Datasheet for Telink

Multi-Standard Wireless SoC

TLSR9218

DS-TLSR9218-E4

Ver 0.8.3

2020/12/07

Keyword

BLE; BLE Mesh; Zigbee; HomeKit; 6LoWPAN; Thread; 2.4GHz; Features; Package

Brief

This datasheet is dedicated for Telink multi-standard wireless IOT SoC TLSR9218.

In this datasheet, function block diagram, key features, and typical applications of TLSR9218 are introduced.

Published by

Telink Semiconductor

Bldg 3, 1500 Zuchongzhi Rd,

Zhangjiang Hi-Tech Park, Shanghai, China

© Telink Semiconductor

All Right Reserved

Legal Disclaimer

This document is provided as-is. Telink Semiconductor reserves the right to make improvements without further notice to this document or any products herein. This document may contain technical inaccuracies or typographical errors. Telink Semiconductor disclaims any and all liability for any errors, inaccuracies or incompleteness contained herein.

Copyright © 2020 Telink Semiconductor (Shanghai) Ltd, Co.

Information

For further information on the technology, product and business term, please contact Telink Semiconductor Company (www.telink-semi.com).

For sales or technical support, please send email to the address of:

telinkcnsales@telink-semi.com

telinkcnsupport@telink-semi.com

Revision History

Version	Change Description
0.8.0	Preliminary release
0.8.1	Remove AI, DPR related information
0.8.2	Update voltage for power consumption in Chapter 1.2.3
0.8.3	1. Update 5 Audio 7 Clock, 9 Interrupt System,10.1 GPIO, 16 AES 2. Add 4.6 Charger, 13.4 Battery Voltage Sampling 3. Modify LSPI to PSPI 4. Other minor edits and corrections

Table of Contents

1 Overview 14
1.1 Block Diagram 14
1.2 Key Features 15
1.2.1 General Features 15
1.2.2 RF Features 16
1.2.3 Features of Power Management Module 16
1.2.4 Bluetooth Features 16
1.2.5 Zigbee Features 17
1.2.6 OpenThread Features 17
1.2.7 HomeKit Features 17
1.2.8 Wireless Audio Features 17
1.2.9 Concurrent Mode Feature 17
1.3 Typical Applications 18
1.4 Ordering Information 18
1.5 Package 18
1.6 Pin Layout 20
2 Key Electrical Specifications 25
2.1 Absolute Maximum Rating 25
2.2 Recommended Operating Conditions 25
2.3 DC Characteristics 25
2.4 AC Characteristics 26
2.5 Analog Input to Digital Output 33
2.6 Digital Input to Analog Output 34
2.7 MICBIAS Characteristics 35
2.8 Digital Microphone Interface Characteristics 36
2.9 Storage Condition 37
3 Reference Design 38
3.1 Schematic 38
3.2 BOM (Bill of Material) 38
4 Memory, MCU and PMU 40
4.1 Memory 40
4.1.1 SRAM 40
4.1.2 Flash 41

4.1.3 E-Fuse 41
4.1.4 Unique ID 41
4.2 MCU 41
4.3 Working Mode 41
4.4 Reset 44
4.5 Power Management 45
4.5.1 Power-on-Reset (POR) and Brown-out Detect 46
4.5.2 Working Mode Switch 49
4.5.3 LDO and DCDC 50
4.5.4 VBAT and VANT Power-Supply Mode 51
4.6 Charger 51
4.7 Wakeup Source 53
5 Audio 56
5.1 Audio CODEC 56
5.2 Audio-in Path 56
5.3 Audio-out Path 57
5.4 Register Description 57
6 BLE/802.15.4/2.4GHz RF Transceiver 62
6.1 Overview 62
6.2 Air Interface Data Rate and RF Channel Frequency 62
6.3 Baseband 63
6.3.1 Packet Format 63
6.3.2 BLE Location Function 64
6.3.3 RSSI and Frequency Offset 64
7 Clock 65
7.1 Clock Sources 65
7.2 System Clock 67
7.3 Module Clock 68
7.3.1 System Timer Clock 68
7.3.2 USB Clock 68
7.3.3 I2S Clock 68
7.3.4 I2S clock frequency, Fi2s_clock, equals to pllclk*I2S_step[6:0]/I2S_mod[7:0].DMIC Clock 68
7.3.5 clkzb32k 68
7.3.6 clk_eoc_src 68
7.3.7 clk_zb_mst 68

7.4 Register Table 69
8 Timer 72
8.1 Timer0 ~ Timer1 72
8.1.1 Mode 0 (System Clock Mode) 72
8.1.2 Mode 1 (GPIO Trigger Mode) 72
8.1.3 Mode 2 (GPIO Pulse Width Mode) 73
8.1.4 Mode 3 (Tick Mode) 74
8.1.5 Watchdog 74
8.1.6 Register Table 75
8.2 32K LTimer 77
8.3 System Timer 77
9 Interrupt System 78
9.1 Interrupt Structure 78
9.2 External Interrupt Sources 79
9.3 Register Description 81
10 Interface 85
10.1 GPIO 85
10.1.1 Basic Configuration 85
10.1.2 Connection Relationship between GPIO and Related Modules 88
10.1.3 Drive Strength 89
10.1.4 Polarity 89
10.1.5 GPIO IRQ Signal 89
10.1.6 GPIO2RISC IRQ Signal 90
10.1.7 Pull-up/Pull-down Resistors 91
10.2 Swire 92
10.3 I2C 93
10.3.1 Communication Protocol 93
10.3.2 I2C Slave Mode 94
10.3.3 I2C Master Mode 95
10.3.3.1 I2C Master Write Transfer in NDMA Mode 95
10.3.3.2 I2C Master Read Transfer in NDMA Mode 95
10.3.3.3 I2C Master Writer Transfer in NDMA Mode 96
10.3.3.4 I2C Master Read Transfer in NDMA Mode 96
10.3.4 Register Description 96
10.4 I2S 99
10.5 HSPI 99

10.5.1 Diagram 99
10.5.2 Features 99
10.5.3 Function Description 100
10.5.3.1 Master Mode 100
10.5.3.2 Slave Mode 100
10.5.3.3 Dual, Quad and 3line I/O 101
10.5.3.4 XIP 101
10.5.3.5 LCD Display Driving 101
10.6 PSPI 105
10.6.1 Diagram 105
10.6.2 Features 105
10.6.3 Function Descriptions 105
10.7 UART 107
10.8 USB 110
11 PWM 117
11.1 Enable PWM 117
11.2 Set PWM Clock 117
11.3 PWM Waveform, Polarity and Output Inversion 117
11.3.1 Waveform of Signal Frame 117
11.3.2 Invert PWM Output 118
11.3.3 Polarity for Signal Frame 118
11.4 PWM Mode 118
11.4.1 Select PWM Modes 118
11.4.2 Continuous Mode 118
11.4.3 Counting Mode 119
11.4.4 IR Mode 119
11.4.5 IR FIFO Mode 120
11.4.6 IR DMA FIFO Mode 121
11.5 PWM Interrupt 122
11.6 Register Description 122
12 Manchester Decoder(MDEC) 127
12.1 Frame Format 127
12.2 Function Description 127
12.2.1 Block Diagram 127
12.2.2 Reset MDEC 128
12.2.3 Select Input Channel 128

12.2.4 Read Result Data 128
12.3 Register Description 128
13 SAR ADC 130
13.1 Power On/Down 130
13.2 ADC Clock 130
13.3 ADC Control in Auto Mode 130
13.3.1 Set Max State and Enable Channel 130
13.3.2 "Set" State 131
13.3.3 "Capture" State 131
13.3.4 Usage Case with Detailed Register Setting 132
13.4 Battery Voltage Sampling 133
13.5 Register Table 133
14 Temperature Sensor 138
15 Low Power Comparator 139
15.1 Power On/Down 139
15.2 Select Input Channel 139
15.3 Select Mode and Input Channel for Reference 139
15.4 Select Scaling Coefficient 140
15.5 Low Power Comparator Output 140
15.6 Register Description 140
16 AES 142
17 Public Key Engine (PKE) 143
17.1 Calculation Model Overview 143
17.2 Function Description 143
17.2.1 Module Description 143
17.2.2 Software Interface (Programming Model) 144
17.3 Register Description 146
18 True Random Number Generator (TRNG) 149
18.1 Model Overview 149
18.2 Interrupt Description 149
18.2.1 CPU Reads RBG_DR without Data 149
18.2.2 Data Valid 150
18.3 Usage Procedure 150
18.3.1 Normal Operation 150
18.3.2 Entropy Source 150

18.4 Register Description 150

List of Figures

Figure 1-1 Block Diagram of the System 14
Figure 1-2 Package of TLSR9218A 19
Figure 1-3 Pin Assignment for TLSR9218A 20
Figure 3-1 Schematic 38
Figure 4-1 Memory Map 40
Figure 4-2 Control Logic of Power up/down 46
Figure 4-3 Initial Power-up Sequence 47
Figure 4-4 Initial Power-down Sequence 48
Figure 4-5 LDO and DCDC 50
Figure 4-6 Charger Diagram 51
Figure 4-7 Charging Timing Diagram 52
Figure 4-8 Wake up Sources 53
Figure 5-1 Audio CODEC 56
Figure 5-2 Audio in Path 57
Figure 5-3 Audio out Path 57
Figure 6-1 Block Diagram of RF Transceiver 62
Figure 7-1 Clock Source 66
Figure 9-1 Block Diagram of PLIC 78
Figure 10-1 Logic Relationship between GPIO and Related Modules 89
Figure 10-2 I2C Timing 94
Figure 10-3 Byte Consisted of Slave Address and R/W Flag Bit 94
Figure 10-4 Read Format in Slave Mode 94
Figure 10-5 Write Format in Slave Mode 95
Figure 10-6 HSPI Diagram 99
Figure 10-7 Master Transfer Mode Format 100
Figure 10-8 Slave Transfer Mode Format 100
Figure 10-9 3-Line Write Sequence 102

Figure 10-10 3-Line Read Sequence 102
Figure 10-11 4-Line Write Sequence 103
Figure 10-12 4-Line Read Sequence 103
Figure 10-13 2-Line Write Sequence 103
Figure 10-14 RGB565 Pixel Transition 104
Figure 10-15 RGB666 Pixel Transition 104
Figure 10-16 RGB888 Pixel Transition 104
Figure 10-17 PSPI Diagram 105
Figure 10-18 PSPI Master Transfer Format 106
Figure 10-19 Slave Transfer Mode Format 106
Figure 10-20 UART Communication 107
Figure 11-1 Signal Frame 117
Figure 11-2 PWM Output Waveform Chart 118
Figure 11-3 Continuous Mode 119
Figure 11-4 Counting Mode (n=0) 119
Figure 11-5 IR Mode (n=0) 120
Figure 11-6 IR Format Examples 121
Figure 12-1 Frame Format 127
Figure 12-2 MDEC Function Block Diagram 127
Figure 13-1 Diagram of ADC 130
Figure 14-1 Block Diagram of Temperature Sensor 138
Figure 15-1 Block Diagram of Low Power Comparator 139
Figure 16-1 AES Address 142
Figure 17-1 Block Diagram of PKE SP Module 144
Figure 18-1 Module Boundary 149

List of Tables

Table 1-1 Ordering Information of TLSR9218 18
Table 1-2 Mechanical Dimension of TLSR9218A 19
Table 1-3 Pin Function of TLSR9218 21
Table 1-4 GPIO Pin Mux 22
Table 2-1 Absolute Maximum Rating 25
Table 2-2 Recommended Operating Conditions 25
Table 2-3 DC Characteristics(VBAT=3.3 V, T = 25 °C) 25
Table 2-4 Digital Inputs/Outputs AC Characteristics (VDD = 3.3 V, T = 25 °C) 26
Table 2-5 RF Performance AC Characteristics(VBAT = 3.3 V, T = 25 °C) 27
Table 2-6 USB Characteristics 31
Table 2-7 RSSI Characteristics 32
Table 2-8 24MHz Crystal Characteristics 32
Table 2-9 32.768 kHz Crystal Characteristics 32
Table 2-10 24 MHz RC Oscillator Characteristics 32
Table 2-11 32 kHz RC Oscillator Characteristics 32
Table 2-12 ADC Characteristics 33
Table 2-13 Analog Microphone / Line Input to ADC Path 33
Table 2-14 Audio DAC to Headphone Output Path 34
Table 2-15 MICBIAS Characteristics 36
Table 2-16 Digital Microphone Interface Characteristics 36
Table 3-1 BOM Table 38
Table 4-1 E-Fuse Information 41
Table 4-2 Working Mode 42
Table 4-3 Retention Analog Registers in Deep Sleep 43
Table 4-4 Register Configuration for Software Reset 44
Table 4-5 Analog Register to Control Delay Counters 46
Table 4-6 Characteristics of Initial Power-up/Power-down Sequence 48

Table 4-7 3.3 V Analog Register for Module Power up/down Control 49

Table 4-8 Analog Register for Wakeup 54

Table 5-1 Audio Registers 58

Table 6-1 Packet Format in Standard 1 Mbps BLE Mode 63

Table 6-2 Packet Format in Standard 2 Mbps BLE Mode 63

Table 6-3 Packet Format in Standard 500 kbps/125 kbps BLE Mode 63

Table 6-4 Packet Format in 802.15.4 Mode 63

Table 6-5 Packet Format in Proprietary Mode 64

Table 7-1 Clock Sources of Each Module 67

Table 7-2 Clock Related Registers 69

Table 8-1 Register Configuration for Timer 0 ~ Timer 1 75

Table 8-2 Register Table for System Timer 77

Table 9-1 Interrupt Sources 79

Table 9-2 Register Configuration for PLIC 81

Table 10-1 GPIO Setting 85

Table 10-2 GPIO Function Mux Configuration Registers 87

Table 10-3 GPIO IRQ Table 90

Table 10-4 Analog Registers for Pull-up/Pull-down Resistor Control 91

Table 10-5 SWIRE Related Registers 93

Table 10-6 I2C Related Registers 96

Table 10-7 Slave Commands 100

Table 10-8 Slave Commands 106

Table 10-9 UART0 Related Registers 107

Table 10-10 USB Related Registers 110

Table 11-1 PWM Registers 122

Table 12-1 MDEC Related Registers 128

Table 13-1 Overall Register Setting 132

Table 13-2 SAR ADC Registers 133

Table 14-1 Analog Register for Temperature Sensor 138

Table 15-1 Analog Register Related to Low Power Comparator 140

Table 16-1 AES Registers 142

Table 17-1 Dual Port Ram Address Map 145

Table 17-2 PKE Related Registers 146

Table 18-1 Register Description 151

1 Overview

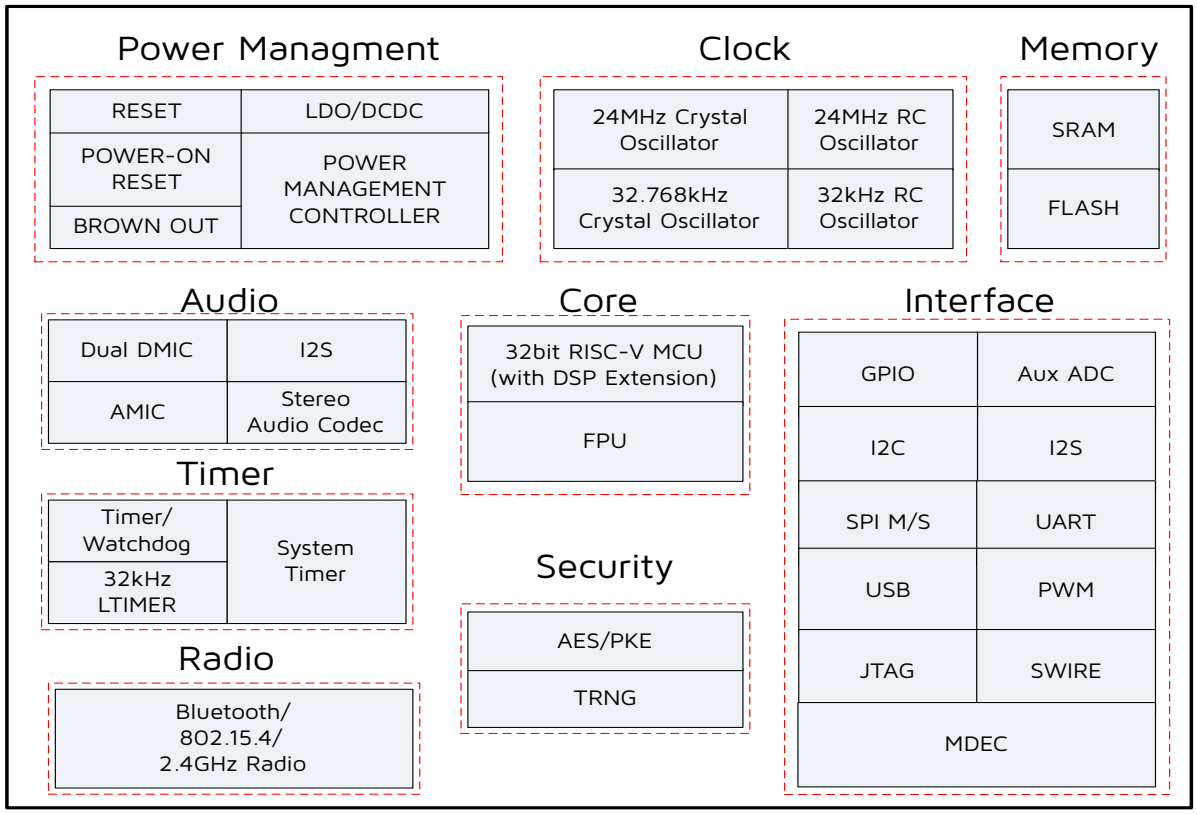
The TLSR9218 supports standards and industrial alliance specifications including Bluetooth 5.2,LE, indoor positioning and BLE Mesh, Zigbee 3.0, HomeKit, 6LoWPAN, Thread and 2.4 GHz proprietary standard.

The TLSR9218 combines the features and functions needed for high quality wireless IoT equipments into a single SoC.

1.1 Block Diagram

The TLSR9218 is designed to offer high integration, ultra-low power application capabilities. The system's block diagram is as shown in Figure 1-1.

Figure 1-1 Block Diagram of the System



The TLSR9218 integrates a powerful 32-bit RISC-V (RISC-Five) MCU, DSP, 2.4 GHz ISM Radio, 256 KB SRAM, 1 MB Flash, stereo audio codec,14 bit AUX ADC, analog and digital Microphone input, PWM, flexible IO interfaces, and other peripheral blocks required for advanced IoT applications.

The TLSR9218 also includes multi-stage power management design allowing ultra-low power operation and making it the ideal candidate for wearable and power-constraint applications.

With the high integration level of TLSR9218, few external components are needed to satisfy customers' ultra-low cost requirements.

1.2 Key Features

1.2.1 General Features

General features are as follows:

1. Support unique ID (UID)
2. 32-bit RISC-V micro-controller
   - Better power-balanced performance than ARM M4
   - Instruction and Data cache controller
   - Maximum running speed up to 96 MHz
   - Integrated DSP extensions instructions
   - Integrated "F" standard extensions for single-precision floating-point
3. Memory architecture
   - Program memory: up to 1 MB Flash
   - Up to 256 KB SRAM including up to 64 KB retention SRAM
4. RTC and other timers
   - Clock source: 24 MHz/32.768 kHz Crystal and 32 kHz/24 MHz embedded RC oscillator
   - Three general 32-bit timers with four selectable modes in active mode
   - Watchdog timer
   - A low-frequency 32 kHz timer available in low power mode
5. A rich set of digital and analog interfaces
   - Up to 26 GPIOs
   - Configurable to select 2-wire SDP or 5-wire JTAG debug interface
   - Dual DMIC (Digital Mic)
   - Support AMIC (Analog Mic), 24 bit 192 kHz I2S
   - Stereo audio codec
   - SPI, I2C, USB 2.0, Swire, UART with hardware flow control Up to 6 channels of differential PWM
   - IR transmitter with DMA
   - 9-channel (only GPIO input), 14-bit Auxiliary ADC
   - Low power comparator
6. Embedded hardware AES and AES-CCM
7. Embedded hardware acceleration for Elliptical curve cryptography (ECC)
8. Embedded Random Number Generator (TRNG)
9. Hardware OTA upgrade and multiple boot switch, allowing convenient product feature roll outs and upgrades
10. Operating temperature range: -40°C~+85°C
11. Completely RoHS-compliant package
    - TLSR9218A, 48-pin QFN 6x6mm
12. Supports all 2.4 GHz IoT standards into a single SoC, including Bluetooth 5.2, BLE Mesh, Zigbee 3.0, Homekit, 6LowPAN, Thread, ANT and 2.4 GHz proprietary technologies

1.2.2 RF Features

RF features include:

1. Bluetooth/802.15.4/2.4 GHz RF transceiver in worldwide 2.4 GHz ISM band
2. Bluetooth Compliant, BLE 1 Mbps and 2 Mbps, Long Range 125 kbps and 500 kbps, IEEE802.15.4 compliant, 250 kbps
3. 2.4 GHz proprietary 1 Mbps/2 Mbps/250 kbps/500 kbps mode with Adaptive Frequency Hopping feature
4. ANT mode
5. Rx Sensitivity: -96 dBm@BLE 1 Mbps, -92.5 dBm @ BLE 2 Mbps mode, -99.5 dBm @ Long Range 125 kbps, -98.5 dBm @ Long Range 500 kbps, , -99.5 dBm@ IEEE802.15.4 250 kbps
6. Tx output power: up to +10 dBm @ BLE mode,
7. 50 Ω matched single-pin antenna input
8. RSSI monitoring with +/-1 dB resolution
9. Auto acknowledgment, retransmission and flow control
10. Support full-function BLE AoA/AoD location features
11. Support PTA (Packet Traffic Arbitrator) for Wi-Fi co-existence

1.2.3 Features of Power Management Module

Features of power management module include:

1. Power supply
   - VBAT (battery): 1.8 V~4.3 V
   - VBUS (USB): 4.5 V~5.5 V
2. Battery monitor for low battery voltage detection
3. Brownout detection/shutoff and Power-On-Reset
4. Multiple-power-state to optimize power consumption
5. Supports USB BC1.2 Battery Charging
6. Power consumption:
   - 11.5 mA with LDOWhole Chip, BLE RX mode: 6.1 mA @ 3.3 V DCDC, 11.5 mA with LDO
   - Whole Chip, BLE TX mode, 6.6 mA @ 0 dBm, 3.3 V DCDC, 12.5 mA @ 0 dBm with LDO
   - Deep sleep with external wakeup (without SRAM retention): 0.7 μA
   - Deep sleep with 32K SRAM retention: 1.7 μA

1.2.4 Bluetooth Features

Bluetooth features include:

1. Bluetooth support with BLE
2. Long range support with 125 kbps and 500 kbps data rate
3. Bluetooth SIG Mesh support
4. Bluetooth based location and indoor positioning support with AoA and AoD (a.k.a Bluetooth 5.1)
5. Bluetooth ISO channel support (a.k.a Bluetooth 5.2) with broadcast and unicast mode
6. Telink extended profile with audio support for voice command based searches

1.2.5 Zigbee Features

Zigbee features include:

1. Zigbee 3.0 platform with Zigbee 3.0 device support, compatible with ZHA and ZLL
2. Green Power support for low power devices

1.2.6 OpenThread Features

OpenThread features include:

1. Implements all Thread networking layers (IPv6, 6LoWPAN, IEEE 802.15.4 with MAC security, Mesh Link Establishment, Mesh Routing)
2. Support Full Thread Device (FTD) and Minimal Thread Device(MTD)
3. Support rich applications
   - IPv6 configuration and raw data interface
   - UDP sockets
   - CoAP client and server

1.2.7 HomeKit Features

HomeKit features include:

1. Single-chip solution with hardware acceleration for all HomeKit security operations
2. Support Apple HomeKit ADK
3. Conformant to latest HomeKit specification
4. Tested against Apple HomeKit Accessory Tester and latest Apple iOS HomeKit HOME application
5. Support for all HAP defined services and characteristics
6. Support for custom defined HAP services and characteristics
7. HomeKit custom update over-the-air (OTA) profile for secure software upgrade over the air implemented

1.2.8 Wireless Audio Features

Wireless audio features include:

1. High performance Stereo Audio Codec with SNR over 96dB and sampling rate up to 192 kHz
2. 24-bit ADC/DAC for Codec
3. Telink-patented technology for true wireless stereo and 1+N hearable devices with synchronized playback and balanced power on all devices
4. Support audio codec for SBC, OPUS, LC3
5. Support voice codec such as G.711, A-Law, u-Law, CVSD, mSBC
6. Support HFP 1.7, HSP 1.2, A2DP 1.3, AVRCP 1.6, SPP 1.2, and PBAP 1.0, ANCS, DIS
7. Support Apple iAP2
8. Support noise suppression and echo cancellation
9. Support Packet Loss Concealment (PLC) for voice processing

1.2.9 Concurrent Mode Feature

In concurrent mode, the chip supports multiple standards working concurrently.

Typical combination is Bluetooth LE + 802.15.4 based standard (e.g. Zigbee, Thread, or 6LoWPAN): BLE and 802.15.4 based stacks can run concurrently with one application state based on time division technology, e.g. BLE stack and Thread stack will run alternately during the divided time slots.

1.3 Typical Applications

The TLSR9218 is an ideal SoC for advanced wireless audio solutions. Its typical applications include, but are not limited to the following:

- Wearable devices
  - Augmented reality glasses
  - Smart watches
  - Smart trackers
  - Wristband
- Audio Solutions
  - Wireless headsets
  - Earbuds

1.4 Ordering Information

Table 1-1 Ordering Information of TLSR9218

Product Series	Package Type	Temperature Range	Ordering No.	Packing Method	Minimum Order Quantity
TLSR9218	QFN48, 6x6x0.75mm	-40°C~+85°C	TLSR9218AER	TR	3000

1.5 Package

Package dimension of TLSR9218A is shown below.

