OVERRUN_PENDING ADC FIFO Overrun IRQ Pending 0: No Pending IRQ 1: FIFO Overrun Pending IRQ Write '1' to clear this interrupt or automatically 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 automatically clear if interrupt condition fails.
15:14	/	/	/
13:8	R	0x0	RXA_CNT ADC FIFO available sample word counter
7:0	/	/	/

#### 6.7.6.6. GPADC FIFO Data Register

Offset: 0x0014			Register Name: GP_FIFO_DATA
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:0	R	UDF	GP_FIFO_DATA GPADC Data in FIFO

#### 6.7.6.7. GPADC Calibration Data Register (Default Value: 0x0000\_0000)

Offset: 0x0018			Register Name: GP_CDATA
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:0	R/W	0x0	GP_CDATA GPADC Calibration Data

#### 6.7.6.8. GPADC Low Interrupt Configure Register (Default Value: 0x0000\_0000)

Offset: 0x0020			Register Name: GP_DATAL_INTC
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3	R/W	0x0	CH3_LOW_IRQ_EN 0: Disable 1: Enable
2	R/W	0x0	CH2_LOW_IRQ_EN 0: Disable 1: Enable
1	R/W	0x0	CH1_LOW_IRQ_EN 0: Disable 1: Enable
0	R/W	0x0	CH0_LOW_IRQ_EN 0: Disable 1: Enable

#### 6.7.6.9. GPADC High Interrupt Configure Register (Default Value: 0x0000\_0000)

Offset: 0x0024			Register Name: GP_DATAH_INTC
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3	R/W	0x0	CH3_HIG_IRQ_EN 0: Disable 1: Enable
2	R/W	0x0	CH2_HIG_IRQ_EN 0: Disable 1: Enable
1	R/W	0x0	CH1_HIG_IRQ_EN 0: Disable 1: Enable
0	R/W	0x0	CH0_HIG_IRQ_EN 0: Disable 1: Enable

#### 6.7.6.10. GPADC DATA Interrupt Configure Register (Default Value: 0x0000\_0000)

Offset: 0x0028			Register Name: GP_DATA_INTC
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3	R/W	0x0	CH3_DATA_IRQ_EN

			0: Disable 1: Enable
2	R/W	0x0	CH2_DATA_IRQ_EN 0: Disable 1: Enable
1	R/W	0x0	CH1_DATA_IRQ_EN 0: Disable 1: Enable
0	R/W	0x0	CH0_DATA_IRQ_EN 0: Disable 1: Enable

#### 6.7.6.11. GPADC Low Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0030			Register Name: GP_DATA_L_INTS
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3	R/W1C	0x0	CH3_LOW_PENGDING 1: Channel 3 Voltage Low Available Pending IRQ Write '1' to clear this interrupt or automatically clear if interrupt condition fails.
2	R/W1C	0x0	CH2_LOW_PENGDING 1: Channel 2 Voltage Low Available Pending IRQ Write '1' to clear this interrupt or automatically clear if interrupt condition fails.
1	R/W1C	0x0	CH1_LOW_PENGDING 1: Channel 1 Voltage Low Available Pending IRQ Write '1' to clear this interrupt or automatically clear if interrupt condition fails.
0	R/W1C	0x0	CH0_LOW_PENGDING 1: Channel 0 Voltage Low Available Pending IRQ Write '1' to clear this interrupt or automatically clear if interrupt condition fails.

#### 6.7.6.12. GPADC High Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0034			Register Name: GP_DATAH_INTS
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3	R/W1C	0x0	CH3_HIG_PENGDING 0: No Pending IRQ 1: Channel 3 Voltage High Available Pending IRQ

			Write '1' to clear this interrupt or automatically clear if interrupt condition fails.
2	R/W1C	0x0	<p>CH2_HIG_PENGDDING                      0: No Pending IRQ                      1: Channel 2 Voltage High Available Pending IRQ</p> <p>Write '1' to clear this interrupt or automatically clear if interrupt condition fails.</p>
1	R/W1C	0x0	<p>CH1_HIG_PENGDDING                      0: No Pending IRQ                      1: Channel 1 Voltage High Available Pending IRQ</p> <p>Write '1' to clear this interrupt or automatically clear if interrupt condition fails.</p>
0	R/W1C	0x0	<p>CHO_HIG_PENGDDING                      0: No Pending IRQ                      1: Channel 0 Voltage High Available Pending IRQ</p> <p>Write '1' to clear this interrupt or automatically clear if interrupt condition fails.</p>

#### 6.7.6.13. GPADC Data Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0038			Register Name: GP_DATA_INTS
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3	R/W1C	0x0	<p>CH3_DATA_PENGDDING                      0: No Pending IRQ                      1: Channel 3 Data Available Pending IRQ</p> <p>Write '1' to clear this interrupt or automatically clear if interrupt condition fails.</p>
2	R/W1C	0x0	<p>CH2_DATA_PENGDDING                      0: No Pending IRQ                      1: Channel 2 Data Available Pending IRQ</p> <p>Write '1' to clear this interrupt or automatically clear if interrupt condition fails.</p>
1	R/W1C	0x0	<p>CH1_DATA_PENGDDING                      0: No Pending IRQ                      1: Channel 1 Data Available Pending IRQ</p> <p>Write '1' to clear this interrupt or automatically clear if interrupt condition fails.</p>
0	R/W1C	0x0	<p>CHO_DATA_PENGDDING                      0: No Pending IRQ                      1: Channel 0 Data Available Pending IRQ</p> <p>Write '1' to clear this interrupt or automatically clear if interrupt condition fails.</p>

#### 6.7.6.14. GPADC CH0 Compare Data Register (Default Value: 0x0BFF\_0400)

Offset: 0x0040			Register Name: GP_CH0_CMP_DATA
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0xBFF	CH0_CMP_HIG_DATA Channel 0 Voltage High Value
15:12	/	/	/
11:0	R/W	0x400	CH0_CMP_LOW_DATA Channel 0 Voltage Low Value

#### 6.7.6.15. GPADC CH1 Compare Data Register (Default Value: 0x0BFF\_0400)

Offset: 0x0044			Register Name: GP_CH1_CMP_DATA
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0xBFF	CH1_CMP_HIG_DATA Channel 1 Voltage High Value
15:12	/	/	/
11:0	R/W	0x400	CH1_CMP_LOW_DATA Channel 1 Voltage Low Value

#### 6.7.6.16. GPADC CH2 Compare Data Register (Default Value: 0x0BFF\_0400)

Offset: 0x0048			Register Name: GP_CH2_CMP_DATA
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0xBFF	CH2_CMP_HIG_DATA Channel 2 Voltage High Value
15:12	/	/	/
11:0	R/W	0x400	CH2_CMP_LOW_DATA Channel 2 Voltage Low Value

#### 6.7.6.17. GPADC CH3 Compare Data Register (Default Value: 0x0BFF\_0400)

Offset: 0x004C			Register Name: GP_CH3_CMP_DATA
Bit	Read/Write	Default/Hex	Description
31:28	/	/	/
27:16	R/W	0xBFF	CH3_CMP_HIG_DATA Channel 3 Voltage High Value
15:12	/	/	/

11:0	R/W	0x400	CH3_CMP_LOW_DATA Channel 3 Voltage Low Value
------	-----	-------	---

#### 6.7.6.18. GPADC CH0 Data Register (Default Value: 0x0000\_0000)

Offset: 0x0080			Register Name: GP_CH0_DATA
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:0	R	0x000	GP_CH0_DATA Channel 0 Data

#### 6.7.6.19. GPADC CH1 Data Register (Default Value: 0x0000\_0000)

Offset: 0x0084			Register Name: GP_CH1_DATA
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:0	R	0x000	GP_CH1_DATA Channel 1 Data

#### 6.7.6.20. GPADC CH2 Data Register (Default Value: 0x0000\_0000)

Offset: 0x0088			Register Name: GP_CH2_DATA
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:0	R	0x0000	GP_CH2_DATA Channel 2 Data

#### 6.7.6.21. GPADC CH3 Data Register (Default Value: 0x0000\_0000)

Offset: 0x008C			Register Name: GP_CH3_DATA
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11:0	R	0x000	GP_CH3_DATA Channel 3 Data

## 6.8. LRADC

### 6.8.1. Overview

The low rate ADC(LRADC) is 6-bit resolution ADC for key application. The LRADC can work up to 2kHz conversion rate.

#### Features:

- One input channel
- 6-bit resolution
- Sample rate up to 2kHz
- Supports hold Key and general Key
- Supports normal,continue and single work mode
- Power supply voltage: AVCC, power reference voltage: 0.75\*AVCC, analog input and detected voltage range: 0~LEVELB(the maximum value is 1.266V)
- Interrupt support

### 6.8.2. Block Diagram

Figure 6-36 shows the block diagram of the LRADC.

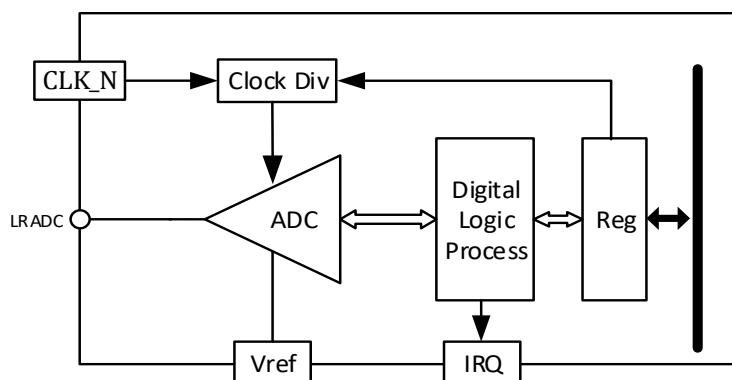


Figure 6- 36. LRADC Block Diagram

### 6.8.3. Operations and Functional Descriptions

#### 6.8.3.1. External Signals

Table 6-19 describes the external signals of the LRADC. The LRADC pin is the analog input signal.

Table 6- 19. LRADC External Signals

Signal	Description	Type
LRADC	Analog Input Channel0	AI

### 6.8.3.2. Clock Sources

The LRADC has one clock source. Table 6-20 describes the clock source for LRADC.

**Table 6- 20. LRADC Clock Sources**

Clock Sources	Description
LOSC	32.768 kHz LOSC

### 6.8.3.3. LRADC Work Mode

#### (1).Normal Mode

ADC gathers 8 samples, the average of the 8 samples is updated in data register, and the data interrupt sign is enabled. It is sampled repeatedly according to this mode until ADC stop.

#### (2).Continue Mode

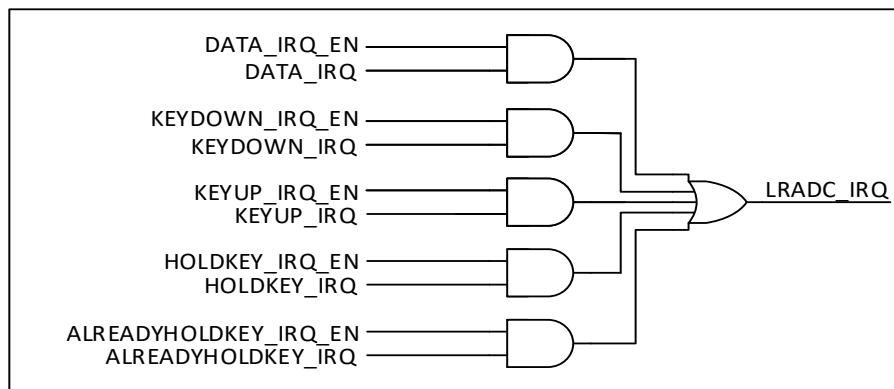
ADC gathers 8 samples every other  $8*(N+1)$  sample cycle. The average of every 8 samples is updated in the data register, and the data interrupt sign is enabled. (N is defined in the bit[19:16] of **LRADC\_CTRL\_REG**).

#### (3).Single Mode

ADC gathers 8 samples, the average of the 8 samples is updated in data register, and the data interrupt sign is enabled, since then ADC stops sample.

### 6.8.3.4. Interrupt

Each LRADC channel has five interrupt sources and five interrupt enable controls.



**Figure 6- 37. LRADC Interrupt**

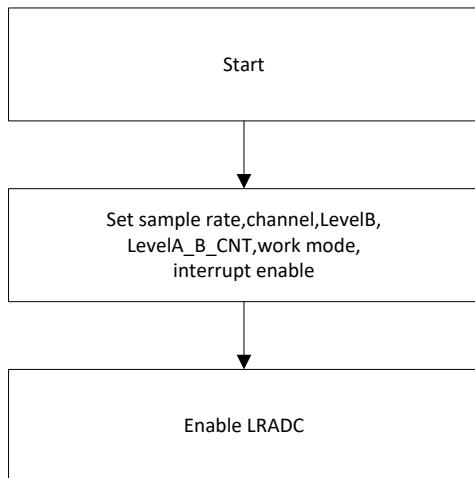
When input voltage is between LEVELA(1.35V) and LEVELB(control by the bit[5:4] of **LRADC\_CTRL**), IRQ1 can be generated. When input voltage is lower than LEVELB, IRQ2 can be generated.

If the controller receives IRQ1, and does not receive IRQ2 at some time, then the controller will generate Hold KEY Interrupt, otherwise DATA\_IRQ Interrupt.

Hold KEY usually is used for self-locking key. When self-locking key holds locking status, the controller receives IRQ2,

then the controller will generate Already Hold Key Interrupt.

#### 6.8.4. Programming Guidelines



**Figure 6- 38. LRADC Initial Process**

- (1) Set CONTINUE\_TIME\_SELECT(LRADC\_CTRL[11:8]) when LRADC works in continue mode.
- (2) The range of input voltage is from 0 to LEVELB(LRADC\_CTRL[5:4]).
- (3) Calculation formula: LRADC\_DATA = Vin/V<sub>REF</sub>\*64, V<sub>REF</sub>=1.35V
- (4) LRADC has 6-bit resolution,1-bit offset error, 1-bit quantizing error. After LRADC calibrates 1-bit offset error, LRADC has 5-bit resolution.