Figure 1-2 Package of TLSR9218A

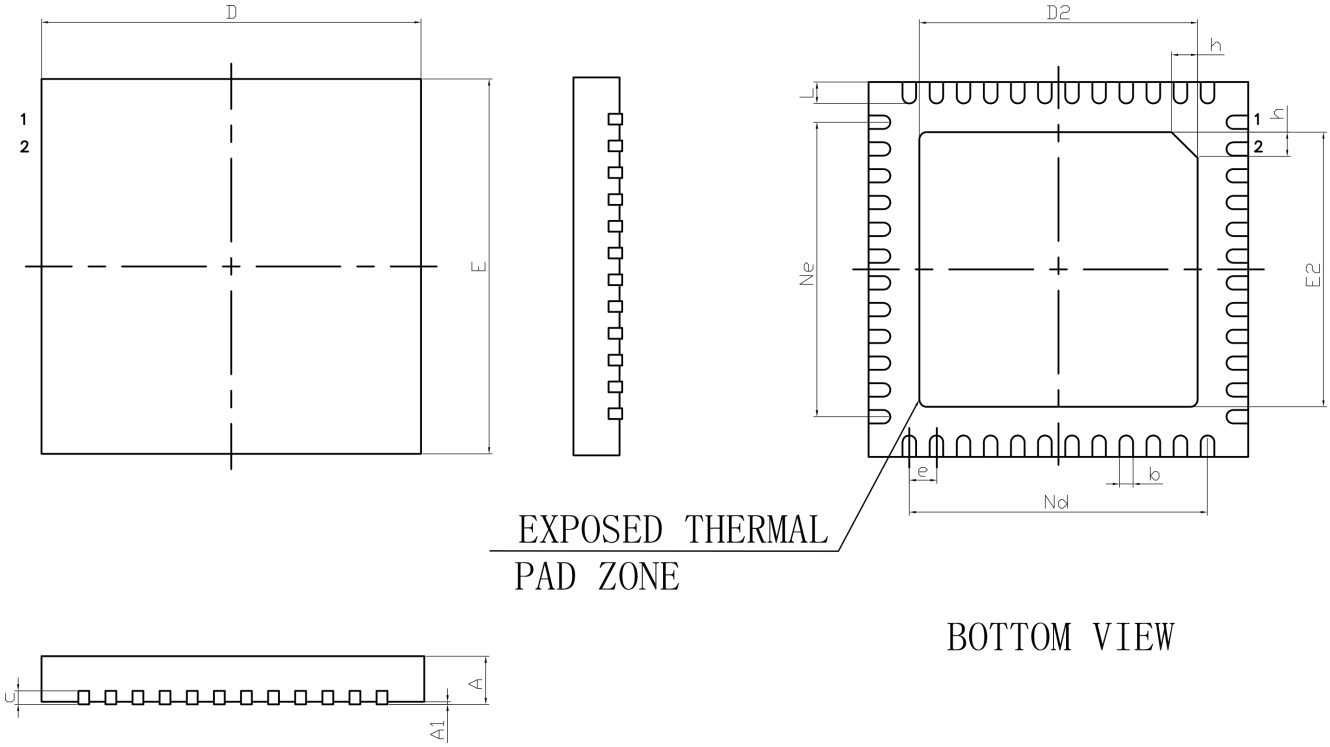


Table 1-2 Mechanical Dimension of TLSR9218A

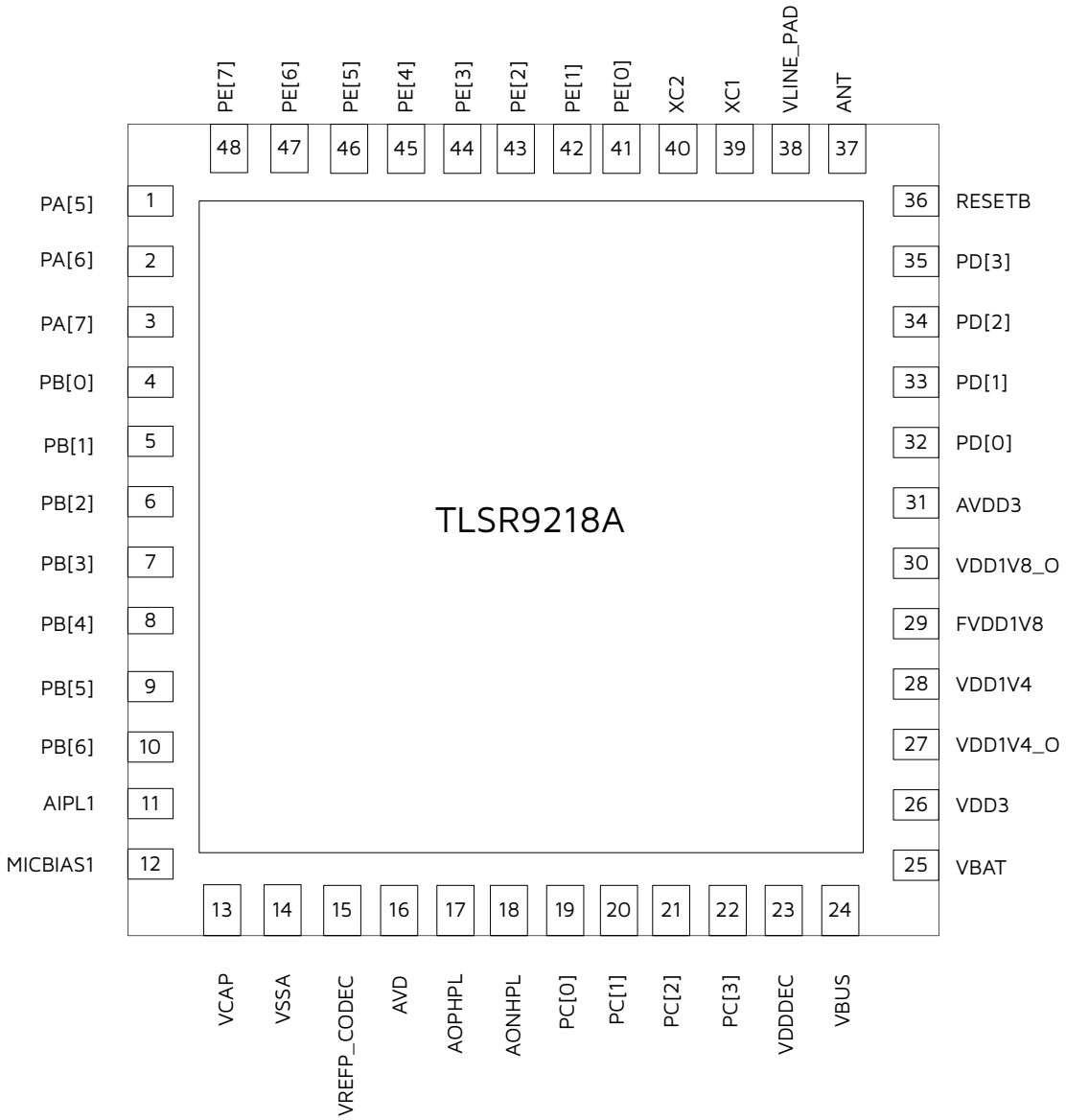
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	-	0.02	0.05
b	0.15	0.20	0.25
c	0.18	0.20	0.23
D	5.90	6.00	6.10
D2	4.10	4.20	4.30
e	0.40BSC		
Nd	4.40BSC		
Ne	4.40BSC		
E	5.90	6.00	6.10
E2	4.10	4.20	4.30
L	0.35	0.40	0.45

SYMBOL	MILLIMETER		
	MIN	NOM	MAX
h	0.30	0.35	0.40
Carrier Size (mil)	177x177		

1.6 Pin Layout

Pin assignment of TLSR9218A is shown below.

Figure 1-3 Pin Assignment for TLSR9218A



Functions of 48 pins for TLSR9218A are described in table below:

Table 1-3 Pin Function of TLSR9218

NO.	Pin Name	Type	Description
1	PA[5]	GPIO	GPIO PA[5]
2	PA[6]	GPIO	GPIO PA[6]
3	PA[7]	GPIO	GPIO PA[7]
4	PB[0]	GPIO	GPIOPB[0]
5	PB[1]	GPIO	GPIO PB[1]
6	PB[2]	GPIO	GPIO PB[2]
7	PB[3]	GPIO	GPIO PB[3]
8	PB[4]	GPIO	GPIO PB[4]
9	PB[5]	GPIO	GPIO PB[5]
10	PB[6]	GPIO	GPIO PB[6]
11	AIPL1	Analog	Left channel single-ended or positive analog input 1
12	MICBIAS1	Analog	Microphone biasing voltage
13	VCAP	Analog	Decoupling cap for internal biasing voltage for internal circuits. This signal is an output of the IC
14	VSSA	GND	Analog negative power supply for internal circuit
15	VREFP_CODEC	Analog	Analog positive supply and reference for internal circuit. This signal is an output of the IC
16	AVD	PWR	Analog positive power supply for embedded linear regulator
17	AOPHPL	Analog	Left channel positive headphone output
18	AONHPL	Analog	Left channel negative headphone output
19	PC[0]	GPIO	GPIO PC[0]
20	PC[1]	GPIO	GPIO PC[1]
21	PC[2]	GPIO	GPIO PC[2]
22	PC[3]	GPIO	GPIO PC[3]
23	VDDDEC	PWR	1.2V output of digital ldo
24	VBUS	PWR	5V VBUS power supply of USB
25	VBAT	PWR	Lion-Battery power supply
26	VDD3	PWR	3.3 V power supply

NO.	Pin Name	Type	Description
27	VDD1V4_O	PWR	1.4V output of DCDC
28	VDD1V4	PWR	1.4V power supply of digital
29	FVDD1V8	PWR	1.8V power supply of flash memory
30	VDD1V8_O	PWR	1.8V output of DCDC
31	AVDD3	PWR	3.3V power supply of DCDC
32	PD[0]	GPIO	GPIO PD[0]
33	PD[1]	GPIO	GPIO PD[1]
34	PD[2]	GPIO	GPIO PD[2]
35	PD[3]	GPIO	GPIO PD[3]
36	RESETB	Reset	Power on reset, active low
37	ANT	Analog	Pin to connect to the Antenna through the matching network
38	VLINE_PAD	PWR	1.4V power supply of RF Transceiver
39	XC1	Analog	Crystal oscillator pin
40	XC2	Analog	Crystal oscillator pin
41	PE[0]	GPIO	GPIO PE[0]
42	PE[1]	GPIO	GPIO PE[1]
43	PE[2]	GPIO	GPIO PE[2]
44	PE[3]	GPIO	GPIO PE[3]
45	PE[4]	GPIO	GPIO PE[4]
46	PE[5]	GPIO	GPIO PE[5]
47	PE[6]	GPIO	GPIO PE[6]
48	PE[7]	GPIO	GPIO PE[7]

GPIO multiple functions are listed in table below:

Table 1-4 GPIO Pin Mux

Pad	Default	Func1	Func2	Func3	Func4	Analog Func
PA[5]	GPIO	-	DAC_I_DAT1_I	-	DM_IO	-
PA[6]	GPIO	-	DAC_Q_DAT1_I	-	DP_IO	-

Pad	Default	Func1	Func2	Func3	Func4	Analog Func
PA[7]	SWS	-	-	-	SWS_IO	-
PB[0]	GPIO	PWM5_O	TX_EN_I	TX_CYC2PA	HSPI_HOLD_N_IO	lp_comp<0>/ sar_in<0>
PB[1]	GPIO	PWM3_O	TX_ON_I	RX_CYC2LNA	CODEC_IRQ_O/ HSPI_WP_N_IO	lp_comp<1>/ sar_in<1>
PB[2]	GPIO	UART0_TX_O	DAC_I_DAT2_I	I2C_SCK_IO	DMIC_DAT_I/ HSPI_DI_IO	lp_comp<2>/ sar_in<2>
PB[3]	GPIO	TX_DAT0_I	UART0_RTX_IO /DAC_I_DAT3_I	I2C_SDA_IO	DMIC_CLK1_O /HSPI_DO_IO	lp_comp<3>/ sar_in<3>
PB[4]	GPIO	TX_DAT2_I	UART0_RTS_O/ DAC_I_DAT4_I	PWM0	DMIC_CLK2_O /HSPI_CK_IO	lp_comp<4>/ sar_in<4>
PB[5]	GPIO	TX_DATA_VALID_I	PWM1_O/ DAC_I_DAT5_I	CLK	CODEC_MCLK_I/ATSEL[0]	lp_comp<5>/ sar_in<5>
PB[6]	GPIO	RATE0_I	UART0_CTS_I/ DAC_I_DAT6_I	MISO_IO1	TX_CYC2PA_O /HSPI_CN_IO	lp_comp<6>/ sar_in<6>
PC[0]	GPIO	PWM0_O	RX_EN_I	CSN	SWM_IO	-
PC[1]	GPIO	DMIC_DAT_I-	TX_CLK_I	ATSEL[0]	I2C_SCK_IO	-
PC[2]	GPIO	ADC_I_DAT0	DMIC_CLK1_O/ DAC_Q_DAT2_I	ATSEL[1]	I2C_SDA_IO	-
PC[3]	GPIO	ADC_I_DAT1	DMIC_CLK2_O/ DAC_Q_DAT3_I	ATSEL[2]	I2C_BCK_IO	-
PD[0]	GPIO	PWM0_N_O	RX_CLK	CSN	UART0_CTS_I	xtl32k_in/ sar_in<8>
PD[1]	GPIO	PWM1_N	ADC_I_DAT3	CLK	UART0_RTS	xtl32k_out/ sar_in<9>
PD[2]	GPIO	PWM2_N	ADC_I_DAT4	MISO_IO1	UART0_TX	-
PD[3]	GPIO	PWM3_N_O	ADC_I_DAT5	MOSI_IO0	UART0_RTX_IO	-
PE[0]	GPIO	PWM3_O	FREQ_CHN_I	UART1_TX	I2C_SCK_IO	-
PE[1]	GPIO	PWM1_O	ADC_Q_DAT3	UART1_CTS_I	I2C_SCK_IO	-

Pad	Default	Func1	Func2	Func3	Func4	Analog Func
PE[2]	GPIO	PWM2_O	ADC_Q_DAT4	UART1_RTX_IO	I2C_SDA_IO	-
PE[3]	GPIO	PWM0_O	ADC_Q_DAT5	UART1_RTS	I2C_SDA_IO	-
PE[4]	GPIO	PWM4_O	ADC_Q_DAT6	RX_CYC2LNA	TDI_I	-
PE[5]	GPIO	PWM5_O	ADC_Q_DAT7	TX_CYC2PA	TDO	-
PE[6]	TMS	-	ADC_Q_DAT8	PWM2_N	TMS_IO	-
PE[7]	TCK	-	ADC_Q_DAT9	PWM3_N	TCK_I	-

2 Key Electrical Specifications

NOTE: The electrical characteristics currently listed in this section are target specifications and only supplied for reference. Some data may be updated according to actual test results.

2.1 Absolute Maximum Rating

Table 2-1 Absolute Maximum Rating

Characteristics	Sym.	Min.	Max.	Unit	Test Condition
Supply Voltage	VBAT	-0.3	4.3	V	-
Voltage on Input Pin	V_{in}	-0.3	VDD+0.3	V	-
Output Voltage	V_{out}	0	VDD	V	-
Storage Temperature Range	T_{Str}	-65	150		-
Soldering Temperature	T_{Sld}	-	260		-

NOTE: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

2.2 Recommended Operating Conditions

Table 2-2 Recommended Operating Conditions

Item	Sym.	Min.	Typ.	Max.	Unit	Condition
Power Supply Voltage	VBAT	1.8	3.7	4.3	V	-
Supply Rise Time from 1.6 V to 1.8 V)	T_R	-	-	10	ms	-
Operating Temperature Range	T_{Opr}	-40	-	85	°C	-

2.3 DC Characteristics

Table 2-3 DC Characteristics(VBAT=3.3 V, T = 25 °C)

Item	Sym.	Min.	Typ.	Max.	Unit	Condition
RX Current	I_{Rx}	-	6.1	-	mA	Whole Chip, BLE mode
TX Current	I_{Rx}	-	6.6	-	mA	Whole Chip @0 dBm, BLE mode

Item	Sym.	Min.	Typ.	Max.	Unit	Condition
RX Current	I_{Rx}	-	11.5	-	mA	Whole Chip, BLE mode, with LDO
TX Current	I_{Rx}	-	12.5	-	mA	Whole Chip @0 dBm, BLE mode, with LDO
Deep sleep with 32kB SRAM retention	I_{Deep1}	-	1.7	-	μA	-
Deep sleep with 64kB SRAM retention		-	TBD	-	μA	-
Deep sleep without SRAM retention	I_{Deep2}	-	0.7	-	μA	-

2.4 AC Characteristics

Table 2-4 Digital Inputs/Outputs AC Characteristics (VDD = 3.3 V, T = 25 °C)

Item	Sym.	Min.	Typ.	Max.	Unit	Condition
Input high voltage	VIH	0.7VDD	-	VDD	V	-
Input low voltage	VIL	VSS	-	0.3VDD	V	-
Output high voltage	VOH	0.9VDD	-	VDD	V	-
Output low voltage	VOL	VSS	-	0.1VDD	V	-

NOTE:

- *The data is test result of engineering sample and may verify for mass production.*
- *VDD stands for IO voltage, e.g, AVDD3, VDD3, or VDD3_DCDC, the range of the IO voltage is 1.8 V ~ 3.6 V, and typical value is 3.3V*

Table 2-5 RF Performance AC Characteristics(VBAT = 3.3 V, T = 25 °C)

Data Rate	Item	Sym.	Min.	Typ.	Max.	Unit	Condition
BLE 1 Mbps RF_RX performance (±250kHz deviation)	Sensitivity	1 Mbps	-	-95.5	-	dBm	-
	Frequency Offset Tolerance	-	-300	-	300	kHz	Wanted signal at -67dBm
	Co-channel rejection	-	-	7	-	dB	-
	In-band blocking rejection (Equal Modulation Interference)	+1/-1 MHz offset	-	-3/-2	-	dB	Wanted signal at -67dBm
		+2/-2 MHz offset	-	-41/-38	-	dB	
		>=3 MHz offset	-	-49	-	dB	
	Image rejection	-	-	-38	-	dB	Wanted signal at -67dBm
BLE 1 Mbps RF_TX performance	Output power, maximum setting	-	-	10	-	dBm	-
	Output power, minimum setting	-	-	-24	-	dBm	-
	Programmable output power range		34			dB	-
	Modulation 20dB bandwidth	-	-	1.4	-	MHz	-

Data Rate	Item	Sym.	Min.	Typ.	Max.	Unit	Condition
BLE 2 Mbps RF_RX performance (±500kHz deviation)	Sensitivity	2 Mbps	-	-92.5	-	dBm	-
	Frequency Offset Tolerance	-	-300	-	300	kHz	Wanted signal at -67dBm
	Co-channel rejection	-	-	8	-	dB	-
	In-band blocking rejection (Equal Modulation Interference)	+2/-2 MHz offset	-	-7/-7	-	dB	Wanted signal at -67dBm
		+4/-4 MHz offset	-	-37/ -38	-	dB	
		>=4 MHz offset	-	-44/ -44	-	dB	
	Image rejection	-	-	-25	-	dB	Wanted signal at -67dBm
BLE 2 Mbps RF_TX performance	Output power, maximum setting	-	-	10	-	dBm	-
	Output power, minimum setting	-	-	-24	-	dBm	-
	Programmable output power range	-	34			dB	-
	Modulation 20dB bandwidth	-	-	2.4	-	MHz	-

Data Rate	Item	Sym.	Min.	Typ.	Max.	Unit	Condition
BLE 500 kbps RF_RX performance (±250kHz deviation)	Sensitivity	500 kbps	-	-98.5	-	dBm	-
	Frequency Offset Tolerance	-	-300	-	300	kHz	-
	Co-channel rejection	-	-	6	-	dB	Wanted signal at -72 dBm
	In-band blocking rejection (Equal Modulation Interference)	+1/-1 MHz offset	-	-3/-3	-	dB	Wanted signal at -72 dBm
		+2/-2 MHz offset	-	-42/ -38	-	dB	
		>=3MHz offset	-	-42	-	dB	
	Image rejection	-	-	-38	-	dB	Wanted signal at -72 dBm
BLE 500 kbps RF_TX performance	Output power, maximum setting	-	-	10	-	dBm	-
	Output power, minimum setting	-	-	-24	-	dBm	-
	Programmable output power range	-	34			dB	-
	Modulation 20dB bandwidth	-	-	1.4	-	MHz	-

Data Rate	Item	Sym.	Min.	Typ.	Max.	Unit	Condition
BLE 125 kbps RF_RX performance (±250kHz deviation)	Sensitivity	125 kbps	-	-99.5	-	dBm	-
	Frequency Offset Tolerance	-	-300	-	300	kHz	-
	Co-channel rejection	-	-	5	-	dB	Wanted signal at -67 dBm
	In-band blocking rejection (Equal Modulation Interference)	+1/-1 MHz offset	-	-3/-3	-	dB	Wanted signal at -67 dBm
		+2/-2 MHz offset	-	-42/ -27	-	dB	
		>=3MHz offset	-	-42/ -36	-	dB	
	Image rejection	-	-	-27	-	dB	Wanted signal at -67 dBm
BLE 125 kbps RF_TX performance	Output power, maximum setting	-	-	10	-	dBm	-
	Output power, minimum setting	-	-	-24	-	dBm	-
	Programmable output power range	-	34			dB	-
	Modulation 20dB bandwidth	-	-	1.4	-	MHz	-

Data Rate	Item	Sym.	Min.	Typ.	Max.	Unit	Condition
Zigbee/RF4CE/ 6LowPan/ Thread RF_RX performance (IEEE802.15.4 250kbps±500 kHz deviation)	Sensitivity	250 kbps	-	-99.5	-	dBm	-
	Frequency Offset Tolerance	-	-300	-	300	kHz	-
	Adjacent channel rejection (-1/+1 channel)	-	-	-42/ -42	-	dB	Wanted signal at -82dBm
	Adjacent channel rejection (-2/ +2 channel)	-	-	-42/ -42	-	dB	Wanted signal at -82dBm
Zigbee/RF4CE/ 6LowPan/ Thread RF_TX performance (IEEE802.15.4 250kbps±500 kHz deviation)	Output power, maximum setting	-	-	10	-	dBm	-
	Output power, minimum setting	-	-	-24	-	dBm	-
	Programmable output power range	-	34			dB	-
	Modulation 20dB bandwidth	-	-	2.7	-	MHz	-
	Error Vector Magnitude	EVM			2%		Max(10 dBm) power output

Table 2-6 USB Characteristics

Item	Sym.	Min.	Typ.	Max.	Unit	Condition
Output Signal Cross-over Voltage	V_{Crs}	1.3	-	2.0	V	-

Table 2-7 RSSI Characteristics

Item	Sym.	Min.	Typ.	Max.	Unit	Condition
RSSI Range	-	-100	-	10	dBm	-
Resolution	-	-	±1	-	dB	-

Table 2-8 24MHz Crystal Characteristics

Item	Sym.	Min.	Typ.	Max.	Unit	Condition
Nominal frequency (parallel resonant)	f_{NOM}	-	24	-	MHz	-
Frequency tolerance	f_{TOL}	-20		+20	ppm	-
Load capacitance	C_L	5	12	18	pF	Programmable on chip load cap
Equivalent series resistance	ESR	-	50	100	ohm	-

Table 2-9 32.768 kHz Crystal Characteristics

Item	Sym.	Min.	Typ.	Max.	Unit	Condition
Nominal frequency (parallel resonant)	f_{NOM}	-	32.768	-	kHz	-
Frequency tolerance	f_{TOL}	-100	-	+100	ppm	-
Equivalent series resistance	ESR	-	50	80	ohm	

Table 2-10 24 MHz RC Oscillator Characteristics

Item	Sym.	Min.	Typ.	Max.	Unit	Condition
Nominal frequency	f_{NOM}	-	24	-	MHz	-
Frequency tolerance	f_{TOL}	-	1	-	%	On chip calibration

Table 2-11 32 kHz RC Oscillator Characteristics

Item	Sym.	Min.	Typ.	Max.	Unit	Condition
Nominal frequency	f_{NOM}	-	32	-	kHz	-
Frequency tolerance	f_{TOL}	-	0.03	-	%	On chip calibration
Calibration time	-	-	3	-	ms	-

Table 2-12 ADC Characteristics

Item	Sym.	Min.	Typ.	Max.	Unit	Condition
Differential nonlinearity	DNL	-	-	1	LSB	10bit resolution mode
Integral nonlinearity	INL	-	-	2	LSB	10bit resolution mode
Signal-to-noise and distortion ratio	SINAD	-	70	-	dB	fin=1 kHz, fS=16 kHz
Effective Number of Bits	ENOB	-	10.5	-	bits	-
Sampling frequency	Fs	-	-	200	ksps	-

2.5 Analog Input to Digital Output

Measurement conditions: Input sine wave with a frequency of 1kHz, measurement bandwidth 20Hz– Fs / 2 for Fs = 8 to 32 kHz, measurement bandwidth 20 Hz – 20 kHz for Fs = 44.1 kHz to 96 kHz, unless otherwise specified.

Table 2-13 Analog Microphone / Line Input to ADC Path

Parameter	Test conditions	Min.	Typ	Max.	Unit
Input level[a]	Full Scale, Gain = 0 dB, boost gain = 20 dB	0.189	0.212	0.239	Vpp
SNR	A-weighted, 1 kHz sine wave @ Full Scale and gain = 0 dB, boost gain = 0 dB	85	90	-	dB
	A-weighted, 1 kHz sine wave @ Full Scale and gain = 0 dB, boost gain = 20 dB	75	80	-	dB
THD	1 kHz sine wave @ Full Scale - 1 dB and gain = 0 dB, boost gain = 0 dB	-	-80	-70	dB
	1 kHz sine wave @ Full Scale - 1 dB and gain = 0 dB, boost gain = 20 dB	-	-70	-60	dB
THD+N[b]	A-weighted, 1 kHz sine wave @ Full Scale -60 dB and gain = 0 dB, boost gain = 0 dB	85	90	-	dB
	A-weighted, 1 kHz sine wave @ Full Scale -60 dB and gain = 0 dB, boost gain = 20 dB	75	80	-	dB
Dynamic Range	A-weighted, 1 kHz sine wave @ Full Scale -60 dB and gain = 0 dB, boost gain = [0-20] dB	-	100	-	dB
PSRR	100 mVpp 1 kHz sine wave is applied to AVD, input data is 0 and gain = 0 dB, boost gain = 20 dB	-	90	-	dB

Parameter	Test conditions	Min.	Typ	Max.	Unit
Boost gain	When activated	0	-	20	dB
Boost gain step	@1 kHz	-	4	-	dB
Boost gain accuracy	@1 kHz	-1		+1	dB
Input resistance	Boost gain = 0 dB, differential configuration	132	160	200	kOhm
	Boost gain = 20 dB, differential configuration	20	26	30	kOhm
Input resistance	Boost gain = 0 dB, single-ended configuration	92	115	138	kOhm
	Boost gain = 20 dB, single-ended configuration	19	24	29	kOhm
Input capacitance	Includes 10 pF for ESD, bonding and package pins cap	-	-	25	pF
Input capacitance	Cbyline	-	1	-	μF

a. The Full Scale input voltage scales with embedded reference: VCAP
b. The specified value is extrapolated by adding 60 dB to the measured SNR

2.6 Digital Input to Analog Output

Measurement conditions: input sine wave with a frequency of 1 kHz, measurement bandwidth 20 Hz–20 kHz, unless otherwise specified.