#### 6.8.5. Register List

Module Name	Base Address
LRADC	0x05070800

Register Name	Offset	Description
LRADC_CTRL	0x0000	LRADC Control Register
LRADC_INTC	0x0004	LRADC Interrupt Control Register
LRADC_INTS	0x0008	LRADC Interrupt Status Register
LRADC_DATA	0x000C	LRADC Data Register

#### 6.8.6. Register Description

##### 6.8.6.1. LRADC Control Register (Default Value: 0x0100\_0168)

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

31: 24	R/W	0x1	FIRST_CONVERT_DLY ADC First Convert Delay Setting ADC conversion is delayed by n samples.
23:20	/	/	/
19:16	R/W	0x0	CONTINUE_TIME_SELECT Continue Mode Time Select One of 8*(N+1) sample as a valuable sample data.
15:14	/	/	/
13:12	R/W	0x0	KEY_MODE_SELECT Key Mode Select 00: Normal Mode 01: Single Mode 10: Continue Mode
11:8	R/W	0x1	LEVELA_B_CNT Level A to Level B time threshold select, judge ADC convert value in level A to level B in n+1 samples.
7	R/W	0x0	LRADC_HOLD_KEY_EN LRADC Hold KEY Enable 0: Disable 1: Enable
6	R/W	0x1	LRADC_CHANNEL_EN LRADC Channel Enable 0: Disable 1: Enable
5:4	R/W	0x2	LEVELB_VOL. Level B Corresponding Data Value Setting (the real voltage value) 00: 0x3C (1.266V) 01: 0x39 (1.202V) 10: 0x36 (1.139V) 11: 0x33 (1.076V)
3:2	R/W	0x2	LRADC_SAMPLE_RATE LRADC Sample Rate 00: 2kHz 01: 1kHz 10: 500Hz 11: 250Hz
1	/	/	/
0	R/W	0x0	LRADC_EN LRADC Enable 0: Disable 1: Enable

#### 6.8.6.2. LRADC Interrupt Control Register (Default Value: 0x0000\_0000)

Offset: 0x0004			Register Name: LRADC_INTC
Bit	Read/Write	Default/Hex	Description
31:5	/	/	/
4	R/W	0x0	<b>ADCO_KEYUP_IRQ_EN</b> ADC0 Key Up IRQ Enable 0: Disable 1: Enable
3	R/W	0x0	<b>ADCO_ALRDY_HOLD_IRQ_EN</b> ADC0 Already Hold Key IRQ Enable 0: Disable 1: Enable
2	R/W	0x0	<b>ADCO_HOLD_IRQ_EN</b> ADC0 Hold Key IRQ Enable 0: Disable 1: Enable
1	R/W	0x0	<b>ADCO_KEYDOWN_EN</b> ADC0 Key Down Enable 0: Disable 1: Enable
0	R/W	0x0	<b>ADCO_DATA_IRQ_EN</b> ADC0 Data IRQ Enable 0: Disable 1: Enable

#### 6.8.6.3. LRADC Interrupt Status Register (Default Value: 0x0000\_0000)

Offset: 0x0008			Register Name: LRADC_INTS
Bit	Read/Write	Default/Hex	Description
31:5	/	/	/
4	R/W1C	0x0	<b>ADCO_KEYUP_PENDING</b> ADC0 Key up Pending Bit When general key is pulled up, and the corresponding interrupt is enabled, the status bit is set. 0: No IRQ 1: IRQ Pending Writing 1 to the bit will clear it and its corresponding interrupt if the interrupt is enabled.
3	R/W1C	0x0	<b>ADCO_ALRDY_HOLD_PENDING</b> ADC0 Already Hold Pending Bit When hold key is pulled down and the general key is pulled down, and the corresponding interrupt is enabled. 0: No IRQ

			1: IRQ Pending Writing 1 to the bit will clear it and its corresponding interrupt if the interrupt is enabled.
2	R/W1C	0x0	ADCO_HOLDKEY_PENDING ADCO Hold Key Pending Bit When hold key is pulled down, and the corresponding interrupt is enabled, the status bit is set and the interrupt line is set. 0: NO IRQ 1: IRQ Pending Writing 1 to the bit will clear it and its corresponding interrupt if the interrupt is enabled.
1	R/W1C	0x0	ADCO_KEYDOWN_PENDING ADCO Key Down IRQ Pending Bit When general key is pulled down, and the corresponding interrupt is enabled, the status bit is set and the interrupt line is set. 0: No IRQ 1: IRQ Pending Writing 1 to the bit will clear it and its corresponding interrupt if the interrupt is enabled.
0	R/W1C	0x0	ADCO_DATA_PENDING ADCO Data IRQ Pending Bit 0: No IRQ 1: IRQ Pending Writing 1 to the bit will clear it and its corresponding interrupt if the interrupt is enabled.

#### 6.8.6.4. LRADC Data Register (Default Value: 0x0000\_003F)

Offset: 0x000C			Register Name: LRADC_DATA
Bit	Read/Write	Default/Hex	Description
31:6	/	/	/
5:0	R	0x3F	LRADC_DATA LRADC Data

## 6.9. PWM

### 6.9.1. Overview

The PWM controller has 8 PWM channels(PWM0,PWM1,PWM2,PWM3,PWM4,PWM5,PWM6,PWM7), and divides to 4 PWM pairs:PWM01 pair,PWM23 pair,PWM45 pair,PWM67 pair. PWM01 pair consists of PWM0 and PWM1, PWM23 pair consists of PWM2 and PWM3, PWM45 pair consists of PWM4 and PWM5, PWM67 pair consists of PWM6 and PWM7.

Each PWM channel supports two functions including PWM output and capture input.The clock sources of PWM channel have OSC24M and APB1.PWM channel can output single-pulse waveform or long-period waveform,the frequency range of the output waveform is from 0Hz to 24/100MHz.PWM also can capture input waveform.PWM channel captures the current value of 16-bit adding-counter at the external rising edge, and loads it to Capture Rise Lock Register, PWM channel captures the current value of 16-bit adding-counter at the external falling edge, and loads it to Capture Fall Lock Register,then the frequency of the external clock can be calculated accurately by the value of Capture Rise Lock Register and Capture Fall Lock Register.

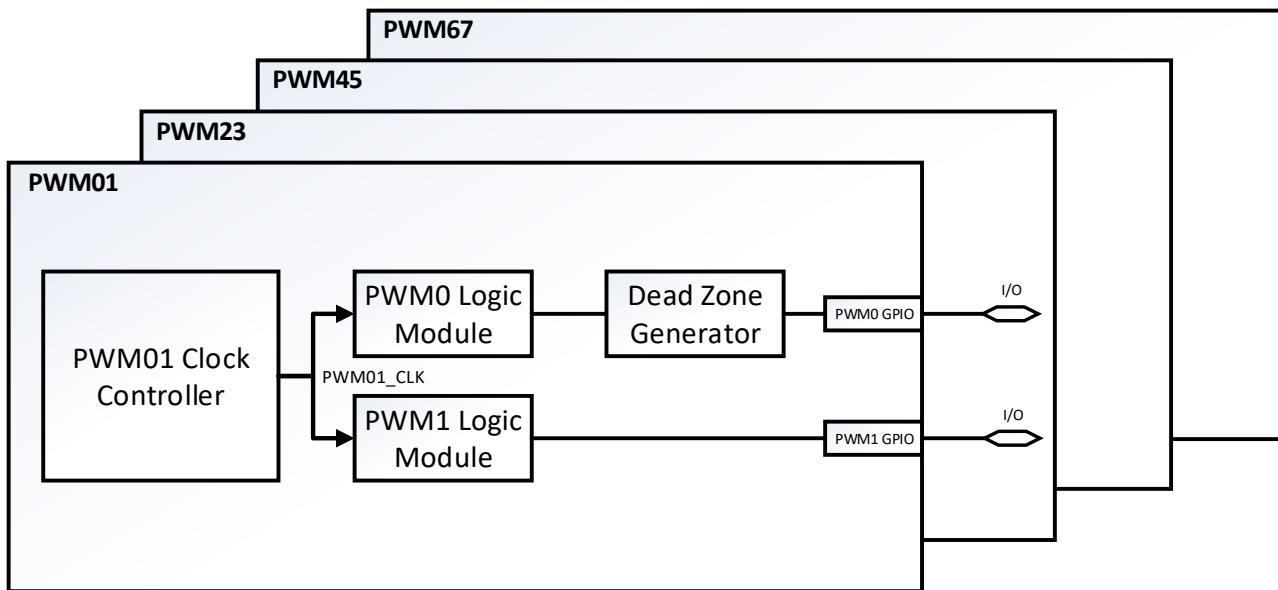
PWM pair can output complementary waveform pair or dead-time PWM pair. When the two channels at a PWM pair have the same prescale, the same period register and opposite active state, then the PWM pair outputs a complementary waveform pair; When the programmable dead-time generator of PWM pair is enabled,then the PWM pair outputs the waveform pair with dead-time, and the dead-time is controllable.

PWM channel can configure to generate interrupt. PWM is as output function, when 16-bit adding-counter is equal to the value of entire cycle, PWM channel can be enabled to generate interrupt. PWM is as input function, when PWM channel captures the external rising edge, PWM channel can be enabled to generate one interrupt; when PWM channel captures the external falling edge, PWM channel can be enabled to generated one interrupt; when PWM channel captures rising edge or falling edge, PWM can trigger interrupt.

The PWM has the following features:

- 8 PWM channels(4 PWM pairs)
- Supports pulse,cycle and complementary pair output
- Supports capture input
- Programming deadzone output
- Build-in the programmable dead-time generator, controllable dead-time
- Three kinds of output waveform: continuous waveform,pulse waveform and complementary pair
- Output frequency range: 0~ 24MHz/100MHz
- Various duty-cycle: 0% ~100%
- Minimum resolution: 1/65536
- Interrupt generation of PWM output and capture input

### 6.9.2. Block Diagram



**Figure 6- 39. PWM Block Diagram**

Each PWM pair consists of 1 clock module, 2 timer logic module and 1 programmable dead-time generator.

### 6.9.3. Operations and Functional Descriptions

#### 6.9.3.1. External Signals

Table 6-21 describes the external signals of the PWM.

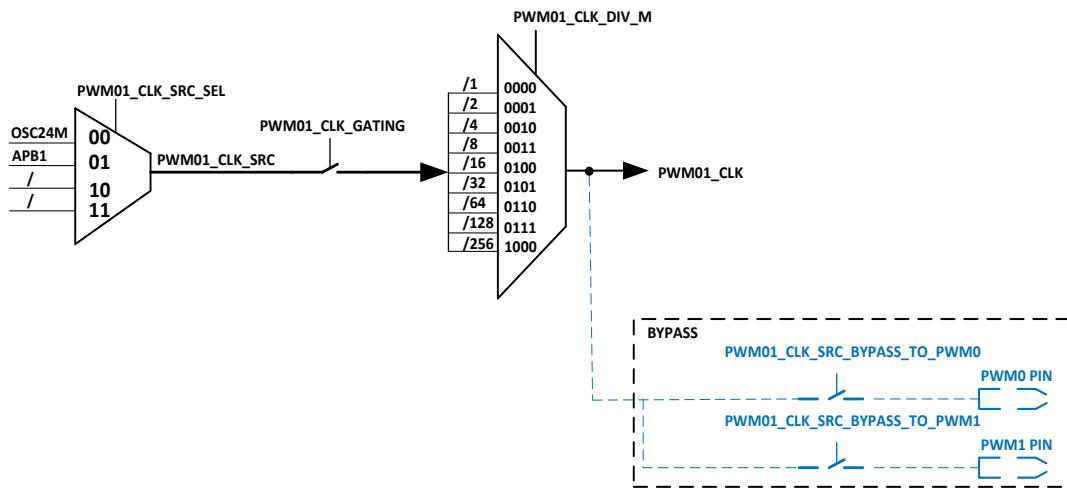
**Table 6- 21. PWM External Signals**

Signal	Description	Type
PWM0	Pulse Width Module Channel0	I/O
PWM1	Pulse Width Module Channel1	I/O
PWM2	Pulse Width Module Channel2	I/O
PWM3	Pulse Width Module Channel3	I/O
PWM4	Pulse Width Module Channel4	I/O
PWM5	Pulse Width Module Channel5	I/O
PWM6	Pulse Width Module Channel6	I/O
PWM7	Pulse Width Module Channel7	I/O

#### 6.9.3.2. Typical Application

- Suitable for display device,such as LCD
- Suitable for electric motor control

### 6.9.3.3. Clock Controller



**Figure 6- 40. PWM01 Clock Controller Diagram**

The clock controller of each PWM pair includes clock source select(PWM01\_CLK\_SRC\_SEL),1~256 scaler(PWM01\_CLK\_DIV\_M), clock source bypass(CLK\_SRC\_BYPASS) and clock switch(PWM01\_CLK\_GATING).

The clock sources of PWM have OSC24M and APB1 Bus. OSC24M comes from external high frequency oscillator, APB1 is APB1 bus clock, usually is 100MHz.

The clock source bypass function is that clock source directly accesses PWM output, the PWM output waveform is the waveform of clock controller output. The BYPASS gridlines in the above figure indicates clock source bypass function, the details about implement, please see Figure 6-41. At last the output clock of the clock controller is sent to PWM logic module.

### 6.9.3.4. PWM Output

Using PWM01 as an example, Figure 6-41 indicates PWM01 output logic module diagram. Other PWM pairs(PWM23, PWM45, PWM67) logic module diagrams are the same as PWM01.

PWM Timer Logic consists of one 16-bit up-counter and one 16-bit comparator. The up-counter is used to control period, and the comparator is used to control duty-cycle. The up-counter and the comparator support cache-loading, PWM output is enabled, the register value of the counter and the comparator can be changed at any time, the changed value is cached to the cache register, when the value of up-counter is equal to **PWM\_ENTIRE\_CYCLE**, the value of the cache register is loaded to the counter and the comparator. Cache-loading is good to avoid unstable PWM output waveform with burr feature when updating the counter value and the comparator value.

PWM supports cycle and pulse waveform output.

**Cycle mode:** When the value of up-counter reaches **PWM\_ENTIRE\_CYCLE**, the value of up-counter is loaded automatically to 0 and the up-counter continues to count, then the output waveform is a continuous waveform.

**Pulse mode:** When the value of up-counter reaches **PWM\_ENTIRE\_CYCLE**, the value of up-counter is loaded automatically to 0 and the up-counter stops counting, then the output waveform is a pulse waveform.

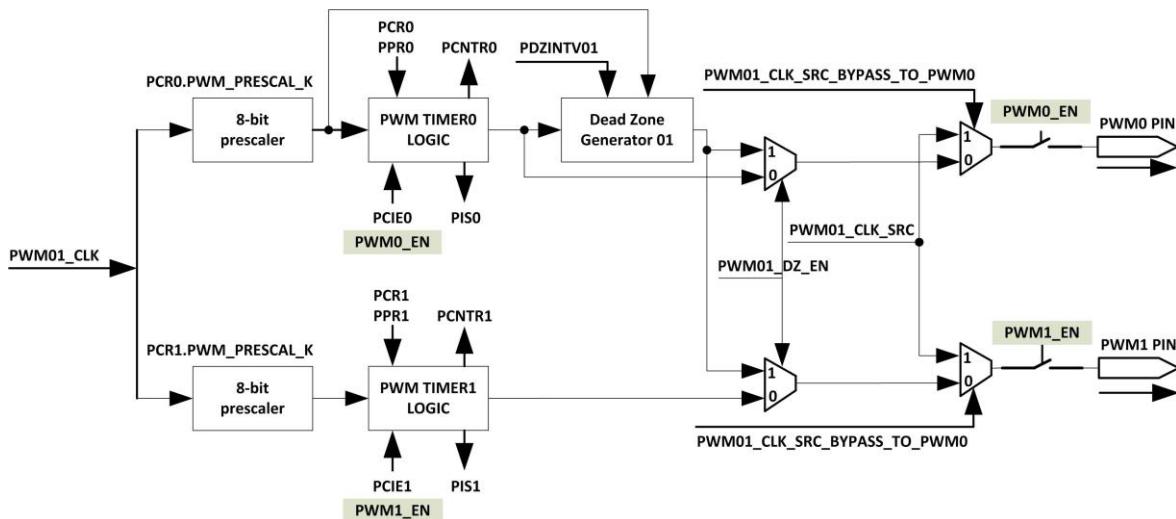


Figure 6-41. PWM01 Output Logic Module Diagram

#### 6.9.3.5. Up-Counter and Comparator

The period, duty-cycle and active state of PWM output waveform are decided by the up-counter and comparator. The rule of the comparator is as follows.

$\text{PCNTR} \geq (\text{PWM\_ENTIRE\_CYCLE} - \text{PWM\_ACT\_CYCLE})$ , output “active state”

$\text{PCNTR} < (\text{PWM\_ENTIRE\_CYCLE} - \text{PWM\_ACT\_CYCLE})$ , output “ $\sim$  (active state)”

##### (1) Active state of PWM0 channel is high level (PCRO. PWM\_ACT\_STA = 1)

When  $\text{PCNTR0} \geq (\text{PPRO. PWM\_ENTIRE\_CYCLE} - \text{PPRO.PWM\_ACT\_CYCLE})$ , then PWM0 outputs 1(high level).