Table 2-14 Audio DAC to Headphone Output Path

Parameter	Test conditions	Min.	Typ	Max.	Unit
Output level	1 kHz sine wave @ Full Scale -1 dB and gain GOL/R= +6 dB, GODL/R= 0 dB, 10 kOhms load	3.78	3.69	4.78	Vpp
	Full Scale and gain GOL/R= -3 dB, GODL/R= 0 dB, 16 Ohms load	1.35	1.5	1.65	Vpp
	Full Scale and gain GOL/R= +6 dB, GODL/R=0 dB, 16 Ohms load	-	3.5	-	Vpp
Maximum output power	16 Ohms load	-	-	95	mW
SNR	A-weighted, 1 kHz sine wave @ Full Scale and gain GOL/R = +6 dB, GODL/R= 0 dB, 10 kOhms load	90	97	-	dB
SNR	A-weighted, 1 kHz sine wave @ Full Scale and gain GOL/R= +6 dB, GODL/R = 0 dB, 16/32 Ohms load	90	-	97	dB
Dynamic Range	A-weighted, 1 kHz sine wave @ Full Scale and gain GOL/R = [-10 + 6] dB, GODL/R = 0 dB, 16 Ohms load	-	106	-	dB

Parameter	Test conditions	Min.	Typ	Max.	Unit
Idle Noise	A-weighted with no signal and gain GOL/R = -10 dB, GODL/R = 0 dB, 16 Ohms load	-	-104.5	-98	dBV
THD	1 kHz sine wave @ Full Scale -1 dB and gain GOL/R = +6 dB, GODL/R = 0 dB, 10 kOhms load	-	-85	-75	dB
	1 kHz sine wave @ Full Scale -1 dB and gain GOL/R = -3 dB, GODL/R = 0 dB, 16 Ohms load	-	-70	-65	dB
THD+N[a]	A-weighted, 1 kHz sine wave @ Full Scale -60 dB and gain GOL/R = +6 dB, GODL/R = 0 dB, 10 kOhms load	90	95	-	dB
Wide Band	Noise1 kHz sine wave @ Full Scale and gain GOL/R = +6 dB, GODL/R = 0 dB, 10 kOhms load, measurement on 20 kHz - 100 kHz	-	75	-	dB
PSRR	100 mVpp 1 kHz is applied to AVD, input data is 0 and gain GOL/R = 0 dB, GODL/R = 0 dB, 10 kOhms load	-	90	-	dB
Analog gain	Gain GOL/R	-19	-	+12	dB
Digital gain	Gain GODL/R	-31	-	+32	dB
Gain step	GOL/R, GODL/R @ 1 kHz	-	1	-	dB
Gain accuracy	GOL/R, GODL/R @ 1 kHz	-0.5	-	+0.5	dB
Pop-up Noise	Active < - > Inactive, 10 kOhms load	-	-60	-	dBV
	Active < - > Inactive, 16 Ohms load	-	-60	-	dBV
Output load resistance (Rl)	-	16	-	-	Ohm
Output load capacitance (Cp)	-	-	-	200	pF

a. The specified value is extrapolated by adding 60 dB to the measured SNR.

2.7 MICBIAS Characteristics

Measurement conditions: input sine wave with a frequency of 1 kHz, measurement bandwidth 20 Hz–20 kHz, unless otherwise specified.

Table 2-15 MICBIAS Characteristics

Parameter	Test Condition	Min.	Typ.	Max.	Unit
MICBIAS Output Level	MICBIAS_V = 0	-	2.08	-	V
	MICBIAS_V = 1	-	1.66	-	
MICBIAS Output Current	-	-	-	4	mA
MICBIAS Output Noise	A-Weighted	-	20	40	uVrms
MICBIAS Decoupling Capacitor	Cmic	0.75	1	1.25	nF
VREFP_CODEC Output Voltage	-	-	2.5	-	V
VCAP Output Voltage	-	-	2	-	V

2.8 Digital Microphone Interface Characteristics

Measurement conditions: Input sine wave with a frequency of 1kHz, MCLK = 12 MHz or 13 MHz, DMIC_CLK = Fmclk/4, measurement bandwidth 20 Hz - Fs/2 for Fs = 8 to 32 kHz, measurement bandwidth 20 Hz – 20 kHz for Fs = 44.1 kHz to 96 kHz, unless otherwise specified.

Table 2-16 Digital Microphone Interface Characteristics

Parameter	Test Condition	Min.	Typ.	Max.	Unit
Input Level[a]	Full Scale max value, Gain GID = 0 dB	84.5	85.6	86.7	%
	Full Scale min value, Gain GID = 0 dB	15.5	14.4	13.3	%
SNR	A-weighted, 1 kHZ sine wave@ Full Scale and gain GIDL, GIDR = 0 dB		100		dB
THD+N	1 kHZ sine wave@ Full Scale - 1 dB and gain GIDL, GIDR = 0 dB		90		dB
THD+N[b]	A-weighted, 1 kHZ sine wave@ Full Scale -60 dB and gain GID = 0 dB		100		dB
Digital Gain	Gain GID when activated	0	-	43	dB
Gain Step	GID @ 1 kHz	-	1	-	dB
Gain Accuracy	GID @ 1 kHz	-0.25	-	+0.25	dB

a. The Full Scale input corresponds to a modulation density of the PDM input

b. The specified value is extrapolated by adding 60 dB to the measured SNR

2.9 Storage Condition

The SoC series is applicable to Moisture Sensitivity Level 3 (based on JEDEC Standard).

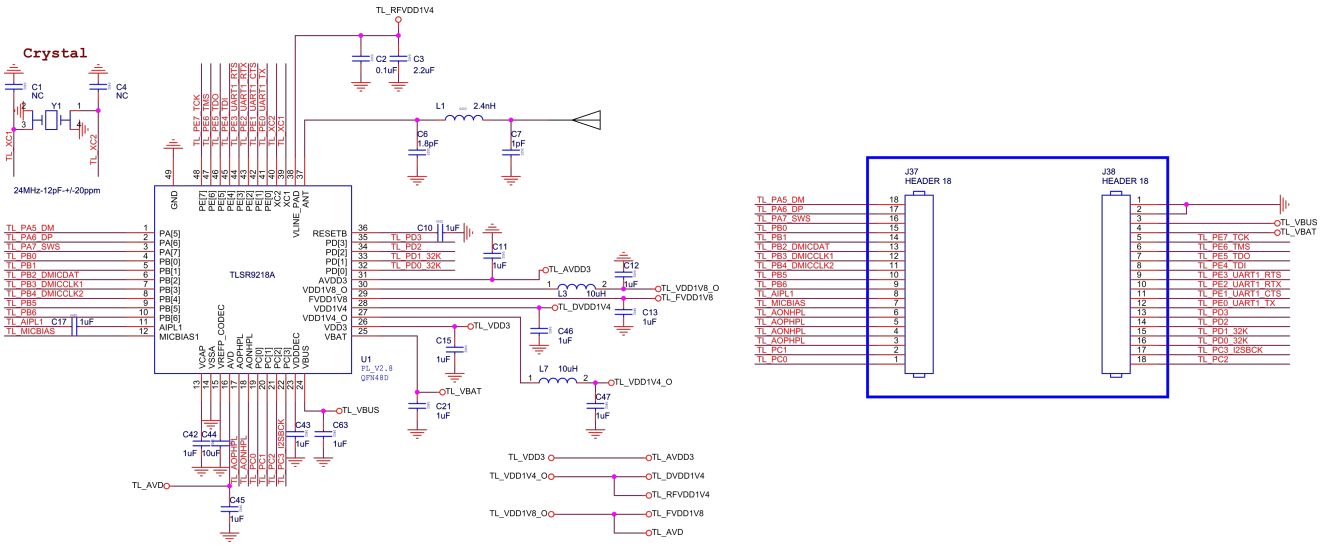
1. Calculated shelf life in sealed moisture barrier bag (MBB): 12 months at <40°C and <90% relative humidity (RH)
2. Peak package body temperature: 260°C
3. After bag is opened, devices that will be subjected to reflow solder or other high temperature process must be
   - Mounted within: 168 hours of factory conditions <=30°C/60% RH, or
   - Stored at <10% RH
4. Devices require bake, before mounting, if:
   - Humidity Indicator Card reads >10% when read at 23 ± 5°C
   - Both of the conditions in 3 are not met
5. If baking is required, devices may be baked for 24 hours at 125 ±5°C

Note: If device containers cannot be subjected to high temperature or shorter bake times are desired, please refer to IPC/JEDEC J-STD-033 for bake condition.

3 Reference Design

3.1 Schematic

Figure 3-1 Schematic



3.2 BOM (Bill of Material)

Table 3-1 BOM Table

Quantity	Reference	Value	Description	PCB Footprint
2	C1, C4	NC	Capacitance,X5R,±10%	0402
1	C2	0.1uF	Capacitance,X5R,±10%	0402
1	C3	2.2uF	Capacitance,X5R,±10%	0402
1	C6	1.8pF	Capacitance,C0G,±0.1pF	0402
1	C7	1pF	Capacitance,C0G,±0.1pF	0402
13	C10, C11, C12, C13, C15, C17, C21, C42, C43, C45, C46, C47, C63	1uF	Capacitance,X5R,±10%	0402
1	C44	10uF	Capacitance,X5R,±10%	0603C
1	L1	2.4nH	High frequency chip inductor,±0.3nH	0402

Quantity	Reference	Value	Description	PCB Footprint
2	L3, L7	10uH	High frequency chip inductor,SMD,20%	1008L
1	U1	TLSR9218A	SOC_RF	qfn_6x6_48pin_0p4_3p3x3p3
1	Y1	24MHz-12pF-+/-20ppm	XTAL SMD 3225,24 MHz,CI=12pF,total tol.±20ppm	OSCCC250X320X110

4 Memory, MCU and PMU

4.1 Memory

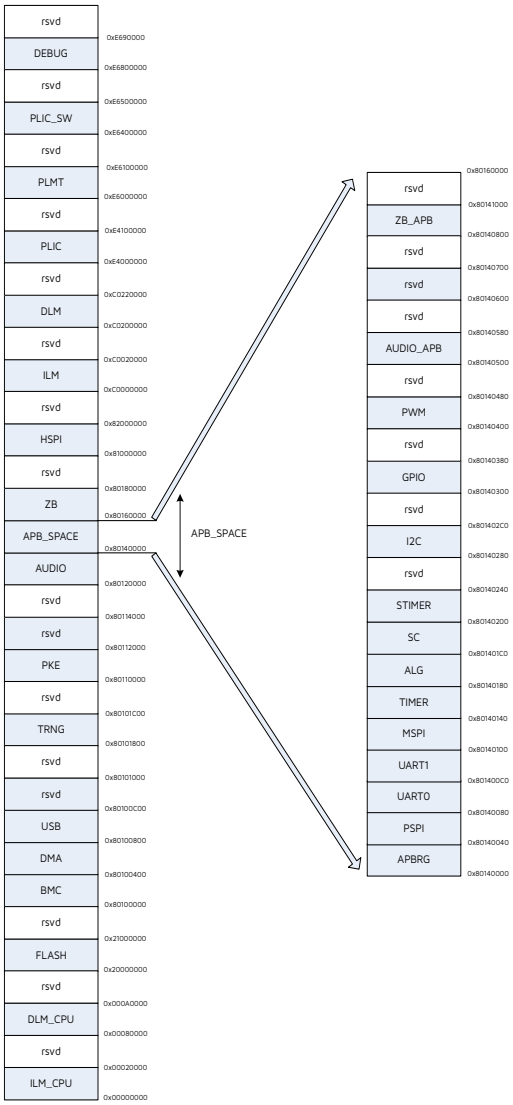
The SoC embeds 128 KB SRAM (including 64KB with retention in deep sleep) as instruction memory, 128 KB SRAM as data memory, and 1MB internal FLASH as program memory.

4.1.1 SRAM

Memory map is shown below.

As shown in Figure 4-1, the SoC embedded 2 SRAM, 128 KB instruction local memory (ILM) and 128 KB data local memory (DLM). For ILM, the lower 64 KB is with retention in deep sleep. ILM and DLM have different addressing address for MCPU (shown as ILM_CPU and DLM_CPU). Please be noted, ILM can store both instruction and data while DLM can only store data, so the instruction stored in local memory should be less than 128 KB.

Figure 4-1 Memory Map



4.1.2 Flash

The internal Flash mainly supports page program, sector/block/chip erase operations, and deep power down operation. Please refer to the corresponding SDK for Flash memory operation details.

MCU uses the separate MSPI_CLK frequency to load instructions, and adopts flash driver to access (read/write) flash with the same speed.

4.1.3 E-Fuse

The non-volatile E-Fuse section is preloaded with 4-byte decryption key and 4-byte E-Fuse configuration, decryption key is shown below.

Table 4-1 E-Fuse Information

E-Fuse bit information	Decryption Key
	Bit0~31

4.1.4 Unique ID

For chip identification and traceability, the flash is preloaded with 128-bit Unique ID (UID). This UID can be read via the interface in SDK.

4.2 MCU

The SoC embeds a 32-bit RISC-V micro-controller, features are listed as following:

1. 5-stage in-order execution pipeline
2. Fast Hardware multiplier
3. Hardware divider
4. Dynamic branch prediction
   - 32-entry branch target buffer (BTB)
5. Performance monitors
6. Misaligned memory accesses
7. RISC-V RV32I base integer instruction set
8. RISC-V RVC standard extension for compressed instructions
9. RISC-V RVM standard extension for integer multiplication and division
10. RISC-V RVA standard extension for atomic instructions
11. RISC-V RVN standard extension for user-level interrupt and exception handling
12. RISC-V "F" standard extensions for single-precision floating-point
13. I & D caches
14. I & D local memories

4.3 Working Mode

The SoC supports six working modes, including Active, Idle, Suspend, Deep Sleep with SRAM retention, Deep Sleep without SRAM retention, and Shutdown.

- The Power Management (PM) module is always active in all working modes.
- For modules such as MCU, RF transceiver (Radio), and SRAM, the state depends on working mode, as shown below.

Table 4-2 Working Mode

Mode	Active	Idle	Suspend	Deep Sleep With SRAM Retention	Deep Sleep without SRAM Retention	Shut Down
MCU	active	stall	stall	off	off	off
Radio	available	available	off	off	off	off
USB	available	available/off	stall/off	off	off	off
Audio	available	available/off	stall/off	off	off	off
Wakeup Time to Active Mode in LDO Mode	-	0 μs	100 μs	Shorter than Deep sleep without retention, almost same as Suspend	1 ms	10 ms
Wakeup Time to Active Mode in DCDC Mode	-	-	-	-	-	-
retention SRAMs (with retention in deep sleep)	full	full	full	full	off	off
Wakeup on RTC (32K Timer wakeup)	-	-	available	available	available	off
Wakeup on pin (IO wakeup)	-	-	available	available	available	off
Wakeup on interrupt	-	available	-	-	-	-

Mode	Active	Idle	Suspend	Deep Sleep With SRAM Retention	Deep Sleep without SRAM Retention	Shut Down
Wakeup on reset pin (RESETB)	-	available	available	available	available	on

NOTE:

- *active: MCU is at working state.*
- *stall: In Idle and Suspend mode, MCU does not work, while its clock is still running.*
- *available for modules: It's selectable to be at working state, or stall/be powered down if it does not need to work.*
- *available/on for wakeup: Corresponding wakeup method is supported.*
- *off for wakeup: Corresponding wakeup method is not supported.*
- *full/off for SRAMs:*
  - *full: Full speed. In Active, Idle and Suspend mode, the two 16kB retention SRAMs are powered on and work normally (can be accessed); in Deep sleep with SRAM retention, the retention SRAMs are powered on, however, the contents of the retention SRAMs can be retained and cannot be accessed.*
  - *off: The retention SRAMs are powered down in Deep sleep without SRAM retention and Shutdown mode.*

Analog registers (0x37 ~ 0x3f) as shown in Table 4-3 are retained in deep sleep mode and can be used to store program state information across deep sleep cycles.

- Analog registers 0x3c~0x3f are non-volatile even when chip enters deep sleep or chip is reset by watchdog or software, i.e. the contents of these registers won't be changed by deep sleep or watchdog reset or chip software reset.
- Analog registers 0x37~0x3b are non-volatile in deep sleep, but will be cleared by watchdog reset or chip software reset.
- After POR (Power-On-Reset), all registers will be cleared to their default values, including these analog registers.

User can set flag in these analog registers correspondingly, so as to check the booting source by reading the flag.

Table 4-3 Retention Analog Registers in Deep Sleep

Address	R/W	Description	Reset Value
afe_0x37	R/W	buffer clean at watchdog	11111111
afe_0x38	R/W	buffer clean at watchdog	00000000

Address	R/W	Description	Reset Value
afe_0x39	R/W	buffer clean at watchdog	00000000
afe_0x3a	R/W	buffer clean at watchdog	00000000
afe_0x3b	R/W	buffer clean at watchdog	00000000
afe_0x3c	R/W	buffer clean at power on	00000000
afe_0x3e	R/W	buffer clean at power on	00000000
afe_0x3f	R/W	buffer clean at power on	00001111

4.4 Reset

The chip supports three types of reset methods, including POR (Power-On-Reset), watchdog reset and software reset.

1. POR: After power on, the whole chip will be reset, and all registers will be cleared to their default values.
2. Watchdog reset: A programmable watchdog is supported to monitor the system. If watchdog reset is triggered, registers except for the retention analog registers 0x3c~0x3f will be cleared.
3. Software reset: It is also feasible to carry out software reset for the whole chip or some modules.
   - Setting address 0x2f[5] as 1b'1 is to reset the whole chip. Similar to watchdog reset, the retention analog registers 0x3c~0x3f are non-volatile, while other registers including 0x37~0x3b will be cleared by chip software reset.
   - Addresses 0x20~0x22 serve to reset individual modules: if some bit is set to logic "1", the corresponding module is reset.

Table 4-4 Register Configuration for Software Reset

Address	R/W	Description	Reset Value
0x20	R/W	Reset control, 1 for reset, 0 for clear [0]: HSPI [1]: I2C [2]: UART0 [3]: USB [4]: PWM0 [5]: RSVD [6]: UART1 [7]: Swire	0x80

Address	R/W	Description	Reset Value
0x21	R/W	[0] RSVD [1] System Timer [2] DMA [3] ALGM [4] PKE [5] RSVD [6] PSPI [7] SPISLV	0x80
0x22	R/W	[0] TIMER [1] AUDIO [2] TRNG [3] RESET MCU DISABLE [4] MCU RESET ENABLE [5] LM [6] RSVD [7] RSVD	0x38
0x23	R/W	[0] ZB [1] ZB_MSTCLK [2] ZB_LPCLK [3] ZB_CRYPT [4] RSVD [5] RSVD [6] SARADC [7] ALG	0x80
0x2f	R/W	[0] suspend enable (RW) [4] ramcrc_clren_tgl [5] rst all (act as watchdog reset) [6] rsvd (mcu low power mode) (W) [7] stall mcu trig If bit[0] set 1, then system will go to suspend. Or only stall mcu (W)	0x00

4.5 Power Management

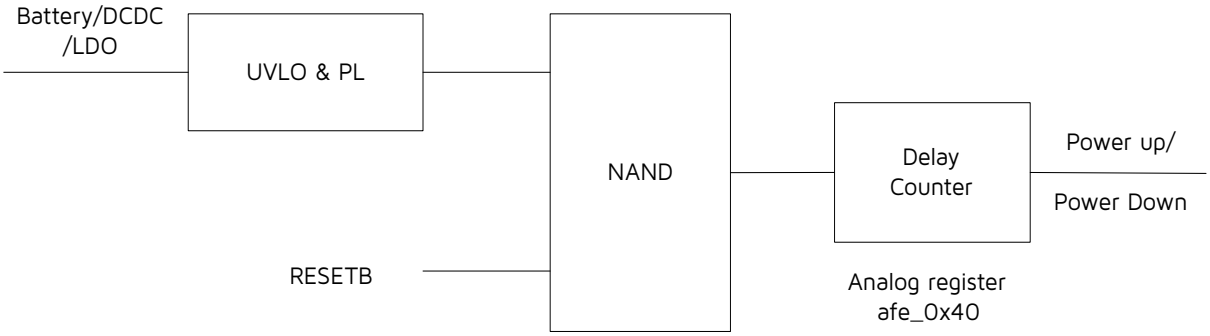
The multiple-stage Power Management (PM) module is flexible to control power state of the whole chip or individual functional blocks such as MCU, RF Transceiver, and peripherals by the following methods:

1. Power-On-Reset (POR) and Brown-out detect
2. Working Mode Switch
3. LDO and DCDC
4. VBAT and VANT Power-Supply Mode

4.5.1 Power-on-Reset (POR) and Brown-out Detect

Figure below shows the control logic of power up/down.

Figure 4-2 Control Logic of Power up/down



As shown in figure above, the whole power up and down is controlled by the UVLO (Ultra-low Voltage Lockout) & PL (Power Logic) module and the external RESETB pin via the logic shown in the above diagram. UVLO takes the external power supply as input and releases the lock only when the power supply voltage is higher than a preset threshold. The RESETB pin has an internal pull-up resistor; an external Cap can be connected on the RESETB pin to control the POR delay.

After both UVLO and RESETB release, there is a further configurable delay before the system reset signal ("Sysrst") is released. The delay is adjusted by analog register afe_0x40. Since the content of afe_0x40 is reset to default only after power cycle, watchdog reset, or software reset, the delay change using afe_0x40 is only applicable when the chip has not gone through these reset conditions. For example, after deep sleep wakeup, the setting in afe_0x40 will take effect.

Register afe_0x40 is described in table below.

Table 4-5 Analog Register to Control Delay Counters

Address	R/W	Description	Reset Value
afe_0x40	R/W	base on 16KHz frequency increase counter(8ms)	10000000

Power up and power down sequences are shown in figures below.

Figure 4-3 Initial Power-up Sequence

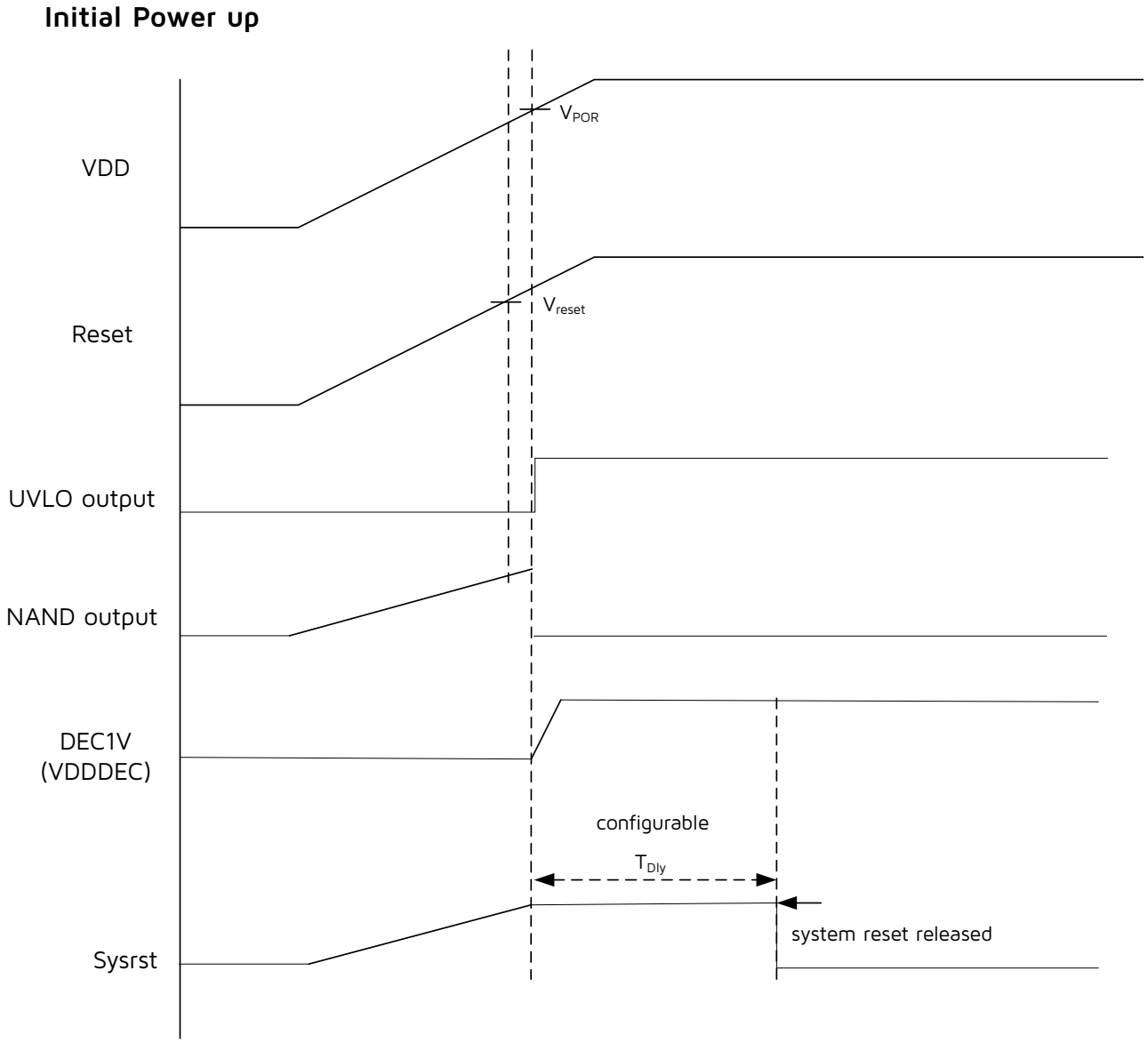


Figure 4-4 Initial Power-down Sequence

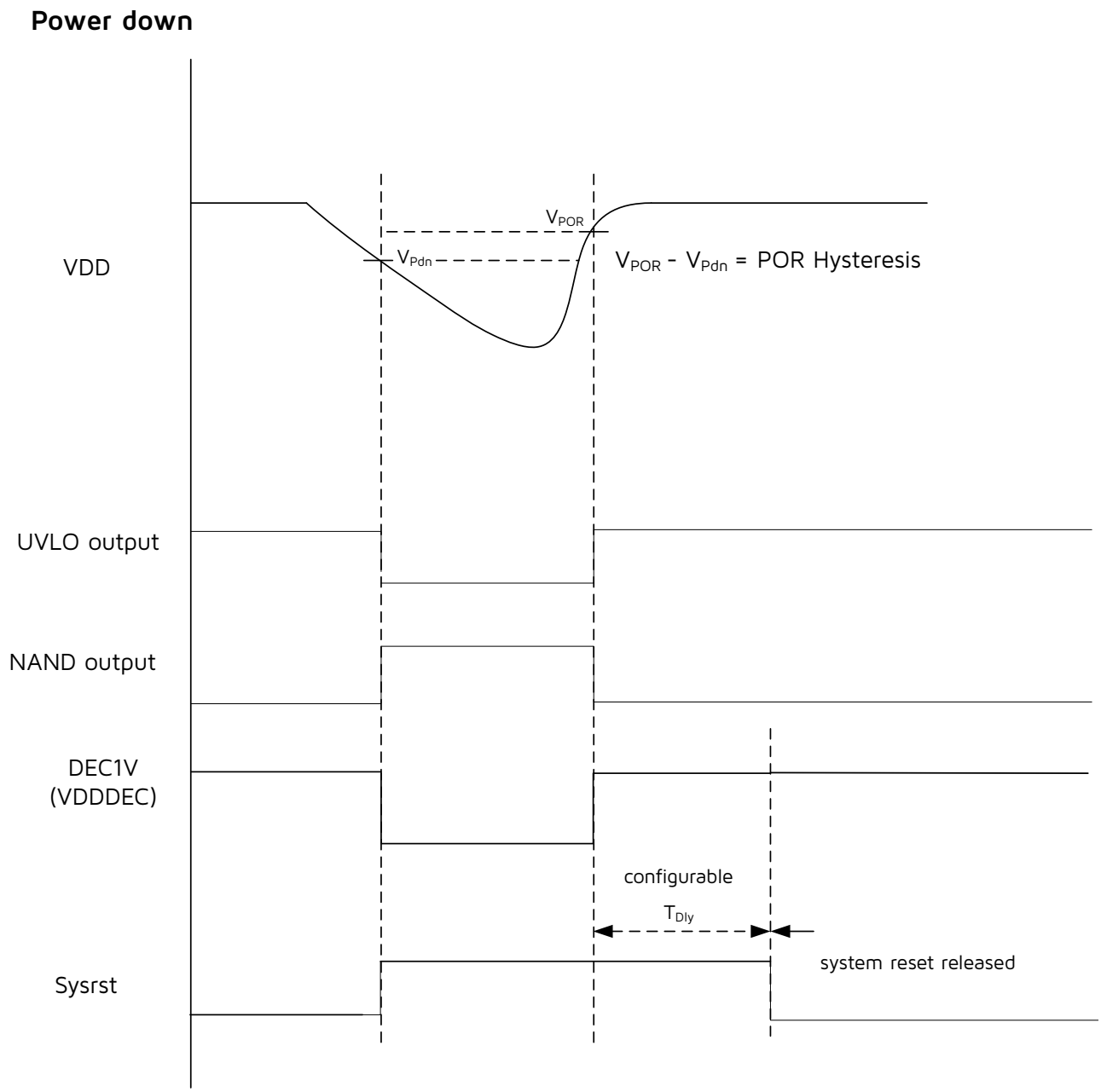


Table 4-6 Characteristics of Initial Power-up/Power-down Sequence

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{POR}	VDD voltage when V_{UVLO} turns to high level	-	1.73	-	V
V_{PDN}	VDD voltage when V_{UVLO} turns to low level	-	1.61	-	V
T_{DLY}	Delay counter value	Configurable via analog register afe_0x40			

4.5.2 Working Mode Switch

In Active mode, MCU is active, all SRAMs are accessible, and other modules are selectable whether to be at working state.

The chip can switch to Idle mode to stall the MCU. In this mode, all SRAMs are still accessible, modules such as RF transceiver, USB are still selectable whether to be at working state. The chip can be triggered to Active mode by interrupt or RESETB pin, and the time to switch to Active mode is negligible.

To decrease power consumption to different levels, the chip can switch to power saving mode (Suspend, Deep sleep with SRAM retention, Deep sleep without SRAM retention, Shutdown) correspondingly.

- In Suspend mode, MCU stalls, all SRAMs are still accessible, the PM module is active, and modules such as RF transceiver, USB are powered down. The chip can be triggered to Active mode by 32K Timer, IO pin or RESETB pin. It takes 100 μs or so to switch from Suspend mode to Active mode.
- In Deep sleep with SRAM retention, the PM module is active, analog and digital modules except for the retention SRAMs are powered down, while the retention SRAMs can be retained and not accessible. The chip can be triggered to Active mode by 32K Timer, IO pin or RESETB pin. The time to switch to Active mode is shorter than Deep sleep without SRAM retention and close to Suspend.
- In Deep sleep without SRAM retention, only the PM module is active, while analog and digital modules including the retention SRAMs are powered down. The chip can be triggered to Active mode by 32K Timer, IO pin or RESETB pin. The time to switch to Active mode is 1 ms or so.
- In Shutdown mode, all digital and analog modules are powered down, and only the PM module is active. The chip can be triggered to Active mode by RESETB pin only. The time to switch to Active mode is 10 ms or so.

User can directly invoke corresponding library function to switch working mode of the chip.

If certain module doesn't need to work, user can power down this module in order to save power.

Table 4-7 3.3 V Analog Register for Module Power up/down Control

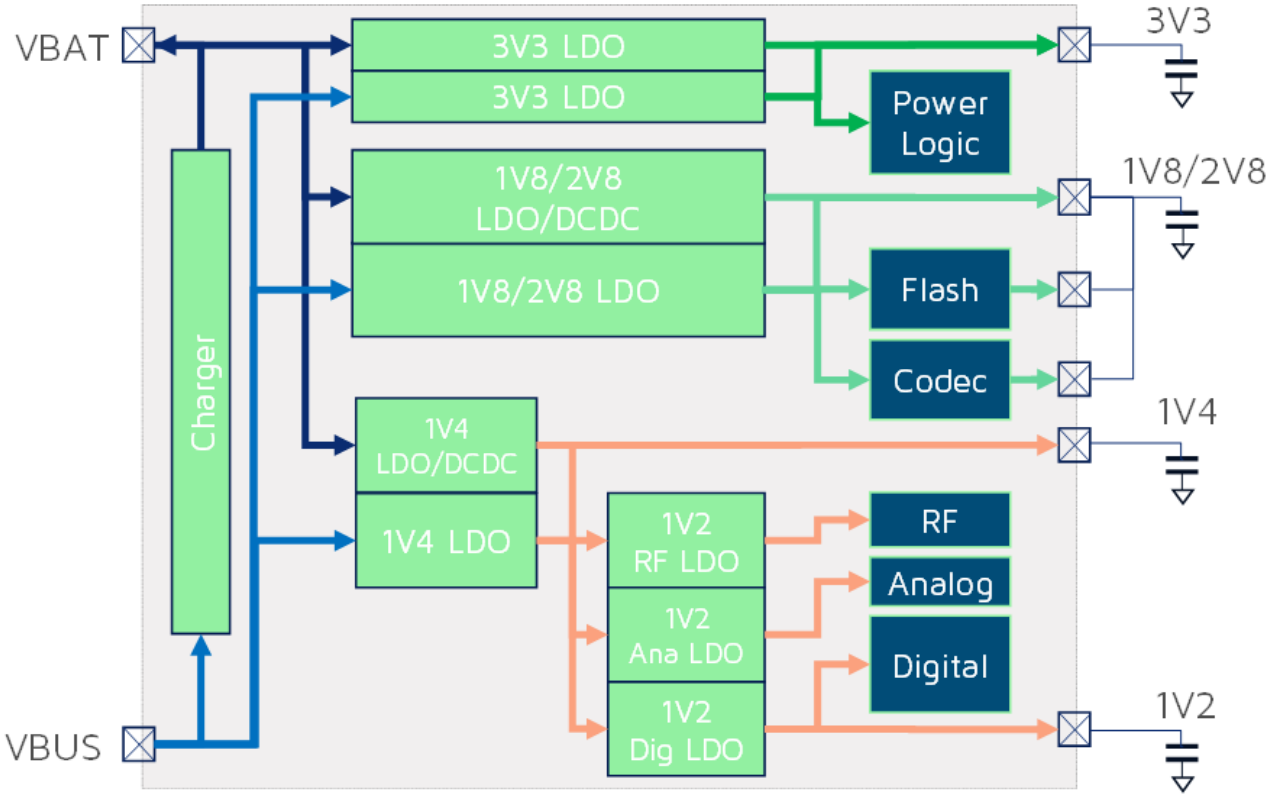
Address	R/W	Description	Reset Value
0x4c	R/W	[0] 1: auto power down 32KRC [1] 1:auto power down 32K xtal [2] 1: auto power down bbpll/ temp_sensor [3] 1:auto power down 24M xtal [4] 1:auto power power logic [5] 1:auto power down dcdc [6] 1:auto power down vbus LDO [7] 1:auto power down ana LDO	0x0

Address	R/W	Description	Reset Value
0x4d	R/W	[0] 1: auto power down low power comparator [1] RSVD [2] RSVD [3] 1:reset xtal quick start cnt [4] 1:auto power suspend ldo [5] 1:auto power retention ldo [6] 1: power down sequence enable [7] 1: enable isolation	0x0

4.5.3 LDO and DCDC

The diagram of LDO and DCDC module is shown as following.

Figure 4-5 LDO and DCDC



As shown in figure above, the SoC embeds two 3.3 V LDO, which can generate 3.3 V voltage output and supply power for Power logic module; one 1.8 V/2.8 V LDO, one 1.8 V/2.8 V DCDC, which can generate 1.8 V/ 2.8 V voltage output and supply power for Flash and CODEC modules; one 1.4 V LDO/DCDC , which can generate 1.4 V voltage output, one 1.4 V LDO, which can generate 1.4 V voltage as the input of the three 1.2

V LDO; the three 1.2 V LDO can generate 1.2 V voltage output and supply power for RF, Analog, and Digital modules respectively.

4.5.4 VBAT and VANT Power-Supply Mode

The chip provides two power-supply modes including VBAT mode and VANT mode.

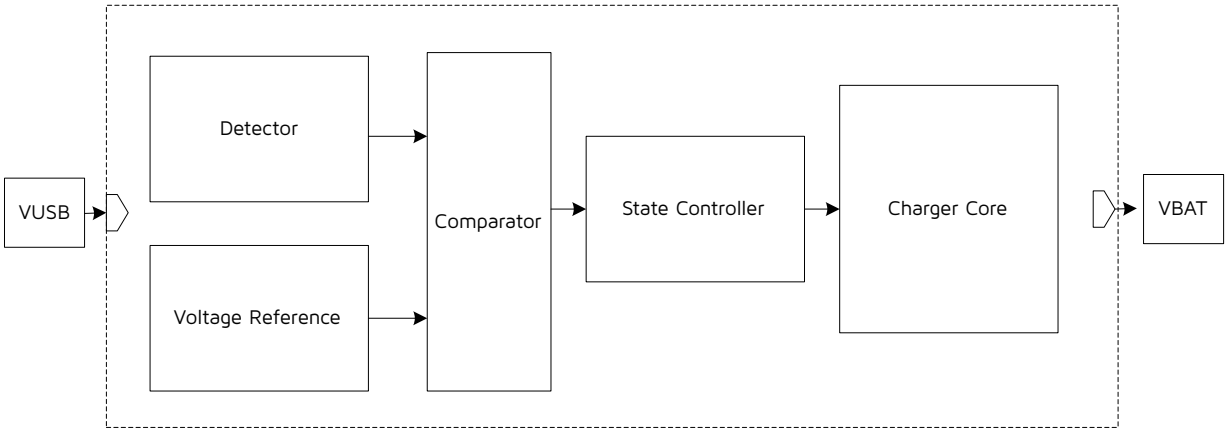
- In VBAT mode, the chip is directly supplied with power by its battery voltage. The maximum output power is related to power supply voltage, for example, the maximum power is 10 dBm or so at 3.3 V power supply.
- In VANT mode, the chip is supplied with 1.2 V voltage by the embedded DCDC and LDO. In this mode, output power won't change with AVDD basically, and the maximum power is 5 dBm or so. Corresponding to the VBAT mode, the VANT mode is more power-saving at the same TX power.

4.6 Charger

The SoC supports linear charger function.

The charger structure is shown as following:

Figure 4-6 Charger Diagram



As shown in figure above, VUSB is the standard charger port, VBAT is the battery port. The detector detects voltage of VUSB, voltage of VBAT, the total current of VUSB, and the temperature change of the SoC; the detecting results will be transfered to different voltage signals, sent to comparator to compare with reference voltages generated by Voltage Reference Module, the comparison results will be sent to state controller to control the charger core.

When VUSB is booting, the charger loops to check whether the values of the detected parameters in the detector exceed the threshold, if yes, the state machine will shut down the charger core and stop charging until the detected parameters return to normal. The thresholds are:

- USB port voltage VUSB should be not higher than 5.6 V
- Battery voltage VBAT should be not higher than 4.35 V
- VBAT should not be higher than VUSB
- The total current drawn from USB should be not higher than 120 mA
- The SoC temperature should not be higher than 125°C

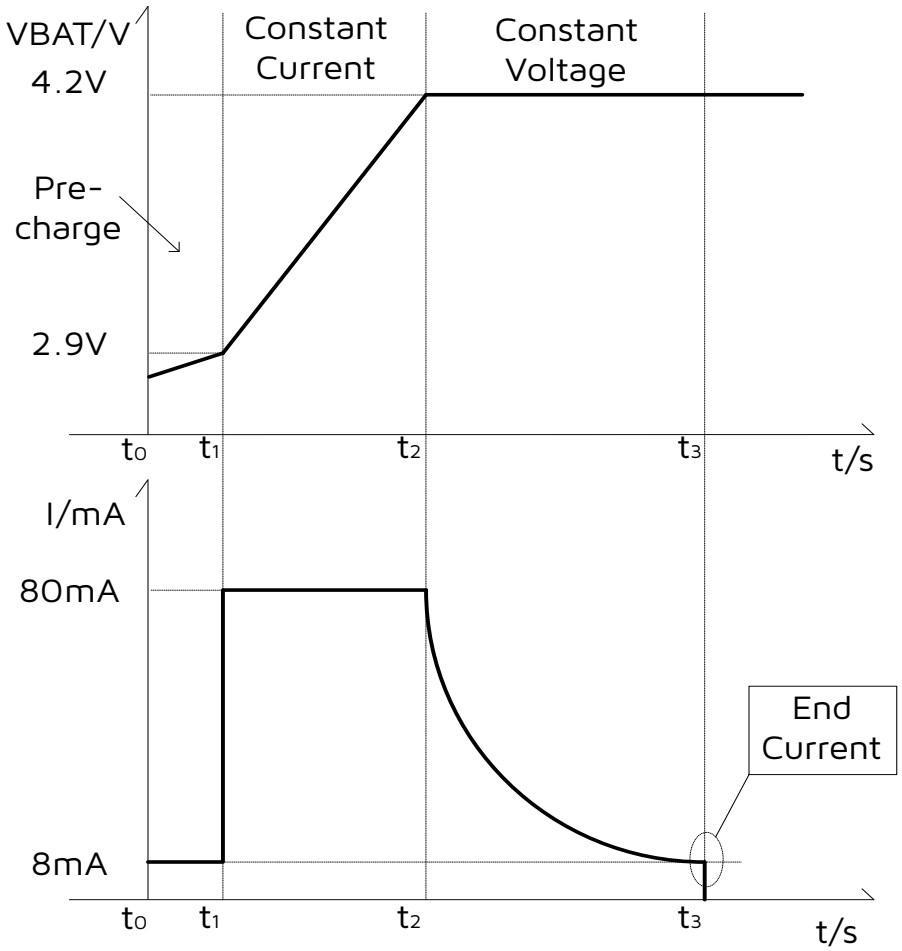
The charging processing can be divided into 4 stages:

1. Pre-charging (trickle changing, TC), when the detector detects VBAT is lower than 2.9 V, the charger core will enter the TC working mode, and the charging current is 1/10 of the constant current charging working mode. Constant current and corresponding TC current can be modified by setting register 0x1a<3:0>, the trimming step is 5 mA, the default value of 0x1a<3:0> is 0x1011, the corresponding constant current is 80 mA.
2. Constant current charging (CC): when VBAT is higher than 2.9 V and lower than 4.2 V, the charger core will enter the CC mode, the charging current will keep to the maximum value, and VBAT increases.
3. Constant voltage charging (CV): when VBAT reaches 4.2 V the charger core will enter the CV mode, the charging voltage will keep 4.2 V and the charing current will decrease.
4. End of charging: when the charging current is lower then 1/10 of the CC current, the charging will end.

The above is the normal charging process, when then charging ends, and VUSB is still connected, when VBAT is lower then 4.05 V, charger will start working again to repeat the above process, this is recharging.

The VBAT timing diagram and charging current timing diagram is shown as below.

Figure 4-7 Charging Timing Diagram



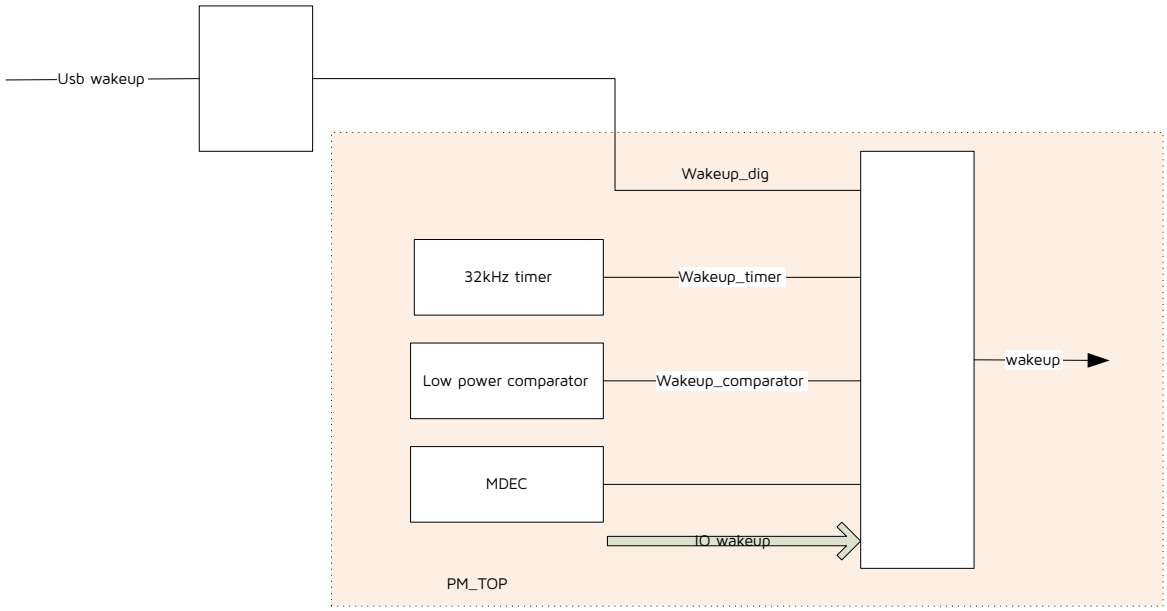
When ending constant current charging and start constant voltage charging, the changing current will drop to about 1/10 of the former constant charging current, and that is the end current. Please be noted, the values shown in the diagram are theoretical values, the actual constant charging current ranges from 75.83 mA to 81.05 mA, and the actual ending current ranges from 12 mA to 14 mA.

As shown above, the charger is designed for lithium battery (voltage ranges from 3.6 V to 4.2 V).

4.7 Wakeup Source

Figure below shows wake up sources of the SoC.

Figure 4-8 Wake up Sources



Each wake up source is detailed below:

USB

This wakeup source can only wake up the system from suspend mode.

Once USB host sends out resuming signal, the system will be woke up.

32 kHZ Timer

This wakeup source is able to wake up the system from suspend mode or two deep sleep modes.

Low Power Comparator

This wakeup source is able to wake up the system from suspend mode or two deep sleep modes.

IO

This wakeup source is able to wake up the system from suspend mode or two deep sleep modes.

MDEC

This wakeup source is able to wake up the system from suspend mode or two deep sleep modes.

Table 4-8 Analog Register for Wakeup

Address	R/W	Description	Reset Value
0x41	R/W	wakeup polarity 0: high level wakeup,1: low level wakeup	0x0
0x42	R/W	wakeup polarity 0: high level wakeup,1: low level wakeup	0x0
0x43	R/W	wakeup polarity 0: high level wakeup,1: low level wakeup	0x0
0x44	R/W	wakeup polarity 0: high level wakeup,1: low level wakeup	0x0
0x45	R/W	wakeup polarity 0: high level wakeup,1: low level wakeup	0x0
0x46	R/W	PA wakeup enable	0x0
0x47	R/W	PB wakeup enable	0x0
0x48	R/W	PC wakeup enable	0x0
0x49	R/W	PD wakeup enable	0x0
0x4a	R/W	PE wakeup enable	0x0
0x4b	R/W	[0] dly sel enable, 1:enable delay controller by pad,delay 0xfb.0: disable pad controller dly, delay is controlled by r_dly; [1] RSVD [2] pad wakeup filter, 1:pad wakeup filter enable 0: disable filter [3] pad wakeup enable [4] dig wakeup enable [5] timer wakeup enable [6] comparator wakeup enable [7] mdec wakeup enable	0x1000000

Address	R/W	Description	Reset Value
0x64	R	write 1 to clean the status: [0]:wkup cmp [1]:wkup timer [2]:wkup dig [3]:wkup pad [4]:wkup mdec [7:5] rsvd	-

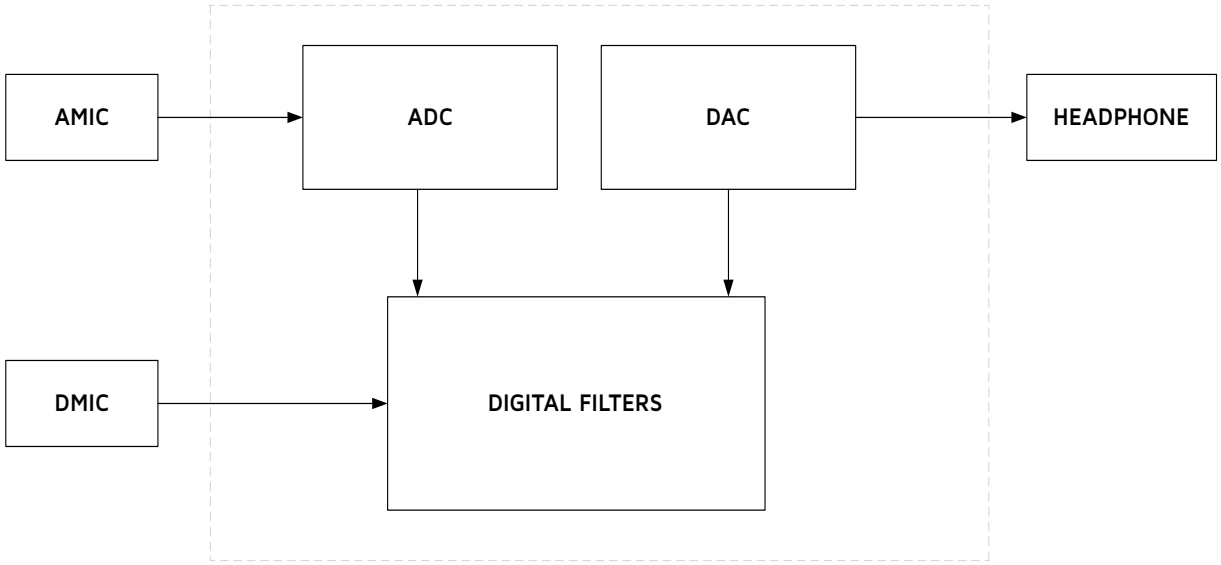
5 Audio

Audio module consists of 3 parts: audio CODEC, audio in path, and audio out path.

5.1 Audio CODEC

Audio CODEC is shown as following:

Figure 5-1 Audio CODEC



As shown in the figure above, audio CODEC consists of one stereo Analog to Digital Converter (ADC), one stereo Digital to Analog Converter (DAC), digital filters and data and control interface.

ADC supports mono/stereo line-in/AMIC/DMIC as input. Line-in is the default input, the analog boost gain ranges from 0 to 20 dB, with stage of 4 dB, the digital programmable gain ranges from 1 to 43 dB, with stage of 1 dB.

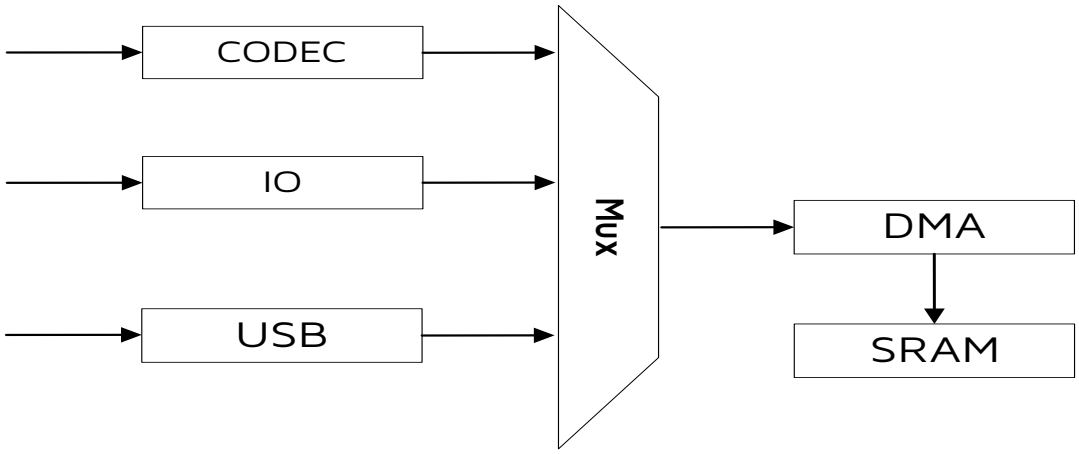
To choose AMIC and DMIC path, related register should be configured, please refer to register table.

DAC supports stereo differential analog output, with analog programmable gain ranging from -19 to +12 dB, and digital programmable gain from -31 to +32 dB, both gain stages are 1 dB.

5.2 Audio-in Path

Audio in path is illustrated in figure below.

Figure 5-2 Audio in Path

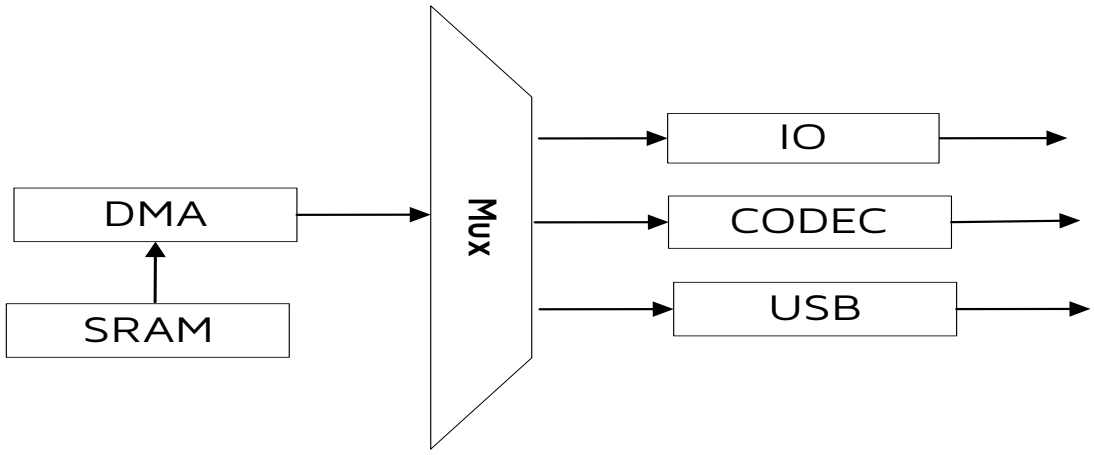


As can be seen in figure above, there are 3 audio input sources, i.e., signal via CODEC, signal from external IO, and USB signal, which can be chosen by configuring registers. Input signal will be produced and then saved in SRAM.

5.3 Audio-out Path

Audio out path is shown in figure below:

Figure 5-3 Audio out Path



As shown in figure above, the audio output signals has 3 output channels, i.e., output via CODEC, external IO, and USB signal.

5.4 Register Description

Audio path related registers are listed in the table below. The base address for the following registers is 0x80140500.