When  $\text{PCNTR0} < (\text{PPRO. PWM\_ENTIRE\_CYCLE} - \text{PPRO.PWM\_ACT\_CYCLE})$ , then PWM0 outputs 0(low level).

The formula of PWM output period and duty-cycle is as follows.

$$T_{\text{period}} = (\text{PWM01\_CLK} / \text{PWMO\_PRESCALE\_K})^{-1} * \text{PPRO.PWM\_ENTIRE\_CYCLE}$$

$$T_{\text{high-level}} = (\text{PWM01\_CLK} / \text{PWMO\_PRESCALE\_K})^{-1} * \text{PPRO.PWM\_ACT\_CYCLE}$$

$$T_{\text{low-level}} = (\text{PWM01\_CLK} / \text{PWMO\_PRESCALE\_K})^{-1} * (\text{PPRO.PWM\_ENTIRE\_CYCLE} - \text{PPRO.PWM\_ACT\_CYCLE})$$

$$\text{Duty-cycle} = (\text{high level time}) / (1 \text{ period time})$$

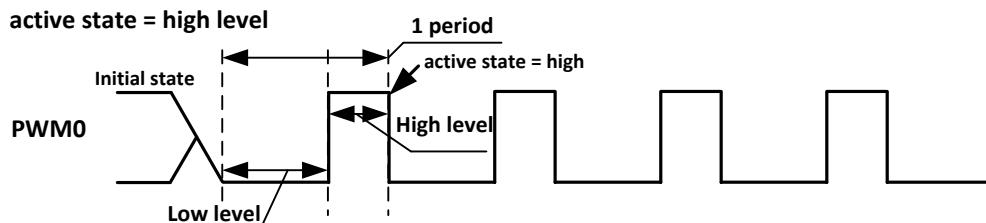


Figure 6-42. PWM0 High Level Active State

## (2) Active state of PWM0 channel is low level (PCR0\_PWM\_ACT\_STA = 0)

When PCNTRO >= (PPR0\_PWM\_ENTIRE\_CYCLE - PPR0\_PWM\_ACT\_CYCLE), then PWM0 outputs 0.

When PCNTRO < (PPR0\_PWM\_ENTIRE\_CYCLE - PPR0\_PWM\_ACT\_CYCLE), then PWM0 outputs 1.

The formula of PWM output period and duty-cycle is as follows.

$$T_{\text{period}} = (\text{PWM01\_CLK} / \text{PWM0\_PREScale\_K})^{-1} * \text{PPR0\_PWM\_ENTIRE\_CYCLE}$$

$$T_{\text{high-level}} = (\text{PWM01\_CLK} / \text{PWM0\_PREScale\_K})^{-1} * (\text{PPR0\_PWM\_ENTIRE\_CYCLE} - \text{PPR0\_PWM\_ACT\_CYCLE})$$

$$T_{\text{low-level}} = (\text{PWM01\_CLK} / \text{PWM0\_PREScale\_K})^{-1} * \text{PPR0\_PWM\_ACT\_CYCLE}$$

$$\text{Duty-cycle} = (\text{high level time}) / (1 \text{ period time})$$

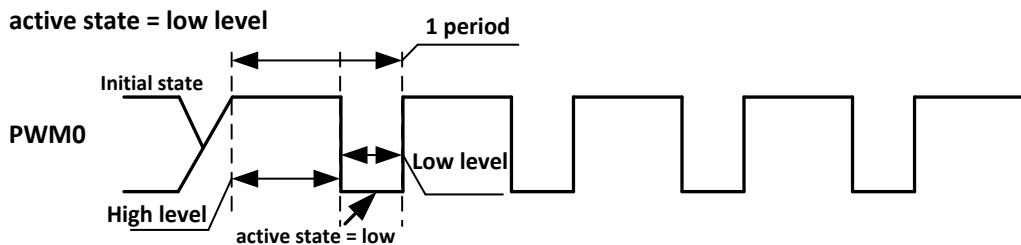


Figure 6- 43. PWM0 Low Level Active State

### 6.9.3.6. Pulse Mode and Cycle Mode

PWM output supports pulse mode and cycle mode. PWM in pulse mode outputs one pulse waveform, but PWM in cycle mode outputs continuous waveform. Figure 6-44 shows the PWM output waveform in pulse mode and cycle mode.

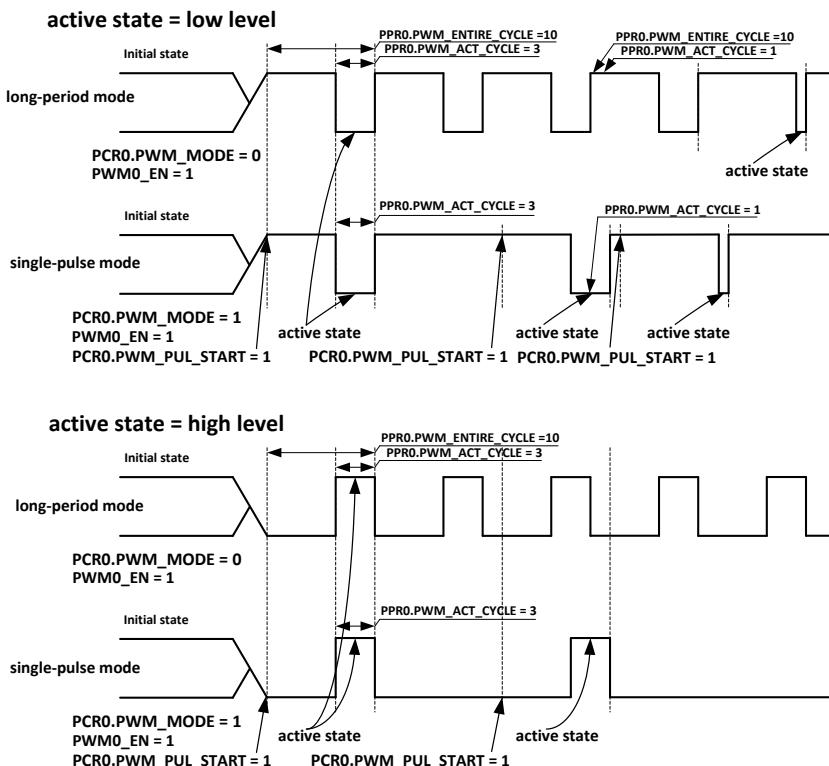


Figure 6- 44. PWM0 Output Waveform in Pulse Mode and Cycle Mode

When PCRO.PWM\_MODE is 0, PWM0 outputs cycle waveform. The calculating formula of  $T_{\text{period}}$  and  $T_{\text{active-state}}$  is as follows.

$$T_{\text{period}} = (\text{PWM01\_CLK} / \text{PWM0\_PRESCALE\_K})^{-1} * \text{PPRO.PWM\_ENTIRE\_CYCLE}$$

$$T_{\text{active state}} = (\text{PWM01\_CLK} / \text{PWM0\_PRESCALE\_K})^{-1} * \text{PPRO.PWM\_ACT\_CYCLE}$$

When PCRO.PWM\_ACT\_STA is 0, the active state of cycle waveform is low level.

When PCRO.PWM\_ACT\_STA is 1, the active state of cycle waveform is high level.

When PCRO.PWM\_MODE is 1, PWM0 outputs pulse waveform. The calculating formula of pulse length is as follows.

$$\text{Pulse length} = \text{PWM01\_CLK} / \text{PWM0\_PRESCALE\_K} * \text{PPRO.PWM\_ACT\_CYCLE}$$

When PCRO.PWM\_ACT\_STA is 0, the pulse level is low level, PWM0 channel outputs low pulse.

When PCRO.PWM\_ACT\_STA is 1, the pulse level is high level, PWM0 channel outputs high pulse.

After PWM0 channel enabled, PCRO.PWM\_PUL\_START need be set to 1 when PWM0 need output pulse waveform, after completed output, PCRO.PWM\_PUL\_START can be cleared to 0 by hardware.

The up-counter and comparator for PWM0 channel support cache loading, after PWM0 channel enabled, whether cycle mode or pulse mode, PPRO value is modified and cached to the buffer register of PPRO, when the up-counter value reaches PPRO.PWM\_ENTIRE\_CYCLE, the value in the buffer register will be loaded to up-counter and comparator, namely the value of up-counter and comparator will be overloaded in the next cycle.

Take Figure 6-44(active state =low level) as an example.

In cycle mode, the initial PPRO.PWM\_ENTIRE\_CYCLE value is 10, the initial PPRO.PWM\_ACT\_CYCLE value is 3. At some time, the value of PPRO.PWM\_ACT\_CYCLE is modified to 1, during the current cycle, the modified PPRO values is cached to PPRO buffer register, at the beginning of the next cycle, the value of PPRO buffer register is loaded into up-counter and comparator, then up-counter starts to work.

In pulse mode, the initial value of PPRO.PWM\_ACT\_CYCLE is 3, in the generation process of a single pulse, the value of PPRO.PWM\_ACT\_CYCLE is modified to 1, during the current cycle, the modified PPRO values is cached to PPRO buffer register, when the value of up-counter reaches PPRO.PWM\_ENTIRE\_CYCLE, then the pulse waveform output ends, the value of PPRO buffer register is loaded into up-counter and comparator, at the next time, after PCRO.PWM\_PUL\_START is set to 1, PPRO modified value has taken effect.

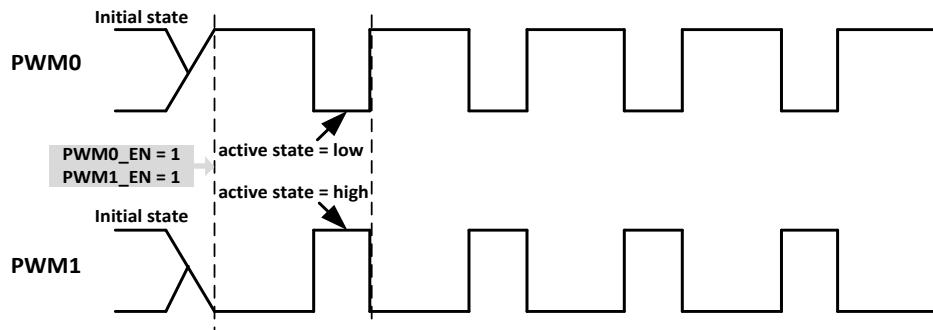


#### NOTE

The time that loading PPRO buffer register value into up-counter and comparator is very short, which can be ignored, and does not affect the PWM output.

#### 6.9.3.7. Complementary Pair Output

Every PWM pair supports complementary pair output and PWM pair with dead-time. Figure 6-45 shows the complementary pair output of PWM01.



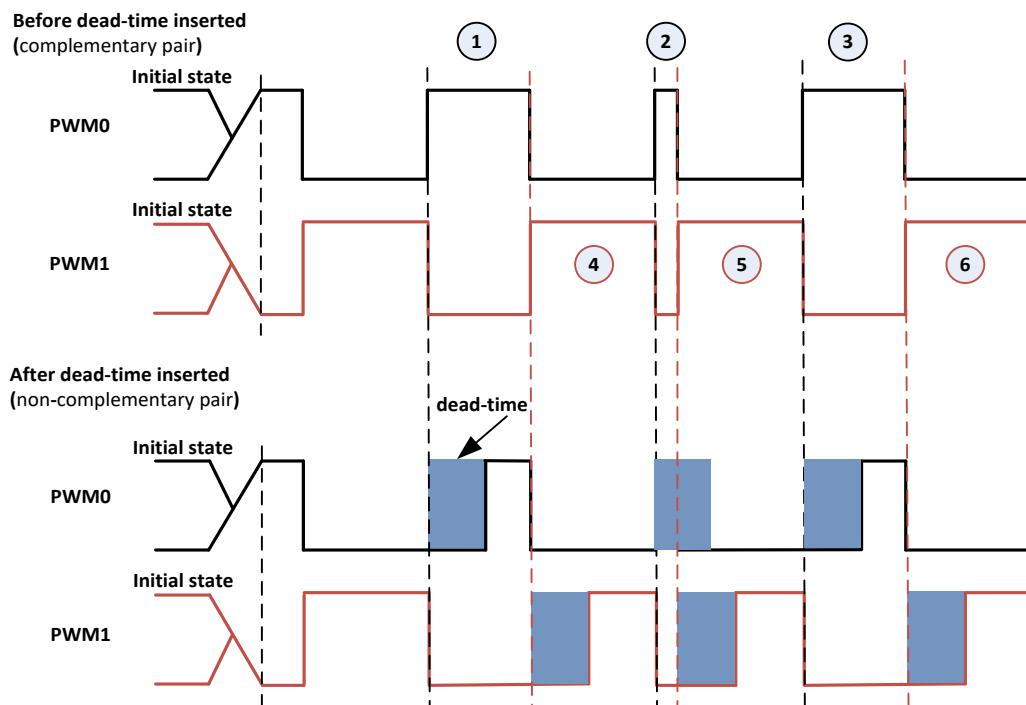
**Figure 6- 45. PWM01 Complementary Pair Output**

The complementary pair output need satisfy the following three conditions:

- The same frequency, the same duty-cycle
- Opposite active state
- Enable two channels of PWM pair at the same time

#### 6.9.3.8. Dead-time Generator

Every PWM pair has a programmable dead-time generator. When the dead-time function of PWM pair enabled, PWM01 output waveform is desided by PWM timer logic and DeadZone Generator. Figure 6-46 shows the output waveform.



**Figure 6- 46. Dead-time Output Waveform**

Before dead-time inserted: a complementary waveform pair of non-inserted dead-time in Dead Zone Generator 01.

After dead-time inserted: 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 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 high level time for mark② in the above figure is less than dead-time,then dead-time will override the high level.The setting of dead-time need consider the period and duty-cycle of output waveform.Dead-time formula is defined as follows:

$$\text{Dead-time} = (\text{PWM01\_CLK} / \text{PWM0\_PRESCALE\_K})^{-1} * \text{PDZINTV01}$$

#### 6.9.3.9. Capture Input

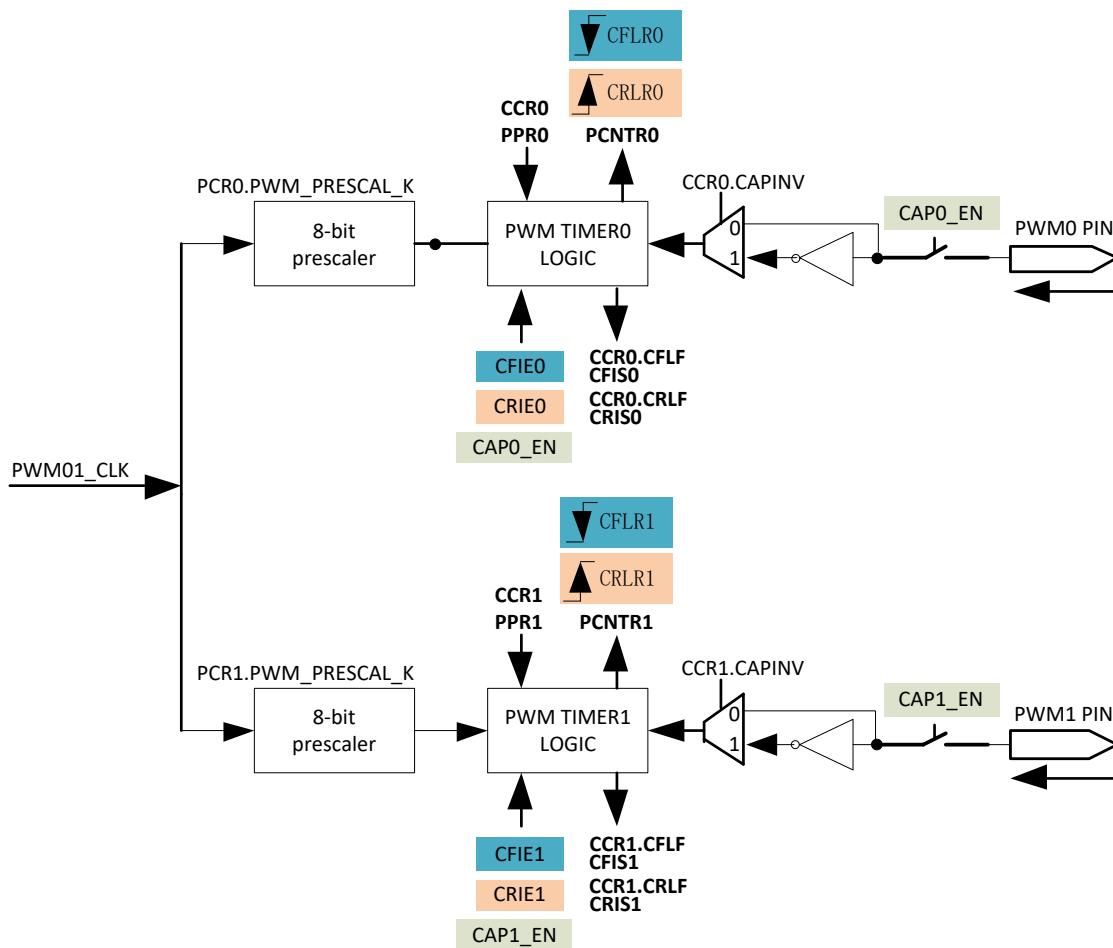


Figure 6- 47. PWM01 Capture Logic Module Diagram

Besides the timer logic module of every PWM channel generates PWM output, it can be used to capture rising edge and falling edge of the external clock.Using PWM0 channel as an example,PWM0 channel has one **CFLR** and one **CRLR** for capturing up-counter value in falling edge, in rising edge,respectively. You can calculate the period of external clock by **CFLR** and **CRLR**.

$$T_{\text{high-level}} = (\text{PWM01\_CLK} / \text{PWM0\_PRESCALE\_K})^{-1} * \text{CRLR0}$$

$$T_{\text{low-level}} = (\text{PWM01\_CLK} / \text{PWM0\_PRESCALE\_K})^{-1} * \text{CFLR0}$$

$$T_{\text{period}} = T_{\text{high-level}} + T_{\text{low-level}}$$

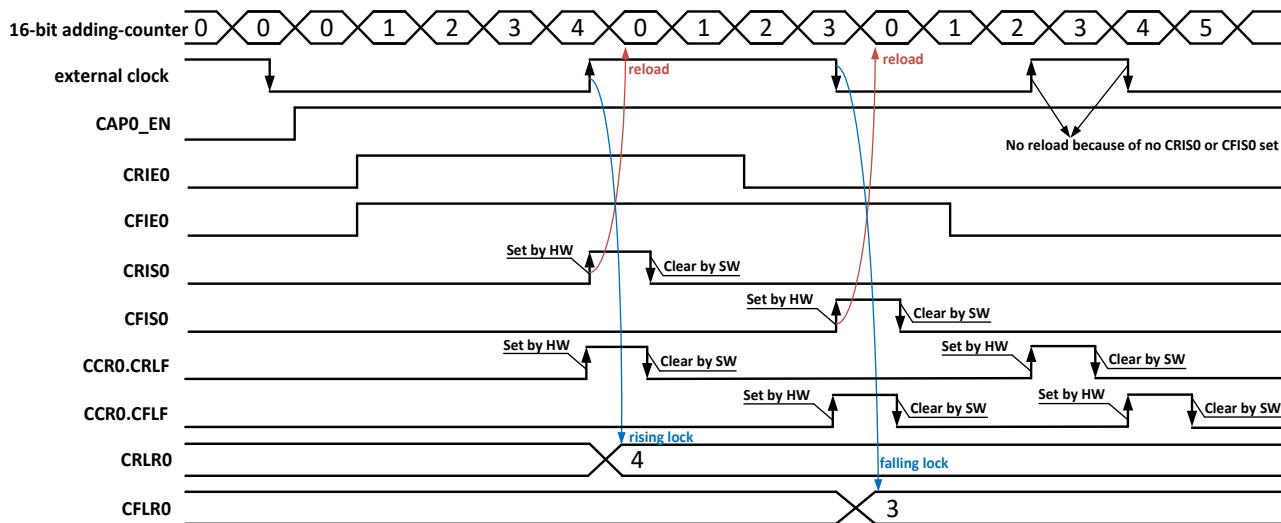


Figure 6-48. PWM0 Channel Capture Timing

When the capture input function of PWM channel is enabled, the up-counter of PWM0 channel starts to work.

When the timer logic module of PWM0 captures one rising edge, the current value of up-counter is locked to **CRLR**, and **CRLF** is set to 1 . If **CRIEO** is 1, then **CRISO** is set to 1, PWM0 channel sends interrupt request, and the up-counter is loaded to 0 and continues to count. If **CRIEO** is 0, the up-counter is not loaded to 0.

When the timer logic module of PWM0 captures one falling edge, the current value of up-counter is locked to **CFLR**, and **CFLF** is set to 1 . If **CFIEO** is 1, then **CFISO** is set to 1, PWM0 channel sends interrupt request, and the up-counter is loaded to 0 and continues to count. If **CFIEO** is 0, the up-counter is not loaded to 0.

#### 6.9.3.10. Interrupt

PWM supports interrupt generation when PWM channel is configured to PWM output or capture input .

For PWM output function, whether pulse mode or cycle mode, if the value of the up-counter reaches **PWM\_ENTIRE\_CYCLE**, the timer logic module will automatically set the PIS(PWM Interrupt Status) bit to 1 by hardware. But the PIS bit is cleared by software.

For capture input function, when the timer logic module of the capture channel0 captures rising edge, and **CRIEO** is 1, then **CRISO** is set to 1; when the timer logic module of the capture channel0 captures falling edge, and **CFIEO** is 1, then **CFISO** is set to 1.

#### 6.9.4. Register List

Module Name	Base Address
PWM	0x0300A000

Register Name	Offset	Description
PIER	0x0000	PWM IRQ Enable Register
PISR	0x0004	PWM IRQ Status Register
CIER	0x0010	Capture IRQ Enable Register
CISR	0x0014	Capture IRQ Status Register
PCCR01	0x0020	PWM01 Clock Configuration Register
PCCR23	0x0024	PWM23 Clock Configuration Register
PCCR45	0x0028	PWM45 Clock Configuration Register
PCCR67	0x002C	PWM67 Clock Configuration Register
PDZCR01	0x0030	PWM01 Dead Zone Control Register
PDZCR23	0x0034	PWM23 Dead Zone Control Register
PDZCR45	0x0038	PWM45 Dead Zone Control Register
PDZCR67	0x003C	PWM67 Dead Zone Control Register
PER	0x0040	PWM Enable Register
CER	0x0044	Capture Enable Register
PCR	0x0060+0x00+N*0x20(N=0~7)	PWM Control Register
PPR	0x0060+0x04+N*0x20(N=0~7)	PWM Period Register
PCNTR	0x0060+0x08+N*0x20(N=0~7)	PWM Count Register
CCR	0x0060+0x0C+N*0x20(N=0~7)	Capture Control Register
CRLR	0x0060+0x10+N*0x20(N=0~7)	Capture Rise Lock Register
CFLR	0x0060+0x14+N*0x20(N=0~7)	Capture Fall Lock Register

### 6.9.5. Register Description

#### 6.9.5.1. PWM IRQ Enable Register (Default Value: 0x0000\_0000)

Offset:0x0000			Register Name: PWM_IRQ_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
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
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

#### 6.9.5.2. PWM IRQ Status Register (Default Value: 0x0000\_0000)

Offset:0x0004			Register Name: PWM_IRQ_STATUS_REG
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W1C	0x0	PIS7 PWM Channel 7 Interrupt Status When PWM channel 7 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 7 interrupt is not pending. Reads 1: PWM channel 7 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 7 interrupt status.
6	R/W1C	0x0	PIS6 PWM Channel 6 Interrupt Status When PWM channel 6 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 6 interrupt is not pending. Reads 1: PWM channel 6 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 6 interrupt status.
5	R/W1C	0x0	PIS5 PWM Channel 5 Interrupt Status When PWM channel 5 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit.

			Reads 0: PWM channel 5 interrupt is not pending. Reads 1: PWM channel 5 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 5 interrupt status.
4	R/W1C	0x0	PIS4 PWM Channel 4 Interrupt Status When PWM channel 4 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 4 interrupt is not pending. Reads 1: PWM channel 4 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 4 interrupt status.
3	R/W1C	0x0	PIS3 PWM Channel 3 Interrupt Status When PWM channel 3 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 3 interrupt is not pending. Reads 1: PWM channel 3 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 3 interrupt status.
2	R/W1C	0x0	PIS2 PWM Channel 2 Interrupt Status When PWM channel 2 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 2 interrupt is not pending. Reads 1: PWM channel 2 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 2 interrupt status.
1	R/W1C	0x0	PIS1 PWM Channel 1 Interrupt Status When PWM channel 1 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 1 interrupt is not pending. Reads 1: PWM channel 1 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 1 interrupt status.
0	R/W1C	0x0	PISO PWM Channel 0 Interrupt Status When PWM channel 0 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: PWM channel 0 interrupt is not pending. Reads 1: PWM channel 0 interrupt is pending. Writes 0: No effect. Writes 1: Clear PWM channel 0 interrupt status.

### 6.9.5.3. PWM Capture IRQ Enable Register (Default Value: 0x0000\_0000)

Offset:0x0010			Register Name: PWM_CAPTURE_IRQ_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R/W	0x0	CFIE7 If the bit is set 1, when capturing channel 7 captures falling edge, it generates a capturing channel 7 pending. 0: Capturing channel 7 fall lock interrupt disable 1: Capturing channel 7 fall lock interrupt enable
14	R/W	0x0	CRIE7 If the bit is set 1, when capturing channel 7 captures rising edge, it generates a capturing channel 7 pending. 0: Capturing channel 7 rise lock interrupt disable 1: Capturing channel 7 rise lock interrupt enable
13	R/W	0x0	CFIE6 If the bit is set 1, when capturing channel 6 captures falling edge, it generates a capturing channel 6 pending. 0: Capturing channel 6 fall lock interrupt disable 1: Capturing channel 6 fall lock interrupt enable
12	R/W	0x0	CRIE6 If the bit is set 1, when capturing channel 6 captures rising edge, it generates a capturing channel 6 pending. 0: Capturing channel 6 rise lock interrupt disable 1: Capturing channel 6 rise lock interrupt enable
11	R/W	0x0	CFIE5 If the bit is set 1, when capturing channel 5 captures falling edge, it generates a capturing channel 5 pending. 0: Capturing channel 5 fall lock interrupt disable 1: Capturing channel 5 fall lock interrupt enable
10	R/W	0x0	CRIE5 If the bit is set 1, when capturing channel 5 captures rising edge, it generates a capturing channel 5 pending. 0: Capturing channel 5 rise lock interrupt disable 1: Capturing channel 5 rise lock interrupt enable
9	R/W	0x0	CFIE4 If the bit is set 1, when capturing channel 4 captures falling edge, it generates a capturing channel 4 pending. 0: Capture channel 4 fall lock interrupt disable 1: Capture channel 4 fall lock interrupt enable
8	R/W	0x0	CRIE4 If the bit is set 1, when capturing channel 4 captures rising edge, it generates a capturing channel 4 pending. 0: Capturing channel 4 rise lock interrupt disable 1: Capturing channel 4 rise lock interrupt enable

7	R/W	0x0	CFIE3 If the bit is set 1, when capturing channel 3 captures falling edge, it generates a capturing channel 3 pending. 0: Capturing channel 3 fall lock interrupt disable 1: Capturing channel 3 fall lock interrupt enable
6	R/W	0x0	CRIE3 If the bit is set 1, when capturing channel 3 captures rising edge, it generates a capturing channel 3 pending. 0: Capturing channel 3 rise lock interrupt disable 1: Capturing channel 3 rise lock interrupt enable
5	R/W	0x0	CFIE2 If the bit is set 1, when capturing channel 2 captures falling edge, it generates a capturing channel 2 pending. 0: Capturing channel 2 fall lock interrupt disable 1: Capturing channel 2 fall lock interrupt enable
4	R/W	0x0	CRIE2 If the bit is set 1, when capturing channel 2 captures rising edge, it generates a capturing channel 2 pending. 0: Capturing channel 2 rise lock interrupt disable 1: Capturing channel 2 rise lock interrupt enable
3	R/W	0x0	CFIE1 If the bit is set 1, when capturing channel 1 captures falling edge, it generates a capturing channel 1 pending. 0: Capturing channel 1 fall lock interrupt disable 1: Capturing channel 1 fall lock interrupt enable
2	R/W	0x0	CRIE1 If the bit is set 1, when capturing channel 1 captures rising edge, it generates a capturing channel 1 pending. 0: Capturing channel 1 rise lock interrupt disable 1: Capturing channel 1 rise lock interrupt enable
1	R/W	0x0	CFIE0 If the bit is set 1, when capturing channel 0 captures falling edge, it generates a capturing channel 0 pending. 0: Capturing channel 0 fall lock interrupt disable 1: Capturing channel 0 fall lock interrupt enable
0	R/W	0x0	CRIE0 If the bit is set 1, when capturing channel 0 captures rising edge, it generates a capturing channel 0 pending. 0: Capturing channel 0 rise lock interrupt disable 1: Capturing channel 0 rise lock interrupt enable

#### 6.9.5.4. PWM Capture IRQ Status Register (Default Value: 0x0000\_0000)

Offset:0x0014	Register Name: PWM_CAPTURE_IRQ_STATUS_REG
---------------	---

Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R/W1C	0x0	<p>CFIS7 Capturing channel 7 falling lock interrupt status. When capturing channel 7 captures falling edge, if capturing channel 7 fall lock interrupt (<b>CFIE7</b>) is enabled, this bit is set 1 by hardware.Writing 1 to clear this bit.</p> <p>Reads 0: Capturing channel 7 interrupt is not pending. Reads 1: Capturing channel 7 interrupt is pending. Writes 0: No effect. Writes 1: Clear capturing channel 7 interrupt status.</p>
14	R/W1C	0x0	<p>CRIS7 Capturing channel 7 rising lock interrupt status. When capturing channel 7 captures rising edge, if capturing channel 7 rise lock interrupt (<b>CRIE7</b>) is enabled, this bit is set 1 by hardware. Write 1 to clear this bit.</p> <p>Reads 0: Capturing channel 7 interrupt is not pending. Reads 1: Capturing channel 7 interrupt is pending. Writes 0: No effect. Writes 1: Clear capture channel 7 interrupt status.</p>
13	R/W1C	0x0	<p>CFIS6 Capturing channel 6 falling lock interrupt status. When capturing channel 6 captures falling edge, if capturing channel 6 fall lock interrupt (<b>CFIE6</b>) is enabled, this bit is set 1 by hardware.Writing 1 to clear this bit.</p> <p>Reads 0: Capturing channel 6 interrupt is not pending. Reads 1: Capturing channel 6 interrupt is pending. Writes 0: No effect. Writes 1: Clear capturing channel 6 interrupt status.</p>
12	R/W1C	0x0	<p>CRIS6 Capturing channel 6 rising lock interrupt status. When capturing channel 6 captures rising edge, if capturing channel 6 rise lock interrupt (<b>CRIE6</b>) is enabled, this bit is set 1 by hardware. Writing 1 to clear this bit.</p> <p>Reads 0: Capturing channel 6 interrupt is not pending. Reads 1: Capturing channel 6 interrupt is pending. Writes 0: No effect. Writes 1: Clear capturing channel 6 interrupt status.</p>
11	R/W1C	0x0	<p>CFIS5 Capturing channel 5 falling lock interrupt status. When capturing channel 5 captures falling edge, if capturing channel 5 fall lock interrupt (<b>CFIE5</b>) is enabled, this bit is set 1 by hardware.Writing 1 to clear this bit.</p> <p>Reads 0: Capturing channel 5 interrupt is not pending. Reads 1: Capturing channel 5 interrupt is pending.</p>