Table 5-1 Audio Registers

Offset	R/W	Description	Reset Value
0x00	R/W	[0]:clk_en, i2s clk enable: 1'b1: enable 1'b0: disable. [1]:clk_div2, i2s clk divide2 enable: 1'b1: enable 1'b0: disable. [2]:mc_clken, codec mc clk enable: 1'b1: enable 1'b0: disable. [3]:mc_clk_inv_o, codec mc clk invert.	0x00
0x01	R/W	[1:0]: format [3:2]: wl [4]: lrp [5]: lrswap [6]: dci_ms [7]: bclkinv	0xca
0x02	R/W	[1:0]: bcm_bits [2]: bclkinv_o [3]: rst_fifo	0x00
0x03	R/W	[1:0]: i2s_cmode [2]: codec_i2s_sel [3]: i2s_out_bit_sel	0x00
0x04	R/W	[1:0]: i2s_ain0_come [3:2]: i2s_ain1_come [6:4]: i2s_ain2_come	0x00
0x05	R/W	[1:0]: ain0_sel [3:2]: aout0_sel [5:4]: ain1_sel [7:6]: aout1_sel	0xff
0x08	R/W	[6:0]: i2c_addr	0x00
0x0a	R/W	[3:0]: aout0_fifo_trig_num [7:4]: ain0_fifo_trig_num	0x17

Offset	R/W	Description	Reset Value
0x0b	R/W	[3:0]: aout1_fifo_trig_num [7:4]: ain1_fifo_trig_num	0x17
0x0c	R/W	[3:0]: aout0_fifo_num	0x00
0x0d	R/W	[3:0]: ain0_fifo_num	0x00
0x0e	R/W	[3:0]: aout1_fifo_num	0x00
0x0f	R/W	[3:0]: ain1_fifo_num	0x00
0x10	R/W	[0]: tx_ptr_sel [4]: rx_ptr_sel	0x00
0x11	R/W	[0]: tx_wptr_en [1]: tx_rptr_en [2]: rx_wptr_en [3]: rx_rptr_en	0x0f
0x1f	R/W	[0]: txfifo_empty [1]: rxfifo_empty	0x03
0x20	R/W	[7:2]: tx_wptr0	0x00
0x21	R/W	[5:0]: tx_wptr1	0x00
0x22	R/W	[7:2]: tx_rptr0	0x00
0x23	R/W	[5:0]: tx_rptr1	0x00
0x24	R/W	[7:0]: txfifo_num0	0x00
0x25	R/W	[5:0]: txfifo_num1	0x00
0x26	R/W	[7:0]: tx_max0l	0x00
0x27	R/W	[5:0]: tx_max0h	0x00
0x28	R/W	[7:2]: rx_wptr0	0x00
0x29	R/W	[5:0]: rx_wptr1	0x00
0x2a	R/W	[7:2]: rx_rptr0	0x00
0x2b	R/W	[5:0]: rx_rptr1	0x00
0x2c	R/W	[7:0]: rxfifo_num0	0x00
0x2d	R/W	[5:0]: rxfifo_num1	0x00
0x2e	R/W	[7:0]: rx_max0l	0x00

Offset	R/W	Description	Reset Value
0x2f	R/W	[5:0]: rx_max0h	0x00
0x30	R/W	[7:0]: thd0_h_l1_l	0x00
0x31	R/W	[5:0]: thd0_h_l1_h	0x00
0x32	R/W	[7:0]: thd0_l_l1_l	0x00
0x33	R/W	[5:0]: thd0_l_l1_h	0x00
0x34	R/W	[3:0]: i2s_dactune_l1	0x00
0x38	R/W	[7:0]: thd0_h_l2_l	0x00
0x39	R/W	[5:0]: thd0_h_l2_h	0x00
0x3a	R/W	[7:0]: thd0_l_l2_l	0x00
0x3b	R/W	[5:0]: thd0_l_l2_h	0x00
0x3c	R/W	[3:0]: i2s_dactune_l2	0x00
0x40	R/W	[7:0]: thd1_h_l1_l	0x00
0x41	R/W	[5:0]: thd1_h_l1_h	0x00
0x42	R/W	[7:0]: thd1_l_l1_l	0x00
0x43	R/W	[5:0]: thd1_l_l1_h	0x00
0x44	R/W	[3:0]: i2s_adctune_l1	0x00
0x48	R/W	[7:0]: thd1_h_l2_l	0x00
0x49	R/W	[5:0]: thd1_h_l2_h	0x00
0x4a	R/W	[7:0]: thd1_l_l2_l	0x00
0x4b	R/W	[5:0]: thd1_l_l2_h	0x00
0x4c	R/W	[3:0]: i2s_adctune_l2	0x00
0x50	R/W	[7:0]: int_pcm_num0	0x00
0x51	R/W	[4:0]: int_pcm_num1	0x00
0x52	R/W	[7:0]: dec_pcm_num0	0x00
0x53	R/W	[4:0]: dec_pcm_num1	0x00
0x54	R/W	[3:0]: pcm_clk_num	0x00

Offset	R/W	Description	Reset Value
0x58	R/W	[0]: codec_mbist_mode [1]: codec_mbist_start [2]: codec_mbist_srstn [3]: codec_mbist_clken	0x00
0x59	R/W	[0]: codec_mbist_end_i [1]: codec_mbist_fail_i [2]: codec_mbist_ferr_i	0x00
0x5a	R/W	[2:0]: codec_mbist_clkdiv	0x00

6 BLE/802.15.4/2.4GHz RF Transceiver

6.1 Overview

The SoC integrates an advanced RF transceiver for 5.2 , 802.15.4, ANT, and 2.4Ghz application.

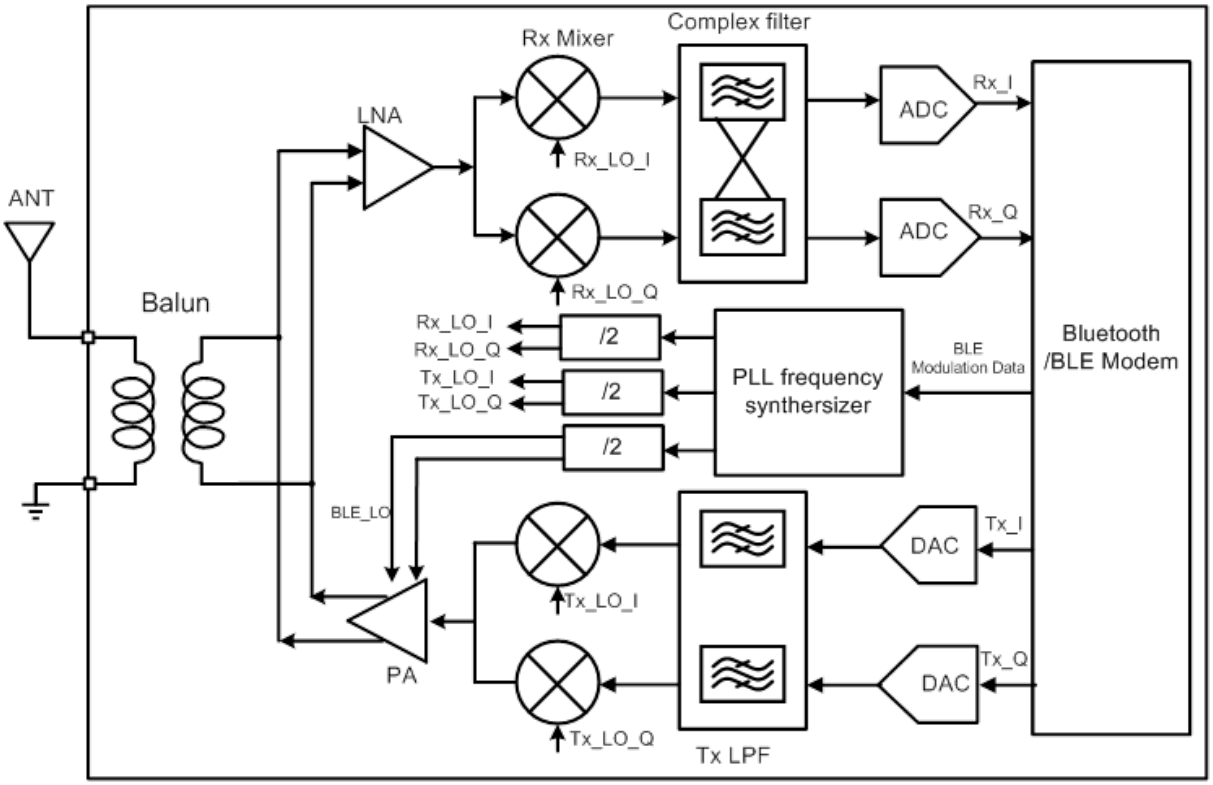
This RF transceiver works in the worldwide 2.4GHz ISM band and it consists of a fully integrated RF synthesizer, a Power Amplifier (PA), a Low Noise Amplifier (LNA), a TX LPF, a Rx complex filter, a TX DAC, a RX ADC, BLE/802.15.4 modulator/Demodulator and on-chip balun.

The BLE mode works in standard-compliant 1Mbps BLE mode, 2Mbps enhancement BLE mode,125Kbps BLE long range mode(S8), 500kbps BLE long range mode(S2).

The Zigbee mode works in IEEE 802.15.4 standard-compliant 250Kbps mode.

The block digram of the transceiver is shown below.

Figure 6-1 Block Diagram of RF Transceiver



6.2 Air Interface Data Rate and RF Channel Frequency

Air interface data rate, the modulated signaling rate for RF transceiver when transmitting and receiving data, is configurable via related register setting: 125kbps, 250kbps, 500kbps, 1Mbps, 2Mbps, 3Mbps.

RF transceiver can operate with frequency ranging from 2.400GHz to 2.4835GHz. The RF channel frequency setting determines the center of the channel.

6.3 Baseband

The baseband is disabled by default. The corresponding API is available for user to power on/down the baseband and enable/disable clock, so that the baseband can be turned on/off flexibly.

The baseband contains dedicated hardware logic to perform fast AGC control, access code correlation, CRC checking, data whitening, encryption/decryption and frequency hopping logic.

The baseband supports all features required by Bluetooth 5 and 802.15.4 specification.

6.3.1 Packet Format

Packet format in standard 1 Mbps BLE mode is shown in table below.

Table 6-1 Packet Format in Standard 1 Mbps BLE Mode

LSB MSB

Preamble (1 octet)	Access Address (4 octets)	PDU (2 ~ 257 octets)	CRC (3 octets)

Packet length 80 bit ~ 2120 bit (80 ~ 2120 μs @ 1 Mbps).

Packet format in standard 2 Mbps BLE mode is shown in table below.

Table 6-2 Packet Format in Standard 2 Mbps BLE Mode

LSB MSB

Preamble (2 octet)	Access Address (4 octets)	PDU (2 ~ 257 octets)	CRC (3 octets)

Packet length 44 bit ~ 1064 bit (44 ~ 1064 μs @ 2 Mbps).

Packet format in standard 500kbps/125kbps BLE mode is shown in table below.

Table 6-3 Packet Format in Standard 500 kbps/125 kbps BLE Mode

LSB MSB

Preamble (10 octet)	Access Address (4 octets)	CI (2 bits)	TERM1 (3 bits)	PDU (2 ~ 257 octets)	CRC (3 octets)	TERM2 (3 bits)

Packet format in 250 kbps 802.15.4 mode is shown in table below.

Table 6-4 Packet Format in 802.15.4 Mode

LSB MSB

Preamble (4~16 octet)	SFD (1 octet)	Frame Length (1 octet)	PSDU (Variable 0~127 octets)	CRC (2 octets)
SHR		PHR	PHY Payload	

Packet format in 2.4 GHz proprietary mode is shown in table below:

Table 6-5 Packet Format in Proprietary Mode

LSB MSB

Preamble (8 octet)	Access Address (configurable 3~5 bytes)	Packet Controller + Payload (1~33 bytes)	CRC (1~2 bytes)

6.3.2 BLE Location Function

In BLE 1M or 2M mode, BLE location features including AoA (Angel of Arrival) and AoD (Angle of Departure) are supported.

In the location mode of operation, the chip transmits a training sequence concatenated to the normal packet transmissions. In AoA mode of operation, the receiving side has multiple antennas and will be switched during the training sequence period. In AoD mode of operation, the transmitting side has multiple antennas and will be switched during the training sequence period. In either mode, the receiving side will be able to determine based on the phase variations of the received training sequences, the angle of location of the peer device.

6.3.3 RSSI and Frequency Offset

The SoC provides accurate RSSI (Receiver Signal Strength Indicator) and frequency offset indication.

- RSSI can be read from the 1 byte at the tail of each received data packet.
- If no data packet is received (e.g. to perform channel energy measurement when no desired signal is present), real-time RSSI can also be read from specific registers which will be updated automatically.
- RSSI monitoring resolution can reach +/-1 dB.
- Frequency offset can be read from the 2 bytes at the tail of the data packet. Valid bits of actual frequency offset may be less than 16 bits, and different valid bits correspond to different tolerance range.

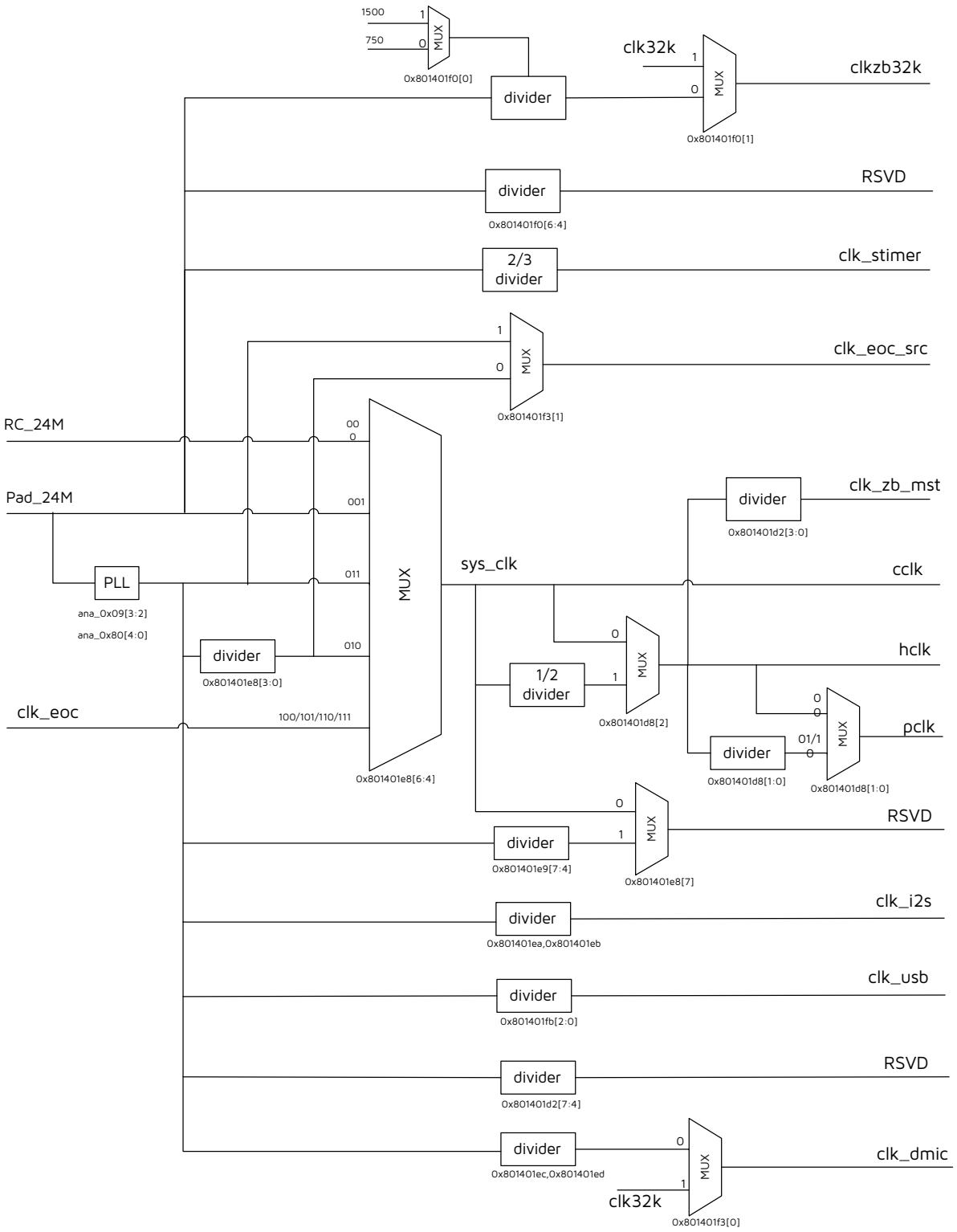
Telink supplies corresponding drivers for user to read RSSI and frequency offset as needed.

7 Clock

7.1 Clock Sources

The SoC's clock sources are a 24 MHz RC oscillator, an clk_eoc module, and an external 24 MHz crystal, as shown below.

Figure 7-1 Clock Source



The clock sources of each module is shown in table below.

Table 7-1 Clock Sources of Each Module

Module	Clock Source(s)
HSPI	hclk
I2C	pclk
UART0	pclk
USB	hclk, clk_usb
PWM	pclk, clk32k
UART1	pclk
SWIRE	hclk
STIMER	pclk, clk32k, clk_stimer
DMA	hclk
ALGM	pclk
PKE	hclk
PLMT	clk32k, hclk
PSPI	pclk
TIMER	pclk
AUDIO	hclk, clk_i2s
TRNG	hclk
MCU	cclk
LM	cclk
ZB	pclk, clkzb32k, clk_zb_mst, hclk
EOC	clk_eoc_src, pclk
GPIO	pclk
CODEC	clk_dmic

7.2 System Clock

There are five selectable clock sources for MCU system clock: RC_24M derived from 24 MHz RC oscillator, 24M crystal, sclk_div, pll clk and clk_eoc. The sources are selectable via register CLKSEL0.

7.3 Module Clock

Registers CLKEN0~CLKEN3 are used to enable or disable clock for various modules. By disable the clocks of unused modules, current consumption could be reduced.

7.3.1 System Timer Clock

System timer clock is derived from 24M crystal oscillator via a 2/3 frequency divider. The clock frequency is fixed as 16MHz.

7.3.2 USB Clock

USB clock is generated by pll_clk via frequency divider, the frequency is calculated with the following equations:

Fclk_usb = Fpllclk/n (n=0x801401fb[2:0], n=2~7)

Fclk_usb = Fpllclk/16 (n=0x801401fb[2:0], n=0)

7.3.3 I2S Clock

I2S clock is generated by pll_clk via frequency divider. Register I2S_STEP[7] should be set as 1'b1 to enable I2S clock. I2S clock frequency dividing factor contains step and mod. Register I2S_STEP[7] and r_i2s_mod_o serve to set I2S clock step[6:0] and mod[7:0] respectively, and mod should be no less than 2*step.

7.3.4 I2S clock frequency, Fi2s_clock, equals to pllclk*I2S_step[6:0]/I2S_mod[7:0].DMIC Clock

DMIC clock is derived from pllclk via a frequency divider.

Register DMIC_STEP should be set as 1'b1 to enable DMIC clock. DMIC clock frequency dividing factor contains step and mod. Register DMIC_STEP and DMIC_MOD serve to set DMIC clock step[6:0] and mod[7:0] respectively, and mod should be no less than 2*step.

DMIC clock frequency, Fdmic_clock, equals to pllclk*DMIC_step[6:0]/I2S_mod[7:0].

7.3.5 clkzb32k

When CLK_DIV[0] = 1, Fclkzb32k=Fpad_24m/1500; when CLK_DIV[0] = 0, Fclkzb32k=Fpad_24m/700;when register CLK_DIV[1] = 1, clkzb32k chooses clk32k, when CLK_DIV[1] = 0, clkzb32 chooses the clock generated by pad_24M via a frequency divider.

7.3.6 clk_eoc_src

clk_eoc_src is the system clock of eoc module. When register SEL[1] = 1, clk_eco_src = pllclk, when register SEL[1] = 0, clk_eco_src = pllclk_div.

7.3.7 clk_zb_mst

clk_zb_mst is generated by hclk via frequency divider.

Fclk_zb_mst = Fhclk/(n+1) (n=CLKEN3[3:0], n=0~15).

7.4 Register Table

Clock related registers are listed in table below. The base address of the following registers is 0x1401c0.

Table 7-2 Clock Related Registers

Address	R/W	Description	Reset Value
0x12	R/W	CLKMOD [3:0]: zb_mst_mod [7:4]: clknpe_mod	0xa3
0x24	R/W	CLKEN0 [0]: hspi [1]: i2c [2]: uart0 [3]: usb [4]: pwm0 [5]: reserved [6]: uart1 [7]: swires-	0x80
0x25	R/W	CLKEN1 [0]:reserved [1]:stimer [2]:dma [3]:algm [4]:pke [5]:machinetime [6]:pspi [7]:-	0xa0
0x26	R/W	CLKEN2 [0]:timer [1]:audio [2]:trng [4]:mcu [5]:lm [7]:eoc	0x30
0x27	R/W	CLKEN3 [0:3]:clk_zb_mst	0x00

Address	R/W	Description	Reset Value
0x28	R/W	CLKSEL0 [3:0]:sclk_div [6:4]:sclk_sel, 000:24M_rc, 001:24M xtal, 010:sclk_div, 011:pll clk, 110:clk_eoc [7]: mspi_clk_sel	0x02
0x29	R/W	CLKSEL1 [7:4]:mspi_div	0x20
0x2a	R/W	I2S_STEP [6:0]:i2s_step [7]: i2s_clk_en	0x01
0x2b	R/W	r_i2s_mod_o	0x02
0x2c	R/W	DMIC_STEP [6:0]:dmic_step [7]: dmic_clk_sel	0x01
0x2d	R/W	DMIC_MOD	0x02
0x2e	R/W	WAKEUPEN [0]: usb_pwdn_i [1]: gpio_wakeup_i [2]: usb resume [3]: standby ex [7:4]:reserved	0x1f
0x2f	R/W	PWDNEN [0]: suspend_en_o [4]: ramcrc_clren_tgl [5]: rst_cnt [7]: stall_en_trg	0x00
0x30	R/W	CLK_DIV [0:2]: clkzb32k_sel [6:4]: r_7816_mod	0x62

Address	R/W	Description	Reset Value
0x33	R/W	SEL [0]: r_dmic_sel_o, 0:dmic_div 1:32k [1]: r_eco_clk_sel_o	0x00

8 Timer

8.1 Timer0 ~ Timer1

The SoC supports two timers: Timer0 ~ Timer1. Timer0 and Timer1 support four modes: Mode 0 (System Clock Mode), Mode 1 (GPIO Trigger Mode), Mode 2 (GPIO Pulse Width Mode) and Mode 3 (Tick Mode), which are selectable via the register TMR_CTRL0 (address 0x140140).

The SoC supports one watchdog timer. The MCU of the SoC embeds a machine timer, using 32K system clock, the address of which is 0xE6000000~0xE60FFFFF.

8.1.1 Mode 0 (System Clock Mode)

In Mode 0, system clock is employed as clock source.

After Timer is enabled, Timer Tick (i.e. counting value) is increased by 1 on each positive edge of system clock from preset initial Tick value. Generally the initial Tick value is set to 0.

Once current Timer Tick value matches the preset Timer Capture (i.e. timing value), an interrupt is generated, Timer stops counting and Timer status is updated.

Steps of setting Timer0 for Mode 0 is taken as an example.

Step 1. Set initial Tick value of Timer0

Set Initial value of Tick via registers TMR_TICK0_0~TMR_TICK0_3, from lowest byte to highest byte respectively. It's recommended to clear initial Timer Tick value to 0.

Step 2. Set Capture value of Timer0

Set registers TMR_CAPT0_0~TMR_CAPT0_3, from lowest byte to highest byte respectively.

Step 3. Set Timer0 to Mode 0 and enable Timer0

Set register TMR_CTRL0 [2:1] to 2b'00 to select Mode 0; Meanwhile set TMR_CTRL0 [0] to 1b'1 to enable Timer0. Timer0 starts counting upward, and Tick value is increased by 1 on each positive edge of system clock until it reaches Timer0 Capture value.

8.1.2 Mode 1 (GPIO Trigger Mode)

In Mode 1, GPIO is employed as clock source. The "**m0**"/"**m1**"/"**m2**" register specifies the GPIO which generates counting signal for Timer0/Timer1/Timer2.

After Timer is enabled, Timer Tick (i.e. counting value) is increased by 1 on each positive/negative (configurable) edge of GPIO from preset initial Tick value. Generally the initial Tick value is set to 0. The "**Polarity**" register specifies the GPIO edge when Timer Tick counting increases.

Once current Timer Tick value matches the preset Timer Capture (i.e. timing value), an interrupt is generated and timer stops counting.

Steps of setting Timer1 for Mode 1 is taken as an example.

Step 1. Set initial Tick value of Timer1

Set Initial value of Tick via registers TMR_TICK1_0~TMR_TICK1_3, from lowest byte to highest byte respectively. It's recommended to clear initial Timer Tick value to 0.

Step 2. Set Capture value of Timer1

Set registers TMR_CAPT1_0~TMR_CAPT1_3, from lowest byte to highest byte respectively.

Step 3. Select GPIO source and edge for Timer1

Select certain GPIO to be the clock source via setting "m1" register.

Select positive edge or negative edge of GPIO input to trigger Timer1 Tick increment via setting "Polarity" register.

Step 4. Set Timer1 to Mode 1 and enable Timer1

Set TMR_CTRL0 [5:4] to 2b'01 to select Mode 1; Meanwhile set TMR_CTRL0 [3] to 1b'1 to enable Timer1. Timer1 starts counting upward, and Timer1 Tick value is increased by 1 on each positive/negative (specified during the 3rd step) edge of GPIO until it reaches Timer1 Capture value.

8.1.3 Mode 2 (GPIO Pulse Width Mode)

In Mode 2, system clock is employed as the unit to measure the width of GPIO pulse. The "**m0**"/"**m1**"/"**m2**" register specifies the GPIO which generates control signal for Timer0/Timer1/Timer2.

After Timer is enabled, Timer Tick is triggered by a positive/negative (configurable) edge of GPIO pulse. Then Timer Tick (i.e. counting value) is increased by 1 on each positive edge of system clock from preset initial Tick value. Generally the initial Tick value is set to 0. The "**Polarity**" register specifies the GPIO edge when Timer Tick starts counting.

While a negative/positive edge of GPIO pulse is detected, an interrupt is generated and timer stops counting.

The GPIO pulse width could be calculated in terms of tick count and period of system clock.

Steps of setting Timer1 for Mode 2 is taken as an example.

Step 1. Set initial Timer1 Tick value

Set Initial value of Tick via registers TMR_TICK1_0~TMR_TICK1_3, from lowest byte to highest byte respectively. It's recommended to clear initial Timer Tick value to 0.

Step 2. Select GPIO source and edge for Timer1

Select certain GPIO to be the clock source via setting "m1" register.

Select positive edge or negative edge of GPIO input to trigger Timer2 counting start via setting "Polarity" register.

Step 3. Set Timer2 to Mode 2 and enable Timer1

Timer1 Tick is triggered by a positive/negative (specified during the 2nd step) edge of GPIO pulse. Timer1 starts counting upward and Timer1 Tick value is increased by 1 on each positive edge of system clock.

While a negative/positive edge of GPIO pulse is detected, an interrupt is generated and Timer1 tick stops.

Step 4. Read current Timer1 Tick value to calculate GPIO pulse width

Read current Timer1 Tick value.

Then GPIO pulse width is calculated as follows:

GPIO Pulse Width = System Clock Period *(Current Timer1 Tick - Initial Timer1 Tick)

For initial Timer1 Tick value is set to the recommended value of 0, then:

GPIO Pulse Width = System Clock Period * Current Timer1 Tick

8.1.4 Mode 3 (Tick Mode)

In Mode 3, system clock is employed.

After Timer is enabled, Timer Tick starts counting upward, and Timer Tick value is increased by 1 on each positive edge of system clock.

This mode could be used as time indicator. There will be no interrupt generated. Timer Tick keeps rolling from 0 to 0xffffffff. When Timer tick overflows, it returns to 0 and starts counting upward again.

Steps of setting Timer0 for Mode 3 is taken as an example.

Step 1. Set initial Tick value of Timer0

Set Initial value of Tick via TMR_TICK0_1 ~TMR_TICK0_3, from lowest byte to highest byte respectively.