			Writes 0: No effect. Reads 1: Clear capturing channel 5 interrupt status.
10	R/W1C	0x0	CRIS5 Capturing channel 5 rising lock interrupt status. When capturing channel 5 captures rising edge, if capturing channel 5 rise lock interrupt ( <b>CRIES5</b> ) is enabled, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: Capturing channel 5 interrupt is not pending. Reads 1: Capturing channel 5 interrupt is pending. Writes 0: No effect. Writes 1: Clear capturing channel 5 interrupt status.
9	R/W1C	0x0	CFIS4 Capturing channel 4 falling lock interrupt status. When capturing channel 4 captures falling edge, if capturing channel 4 fall lock interrupt ( <b>CFIE4</b> ) is enabled, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: Capturing channel 4 interrupt is not pending. Reads 1: Capturing channel 4 interrupt is pending. Writes 0: No effect. Writes 1: Clear capturing channel 4 interrupt status.
8	R/W1C	0x0	CRIS4 Capturing channel 4 rising lock interrupt status. When capturing channel 4 captures rising edge, if capturing channel 4 rise lock interrupt ( <b>CRIE4</b> ) is enabled, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: Capturing channel 4 interrupt is not pending. Reads 1: Capturing channel 4 interrupt is pending. Writes 0: No effect. Writes 1: Clear capturing channel 4 interrupt status.
7	R/W1C	0x0	CFIS3 Capture channel 3 falling lock interrupt status. When capture channel 3 captures falling edge, if capture channel 3 fall lock interrupt ( <b>CFIE3</b> ) is enabled, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: Capture channel 3 interrupt is not pending. Reads 1: Capture channel 3 interrupt is pending. Writes 0: no effect. Writes 1: Clear capture channel 3 interrupt status.
6	R/W1C	0x0	CRIS3 Capture channel 3 rising lock interrupt status. When capture channel 3 captures rising edge, if capture channel 3 rise lock interrupt ( <b>CRIE3</b> ) is enabled, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: Capture channel 3 interrupt is not pending. Reads 1: Capture channel 3 interrupt is pending.

			Writes 0: no effect. Writes 1: Clear capture channel 3 interrupt status.
5	R/W1C	0x0	CFIS2 Capture channel 2 falling lock interrupt status. When capture channel 2 captures falling edge, if capture channel 2 fall lock interrupt ( <b>CFIE2</b> ) is enabled, this bit is set 1 by hardware.Writing 1 to clear this bit. Reads 0: Capture channel 2 interrupt is not pending. Reads 1: Capture channel 2 interrupt is pending. Writes 0: no effect. Writes 1: Clear capture channel 2 interrupt status.
4	R/W1C	0x0	CRIS2 Capture channel 2 rising lock interrupt status. When capture channel 2 captures rising edge, if capture channel 2 rise lock interrupt ( <b>CRIE2</b> ) is enabled, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: Capture channel 2 interrupt is not pending. Reads 1: Capture channel 2 interrupt is pending. Writes 0: no effect. Writes 1: Clear capture channel 2 interrupt status.
3	R/W1C	0x0	CFIS1 Capture channel 1 falling lock interrupt status. When capture channel 1 captures falling edge, if capture channel 1 fall lock interrupt ( <b>CFIE1</b> ) is enabled, this bit is set 1 by hardware.Writing 1 to clear this bit. Reads 0: Capture channel 1 interrupt is not pending. Reads 1: Capture channel 1 interrupt is pending. Writes 0: no effect. Writes 1: Clear capture channel 1 interrupt status.
2	R/W1C	0x0	CRIS1 Capture channel 1 rising lock interrupt status. When capture channel 1 captures rising edge, if capture channel 1 rise lock interrupt ( <b>CRIE1</b> ) is enabled, this bit is set 1 by hardware. Writing 1 to clear this bit. Reads 0: Capture channel 1 interrupt is not pending. Reads 1: Capture channel 1 interrupt is pending. Writes 0: no effect. Writes 1: Clear capture channel 1 interrupt status.
1	R/W1C	0x0	CFISO Capture channel 0 falling lock interrupt status. When capture channel 0 captures falling edge, if capture channel 0 fall lock interrupt ( <b>CFIE0</b> ) is enabled, this bit is set 1 by hardware.Writing 1 to clear this bit. Reads 0: Capture channel 0 interrupt is not pending. Reads 1: Capture channel 0 interrupt is pending.

			Writes 0: no effect. Writes 1: Clear capture channel 0 interrupt status.
0	R/W1C	0x0	<p>CRISO Capture channel 0 rising lock interrupt status. When capture channel 0 captures rising edge, if capture channel 0 rise lock interrupt (<b>CRIE0</b>) is enabled, this bit is set 1 by hardware. Writing 1 to clear this bit.</p> <p>Reads 0: Capture channel 0 interrupt is not pending. Reads 1: Capture channel 0 interrupt is pending. Writes 0: no effect. Writes 1: Clear capture channel 0 interrupt status.</p>

#### 6.9.5.5. PWM01 Clock Configuration Register (Default Value: 0x0000\_0000)

Offset:0x0020			Register Name: PWM01_CLOCK_CFG_REG
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8:7	R/W	0x0	<p>PWM01_CLK_SRC Select PWM01 Clock Source 00: OSC24M 01: APB1 Others: Reserved</p>
6	R/W	0x0	<p>PWM01_CLK_SRC_BYPASS_TO_PWM1 Bypass PWM01 Clock Source to PWM1 Output 0: Not bypass 1: Bypass</p>
5	R/W	0x0	<p>PWM01_CLK_SRC_BYPASS_TO_PWM0 Bypass PWM01 Clock Source to PWM0 Output 0: Not bypass 1: Bypass</p>
4	R/W	0x0	<p>PWM01_CLK_GATING Gating Clock for PWM01 0: Mask 1: Pass</p>
3:0	R/W	0x0	<p>PWM01_CLK_DIV_M PWM01 Clock Divide M 0000: /1 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128</p>

			1000: /256 others: Reserved
--	--	--	--------------------------------

#### 6.9.5.6. PWM23 Clock Configuration Register (Default Value: 0x0000\_0000)

Offset:0x0024			Register Name: PWM23_CLOCK_CFG_REG
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8:7	R/W	0x0	PWM23_CLK_SRC_SEL Select PWM23 Clock Source 00: OSC24M 01: APB1 Others: Reserved
6	R/W	0x0	PWM23_CLK_SRC_BYPASS_TO_PWM3 Bypass PWM23 Clock Source to PWM3 Output 0: Not bypass 1: Bypass
5	R/W	0x0	PWM23_CLK_SRC_BYPASS_TO_PWM2 Bypass PWM23 Clock Source to PWM2 Output 0: Not bypass 1: Bypass
4	R/W	0x0	PWM23_CLK_GATING Gating Clock for PWM23 0: Mask 1: Pass
3:0	R/W	0x0	PWM23_CLK_DIV_M PWM23 Clock Divide M 0000: /1 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 others: Reserved

#### 6.9.5.7. PWM45 Clock Configuration Register (Default Value: 0x0000\_0000)

Offset:0x0028			Register Name: PWM45_CLOCK_CFG_REG
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/

8:7	R/W	0x0	PWM45_CLK_SRC_SEL Select PWM45 Clock Source 00: OSC24M 01: APB1 Others: Reserved
6	R/W	0x0	PWM45_CLK_SRC_BYPASS_TO_PWM5 Bypass PWM45 Clock Source to PWM5 Output 0: Not bypass 1: Bypass
5	R/W	0x0	PWM45_CLK_SRC_BYPASS_TO_PWM4 Bypass PWM45 Clock Source to PWM4 Output 0: Not bypass 1: Bypass
4	R/W	0x0	PWM45_CLK_GATING Gating Clock for PWM45 0: Mask 1: Pass
3:0	R/W	0x0	PWM45_CLK_DIV_M PWM45 Clock Divide M 0000: /1 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 others: Reserved

#### 6.9.5.8. PWM67 Clock Configuration Register (Default Value: 0x0000\_0000)

Offset:0x002C			Register Name: PWM67_CLOCK_CFG_REG
Bit	Read/Write	Default/Hex	Description
31:9	/	/	/
8:7	R/W	0x0	PWM67_CLK_SRC_SEL Select PWM67 Clock Source 00: OSC24M 01: APB1 Others: Reserved
6	R/W	0x0	PWM67_CLK_SRC_BYPASS_TO_PWM7 Bypass PWM67 Clock Source to PWM7 Output 0: Not bypass 1: Bypass

5	R/W	0x0	PWM67_CLK_SRC_BYPASS_TO_PWM6 Bypass PWM67 Clock Source to PWM6 Output 0: Not bypass 1: Bypass
4	R/W	0x0	PWM67_CLK_GATING Gating Clock for PWM67 0: Mask 1: Pass
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

#### 6.9.5.9. PWM01 Dead Zone Control Register (Default Value: 0x0000\_0000)

Offset:0x0030			Register Name: PWM01_CLOCK_CFG_REG
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

#### 6.9.5.10. PWM23 Dead Zone Control Register (Default Value: 0x0000\_0000)

Offset:0x0034			Register Name: PWM23_CLOCK_CFG_REG
Bit	Read/Write	Default/Hex	Description
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
--	--	--	---

#### 6.9.5.11. PWM45 Dead Zone Control Register (Default Value: 0x0000\_0000)

Offset:0x0038			Register Name: PWM45_CLOCK_CFG_REG
Bit	Read/Write	Default/Hex	Description
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

#### 6.9.5.12. PWM67 Dead Zone Control Register (Default Value: 0x0000\_0000)

Offset:0x003C			Register Name: PWM67_CLOCK_CFG_REG
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

#### 6.9.5.13. PWM Enable Register (Default Value: 0x0000\_0000)

Offset:0x0040			Register Name: PWM_ENABLE_REG
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
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

#### 6.9.5.14. PWM Capture Enable Register (Default Value: 0x0000\_0000)

Offset:0x0044			Register Name: PWM_CAPTURE_ENABLE_REG
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	CAP7_EN When capture function is enabled,the 16-bit up-counter starts working and 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 capture function is enabled, the 16-bit up-counter starts working and 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 capture function is enabled, the 16-bit up-counter starts working and 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 capture function is enabled, the 16-bit up-counter starts working and capture channel4 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable
3	R/W	0x0	CAP3_EN When capture function is enabled, the 16-bit up-counter starts working and capture channel3 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable
2	R/W	0x0	CAP2_EN When capture function is enabled, the 16-bit up-counter starts working and capture channel2 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable
1	R/W	0x0	CAP1_EN When capture function is enabled, the 16-bit up-counter starts working and capture channel1 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable
0	R/W	0x0	CAP0_EN When capture function is enabled, the 16-bit up-counter starts working and capture channel0 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable

#### 6.9.5.15. PWM Control Register (Default Value: 0x0000\_0000)

Offset:0x0060+0x0+N*0x20(N=0~7)			Register Name: PWM_CTL_REG
Bit	Read/Write	Default/Hex	Description
31:12	/	/	/
11	R	0x0	PWM_PERIOD_RDY PWM Period Register Ready 0: PWM period register is ready to write 1: PWM period register is busy
10	R/WAC	0x0	PWM_PUL_START PWM Pulse Output Start 0: No effect 1: Output 1 pulse After finishing configuration for outputting pulse, set this bit once and then PWM would output one pulse. After the pulse is finished, the bit will be cleared automatically.
9	R/W	0x0	PWM_MODE PWM Output Mode Select 0: Cycle mode 1: Pulse mode
8	R/W	0x0	PWM_ACT_STA PWM Active State 0: Low Level 1: High Level
7:0	R/W	0x0	PWM_PRESCAL_K PWM pre-scale K, actual pre-scale is (K+1). K = 0, actual pre-scale: 1 K = 1, actual pre-scale: 2 K = 2, actual pre-scale: 3 K = 3, actual pre-scale: 4 ..... K = 255, actual pre-scale: 256

#### 6.9.5.16. PWM Period Register

Offset:0x0060+0x04+N*0x20(N=0~7)			Register Name: PWM_PRD_REG
Bit	Read/Write	Default/Hex	Description
31:16	R/W	UDF	PWM_ENTIRE_CYCLE Number of the entire cycles in the PWM clock. 0: 1 cycle 1: 2 cycles ... N: N+1 cycles If the register need to be modified dynamically, the PCLK should be faster

			than the PWM CLK.
15:0	R/W	UDF	<p>PWM_ACT_CYCLE                      Number of the active cycles in the PWM clock.                      0: 0 cycle                      1: 1 cycle                      ...                      N: N cycles</p>

#### 6.9.5.17. PWM Counter Register

Offset:0x0060+0x08+N*0x20(N=0~7)			Register Name: PWM_CTR_REG
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R	UDF	<p>PWM output or capture input.                      The field indicates the current value of the PWM 16-bit up-counter.</p>

#### 6.9.5.18. PWM Capture Control Register (Default Value: 0x0000\_0000)

Offset:0x0060+0x0C+N*0x20(N=0~7)			Register Name: PWM_CAPTURE_CTL_REG
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2	R/W1C	0x0	<p>CRLF                      When capturing channel captures the rising edge, the current value of the 16-bit up-counter is latched to <b>CRLR</b> and the bit is set to 1 by hardware. Writing 1 to clear the bit.</p>
1	R/W1C	0x0	<p>CFLF                      When capturing channel captures the falling edge, the current value of the 16-bit up-counter is latched to <b>CFLR</b> and the bit is set to 1 by hardware. Writing 1 to clear the bit.</p>
0	R/W	0x0	<p>CAPINV                      Inverting the input signal from the capturing channel before the 16-bit counter of the capturing channel.                      0: Not inverse                      1: Inverse</p>

#### 6.9.5.19. PWM Capture Rise Lock Register

Offset:0x0060+0x10+N*0x20(N=0~7)			Register Name: PWM_CAPTURE_RISE_LOCK_REG
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R	UDF	CRLR

			When the capturing channel captures the rising edge, the current value of the 16-bit up-counter is latched to the register.
--	--	--	---

#### 6.9.5.20. PWM Capture Fall Lock Register

Offset:0x0060+0x14+N*0x20(N=0~7)		Register Name: PWM_CAPTURE_FALL_LOCK_REG	
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R	UDF	CFLR When the capturing channel captures the falling edge, the current value of the 16-bit up-counter is latched to the register.

## 6.10. LEDC

### 6.10.1. Overview

The LEDC is used to control external LED lamp.

The LEDC has the following features:

- Configurable LED output high/low level width
- Configurable LED reset time
- LEDC data supports DMA configuration mode and CPU configuration mode
- Maximum 1024 LED serial connect
- LED data transfer rate up to 800 kbit/s
- Configurable RGB display mode
- Non-data output default level is configurable

### 6.10.2. Block Diagram

Figure 6-49 shows a block diagram of the LEDC.

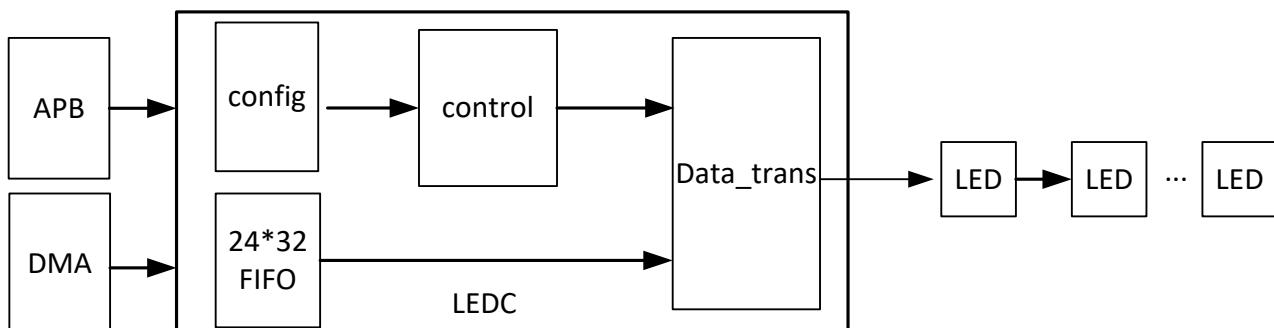


Figure 6- 49. LEDC Block Diagram

config: register configuration

control: LEDC timing control and status control

FIFO: 24\*32 data FIFO

Data\_trans: input data converts to 0 character and 1 character

### 6.10.3. Operations and Functional Descriptions

#### 6.10.3.1. External Signals

Table 6-22 describes the external signals of the LEDC.

**Table 6- 22. LEDC External Signals**

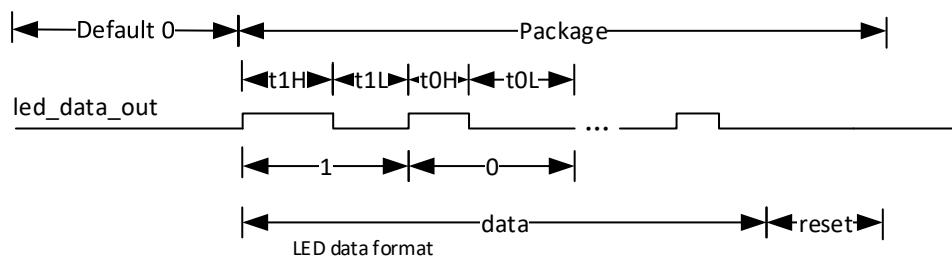
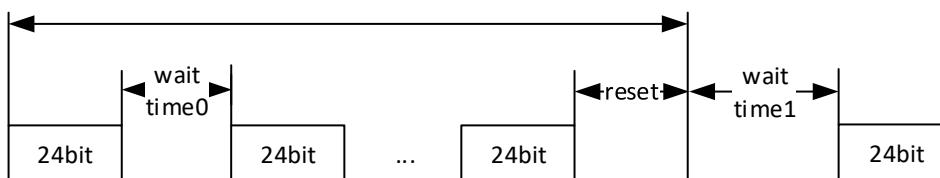
Signal	Description	Type
LEDC-DO	Intelligent Control LED Signal Output	O

#### 6.10.3.2. Clock Sources

**Table 6- 23. LEDC Clock Sources**

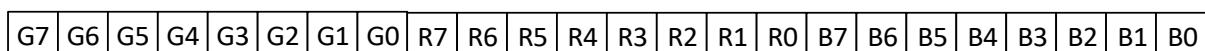
Clock Sources	Description
LEDC_CLK	24MHz(fixed)
APB_CLK	250MHz(changed), support range:24MHz~250MHz, typical value:200MHz

#### 6.10.3.3. LEDC Timing

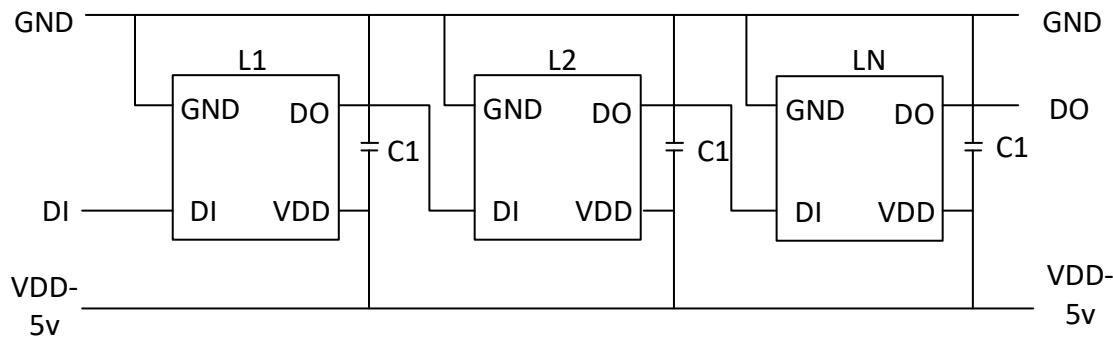

**Figure 6- 50. LEDC Package Output Timing Diagram**

**Figure 6- 51. LEDC 1-frame Output Timing Diagram**

#### 6.10.3.4. 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 6- 52. LEDC Input Data Structure**

#### 6.10.3.5. LEDC Typical Circuit



C1: filter capacitor of LED , usually is 100NF

Figure 6- 53. LEDC Typical Circuit

#### 6.10.3.6. LEDC Data Input Code

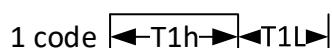


Figure 6- 54. LEDC Data Input Code

#### 6.10.3.7. LEDC Data Transfer Time

The time parameter of typical LED datasheet:

T0H	0 code, high-level time	220ns-380ns
T0L	0 code, low-level time	580ns-1.6us
T1H	1 code, high-level time	580ns-1.6us
T1L	1 code, low-level time	220ns-420ns
RESET	Frame unit, low-level time	> 280us

#### 6.10.3.8. LEDC Data Transfer Mode

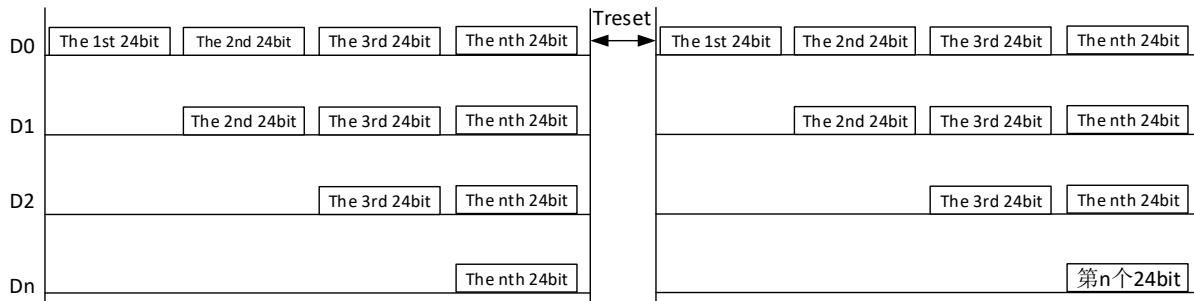


Figure 6- 55. LEDC Data Transfer Mode

#### 6.10.3.9. LEDC Parameter

(1) PAD rate > 800kbit/s

(2) LED number supported:

$T_0\text{-code}$ : 800ns~1980ns,  $T_1\text{-code}$ : 800ns~2020ns

When LED refresh rate is 30 frame/s, LED number supported is  $(1s/30-280\mu s)/((800ns\sim 2020ns)*24) = 1023\sim 681$ .

When LED refresh rate is 60 frame/s, LED number supported is  $(1s/60-280\mu s)/((800ns\sim 2020ns)*24) = 853\sim 337$ .

#### 6.10.3.10. LEDC Data Transfer

LEDC supports DMA data transfer mode or CPU data transfer mode. DMA data transfer mode is set by LEDC\_DMA\_EN.

- **Data transfer in DMA mode**

When the valid space of internal FIFO is greater than the setting FIFO free space threshold, LEDC sends DMA\_REQ to require DMA to transfer data from DRAM to LEDC. The maximum data transfer size in DMA mode is 16 word.(The internal FIFO level is 32.)

- **Data transfer in CPU mode**

When the valid space of internal FIFO is greater than the setting FIFO free space threshold, 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.)

#### 6.10.3.11. LEDC Interrupt

Module Name	Description
FIFO_OVERFLOW_INT	FIFO overflow interrupt. The data written by external is more than the maximum storage space of LED FIFO, LEDC will be in data loss state. At this time, software needs to deal with the abnormal situation. The processing mode is as follows.

	Software can query LED_FIFO_DATA_REG to determine which data has been stored in internal FIFO of LEDC. LEDC performs soft_reset operation to refresh all data.
WAITDATA_TIMEOUT_INT	<p>Wait data timeout interrupt</p> <p>When internal FIFO of LEDC cannot get data because of some abnormal situation, after led_wait_data_time, timeout interrupt is set, LEDC is in WAIT_DATA state, LEDC outputs a level state configured by LED_POLARITY; in the course of wait_data, if new data arrives, 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), LED may enter in refresh state, data has not been sent.</p>
FIFO_CPUREQ_INT	<p>FIFO request CPU data interrupt</p> <p>When FIFO data is less than threshold, the interrupt will be reported to CPU.</p>
LEDC_TRANS_FINISH_INT	<p>Data transfer complete interrupt</p> <p>The value indicates that the data configured as total_data_length has been transferred complete.</p>

LEDC interrupt usage scenario:

(1) CPU mode

Software can enable GLOBAL\_INT\_EN,FIFO\_CPUREQ\_INT\_EN,WAITDATA\_TIMEOUT\_INT\_EN,FIFO\_OVERFLOW\_INT\_EN,LED\_C\_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.

(2) DMA mode

Software can enable GLOBAL\_INT\_EN,WAITDATA\_TIMEOUT\_INT\_EN,FIFO\_OVERFLOW\_INT\_EN,LED\_C\_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.

## 6.10.4. Programming Guidelines

### 6.10.4.1. LEDC Normal Configuration Process

**Step1** Configure LEDC\_CLK and bus pclk.

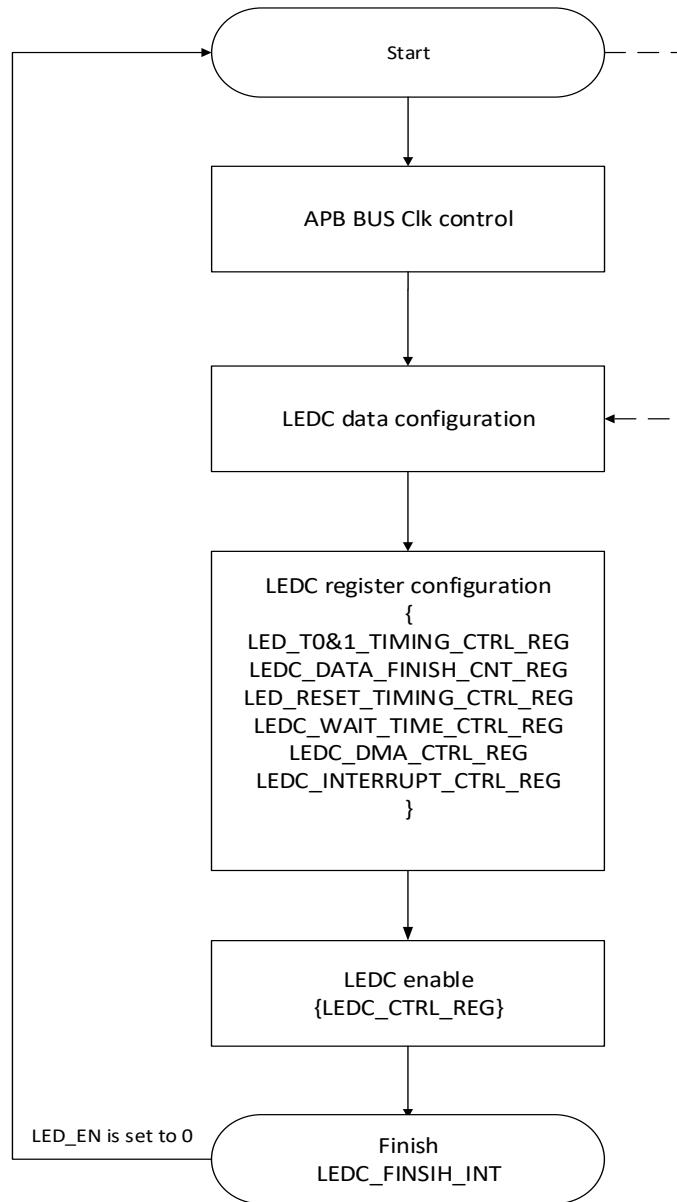
**Step2** Configure the written LEDC data.

**Step3** Configure LED\_T10\_TIMING\_CTRL\_REG, LEDC\_DATA\_FINISH\_CNT\_REG, LED\_RESET\_TIMING\_CTRL\_REG, LEDC\_WAIT\_TIME\_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.

**Step4** Configure LEDC\_CTRL\_REG to enable LEDC\_EN, LEDC will start to output data.

**Step5** When 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.

**Step6** Repeat step1,2,3,4 to re-execute a new round of configuration, enable LEDC\_EN, LEDC will start new data transfer.



**Figure 6- 56. LEDC Normal Configuration Process**

#### 6.10.4.2. LEDC Abnormal Scene Processing Flow

##### (1) WAITDATA\_TIMEOUT Abnormal Status

**Step1** WAITDATA\_TIMEOUT indicates internal FIFO data request of LEDC cannot obtain response, at this time if the default output level is low level, then external LED may think there was a reset operation and cause LED data to be flushed incorrectly.

**Step2** LEDC needs be performed soft\_reset operation, that is LEDC\_SOFT\_RESET=1; after soft\_reset, LEDC\_EN will be pulled-down automatically, all internal status register and control state machine will return to idle state, LEDC FIFO read & write point is cleared to 0, LEDC interrupt is cleared.

**Step3** Setting reset\_led\_en to 1 indicates LEDC can send proactive a reset operation to ensure external LED lamp in right state.

**Step4** Software reads the state of reset\_led\_en, when is 1, it indicates LEDC does not perform the transmission of LED

reset operation; when is 0, LEDC completes the transmission of LED reset operation.

**Step5** When LEDC reset operation finishes, the LEDC data and register configuration need be reoperated to start retransmission data operation.