Step 2. Set Timer0 to Mode 3 and enable Timer0

Set TMR_CTRL0 [2:1] to 2b'11 to select Mode 3, meanwhile set address TMR_CTRL0 [0] to 1b'1 to enable Timer0. Timer0 Tick starts to roll.

Step 3. Read current Timer0 Tick value

Current Timer0 Tick value can be read fromTMR_TICK0_1 ~TMR_TICK0_3.

8.1.5 Watchdog

Programmable watchdog could reset chip from unexpected hang up or malfunction.

Watchdog Capture has 24bits, which consists of WT_TARGET_1~WT_TARGET_3 as byte 1 ~byte 3. Chip will be reset when TMR_CTRL3[3] is set to 1.

Step 1. Set WT_TARGET_1~WT_TARGET_3

Set registers WT_TARGET_1~WT_TARGET_3, from lowest byte to highest byte respectively.

Step 2. Enable Watchdog

Set TMR_CTRL2 [7] to 1b'1 to enable Watchdog.

During normal working condition, TMR_CTRL3[3] need write 1 to clean the watchdog before the watchdog hits WT_TARGET3-1, or it will reboot the whole chip, and the TMR_CTRL3[2] will be assert to 1, this bit will be clean when write 1.

8.1.6 Register Table

Timer related register are listed in table below. The base address for the following registers is 0x 140140.

Table 8-1 Register Configuration for Timer 0 ~ Timer 1

Offset	R/W	Description	Reset Value
0x00	R/W	TMR_CTRL0 [1:0] 0:tmr0m0,using pclk 1:tmr0m1, count gpio2risc0 posedge 2:tmr0m2 count gpio2risc0 high width 3:tmr0m3,tick [2] Timer0 enable [3] Timer0 nowrap [5:4] 0:tmr1m0,using pclk 1:tmr1m1, count gpio2risc1 posedge 2:tmr1m2 count gpio2risc1 high width 3:tmr1m3,tick [6] Timer2 enable [7] Timer1 nowrap	0x0
0x02	R/W	TMR_CTRL2 [7] watchdog_en	0x0
0x03	R/W	TMR_CTRL3 [0] tmr0_o=tmr0 [1] tmr1_o=tmr1 [2] hit watchdog target [3] clear wd_cnt [7] software_irq	0x0
0x04	R/W	TMR_CAPT0_0 capt0[7:0] Byte 0 of timer0 capture	0x0
0x05	R/W	TMR_CAPT0_1 capt0[15:8] Byte 1 of timer0 capture	0x0
0x06	R/W	TMR_CAPT0_2 capt0[23:16] Byte 2 of timer0 capture	0x0
0x07	R/W	TMR_CAPT0_3 capt0[31:24] Byte 3 of timer0 capture	0x0
0x08	R/W	TMR_CAPT1_0 capt1[7:0] Byte 0 of timer1 capture	0x0

Offset	R/W	Description	Reset Value
0x09	R/W	TMR_CAPT1_1 capt1[15:8] Byte 1 of timer1 capture	0x0
0x0a	R/W	TMR_CAPT1_2 capt1[23:16] Byte 2 of timer1 capture	0x0
0x0b	R/W	TMR_CAPT1_3 capt1[31:24] Byte 3 of timer1 capture	0x0
0x0d	R/W	WT_TARGET_1 watchdog_target2[15:8] Byte 1 of watchdog target value	0x0
0x0e	R/W	WT_TARGET_2 watchdog_target2[23:16] Byte 2 of watchdog target value	0x0
0x0f	R/W	WT_TARGET_3 watchdog_target2[31:24] Byte 3 of watchdog target value	0x0
0x10	R	TMR_TICK0_0 ticko[7:0] Byte 0 of timer0 ticker	0x0
0x11	R	TMR_TICK0_1 ticko[15:8] Byte 1 of timer0 ticker	0x0
0x12	R	TMR_TICK0_2 ticko[23:16] Byte 2 of timer0 ticker	0x0
0x13	R	TMR_TICK0_3 ticko[31:24] Byte 3 of timer0 ticker	0x0
0x14	R	TMR_TICK1_0 tick1[7:0] Byte 0 of timer1 ticker	0x0
0x15	R	TMR_TICK1_1 tick1[15:8] Byte 1 of timer1 ticker	0x0
0x16	R	TMR_TICK1_2 tick1[23:16] Byte 2 of timer1 ticker	0x0
0x17	R	TMR_TICK1_3 tick1[31:24] Byte 3 of timer1 ticker	0x0

8.2 32K LTimer

The SoC also supports a low frequency (32kHz) LTIMER in suspend mode or deep sleep mode. This timer can be used as one kind of wakeup source.

8.3 System Timer

The SoC also supports a System Timer, the clock frequency for System Timer is fixed as 16MHz irrespective of system clock.

In suspend mode, both System Timer and Timer0 ~ Timer1 stop counting, and 32K Timer starts counting. When the chip restores to active mode, Timer0 ~ Timer1 will continue counting from the number when they stops; In contrast, System Timer will continue counting from an adjusted number which is a sum of the number when it stops and an offset calculated from the counting value of 32K Timer during suspend mode.

System timer related registers are listed in table below. The base address for the registers is 0x140200.

Table 8-2 Register Table for System Timer

Offset	R/W	Description	Reset Value
0x00	R/W	SYS_TIMER0	0x0
0x01	R/W	SYS_TIMER1	0x0
0x02	R/W	SYS_TIMER2	0x0
0x03	R/W	SYS_TIMER3	0x0
0x0a	R/W	SYS_TIMER_CTRL	0x0
0x0b	R/W	SYS_TIMER_ST	0x0

9 Interrupt System

9.1 Interrupt Structure

The SoC provides three interrupt inputs: Timer interrupt, software interrupt, and external interrupt. External interrupts are arbitrated and distributed by a platform-level interrupt controller (PLIC) to the processor core. Each external interrupt source can be assigned its own priority, and the RISC-V processor core could select which external interrupt sources it would handle. PLIC routes the highest priority interrupt source to the target processor.

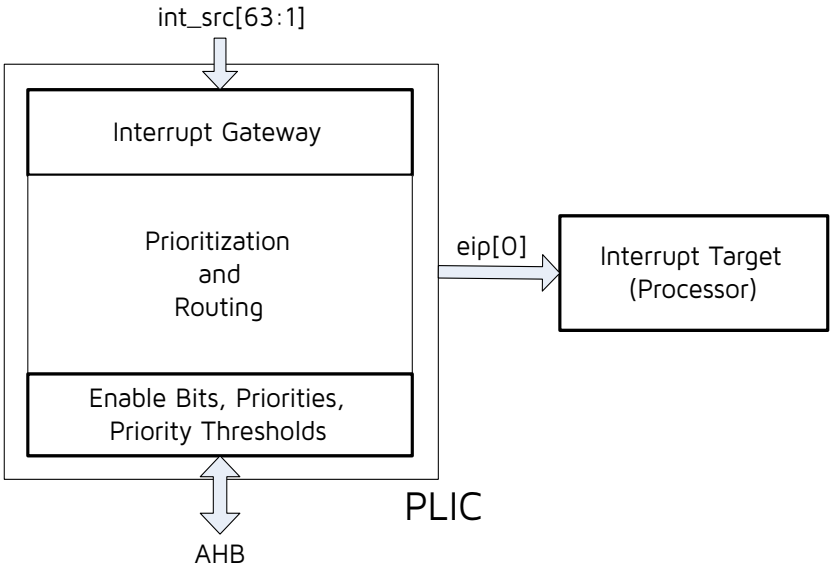
The Machine Interrupt Pending Control and State Register (MIP CSR) contains pending bits of these three interrupts, and the Machine Interrupt Enable Control and State Register (MIE CSR) contains enable bits of these interrupts. The processor can selectively enable interrupts by manipulating the MIE CSR, or globally disable interrupts by clearing the MIE bit.

Platform-Level Interrupt Controller (PLIC) prioritizes and distributes global interrupts. It is compatible with RISC-V PLIC with the following features:

- Software-programmable interrupt generation
- Preemptive priority interrupt extension
- Vectored interrupt extension

The following figure shows the block diagram of PLIC. Interrupt source (e.g., devices) send interrupt requests to PLIC through the int_src signals and they are converted to interrupt requests by the interrupt gateway. Interrupt requests are prioritized and routed to interrupt targets (e.g., RISC-V processor core) according to interrupt settings. Interrupt settings include enable bits, priorities, and priority thresholds, and these settings are programmable through the bus interface.

Figure 9-1 Block Diagram of PLIC



9.2 External Interrupt Sources

There are 63 external interrupt sources, listed in table below.

Table 9-1 Interrupt Sources

No.	Type	Interrupt Source
1	syn	irq_stimer_lev
2	syn	alg_irq
3	syn	tmr1_irq
4	syn	tmr0_irq
5	syn	irq_dma
6	syn	irq_bmc
7	syn	irq_udc_[0]
8	syn	irq_udc[1]
9	syn	irq_udc[2]
10	syn	irq_udc[3]
11	syn	irq_udc[4]
12	syn	irq_zb_dm
13	syn	irq_zb_ble
14	syn	irq_zb_bt
15	syn	irq_zb_rt
16	syn	irq_pwm
17	syn	irq_pke
18	syn	irq_uart1
19	syn	irq_uart
20	syn	irq_dfifo
21	syn	irq_i2c
22	syn	irq_spi_ahb
23	syn	irq_spi_apb
24	syn	usb_pwdn
25	syn	irq_gpio

No.	Type	Interrupt Source
26	syn	gpio2risc[0]
27	syn	gpio2risc[1]
28	syn	soft_irq
29	syn	-
30	syn	-
31	syn	-
32	syn	-
33	syn	-
34	syn	usb_250us
35	syn	usb_reset
36	syn	-
37	syn	-
38	syn	-
39	syn	-
40	syn	-
41	syn	-
42	syn	-
43	syn	-
44	syn	-
45	syn	-
46	syn	-
47	syn	-
48	syn	-
49	syn	-
50	syn	-
51	syn	-
52	syn	-
53	syn	-

No.	Type	Interrupt Source
54	syn	-
55	syn	-
56	syn	-
57	syn	-
58	syn	-
59	syn	-
60	syn	-
61	asyn	-
62	asyn	pm_irq_tm
63	asyn	emq_irq

9.3 Register Description

PLIC related register are listed in table below. The base address for the following registers is 0xE4000000.

Table 9-2 Register Configuration for PLIC

Offset	R/W	Description	Reset Value
0x00	R/W	Feature Enable Register [0]: PREEMPT, Preemptive priority interrupt enable [1]: VECTORED, Vector mode enable	0x00
0x04*n	R/W	Interrupt Source n Priority [31:0]: Interrupt source n priority. 0: Never interrupt, 1-3: Interrupt source priority. The larger the value, the higher the priority.	0x01

Offset	R/W	Description	Reset Value
0x1000	R/W	Interrupt source 1~31 Pending. The register provide the interrupt pending status of interrupt sources 1~31, and a way for software to trigger an interrupt without relying on external devices. Every interrupt source occupies 1 bit. [31:1]: interrupt pending status of interrupt sources 1~31.	0x00
0x1004	R/W	Interrupt source 32~63 Pending. The register provide the interrupt pending status of interrupt sources 32~63, and a way for software to trigger an interrupt without relying on external devices. Every interrupt source occupies 1 bit. [31:0]: interrupt pending status of interrupt sources 32~63.	0x00
0x1080	R/O	Interrupt Trigger Type. These registers are read-only and indicate the configured interrupt trigger type of interrupt sources 1~31. Every interrupt source occupies 1 bit. [31:1]: Edge-triggered interrupt, 1: Edge-triggered interrupt, 0: Level-triggered interrupt	0x00
0x1084	R/O	Interrupt Trigger Type. These registers are read-only and indicate the configured interrupt trigger type of interrupt sources 32~63. Every interrupt source occupies 1 bit. [31:0]: Edge-triggered interrupt, 1: Edge-triggered interrupt, 0: Level-triggered interrupt	0x00

Offset	R/W	Description	Reset Value
0x1100	R/O	Number of Interrupt and Target Configuration Register. [15:0]: The number of supported interrupt sources [31:16]: The number of supported targets	0x0002003F
0x1104	R/O	Version & Maximum Priority Configuration Register. [15:0]: The version of the PLIC design [31:16]: The maximum priority supported	0x00030001
0x2000	R/W	Interrupt Enable Bits for interrupt sources 1~31. Every interrupt source occupies 1 bit. [31:1]: Interrupt Enable Bits for interrupt sources 1~31	0x00
0x2004	R/W	Interrupt Enable Bits for interrupt sources 32~63. Every interrupt source occupies 1 bit. [31:0]: Interrupt Enable Bits for interrupt sources 32~63.	0x00
0x200000	R/W	Priority Threshold. [31:0]: THRESHOLD, Interrupt priority threshold	0x0
0x200004	R/W	Claim and Complete Register. [9:0]: INTERRUPT_ID, On reads, indicating the interrupt source that has being claimed. On writes, indicating the interrupt source that has been handled (completed).	0x0

Offset	R/W	Description	Reset Value
0x200400	R/W	The register is read/writable registers for accessing the preempted priority stack. The purpose of the register is for saving and restoring priorities of the nested/ preempted interrupts. [3:0]: Each bit indicates if the corresponding priority level has been preempted by a higher-priority interrupt.	0x00

10 Interface

10.1 GPIO

The SoC supports up to 26 GPIOs. All digital IOs can be used as general purpose IOs.

10.1.1 Basic Configuration

All GPIOs can be configured with related registers, as described as following.

Table 10-1 GPIO Setting

Pad	Input	IE	OEN	Output/PE	Polarity	DS	Act as GPIO
PA[7]	0x140300 [7]	0x140301 [7]	0x140302 [7]	0x140303 [7]	0x140304 [7]	0x140305 [7]	0x140306 [7]
PB[0]	0x140308 [0]	0x140309 [0]	0x14030a [0]	0x14030b [0]	0x14030c [0]	0x14030d [0]	0x14030e [0]
PB[1]	0x140308 [1]	0x140301 [1]	0x14030a [1]	0x14030b [1]	0x14030c [1]	0x14030d [1]	0x14030e [1]
PB[2]	0x140308 [2]	0x140301 [2]	0x14030a [2]	0x14030b [2]	0x14030c [2]	0x14030d [2]	0x14030e [2]
PB[3]	0x140308 [3]	0x140301 [3]	0x14030a [3]	0x14030b [3]	0x14030c [3]	0x14030d [3]	0x14030e [3]
PB[4]	0x140308 [4]	0x140301 [4]	0x14030a [4]	0x14030b [4]	0x14030c [4]	0x14030d [4]	0x14030e [4]
PB[5]	0x140308 [5]	0x140301 [5]	0x14030a [5]	0x14030b [5]	0x14030c [5]	0x14030d [5]	0x14030e [5]
PB[6]	0x140308 [6]	0x140301 [6]	0x14030a [6]	0x14030b [6]	0x14030c [6]	0x14030d [6]	0x14030e [6]
PC[0]	0x140310 [0]	c1[0]	0x140312 [0]	0x140313 [0]	0x140314 [0]	c3[0]	0x140316 [0]
PC[1]	0x140310 [1]	c1[1]	0x140312 [1]	0x140313 [1]	0x140314 [1]	c3[1]	0x140316 [1]
PC[2]	0x140310 [2]	c1[2]	0x140312 [2]	0x140313 [2]	0x140314 [2]	c3[2]	0x140316 [2]
PC[3]	0x140310 [3]	c1[3]	0x140312 [3]	0x140313 [3]	0x140314 [3]	c3[3]	0x140316 [3]

Pad	Input	IE	OEN	Output/PE	Polarity	DS	Act as GPIO
PD[0]	0x140318 [0]	c4[0]	0x14031a [0]	0x14031b [0]	0x14031c [0]	c6[0]	0x14031e [0]
PD[1]	0x140318 [1]	c4[1]	0x14031a [1]	0x14031b [1]	0x14031c [1]	c6[1]	0x14031e [1]
PD[2]	0x140318 [2]	c4[2]	0x14031a [2]	0x14031b [2]	0x14031c [2]	c6[2]	0x14031e [2]
PD[3]	0x140318 [3]	c4[3]	0x14031a [3]	0x14031b [3]	0x14031c [3]	c6[3]	0x14031e [3]
PE[0]	0x140320 [0]	0x140321 [0]	0x140322 [0]	0x140323 [0]	0x140324 [0]	0x140325 [0]	0x140326 [0]
PE[1]	0x140320 [1]	0x140321 [1]	0x140322 [1]	0x140323 [1]	0x140324 [1]	0x140325 [1]	0x140326 [1]
PE[2]	0x140320 [2]	0x140321 [2]	0x140322 [2]	0x140323 [2]	0x140324 [2]	0x140325 [2]	0x140326 [2]
PE[3]	0x140320 [3]	0x140321 [3]	0x140322 [3]	0x140323 [3]	0x140324 [3]	0x140325 [3]	0x140326 [3]
PE[4]	0x140320 [4]	0x140321 [4]	0x140322 [4]	0x140323 [4]	0x140324 [4]	0x140325 [4]	0x140326 [4]
PE[5]	0x140320 [5]	0x140321 [5]	0x140322 [5]	0x140323 [5]	0x140324 [5]	0x140325 [5]	0x140326 [5]
PE[6]	0x140320 [6]	0x140321 [6]	0x140322 [6]	0x140323 [6]	0x140324 [6]	0x140325 [6]	0x140326 [6]
PE[7]	0x140320 [7]	0x140321 [7]	0x140322 [7]	0x140323 [7]	0x140324 [7]	0x140325 [7]	0x140326 [7]

NOTE:

- *IE: Input enable, high active. 1: enable input, 0: disable input.*
- *OEN: Output enable, low active. 0: enable output, 1: disable output.*
- *Output: configure GPO output.*
- *Input: read GPI input.*
- *DS: Drive strength. 1: maximum DS level (default), 0: minimal DS level.*
- *Act as GPIO: enable (1) or disable (0) GPIO function.*
- *Polarity: see section below*
- *c1, c3, c4, c6 are analog registers*

Table 10-2 GPIO Function Mux Configuration Registers

Address	R/W	Description	Default Value
0x140330	RW	[1:0]: function control bits of PA[0] [3:2]: function control bits of PA[1] [5:4]: function control bits of PA[2] [7:6]: function control bits of PA[3]	0x00
0x140331	RW	[1:0]: function control bits of PA[4] [3:2]: function control bits of PA[5] [5:4]: function control bits of PA[6] [7:6]: function control bits of PA[7]	0x00
0x140332	RW	[1:0]: function control bits of PB[0] [3:2]: function control bits of PB[1] [5:4]: function control bits of PB[2] [7:6]: function control bits of PB[3]	0x00
0x140333	RW	[1:0]: function control bits of PB[4] [3:2]: function control bits of PB[5] [5:4]: function control bits of PB[6] [7:6]: function control bits of PB[7]	0x00
0x140334	RW	[1:0]: function control bits of PC[0] [3:2]: function control bits of PC[1] [5:4]: function control bits of PC[2] [7:6]: function control bits of PC[3]	0x00

Address	R/W	Description	Default Value
0x140335	RW	[1:0]: function control bits of PC[4] [3:2]: function control bits of PC[5] [5:4]: function control bits of PC[6] [7:6]: function control bits of PC[7]	0x00
0x140336	RW	[1:0]: function control bits of PD[0] [3:2]: function control bits of PD[1] [5:4]: function control bits of PD[2] [7:6]: function control bits of PD[3]	0x00
0x140337	RW	[1:0]: function control bits of PD[4] [3:2]: function control bits of PD[5] [5:4]: function control bits of PD[6] [7:6]: function control bits of PD[7]	0x00
0x140350	RW	[1:0]: function control bits of PE[0] [3:2]: function control bits of PE[1] [5:4]: function control bits of PE[2] [7:6]: function control bits of PE[3]	0x00
0x140351	RW	[1:0]: function control bits of PE[4] [3:2]: function control bits of PE[5] [5:4]: function control bits of PE[6] [7:6]: function control bits of PE[7]	0x00
0x140356	RW	[1:0]: function control bits of PE[0] [3:2]: function control bits of PE[1] [5:4]: function control bits of PE[2] [7:6]: function control bits of PE[3]	0x00
0x140356	RW	[1:0]: function control bits of PE[4] [3:2]: function control bits of PE[5]	0x00

10.1.2 Connection Relationship between GPIO and Related Modules

GPIO can be used to generate GPIO interrupt signal for interrupt system, counting or control signal for Timer/ Counter module, or GPIO2RISC interrupt signal for interrupt system.

For the "Exclusive Or (XOR)" operation result for input signal from any GPIO pin and respective "Polarity" value, on one hand, it takes "And" operation with "irq" and generates GPIO interrupt request signal; on the other hand, it takes "And" operation with "m0/m1", and generates counting signal in Mode 1 or control signal in Mode 2 for Timer0/Timer1, or generates GPIO2RISC[0]/GPIO2RISC[1] interrupt request signal.

GPIO interrupt request signal = | ((input ^ polarity) & irq);

Counting (Mode 1) or control (Mode 2) signal for Timer0 = | ((input ^ polarity) & m0);

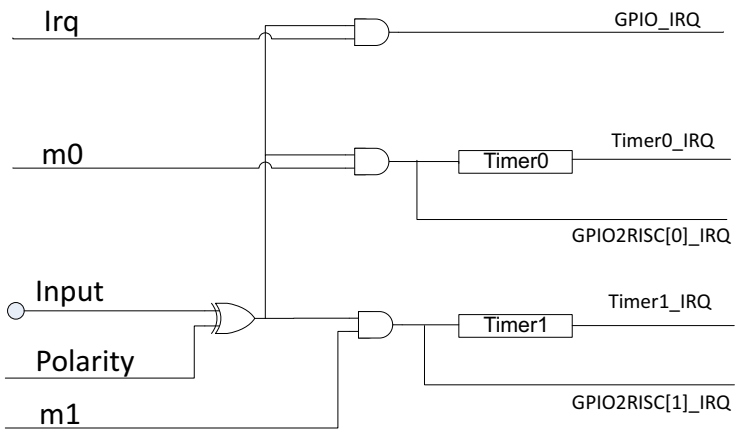
Counting (Mode 1) or control (Mode 2) signal for Timer1 = | ((input ^ polarity) & m1);

GPIO2RISC[0] interrupt request signal = | ((input ^ polarity) & m0);

GPIO2RISC[1] interrupt request signal = | ((input ^ polarity) & m1).

As is shown in figure below.

Figure 10-1 Logic Relationship between GPIO and Related Modules



10.1.3 Drive Strength

The registers in the "DS" column are used to configure the corresponding pin's driving strength: "1" indicates maximum drive level, while "0" indicates minimal drive level.

The "DS" configuration will take effect when the pin is used as output. It's set as the strongest driving level by default. In actual applications, driving strength can be decreased to lower level if necessary.

10.1.4 Polarity

By configuring "Polarity" registers, user can determine GPIO edges in Timer modes. In Timer Mode 1, it determines GPIO edge when Timer Tick counting increases. In Timer Mode 2, it determines GPIO edge when Timer Tick starts counting.

Users can read addresses to see which GPIO asserts counting signals (Mode 1)/control signal (Mode 2) for Timers.

10.1.5 GPIO IRQ Signal

Select GPIO interrupt trigger edge (positive edge or negative edge) via configuring "Polarity", and set corresponding GPIO interrupt enabling bit "Irq".

10.1.6 GPIO2RISC IRQ Signal

Select GPIO2RISC interrupt trigger edge (positive edge or negative edge) via configuring "Polarity", and set corresponding GPIO enabling bit "m0"/"m1", then enable GPIO2RISC[0]/GPIO2RISC[1] interrupt.

Table 10-3 GPIO IRQ Table

Pad	Input	IRQ	m0	m1	Polarity
PA[7]	0x140300 [7]	0x140307[7]	0x140338[7]	0x140340[7]	0x140304 [7]
PB[0]	0x140308 [0]	0x14030f[0]	0x140339[0]	0x140341[0]	0x14030c [0]
PB[1]	0x140308 [1]	0x14030f[1]	0x140339[1]	0x140341[1]	0x14030c [1]
PB[2]	0x140308 [2]	0x14030f[2]	0x140339[2]	0x140341[2]	0x14030c [2]
PB[3]	0x140308 [3]	0x14030f[3]	0x140339[3]	0x140341[3]	0x14030c [3]
PB[4]	0x140308 [4]	0x14030f[4]	0x140339[4]	0x140341[4]	0x14030c [4]
PB[5]	0x140308 [5]	0x14030f[5]	0x140339[5]	0x140341[5]	0x14030c [5]
PB[6]	0x140308 [6]	0x14030f[6]	0x140339[6]	0x140341[6]	0x14030c [6]
PC[0]	0x140310 [0]	0x140317[0]	0x14033a[0]	0x140342[0]	0x140314 [0]
PC[1]	0x140310 [1]	0x140317[1]	0x14033a[1]	0x140342[1]	0x140314 [1]
PC[2]	0x140310 [2]	0x140317[2]	0x14033a[2]	0x140342[2]	0x140314 [2]
PC[3]	0x140310 [3]	0x140317[3]	0x14033a[3]	0x140342[3]	0x140314 [3]
PD[0]	0x140318 [0]	0x14031f[0]	0x14033b[0]	0x140343[0]	0x14031c [0]
PD[1]	0x140318 [1]	0x14031f[1]	0x14033b[1]	0x140343[1]	0x14031c [1]
PD[2]	0x140318 [2]	0x14031f[2]	0x14033b[2]	0x140343[2]	0x14031c [2]
PD[3]	0x140318 [3]	0x14031f[3]	0x14033b[3]	0x140343[3]	0x14031c [3]
PE[0]	0x140320 [0]	0x140327[0]	0x14033c[0]	0x140344[0]	0x140324 [0]
PE[1]	0x140320 [1]	0x140327[1]	0x14033c[1]	0x140344[1]	0x140324 [1]
PE[2]	0x140320 [2]	0x140327[2]	0x14033c[2]	0x140344[2]	0x140324 [2]
PE[3]	0x140320 [3]	0x140327[3]	0x14033c[3]	0x140344[3]	0x140324 [3]
PE[4]	0x140320 [4]	0x140327[4]	0x14033c[4]	0x140344[4]	0x140324 [4]
PE[5]	0x140320 [5]	0x140327[5]	0x14033c[5]	0x140344[5]	0x140324 [5]
PE[6]	0x140320 [6]	0x140327[6]	0x14033c[6]	0x140344[6]	0x140324 [6]
PE[7]	0x140320 [7]	0x140327[7]	0x14033c[7]	0x140344[7]	0x140324 [7]

10.1.7 Pull-up/Pull-down Resistors

All GPIOs support configurable pull-up resistor of rank x1 and x100 or pull-down resistor of rank x10 which are all disabled by default. Analog registers afe_0x0e<7:0> ~ afe_0x17<7:0> serve to control the pull-up/pull-down resistor for each GPIO, as shown in table below.