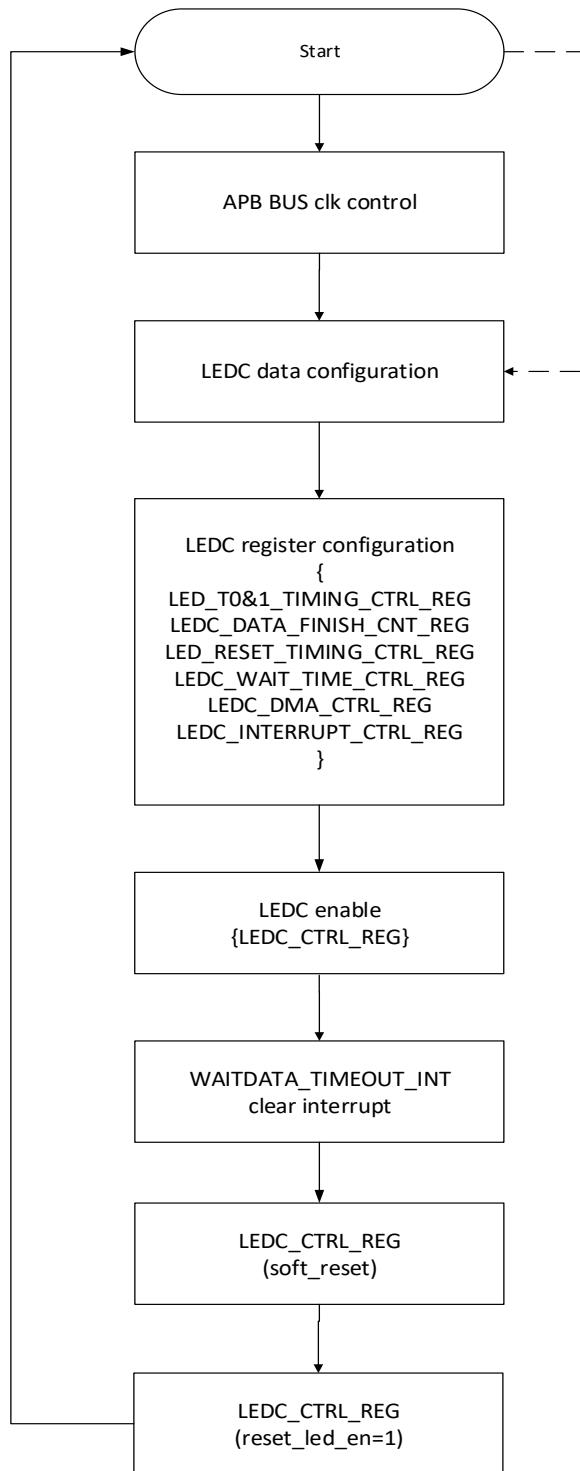


Figure 6- 57. LEDC Timeout Abnormal Processing Flow

## (2) FIFO Overflow Abnormal Status

**Step1** FIFO\_OVERFLOW\_INT indicates software configured data exceeds LEDC FIFO space, at this time redundant data will be lost.

**Step2** Software needs read data in LEDC\_FIFO\_DATA\_REG register to confirm the lost data.

**Step3** Software re-configures the lost data to LEDC.

**Step4** If software uses soft\_reset operation, the operation is the same with timeout abnormal processing flow.

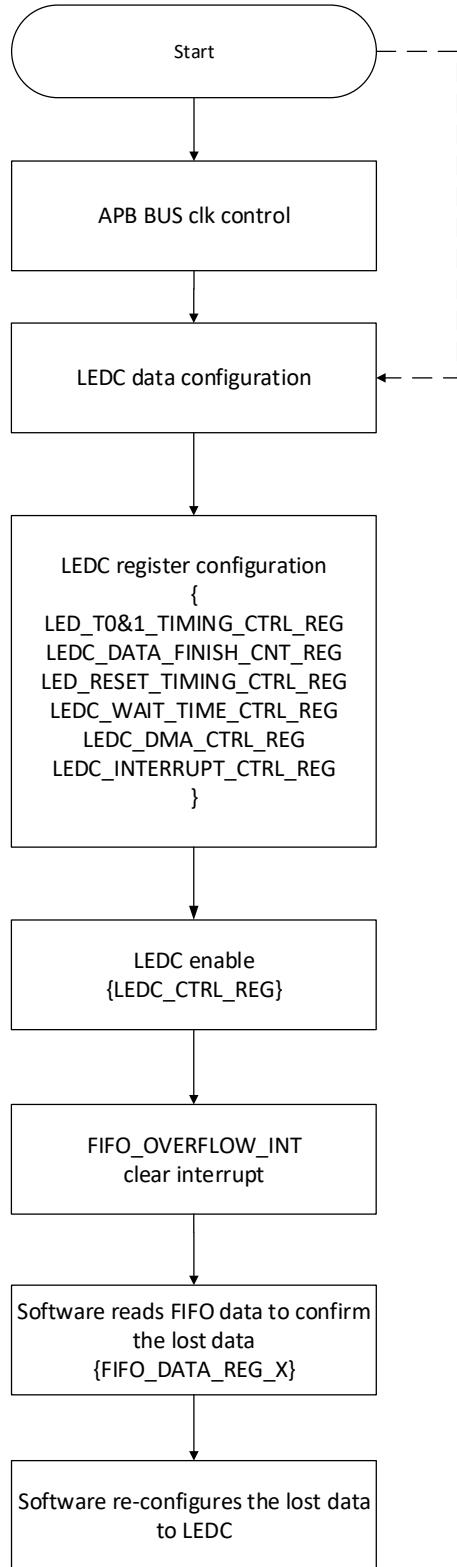


Figure 6- 58. FIFO Overflow Abnormal Processing Flow

### 6.10.5. Register List

Module Name	Base Address
LEDC	0x06700000

Register Name	Offset	Description
LEDC_CTRL_REG	0x0000	LEDC Control Register
LED_T1&0_TIMING_CTRL_REG	0x0004	LED T0 Timing Control Register
LEDC_DATA_FINISH_CNT_REG	0x0008	LEDC Data Finish Counter Register
LED_RESET_TIMING_CTRL_REG	0x000C	LED Reset Timing Control Register
LEDC_WAIT_TIME0_CTRL_REG	0x0010	Wait Time0 Control Register
LEDC_DATA_REG	0x0014	LEDC Data Register
LEDC_DMA_CTRL_REG	0x0018	DMA Control Register
LEDC_INT_CTRL_REG	0x001C	Interrupt Control Register
LEDC_INT_STS_REG	0x0020	Interrupt Status Register
LEDC_WAIT_TIME1_CTRL_REG	0x0028	Wait Time1 Control Register
LEDC_FIFO_DATA_REG	0x0030+4*N	LEDC FIFO Data Register

### 6.10.6. Register Description

#### 6.10.6.1. 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	<p>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).</p>
15:11	/	/	/
10	R/W	0x0	<p>RESET_LED_EN Write operation: Software writes 1, CPU triggers LEDC to transfer a reset to LED.</p> <p> <b>NOTE</b></p> <p><b>Only when LEDC is in IDLE status, reset can be performed. After reset finished, the control state machine returns to the IDLE status. To return</b></p>

			<p><b>LEDC to the IDLE status, it also needs to be used with SOFT_RESET.</b></p> <p><b>When software sets the bit, software can read the bit to check if reset is complete.</b></p> <p>Read operation:</p> <p>0: LEDC completes the transmission of LED reset operation</p> <p>1: LEDC does not complete the transmission of LED reset operation</p>																																																																	
9	/	/	/																																																																	
8:6	R/W	0x0	<p>LED_RGB_MODE</p> <table> <tr><td>000</td><td>GRB(bypass)</td></tr> <tr><td>001</td><td>GBR</td></tr> <tr><td>010</td><td>RGB</td></tr> <tr><td>011</td><td>BGR</td></tr> <tr><td>100</td><td>RBG</td></tr> <tr><td>101</td><td>BRG</td></tr> </table> <p>By default, software configures data to LEDC according to GRB(MSB) mode, LEDC internal combines data to output to the external LED.</p> <p>Other modes configures as follows.</p> <table border="1"> <thead> <tr> <th>Software Input Mode</th> <th>Configuration</th> <th>LEDC Output Mode</th> </tr> </thead> <tbody> <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>BGR</td> </tr> <tr> <td>100</td> <td>RBG</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>RGB</td> </tr> <tr> <td>100</td> <td>BRG</td> </tr> <tr> <td>101</td> <td>RBG</td> </tr> <tr> <td rowspan="6">RGB</td> <td>000</td> <td>RGB</td> </tr> <tr> <td>001</td> <td>RBG</td> </tr> <tr> <td>010</td> <td>GRB</td> </tr> <tr> <td>011</td> <td>BGR</td> </tr> <tr> <td>100</td> <td>GBR</td> </tr> <tr> <td>101</td> <td>BRG</td> </tr> <tr> <td rowspan="5">RBG</td> <td>000</td> <td>RBG</td> </tr> <tr> <td>001</td> <td>RGB</td> </tr> <tr> <td>010</td> <td>BRG</td> </tr> <tr> <td>011</td> <td>GRB</td> </tr> <tr> <td>100</td> <td>BGR</td> </tr> </tbody> </table>	000	GRB(bypass)	001	GBR	010	RGB	011	BGR	100	RBG	101	BRG	Software Input Mode	Configuration	LEDC Output Mode	GRB	000	GRB	001	GBR	010	RGB	011	BGR	100	RBG	101	BRG	GBR	000	GBR	001	GRB	010	BGR	011	RGB	100	BRG	101	RBG	RGB	000	RGB	001	RBG	010	GRB	011	BGR	100	GBR	101	BRG	RBG	000	RBG	001	RGB	010	BRG	011	GRB	100	BGR
000	GRB(bypass)																																																																			
001	GBR																																																																			
010	RGB																																																																			
011	BGR																																																																			
100	RBG																																																																			
101	BRG																																																																			
Software Input Mode	Configuration	LEDC Output Mode																																																																		
GRB	000	GRB																																																																		
	001	GBR																																																																		
	010	RGB																																																																		
	011	BGR																																																																		
	100	RBG																																																																		
	101	BRG																																																																		
GBR	000	GBR																																																																		
	001	GRB																																																																		
	010	BGR																																																																		
	011	RGB																																																																		
	100	BRG																																																																		
	101	RBG																																																																		
RGB	000	RGB																																																																		
	001	RBG																																																																		
	010	GRB																																																																		
	011	BGR																																																																		
	100	GBR																																																																		
	101	BRG																																																																		
RBG	000	RBG																																																																		
	001	RGB																																																																		
	010	BRG																																																																		
	011	GRB																																																																		
	100	BGR																																																																		

				101	GBR		
BGR				000	BGR		
				001	BRG		
				010	GBR		
				011	RBG		
				100	GRB		
				101	RGB		
				000	BRG		
BRG				001	BGR		
				010	RBG		
				011	GBR		
				100	RGB		
				101	GRB		
				000	BRG		
				001	BGR		
5	R/W	0x1		LED_MSB_TOP Adjust sequence of the combined GRB data 0: LSB 1: MSB			
4	R/W	0x1		LED_MSB_G MSB control for Green data 0: LSB 1: MSB			
3	R/W	0x1		LED_MSB_R MSB control for Red data 0: LSB 1: MSB			
2	R/W	0x1		LED_MSB_B MSB control for Blue data 0: LSB 1: MSB			
1	R/W1C	0x0		LEDC_SOFT_RESET LEDC soft reset Write 1 to clear it automatically. The range of LEDC soft reset: all internal status registers, the control state machine returns to in idle status, LEDC FIFO read & write point is cleared to 0, LEDC interrupt is cleared, the affected registers are follow. 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	<p>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.</p>

#### 6.10.6.2. LED\_T01\_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	<p>T1H_TIME LED T1H time Unit: cycle(24MHz), T1H_TIME =42ns*(N+1) The default value is 800ns, the range is 80ns~2560ns. N: 1~3F When is 0, T1H_TIME = 3F</p>
20:16	R/W	0x6	<p>T1L_TIME LED T1L time Unit: cycle(24MHz), T1L_TIME =42ns*(N+1) The default value is 280ns, the range is 80ns~1280ns. N: 1~1F When is 0, T1L_TIME = 1F</p>
15:11	/	/	/
10:6	R/W	0x7	<p>TOH_TIME LED TOh time Unit: cycle(24MHz), TOH_TIME =42ns*(N+1) The default value is 280ns, the range is 80ns~1280ns. N: 1~1F When is 0, TOH_TIME = 1F</p>
5:0	R/W	0x13	<p>TOL_TIME LED T0l time Unit: cycle(24MHz), TOL_TIME =42ns*(N+1) The default value is 800ns, the range is 80ns~2560ns. N: 1~3F When is 0, TOL_TIME = 3F</p>

#### 6.10.6.3. 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	<b>LED_WAIT_DATA_TIME</b> The value is the time that internal FIFO in LEDC is waiting data, when the setting time is exceeded , LEDC will send waitdata_timeout_int interrupt. (This is a abnormal situation, software needs to reset LEDC.) The value is 300us by default. The adjust range is from 80ns to 655us, 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	<b>LED_DATA_FINISH_CNT</b> The value is the total LED data that have been sent .(Range: 0~8k)

#### 6.10.6.4. LED 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	<b>TR_TIME</b> Reset time control of LED lamp Unit: cycle(24MHz), tr_time=42ns*(N+1) The default value is 300us. The adjust range is from 80ns to 327us. N: 1~1FFF
15:10	/	/	/
9:0	R/W	0x0	<b>LED_NUM</b> 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.

#### 6.10.6.5. 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	<b>WAIT_TIM0_EN</b> WAIT_TIM0 enable When it is 1, the controller automatically insert waiting time before LED package data. 0: Disable 1: Enable

7:0	R/W	0xFF	<b>TOTAL_WAIT_TIME0</b> Waiting time between 2 LED datas, LEDC output is low level. The adjust range is from 80ns to 10us. wait_time0=42ns*(N+1) Unit: cycle(24MHz) N: 1~FF
-----	-----	------	---

#### 6.10.6.6. 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)

#### 6.10.6.7. LEDC DMA Control Register (Default: 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
4:0	R/W	0x0F	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, DMA or CPU request will generate. The default value is 15. The adjust value is from 1 to 31. The recommended configuration is 7 or 15. When the configuration value is 0, LEDFIFO_TRIG_LEVEL=F.

#### 6.10.6.8. 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 software is more than internal FIFO level of LEDC, LEDC is in 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

#### 6.10.6.9. LEDC Interrupt Status Register (Default Value: 0x0000\_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, LEDC will be in data loss state. At this time, software needs to deal with the abnormal situation. The processing mode is as follows. Software can query LED_FIFO_DATA_REG to determine which data has been stored in internal FIFO of LEDC. LEDC performs soft_reset operation to refresh all data. 0: FIFO not overflow 1: FIFO overflow

3	R/W1C	0x0	<b>WAITDATA_TIMEOUT_INT</b> When internal FIFO of LEDC cannot get data because of some abnormal situation, after led_wait_data_time, timeout interrupt is set, LEDC is in WAIT_DATA state, LEDC outputs a level state configured by LED_POLARITY; in the course of wait_data, if new data arrives, 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), LED may enter in refresh state, data has not been sent. 0: LEDC not timeout 1: LEDC timeout
2	/	/	/
1	R/W1C	0x0	<b>FIFO_CPUREQ_INT</b> FIFO request CPU data interrupt When FIFO data is less than threshold, the interrupt will be reported to CPU. 0: FIFO does not request that CPU transfers data 1: FIFO requests that CPU transfers data
0	R/W1C	0x0	<b>LED_TRANS_FINISH_INT</b> Data transfer complete interrupt The value indicates that the data configured as total_data_length has been transferred complete. 0: Data not transfer complete 1: Data transfer complete

#### 6.10.6.10. LEDC Wait Time 1 Control Register (Default Value: 0x1FF\_FFFF)

Offset: 0x0028			Register Name: LEDC_WAIT_TIME1_CTRL_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	<b>WAIT_TIM1_EN</b> 0: Disable 1: Enable <b>WAIT_TIME1 enable</b> When the bit is 1, the controller automatically insert waiting time before LED frame data.
30:0	R/W	0x1FF_FFFF	<b>TOTAL_WAIT_TIME1</b> Waiting time between 2 frame datas, LEDC output is low level. The adjust range is from 80ns to 85s. wait_time1=42ns*(N+1) Unit: cycle(24MHz) N: 80~7FFF_FFFF

#### 6.10.6.11. LEDC FIFO Data Register X (Default Value: 0x0000\_0000)

Offset: 0x0030+N*0x4(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.

## Contents

---

Chapter 7 Security System.....	535
7.1. Crypto Engine.....	535
7.1.1. Overview .....	535
7.1.2. Block Diagram .....	535
7.1.3. Operations and Functional Descriptions .....	536
7.1.4. Register List .....	541
7.1.5. Register Description .....	541
7.2. Security ID .....	544

## Figures

---

Figure 7- 1. CE Block Diagram .....	536
Figure 7- 2. Task Chaining .....	537
Figure 7- 3. Task Request Process.....	540

# Chapter 7 Security System

---

## 7.1. Crypto Engine

### 7.1.1. Overview

The Crypto Engine(CE) module is one encryption/decryption algorithm accelerator. It supports kinds of symmetric, asymmetric, Hash, and RNG algorithms. There are two software interfaces for secure and non-secure world each. The software interface is simple for configuration, only setting interrupt control, task description address and load tag. Algorithm control information is written in memory by task descriptor, then CE automatically reads it when executing request. It supports parallel requests from 4 channels, and has an internal DMA controller to transfer data between CE and memory.

The CE has the following features:

- Symmetrical algorithm: AES, DES, 3DES
- Hash algorithm: MD5, SHA1, SHA224, SHA256, SHA384, SHA512, HMAC-SHA1, HMAC-SHA256
- Asymmetrical algorithm: RSA512/1024/2048bit
- 160-bit hardware PRNG with 175-bit seed
- 256-bit hardware TRNG
- ECB, CBC, CTR, CTS, OFB, CFB modes for AES
- ECB, CBC, CTR modes for DES/3DES
- 16-, 32-, 64-, 128-bit wide size for AES CTR
- 1-, 8-, 64-, 128-bit width for AES-CFB
- 16-, 32-, 64-bit wide size for DES/3DES CTR
- 128-, 192-, 256-bit key size for AES
- Multi-package mode for MD5/SHA1/SHA224/SHA256/SHA384/SHA512
- Internal DMA controller for data transfer with memory
- Supports secure and non-secure interfaces respectively

### 7.1.2. Block Diagram

The following figure shows the block diagram of Crypto Engine.

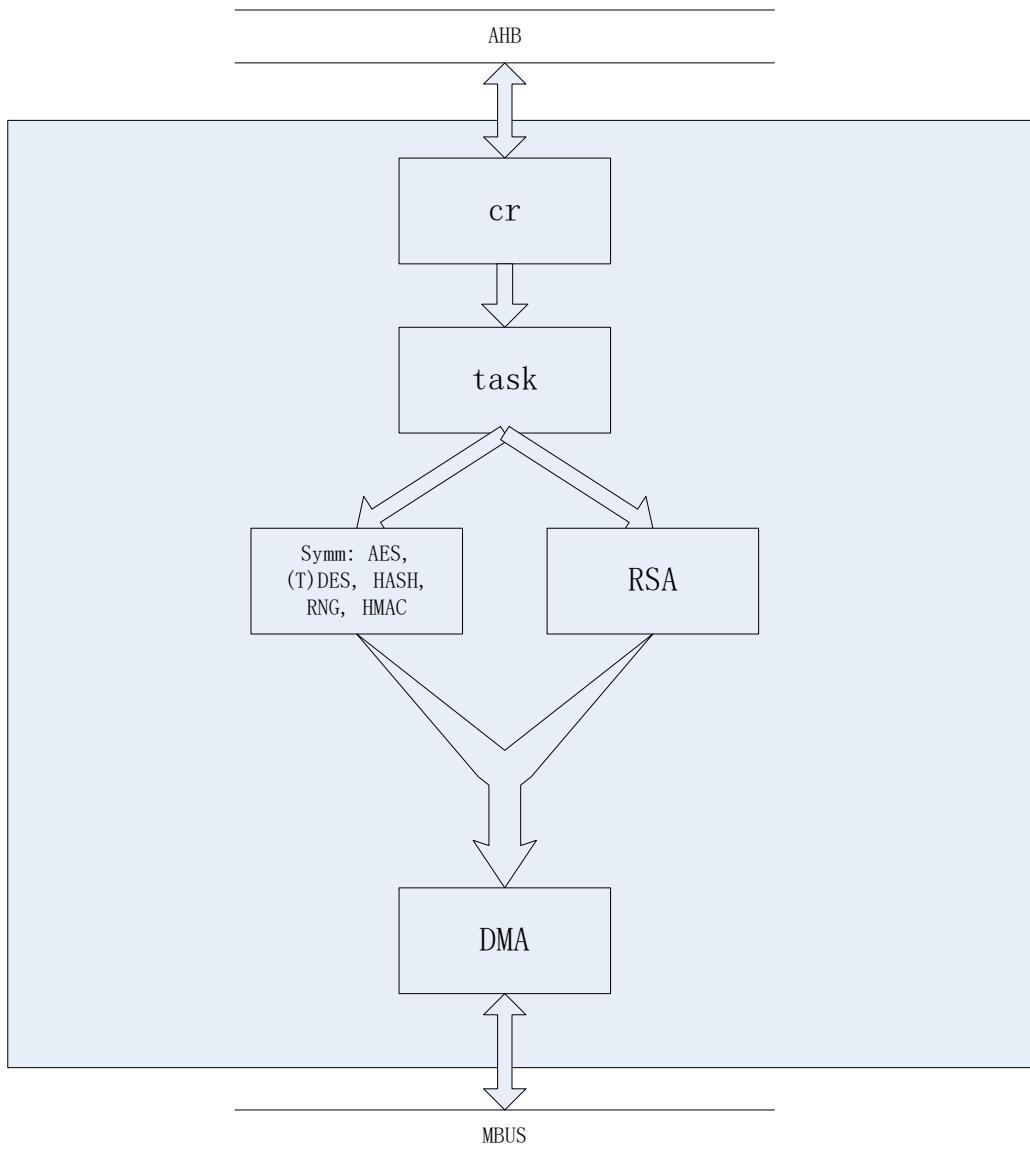
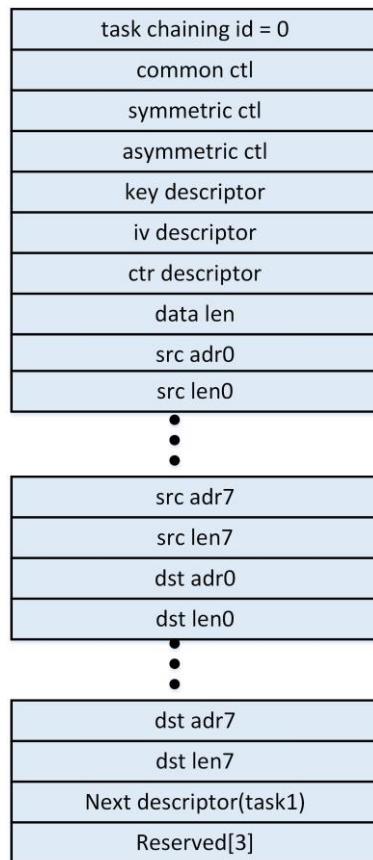


Figure 7- 1. CE Block Diagram

### 7.1.3. Operations and Functional Descriptions

#### 7.1.3.1. Task Descriptor

Software make request through task descriptor, providing algorithm type, mode, key address, source/destination address and size, etc. The task descriptor is as follows.

**Figure 7- 2. Task Chaining**

Task chaining id supports 0~3.

#### 7.1.3.2. Task Descriptor Queue Common Control

Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	Interrupt enable for current task 0: disable interrupt 1: enable interrupt
30:17	/	/	/
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

7	/	/	/
6:0	R/W	0x0	Algorithm Type 0x0: AES 0x1: DES 0x2: Triple DES (3DES) 0x10: MD5 0x11: SHA-1 0x12: SHA-224 0x13: SHA-256 0x14: SHA-384 0x15: SHA-512 0x16: HMAC-SHA1 0x17: HMAC-SHA256 0x20: RSA 0x30: TRNG 0x31: PRNG Others: reserved

#### 7.1.3.3. Task Descriptor Queue Symmetric Control

Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:20	R/W	0x0	<b>KEY_SELECT</b> 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	<b>CFB_WIDTH</b> For AES-CFB width 00: CFB1 01: CFB8 10: CFB64 11: CFB128
17	R/W	0x0	<b>PRNG_LD</b> Load new 15bits key into Ifsr for PRNG
16	R/W	0x0	<b>AES CTS last package flag</b> When setting to '1', it means this is the last package for AES-CTS mode(the size of the last package >128bit).
15:12	/	/	/
11:8	R/W	0x0	<b>ALGORITHM_MODE</b> CE algorithm mode

			0000: Electronic Code Book (ECB) mode 0001: Cipher Block Chaining (CBC) mode 0010: Counter (CTR) mode 0011: CipherText Stealing (CTS) mode 0100: Output feedback (OFB)mode 0101: Cipher feedback (CFB)mode 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

#### 7.1.3.4. Task Descriptor Queue Asymmetric Control

Bit	Read/Write	Default/Hex	Description
31	/	/	/
30:28	R/W	0x0	RSA Public Modulus Width 000:512-bit 001:1024-bit 010:2048-bit Other: reserved
27:0	/	/	/

#### 7.1.3.5. Task Request

Basically, there are 4 steps for one task handling from software.

**Step1:** Software should configure task descriptor in memory, including all fields in descriptor. Channel id corresponds to one channel in CE. According to algorithm type, software should set the fields in common control, symmetric control, asymmetric control, then provide key/iv/ctr address and the data length of this task. Source and destination sg address and size are set based on upper application. If there is another task concatenating after this task, then set its descriptor address at next descriptor field.

**Step 2:** Software should set registers, including task descriptor address, interrupt control.

**Step 3:** Software reads load register to ensure that the bit0 is zero, then starts request by pulling up the bit0 of the load register.

**Step 4:** Wait interrupt status.

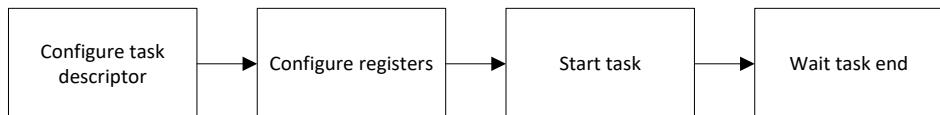


Figure 7- 3. Task Request Process

#### 7.1.3.6. Data Length Setting

Data length field in task descriptor has different meaning for different algorithms.

For AES-CTS, data length field indicates valid source data byte number, for others indicate source data words number.

For PRNG, data length should be 5 words aligned.

For TRNG should be 8 words aligned.

Data size in source and destination sg is as words, whose value should corresponds with data length field, or else CE will report error and stop execution.

#### 7.1.3.7. Security Operation

When CPU issues request to CE module, CE module will save the secure mode of CPU. When executing this request, this state bit works as access tag for inner and system resource. For HUK/RSSK/SSK from SID, only secure mode can access, or else these keys will be used as 0. For access to SID and keysram module through AHB bus, only secure mode can success, or else will read 0 or can not write. When issuing MBUS read and write requests, CE will use send this secure mode bit to BUS, so secure request can access secure and non secure space, but non secure request only can access non secure space.

#### 7.1.3.8. Error Check

CE module includes error detection for task configuration, data computing error, and authentication invalid. When algorithm type in task description is read into module, CE will check if this type is supported through checking algorithm type field in common control. If type value is out of scope, CE will issue interrupt signal and set error state. Each type has certain input and output data size. After getting task descriptor, input size and output size configuration will be checked to avoid size error. If size configuration is wrong, CE will issue interrupt signal and set error state.

#### 7.1.3.9. Clock Requirement

Clock Name	Description	Requirement
hclk	AHB bus clock	24MHz ~ 200MHz
mclk	MBUS clk	24MHz ~ 400MHz
ce_clk	CE work clock	24MHz ~ 300MHz

#### 7.1.4. Register List

Module Name	Base Address
CE_NS	0x01904000
CE_S	0x01904800

Register Name	Offset	Description
CE_TDA	0x0000	Task Descriptor Address
CE_CTL	0x0004	Control Register
CE_ICR	0x0008	Interrupt Control Register
CE_ISR	0x000C	Interrupt Status Register
CE_TLR	0x0010	Task Load Register
CE_TSR	0x0014	Task Status Register
CE_ESR	0x0018	Error Status Register
CE_CSA	0x0024	DMA Current Source Address
CE_CDA	0x0028	DMA Current Destination Address
CE_TPR	0x002C	Throughput Register

#### 7.1.5. Register Description

##### 7.1.5.1. CE Task Descriptor Address Register(Default Value: 0x0000\_0000)

Offset: 0x0000			Register Name: CE_TDA
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Task Descriptor Address

##### 7.1.5.2. 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 Gating Enable(only for CE_S) 0: RSA clk gating enable 1: RSA clk gating disable
2:0	/	/	/

##### 7.1.5.3. CE Interrupt Control Register(Default Value: 0x0000\_0000)

Offset: 0x0008			Register Name: CE_ICR
Bit	Read/Write	Default/Hex	Description

31:4	/	/	/
3:0	R/W	0x0	Task Channel3~0 Interrupt Enable 0: Disable 1: Enable

#### 7.1.5.4. CE Interrupt Status Register(Default Value: 0x0000\_0000)

Offset: 0x000C			Register Name: CE_ISR
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3:0	R/W1C	0x0	Task Channel3~0 End Pending 0: Not finished 1: Finished It indicates if task has been completed . Write '1' to clear it.

#### 7.1.5.5. 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 setting, CE can load the descriptor of task if task FIFO is not full.

#### 7.1.5.6. 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	Running Channel Number 00: Task channel0 01: Task channel1 10: Task channel2 11: Task channel3

#### 7.1.5.7. 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/W1C	0x0	Task Channel3 Error Type xxx1: Algorithm not support xx1x: Data length error x1xx: Keysram access error for AES. Write '1' to clear. 1xxx: Reserved
11:8	R/W1C	0x0	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/W1C	0x0	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/W1C	0x0	Task Channel0 Error Type xxx1: Algorithm not support xx1x: Data length error x1xx: Keysram access error for AES. Write '1' to clear. 1xxx: Reserved

#### 7.1.5.8. 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	Current source address

#### 7.1.5.9. 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	Current destination address

#### 7.1.5.10. CE Throughput Register(Default Value: 0x0000\_0000)

Offset: 0x002C			Register Name: CE_TPR
Bit	Read/Write	Default/Hex	Description
31:0	R/WC	0x0	It indicates the throughput writing to this register at last time. Writing to this register will clear it to 0.

## 7.2. Security ID

The Security ID(SID) is 1Kbit electrical efuse for saving key, which includes chip ID, thermal sensor, HASH code and security key,etc.

The SID module has the following features:

- The module register is non-secure forever, efuse has secure zone and non-secure zone
- A fuse only can program one time
- Loading the key to CE

For complete SID information, refer to the **Allwinner R328-S3 SID Specification**.