Table 10-4 Analog Registers for Pull-up/Pull-down Resistor Control

Address	R/W	Description	Default Value
0x0e	R/W	GPIO_A<3:0> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up	00000000
0x0f	R/W	GPIO_A<7:4> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up	00000000
0x10	R/W	GPIO_B<3:0> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up	00000000
0x11	R/W	GPIO_B<7:4> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up	00000000
0x12	R/W	GPIO_C<3:0> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up	00000000

Address	R/W	Description	Default Value
0x13	R/W	GPIO_C<7:4> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up	00000000
0x14	R/W	GPIO_D<3:0> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up	00000000
0x15	R/W	GPIO_D<7:4> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up	00000000
0x16	R/W	GPIO_E<3:0> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up	00000000
0x17	R/W	GPIO_E<7:4> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up	00000000

10.2 Swire

The SoC supports Single Wire Slave interface. SWM (Single Wire Master) and SWS (Single Wire Slave) represent the master and slave device of the single wire communication system developed by Telink. The maximum data rate can be up to 2Mbps.

SWS usage is not supported in power-saving mode (deep sleep or suspend).

SWS related registers are listed as following, the base address of the following registers is 0x80100c00.

Table 10-5 SWIRE Related Registers

Offset	R/W	Description	Default Value
0x00	R	swire_data	0x00
0x01	RW	SWIRE_CTL [0]:swire_wr [1]:swire_rd [2]:swire_cmd [3]:swire_err_flag [4]:swire_eop [6]:swire_usb_det [7]:swire_usb_en	0x80
0x02	RW	SWIRE_CTL2 [6:0]: swire_clk_div	0x05
0x03	RW	[1:0]: function control bits of PB[4] [3:2]: function control bits of PB[5] [5:4]: function control bits of PB[6] [7:6]: function control bits of PB[7]	0x00

10.3 I2C

The SoC embeds I2C hardware module, which could act as Master mode or Slave mode. I2C is a popular inter-IC interface requiring only 2 bus lines, a serial data line (SDA) and a serial clock line (SCL).

10.3.1 Communication Protocol

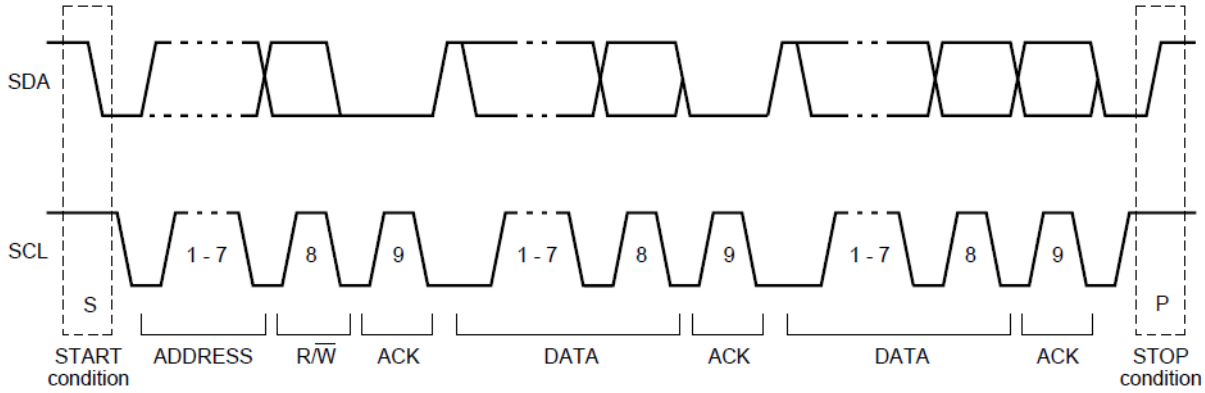
Telink I2C module supports standard mode (100kbps) and Fast-mode (400kbps) with restriction that system clock must be by at least 10x of data rate.

Two wires, SDA and SCL (SCK) carry information between Master device and Slave device connected to the bus. Each device is recognized by unique address (ID). Master device is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. Slave device is the device addressed by a Master.

Both SDA and SCL are bidirectional lines connected to a positive supply voltage via a pull-up resister. It's recommended to use external 3.3kohm pull-up resistor. For standard mode, the internal pull-up resistor of rank x1 can be used instead of the external 3.3 kohm pull-up.

When the bus is free, both lines are HIGH. It's noted that data in SDA line must keep stable when clock signal in SCL line is at high level, and level state in SDA line is only allowed to change when clock signal in SCL line is at low level.

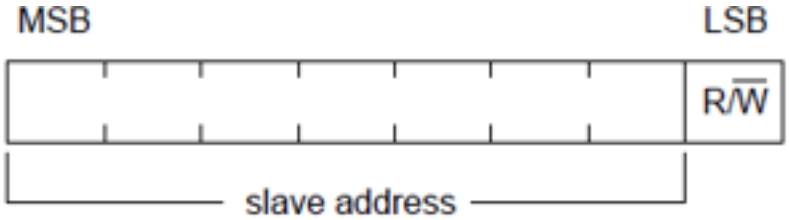
Figure 10-2 I2C Timing



10.3.2 I2C Slave Mode

I2C module acts as Slave mode by default. I2C slave address can be configured via register I2C_ID (address 0x01) [7:1], as shown below.

Figure 10-3 Byte Consisted of Slave Address and R/W Flag Bit



I2C slave mode supports two sub modes including Direct Memory Access (DMA) mode and No direct Memory Access (NDMA).

In I2C Slave mode, Master could initiate transaction anytime. I2C slave module will reply with ACK automatically. To monitor the start of I2C transaction, user could set interrupt from GPIO for SCA or SCL.

Read and write format of Slave modes are shown as below.

DMA and NDMA access buffer through dma and ahb, respectively.

Figure 10-4 Read Format in Slave Mode

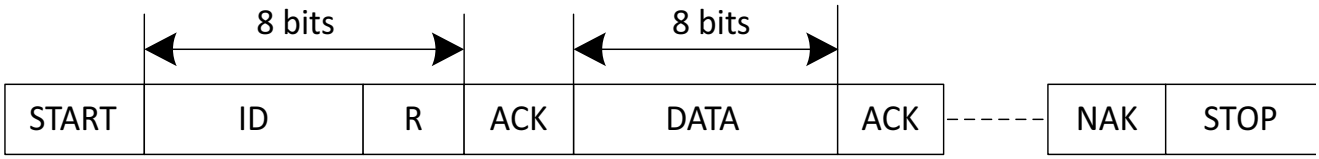
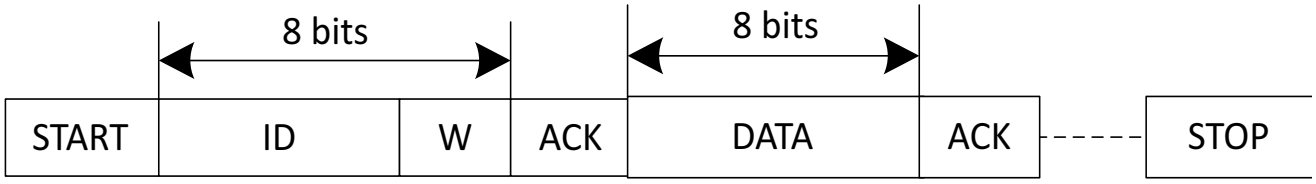


Figure 10-5 Write Format in Slave Mode



10.3.3 I2C Master Mode

Register I2CSCT0[1] should be set to 1b'1 to enable I2C master mode.

Register I2CSP sets I2C Master clock: F_{I2C} = (System Clock / (4 *clock speed configured in register I2CSP).

A complete I2C protocol contains START, Slave Address, R/W bit, data, ACK and STOP. Slave address could be configured via I2C_ID [7:1].

I2C Master could send START, Slave Address, R/W bit, data and STOP cycle by configuring SLAVE_STRECH_EN. I2C master will send enabled cycles in the correct sequence.

Register I2CMST serves to indicate whether Master/Master packet is busy, as well as Master received status. Bit[0] will be set to 1 when one byte is being sent, and the bit can be automatically cleared after a start signal/ address byte/acknowledge signal/data /stop signal is sent. Bit[1] is set to 1 when the start signal is sent, and the bit will be automatically cleared after the stop signal is sent. Bit[2] indicates whether to succeed in sending acknowledgement signal.

10.3.3.1 I2C Master Write Transfer in NDMA Mode

I2C Master has 8-byte buffer for write data, which are I2C_data_buf0, I2C_data_buf1, I2C_data_buf2 and I 2C_data_buf3. Write transfer will be completed by I2C master module.

For example, to implement an I2C write transfer with 4-byte data, which contains START, Slave Address, Write bit, ACK from Slave, 1st byte, ACK from slave, 2nd byte, ACK from slave, 3rd byte, ACK from slave,4th byte, ACK from slave and STOP, user needs to configure I2C slave address to I2C_ID[7:1], 1st byte address to buff. To start I2C write transfer, I2CSCT1 is configured to 0x13 (0001 0011). I2C Master will launch START, Slave address. 1 word data to buff; I2CSCT1 is configured to 0x24 (0000 0024).Write bit, load ACK to I2CMST[2], send buff data, load ACK to I2CMST[2] and then STOP sequentially.

I2c supports a single write of 255 bytes.

10.3.3.2 I2C Master Read Transfer in NDMA Mode

I2C Master has 8 byte buffer for read data, which is fifo (0x08). Read transfer will be completed by I2C Master.

For example, to implement an I2C read transfer with 4 byte data, which contains START, Slave Address, Read bit, ACK from Slave, 4 byte from Slave, ACK by master and STOP, user needs to configure I2C slave address to I2C_ID[7:1]. To start I2C read transfer, I2CSCT1 is configured to 0x7b (0111 1011). I2C Master will launch START, Slave address, Read bit, load ACK to I2CMST[2], load data to I2CDR, reply ACK and then STOP sequentially.

I2C supports a single read of 255 bytes.

10.3.3.3 I2C Master Writer Transfer in NDMA Mode

The data to be sent is put into SRAM, set tx dma config, user needs to configure I2C slave address to I2C_ID [7:1], 1st byte address to buff. To start I2C write transfer, I2CSCT1 is configured to 0x13 (0001 0011). I2C Master will launch START, Slave address. 1 word data to buff, I2CSCT1 is configured to 0x24 (0000 0024). Write bit, load ACK to I2CMST[2], send buff data, load ACK to I2CMST[2] and then STOP sequentially.

I2c supports a single write of 255bytes.

10.3.3.4 I2C Master Read Transfer in NDMA Mode

For example, set rx dma config, user needs to configure I2C slave address to I2C_ID[7:1]. To start I2C read transfer, I2CSCT1 is configured to 0x7b (0111 1011). I2C Master will launch START, Slave address, Read bit, load ACK to I2CMST[2], load data to I2CDR, reply ACK and then STOP sequentially.

I2c supports a single write of 255 bytes.

10.3.4 Register Description

The I2C related registers are listed as following, the base address of the following registers is 0x80140280.

Table 10-6 I2C Related Registers

Address	R/W	Description	Default Value
0x01	RW	I2CSP I2C master clock speed	1f
0x01	RW	I2CID I2C ID	5c
0x02	R	I2CMST [0]: master busy [1]: master packet busy [2]: master received status: 1 for nak; 0 for ack [5:3]: master state of the base [7:5]: slave state of the base	30

Address	R/W	Description	Default Value
0x03	RW	I2CSCT [0]: I2C master enable [1]: clk stretch enable, suspend transmission by pulling SCL down to low level, and continue transmission after SCL is released to high level. [2]: rx interrupt enable [3]: tx interrupt enable [4]: mask_txdone [5]: mask_rxdone [6]:rnack_en, The last byte data read is automatically returned to nack [7]:Delay sda and oen before ack (ID, ADDRESS, DATAW)	00
0x04	RW	I2CSCT [0]: launch ID cycle [1]: launch address cycle [2]: launch data write cycle [3]: launch data read cycle [4]: launch start cycle [5]: launch stop cycle [6]: enable read ID [7]: enable ACK in read command	00
0x05	RW	I2CTRIG [3:0]: rx_irq_trig level [7:4]: tx_irq_trig level	44
0x06	RW	I2CLEN Config buffer send and receive byte number: default 1 byte	01

Address	R/W	Description	Default Value
0x07	RW	SLAVE_STRECH_EN [0]: slave auto stretch clk eanble [1]: slave manul stretch clk [2]: (w) clear slave stretch [6]: standard mode and system clock 48M,maintain ss_scl setup time Max [7]: fast mode:ss_scl setup time small	00
0x08	RW	I2C_data_buf0 write/read buffer[7:0]	-
0x09	RW	I2C_data_buf1 Write/read buffer[15:8]	-
0x0a	RW	I2C_data_buf2 Write/read buffer[23:16]	-
0x0b	RW	I2C_data_buf3 Write/read buffer[31:24]	-
0x0c	R	Buf_cnt [3:0]: rx_buf_cnt [7:4]: tx_buf_cnt	-
0x0d	R	I2c_sts [2:0]: rbcnt [3]: i2c_irq [6:4]: wbcnt	-
0x0e	R	irq_sts [0]: txdone [1]: tx_buf_irq [2]: rxdone [3]: rx_buf_irq [4]: tx_en(W)	-
0x0f	R	rx_fifo_len rx fifo receive byte number	-

10.4 I2S

The I2S module supports I2S, LJ, RJ, and DSP input formats; supports 16 bits, 20 bits, 24 bits and 32 bits input data bandwidth.

10.5 HSPI

10.5.1 Diagram

HSPI diagram is shown as following:

As shown in the diagram, AHB_BUS is used to configure the direct address mapping of registers and XIP. DMA_BUS is the bus between the SPI module and the DMA module. SPI_BUS is the SPI interface connected to the pad.

u_spi_regif is to parse the AHB protocol and send it to the u_spi_reg module for register configuration.

The u_spi_ctrl module selects the state and mode according to the value of the register configuration, and controls the format of the transmitted data.

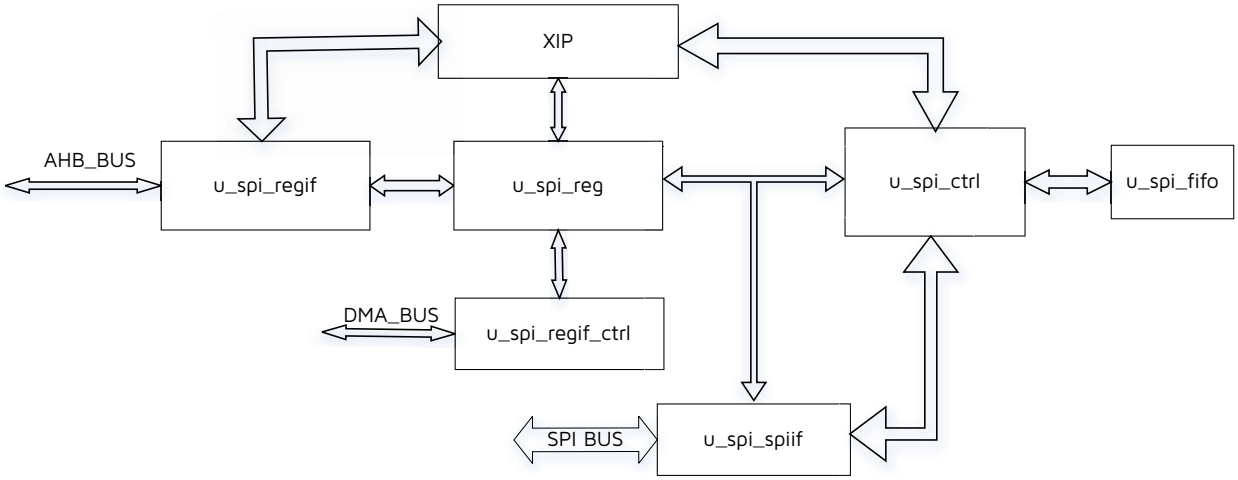
The u_spi_spiif module adjusts the characteristics of the SPI rate or polarity of transmission and reception according to the configuration.

The u_spi_regif_ctrl module is mainly used to control and analyze signals related to DMA.

The XIP module takes effect when it is configured in XIP mode, and its role is to directly map ahb_bus to SPI_BUS.

u_spi_fifo serves as a buffer for sending and receiving data.

Figure 10-6 HSPI Diagram



10.5.2 Features

The SoC embeds HSPI for high-speed applications, features of HSPI are listed as following:

- Supports SPI Master/Slave mode
- Supports Dual line, Quad line and 3 line I/O SPI interface

- Supports XIP function
- Supports LCD driving with SPI ports
- Supports DMA transmission

10.5.3 Function Description

10.5.3.1 Master Mode

Users can define transmit data format by configure TransMode registers, the transmitted data will be written in to HSPI FIFO via software or DMA.

Master transfer format is shown as following:

Figure 10-7 Master Transfer Mode Format

8bit cmd(default disable)	1~4bytes Addr(default disable)	Transfer mode

Set the cmd_en bit of the SPI_MODE2 register to 1 to indicate that cmd phase is enabled.

See SPI_TRANS0 register for transfer mode configuration.

The SPI output clock of Master mode can be divided by register, which can be up to ahb_clock.

10.5.3.2 Slave Mode

The format that Slave receives is fixed, so the Master needs to send in the prescribed format.

Slave transfer format is shown as following:

Figure 10-8 Slave Transfer Mode Format

8bit slave command	8bit dummy	Slave data

Slave judges the read and write operations of the Master according to the received command.

Slave commands are listed in the table below:

Table 10-7 Slave Commands

Slave Command	OP Code	Slave Data
Read status single io	0x05	8bit state(slave ready:0x5a or not ready: 0x00)
Read status dual io	0x15	8bit state(slave ready:0x5a or not ready:0x00)
Read status quad io	0x25	8bit state(slave ready:0x5a or not ready:0x00)
Read data single io	0x0b	Reply data from txfifo
Read data dual io	0x0c	Reply data from txfifo
Read data quad io	0x0e	Reply data from txfifo

Slave Command	OP Code	Slave Data
Write data single io	0x51	Data saved to rxfifo
Write data dual io	0x52	Data saved to rxfifo
Write data quad io	0x54	Data saved to rxfifo
User-defined	Any 8bit numbers other than the listed OP codes	Depending on the transfer control Register

The SPI input clock should be in the following range for Slave mode: master spi_clk frequency <= 1/4 slave ahb_clk frequency

10.5.3.3 Dual, Quad and 3line I/O

Master's Dual and Quad I/O are configured via the following registers:

spi_dual, spi_quad, cmd_fmt, Addr_fmt.

Spi_dual and spi_quad are used for the Data section. Writing 1 to cmd_fmt means that the I/O mode of the command is the same as that of the Data section, and writing 1 to Addr_fmt means that the I/O mode of the Addr is the same as the Data section.

Master's 3line I/O mode indicates that mosi is a bidirectional I/O. Configured by spi_lsb of register SPIMODE0. SPI_CSN, SPI_CLK, SPI_MOSI form a group of SPI interfaces.

Slave's Dual and Quad I/O are determined based on the command analyzed by Slave. But Slave's command and dummy are fixed only according to single I/O.

Slave also supports 3line mode, and the slave command is only available in single IO mode. The spi_lsb of the SPIMODE0 configuration register is also required. SPI_CSN, SPI_CLK, and SPI_MOSI form a group of SPI interfaces.

10.5.3.4 XIP

The function of XIP is to map the reading and writing of AHB bus to the timing of SPI.

Before the XIP transmission, the SPI mode should be configured, such as dual (quad) I/O, transmode, write/read command, etc. After the SPI modes are all configured, enable the register xip_enable, at this time the ahb bus request can be converted into SPI timing.

10.5.3.5 LCD Display Driving

HSPI can drive LCD display with SPI interface.

There are 3 ways to drive LCD display with SPI: 3-line, 4-line, 2-line.

These 3 methods and all RGB data formats (565, 666, 888) are all supported.

The 4-line format requires an additional GPIO to represent the D/CX signal.

Read/Write Sequences

Read/write sequences of 3-line, 4-line and 2-line serial interfaces are shown in figures below. Please be noted, 2-Line serial interface is designed for writing data in LCD screen, so only writing sequence is shown.

Figure 10-9 3-Line Write Sequence

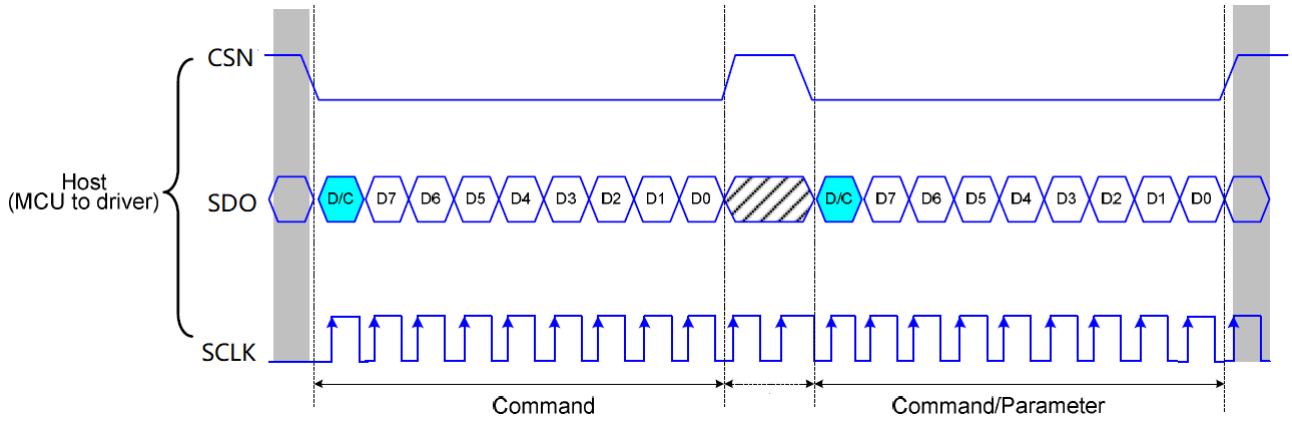


Figure 10-10 3-Line Read Sequence

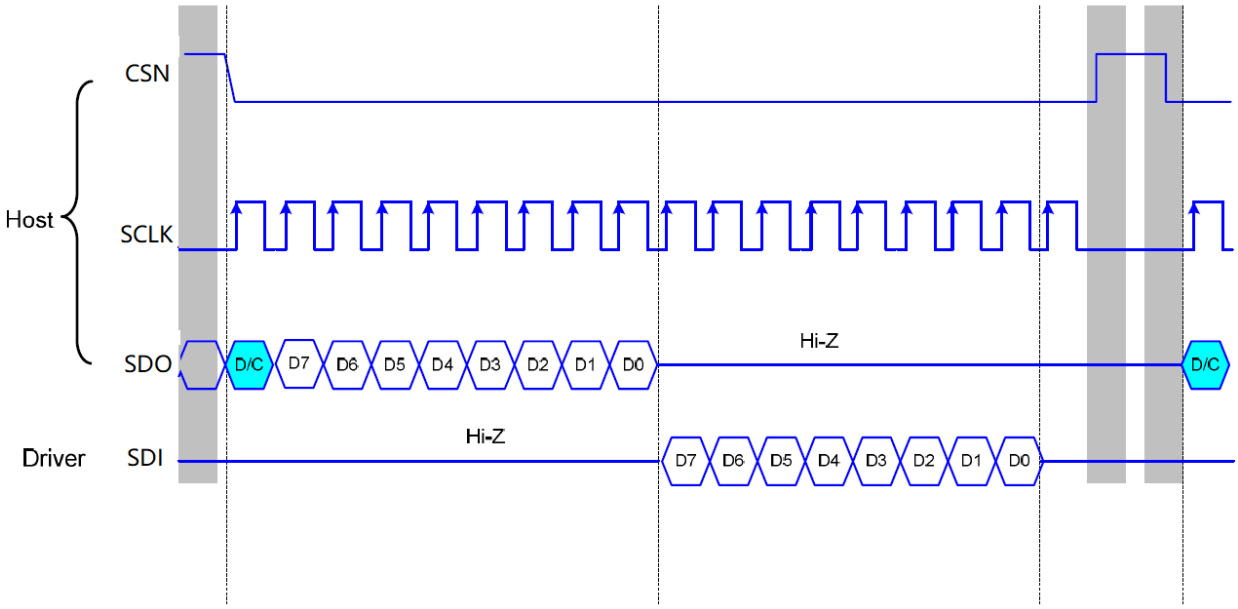


Figure 10-11 4-Line Write Sequence

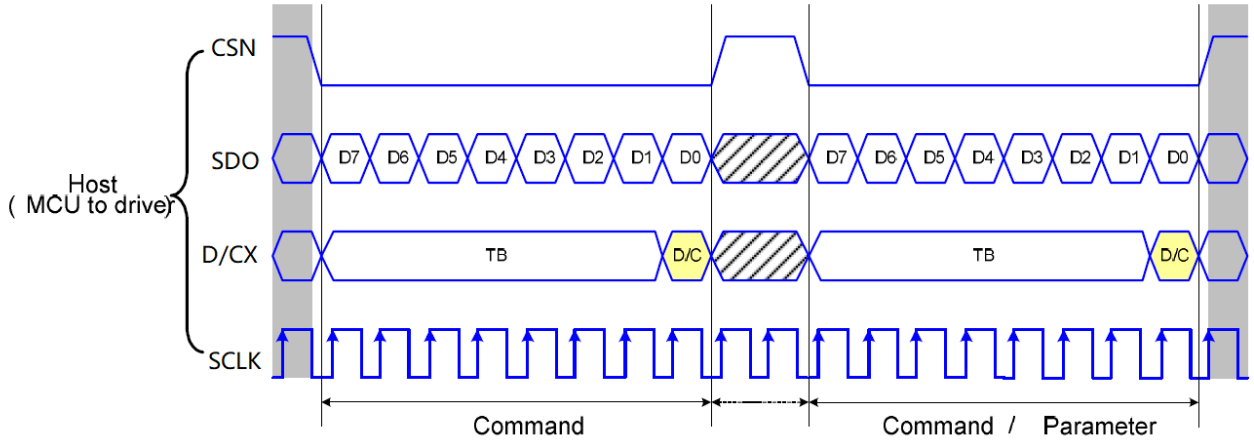


Figure 10-12 4-Line Read Sequence

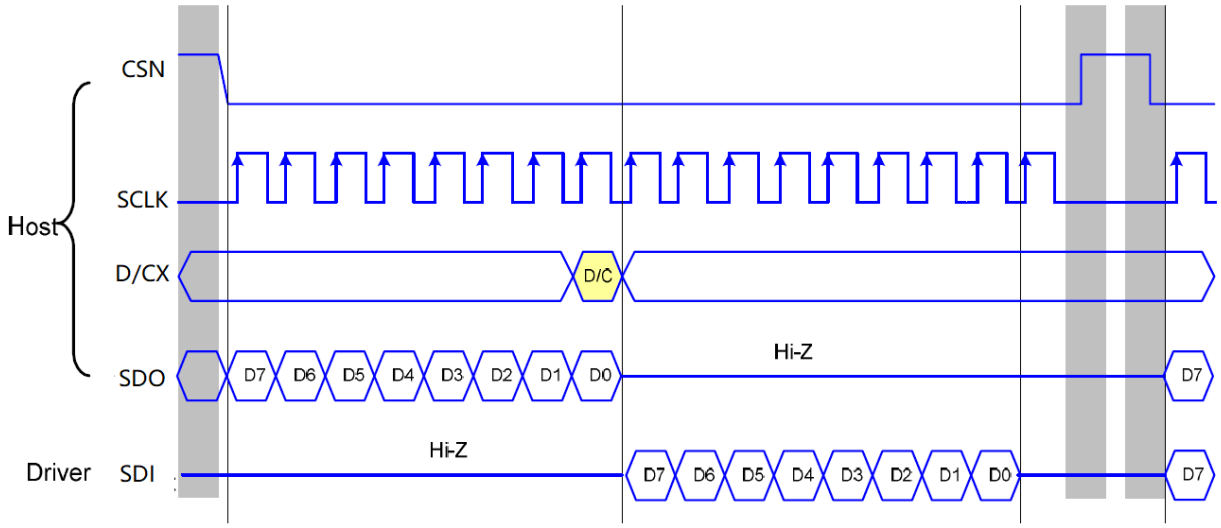
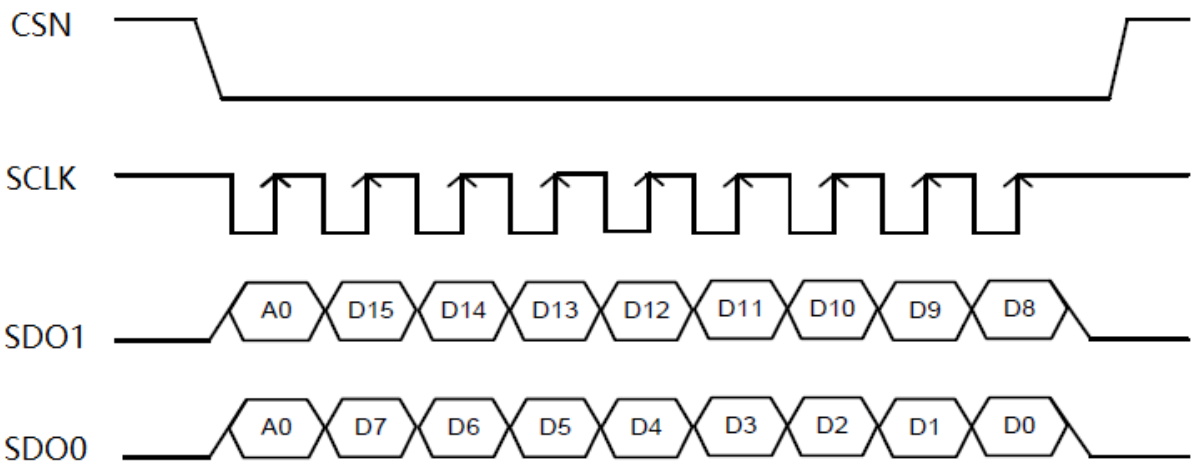


Figure 10-13 2-Line Write Sequence



RGB Formats

RGB565, RGB666 and RGB888 pixel transitions are shown in figures below.

Figure 10-14 RGB565 Pixel Transition

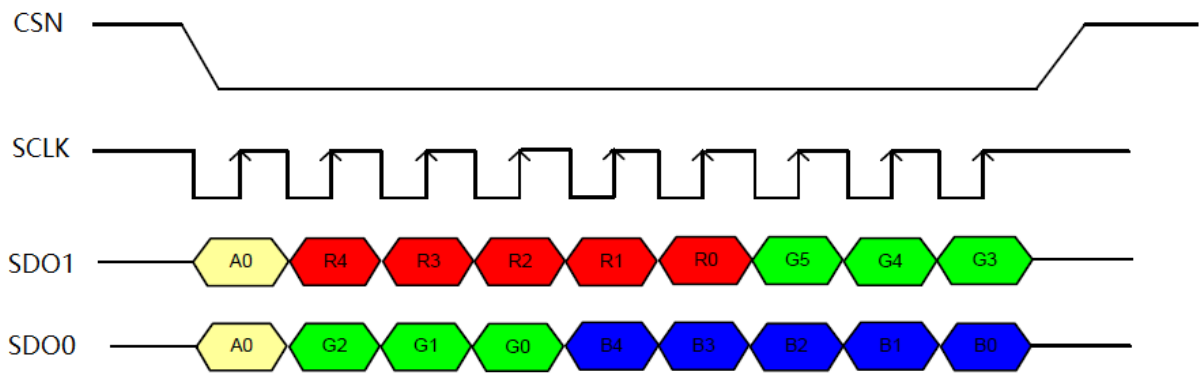


Figure 10-15 RGB666 Pixel Transition

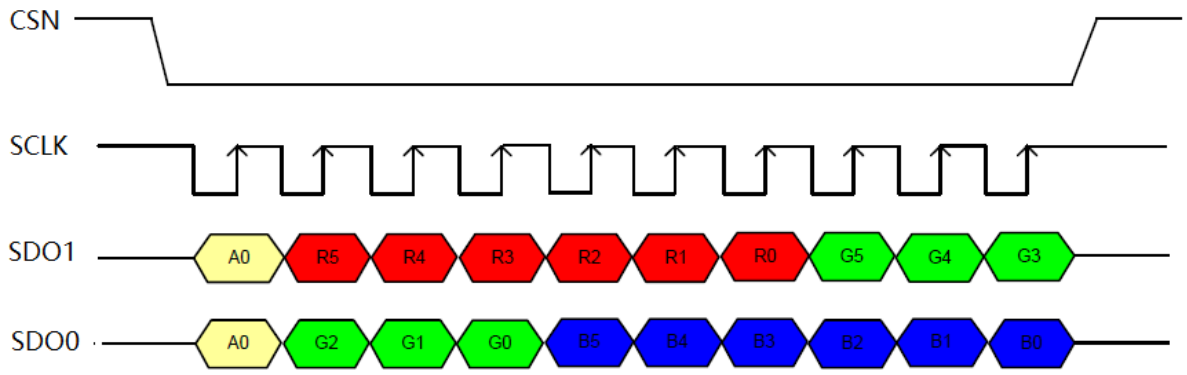
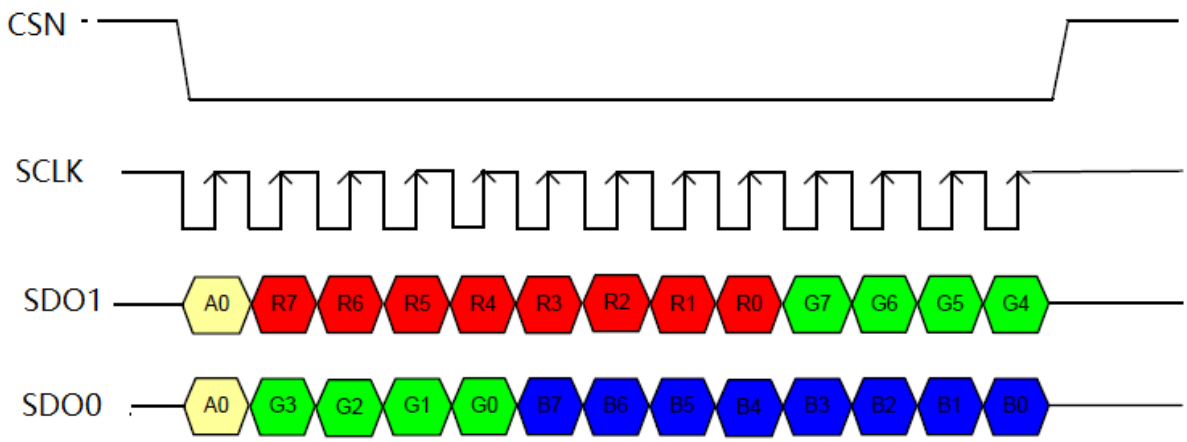


Figure 10-16 RGB888 Pixel Transition

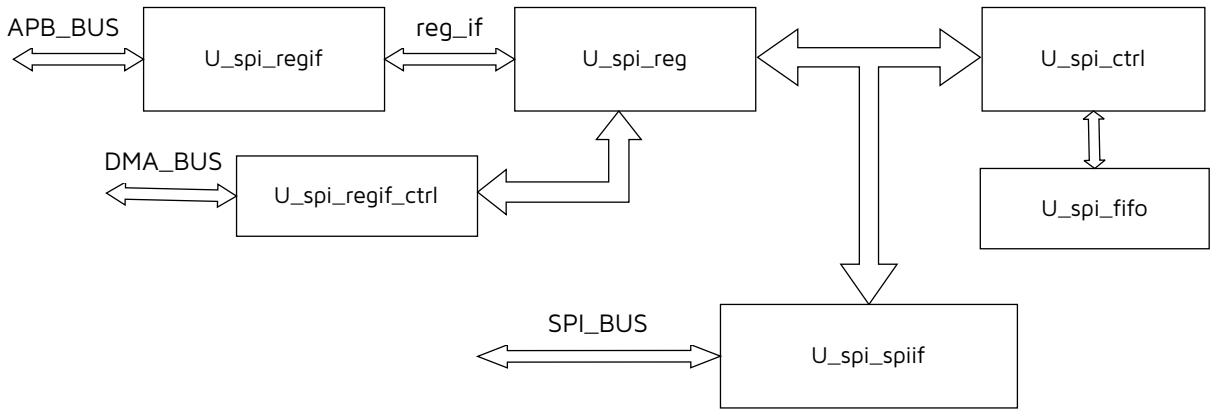


10.6 PSPI

10.6.1 Diagram

PSPI diagram is shown as following:

Figure 10-17 PSPI Diagram



As shown in the diagram, the bus between the SPI module and the DMA module. SPI_BUS is the SPI interface connected to the pad.

u_spi_regif is to parse the APB protocol and send it to the u_spi_reg module for register configuration.

The u_spi_ctrl module selects the state and mode according to the value of the register configuration, and controls the format of the transmitted data.

The u_spi_spiif module adjusts the characteristics of the SPI rate or polarity of transmission and reception according to the configuration.

The u_spi_regif_ctrl module is mainly used to control and analyze signals related to DMA.

u_spi_fifo serves as a buffer for sending and receiving data.

10.6.2 Features

The SoC embeds PSPI for low-power consumption applications, features of PSPI are listed as following:

- Supports SPI Master/Slave mode
- Supports Dual line and 3 line I/O SPI interface
- Supports DMA transmission

10.6.3 Function Descriptions

Master Mode

Users can define transmit data format by configure TransMode registers, the transmitted data will be written in to PSPI FIFO via software or DMA.

Master transfer format is shown as following:

Figure 10-18 PSPI Master Transfer Format

8bit cmd(default disable)	Transfer mode

Write 1 to the cmd_en bit of the SPI_MODE2 register to indicate that cmd phase is enabled.

See SPI_TRANS0 register for transfer mode configuration.

The SPI output clock of Master mode can be divided by register, and the fastest can reach abp_clock.

Slave Mode

The format that Slave receives is fixed, so the Master needs to send in the prescribed format.

Slave transfer format is shown as following:

Figure 10-19 Slave Transfer Mode Format

8bit slave command	8bit dummy	Slave data

Slave judges the read and write operations of the Master according to the received command.

Slave commands are listed in the table below:

Table 10-8 Slave Commands

Slave Command	OP Code	Slave Data
Read status single io	0x05	8bit state(slave ready:0x5a or not ready: 0x00)
Read status dual io	0x15	8bit state(slave ready:0x5a or not ready:0x00)
Read data single io	0x0b	Reply data from txfifo
Read data dual io	0x0c	Reply data from txfifo
Write data single io	0x51	Data saved to rxfifo
Write data dual io	0x52	Data saved to rxfifo
User-defined	Any 8bit numbers other than the listed OP codes	Depending on the transfer control Register

The SPI input clock should be in the following range for Slave mode: master spi_clk frequency <= 1/4 slave ahb_clk frequency

Dual, 3line I/O

The Dual I/O of the Master is configured through registers, and the Spi_dual register is used for the I/O attributes of the Data segment. Command is a fixed single I/O.

Master's 3line I/O mode indicates that mosi is a bidirectional I/O. Configured by spi_lsb of register SPIMODE0. SPI_CSN, SPI_CLK, SPI_MOSI form a group of SPI interfaces.

Slave's Dual I/O is determined based on the command analyzed by Slave. But Slave's command and dummy are fixed only according to single I/O.

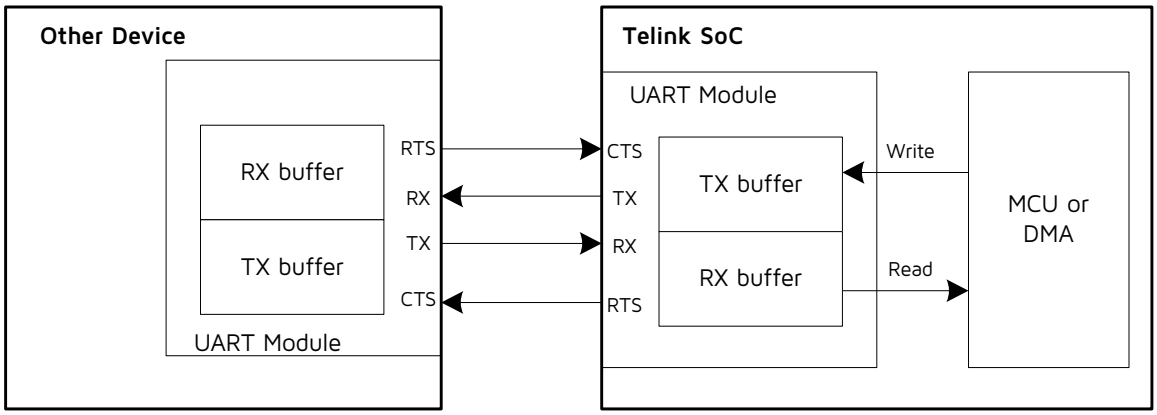
Slave also supports 3line mode, and the slave command is only available in single io mode. The spi_lsb of the SPIMODE0 configuration register is also required. SPI_CSN, SPI_CLK, and SPI_MOSI form a group of SPI interfaces.

10.7 UART

The SoC embeds UART (Universal Asynchronous Receiver/Transmitter) to implement full-duplex transmission and reception via UART TX and RX interface. Both TX and RX interface are 4-layer FIFO (First In First Out) interface. The SoC Supports 2-channel UART, UART0 and UART1.

Hardware flow control is supported via RTS and CTS.

Figure 10-20 UART Communication



As shown in figure above, data to be sent is first written into TX buffer by MCU or DMA, then UART module transmits the data from TX buffer to other device via pin TX. Data to be read from other device is first received via pin RX and sent to RX buffer, then the data is read by MCU or DMA.

If RX buffer of the UART is close to full, the SoC will send a signal (configurable high or low level) via pin RTS to inform other device that it should stop sending data. Similarly, if the SoC receives a signal from pin CTS, it indicates that RX buffer of other device is close to full and the SoC should stop sending data.

UART related registers are listed in tables below.

For UART0 related register, the base address is 0x8140080, for UART1 related register, the base address is 0x8140c0.

Table 10-9 UART0 Related Registers

Offset	R/W	Description	Default Value
0x00	R	UART_DATA_BUF0 Write/read buffer[7:0]	0x00

Offset	R/W	Description	Default Value
0x01	R	UART_DATA_BUF1 Write/read buffer[15:8]	0x00
0x02	R	UART_DATA_BUF2 Write/read buffer[23:16]	0x00
0x03	R	UART_DATA_BUF3 Write/read buffer[31:24]	0x00
0x04	RW	UART_CLK_DIV_L uart_cli_div[7:0]:uart clk div register	0xff
0x05	RW	UART_CLK_DIV_H uart_cli_div[15:8]:uart_sclk = sclk/ (uart_clk_div[14:0]+1) uart_clk_div[15] :1:enable clock divider,0: disable.	0x0f
0x06	RW	UART_CTRL0 [3:0] bpwc, bit width, should be larger than 2 Buadrate = uart_sclk/(bpwc+1) [6] rx interrupt enable [7] tx interrupt enable	0x0f
0x07	RW	UART_CTRL1 [0] cts select, 0: cts_i, 1: cts _i inverter [1] cts enable, 1: enable, 0, disable [2] Parity, 1: enable, 0:disable [3] even Parity or odd [5:4] stop bit, 00: 1 bit, 01, 1.5bit 1x: 2bits [6] ttl enable [7] uart tx, rx loopback	0x0e
0x08	RW	UART_CTRL2 [3:0] rts trig level [4] rts Parity [5] rts manual value [6] rts manual enable [7] rts enable	0xa5

Offset	R/W	Description	Default Value
0x09	RW	UART_CTRL3 [3:0] rx_irq_trig level [7:4] tx_irq_trig level	0x44
0x0a	RW	UART_RXTIMEOUT_O_L r_rxtimeout_o[7:0]:The setting is transfer one bytes need cycles base on uart_clk. For example, if transfer one bytes (1start bit+8bits data+1 priority bit+2stop bits) total 12 bits, this register setting should be (bpwc+1)*12.	0xc0
0x0b	RW	UART_RXTIMEOUT_O_H [1:0] r_rxtimeout_o[9:8]:R_rxtimeout 2'b00:rx timeout time is r_rxtimeout[7:0] 2'b01:rx timeout time is r_rxtimeout[7:0]*2 2'b10:rx timeout time is r_rxtimeout[7:0]*3 3'b11: rx timeout time is r_rxtimeout[7:0]*4 R_rxtimeout is for rx dma to decide the end of each transaction. Supposed the interval between each byte in one transaction is very short. [2] mask_rxdone [6] mask_txdone [7] mask_err	0x01
0x0c	Volatile	UART_BUFCNT [3:0] r_buf_cnt [7:4] t_buf_cnt	0x00
0x0d	Volatile	UART_STATUS [2:0] rbcnt [3] irq [6:4] R: wbcnt, W: [6] write 1 to clear rx [7] R: rx_err, W:[7] write 1 to clear tx	0x00
0x0e	Volatile	UART_TXRX_STATUS [0] txdone [1] txbuf_irq [2] rxdone [3] rxbuf_irq	0x00

Offset	R/W	Description	Default Value
0x0f	Volatile	UART_STATE [2:0] tx state machine [7:4] rx state machine	0x00

10.8 USB

The SoC has a full-speed (12Mbps) USB interface for communicating with other compatible digital devices. The USB interface acts as a USB peripheral, responding to requests from a master host controller. The chip contains internal 1.5kohm pull up resistor for the DP pin.

Telink USB interface supports the Universal Serial Bus Specification, Revision v2.0 (USB v2.0 Specification).

The chip supports 9 endpoints, including control endpoint 0 and 8 configurable data endpoints. Endpoint 1, 2, 3, 4, 7 and 8 can be configured as input endpoint, while endpoint 5 and 6 can be configured as output endpoint. In audio class application, only endpoint 6 supports iso out mode, while endpoint 7 supports iso in mode. In other applications, each endpoint can be configured as bulk, interrupt and iso mode. For control endpoint 0, the chip's hardware vendor command is configurable.

Optional suspend mode:

- Selectable as USB suspend mode or chip suspend mode, support remote wakeup.
- Current draw in suspend mode complied with USB v2.0 Specification.
- USB pins (DM, DP) can be used as GPIO function in suspend mode.
- Resume and detach detect: Recognize USB device by detecting the voltage on the DP pin with configurable 1.5K pull-up resistor.
- USB pins configurable as wakeup GPIOs.

The USB interface belongs to an independent power domain, and it can be configured to power down independently.

The USB related registers are listed in table below, the base address is 0x80100800.

Table 10-10 USB Related Registers

Offset	R/W	Description	Default Value
0x00	VOLATILE	EDP0PTR [3:0]: reg_ptr, Endpoint 0 buffer point	0x00
0x01	VOLATILE	EDP0DAT [7:0]:buff,Endpoint 0 buffer data access address	0x00

Offset	R/W	Description	Default Value
0x02	VOLATILE	EDP0CT [0]:ack_data, Ack data [1]:stall_data, Stall data [2]:ack_status, Ack status [3]:stall_status, Stall status	0x00
0x03	R	EDP0ST [3:0]:udc_cnt,number of data transferred [4]:irq_setup, udc_irq_o={irq_setinf,irq_status,irq_data,irq_setup}: setup interrupt flag [5]:irq_data, data interrupt flag [6]:irq_status, status interrupt flag [7]:irq_setinf, set interface interrupt flag	0x00
0x04	RW	EDP0MODE [0]:r_en_sadr,enable auto decoding set_address command [1]:r_en_cfg ,enable auto decoding set_config command [2]:r_en_inf ,enable auto decoding set_interface command [3]:r_en_sta ,enable auto decoding get_status command [4]:r_en_frm ,enable auto decoding sync_frame command [5]:r_en_desc,enable auto decoding get_descriptor command [6]:r_en_fea,enable auto decoding set_feature command [7]:r_en_hw,enable auto decoding standard command	0xff
0x05	RW	USBCT [0]:r_clk_sel_0use auto calibrate clock if 1, use system clock if 0 [1]:low_speedlow speed mode if 1; full speed mode if 0 [2]:r_clk_sel_2low jitter mode if 1 [3]:test_modeusb test mode [7:4]:r_clk_sel_o2 for select 48M RC clock; 1 for 400M RC	0x01

Offset	R/W	Description	Default Value
0x06	R	CALCYCL [7:0]:r_clk_div_il,r_clk_div_i	0x00
0x07	R	CALCYCH [2:0]:r_clk_div_ih,r_clk_div_i	0x00
0x0a	RW	MDEV [0]:r_mdev,self power 1: self power 0: bus power [1]:suspend_i,USB suspend status read only [2]:wakeup_feature_o,wakeup feature read only [3]:r_vend,r_vnd[0] vendor cmd offset (byte1[7] == r_vnd[0] means vendor cmd) [4]:r_vend_disable,1 for disable vendor cmd	0x00
0x0b	RW	EDP0SIE [6:0]:sie_adr_i,sie_adr_i[6:0] [7]:r_config	0x00
0x0c	RW	SUSPENDCYC, r_suspenf_cnt	0x18
0x0d	R	INFALT [7:0];r_infalt,Interface and alternate setting number in last SET_INTERFACE command	0x00
0x0e	RW	EDPS_EN [7:0]:edps_en	0xff
0x0f	RW	IRQ_MASK [2:0]:r_maskmask[0]: irq_reset; mask[1]:irq_250us; mask[2]:irq_suspend; [4:3]:r_lvllvl[0]:0-->irq_reset_edge;1-->usb_reset_i; lvl[1]:0-->irq_250us_edge; 1-->usb_250us_i; [5]:irq_reset_oW:Clear usb reset edge interrupted [6]:irq_250us_oW:Clear usb 250us edge interrupted [7]:irq_suspend_oUSB suspend status read only: mask & suspend	0x04
0x10	VOLATILE	EDPSPTR [7:0]:rd_ptrl	0x00
0x11	VOLATILE	EDPS1PTR [7:0]:rd_ptrl	0x00

Offset	R/W	Description	Default Value
0x12	VOLATILE	EDPS2PTR [7:0]:rd_ptrl	0x00
0x13	VOLATILE	EDPS3PTR [7:0]:rd_ptrl	0x00
0x14	VOLATILE	EDPS4PTR [7:0]:rd_ptrl	0x00
0x15	VOLATILE	EDPS5PTR [7:0]:rd_ptrl	0x00
0x16	VOLATILE	EDPS6PTR [7:0]:rd_ptrl	0x00
0x17	VOLATILE	EDPS7PTR [7:0]:rd_ptrl	0x00
0x18	VOLATILE	EDPSDATA [7:0]:sr_q	0x00
0x19	VOLATILE	EDPS1DATA [7:0]:sr_q	0x00
0x1a	VOLATILE	EDPS2DATA [7:0]:sr_q	0x00
0x1b	VOLATILE	EDPS3DATA [7:0]:sr_q	0x00
0x1c	VOLATILE	EDPS4DATA [7:0]:sr_q	0x00
0x1d	VOLATILE	EDPS5DATA [7:0]:sr_q	0x00
0x1e	VOLATILE	EDPS6DATA [7:0]:sr_q	0x00
0x1f	VOLATILE	EDPS7DATA [7:0]:sr_q	0x00

Offset	R/W	Description	Default Value
0x20	VOLATILE	EDPSCT [0]:rd_ack [1]:rd_stall [2]:set_data0,Set Data0 [3]:set_data1,Set Data1 [7]:edp8_dma_eofLaunch EOF for FIFO mode (W) (no support)	0x00
0x21	VOLATILE	EDP1SCT [0]:rd_ack [1]:rd_stall [2]:set_data0,Set Data0 [3]:set_data1,Set Data1	0x00
0x22	VOLATILE	EDP2SCT [0]:rd_ack [1]:rd_stall [2]:set_data0?Set Data0 [3]:set_data1?Set Data1	0x00
0x23	VOLATILE	EDP3SCT [0]:rd_ack [1]:rd_stall [2]:set_data0,Set Data0 [3]:set_data1,Set Data1	0x00
0x24	VOLATILE	EDP4SCT [0]:rd_ack [1]:rd_stall [2]:set_data0,Set Data0 [3]:set_data1,Set Data1	0x00
0x25	VOLATILE	EDP5SCT [0]:rd_ack [1]:rd_stall [2]:set_data0,Set Data0 [3]:set_data1,Set Data1	0x00

Offset	R/W	Description	Default Value
0x26	VOLATILE	EDP6SCT [0]:rd_ack,ACK [1]:rd_stall,Stall [2]:set_data0,Set Data0 [3]:set_data1,Set Data1 [6]:rd_mono_aout,MONO mode [7]:rd_en_aout,Audio ISO out enable	-
0x27	VOLATILE	EDP7SCT [0]:rd_ack,ACK [1]:rd_stall,Stall [2]:set_data0,Set Data0 [3]:set_data1,Set Data1 [6]:rd_mono_aout,MONO mode [7]:rd_en_aout,Audio ISO out enable	-
0x28	VOLATILE	EDPSADR,rd_adr	0x80
0x29	RW	EDPS1ADR,Endpoint 1 buffer address	0x00
0x2a	RW	EDPS2ADR,Endpoint 2 buffer address	0x08
0x2b	RW	EDPS3ADR,Endpoint 3 buffer address	0x10
0x2c	RW	EDPS4ADR,Endpoint 4 buffer address	0x40
0x2d	RW	EDPS5ADR,Endpoint 5 buffer address	0xc0
0x2e	RW	EDPS6ADR,Endpoint 6 buffer address	0x20
0x2f	RW	EDPS7ADR,Endpoint 7 buffer address	0x30
0x30	RW	USBRAM [0]:sr_cen,CEN in power down mode [1]:sr_clk,CLK in power down mode [2]:r_ram2,Reserved [3]:wen_i,WEN in power down mode [4]:r_ram4,CEN in function mode	0x18
0x38	RW	USBSOEnable endpoint ISO mode	0xc0
0x39	RW	USBIRQ [7:0]:r_irq,Endpoint data transfer interrupt	0x00
0x3a	RW	USBMASK,Endpoint interrupt mask	0xff

Offset	R/W	Description	Default Value
0x3b	RW	USBMAX0,Maximum endpoint 8 transfer number	0x10
0x3c	RW	USBMIN0,Minimum threshold to ACK endpoint 8 transfer	0x40
0x3d	RW	USBFIFO [0]:r_fifo0,Endpoint 0 FIFO mode [1]:full0,Full flag [2]:r_mode00 [3]:edp8_eof [6:4]:edp8_dma_eof [7]:r_mode05	-
0x3e	RW	USBMAX [6:0]:max_in, USBMAX*8	0x08
0x3f	VOLATILE	USBTICK	0x00

11 PWM

The SoC supports 6-channel PWM (Pulse-Width-Modulation) output. Each PWM#n (n=0~5) has its corresponding inverted output at PWM#n_N pin.

11.1 Enable PWM

Register PWM_EN[5:1] and PWM_EN0[0] serves to enable PWM5~PWM0 respectively via writing "1" for the corresponding bits.

11.2 Set PWM Clock

PWM clock derives from system clock. Register PWM_CLKDIV serves to set the frequency dividing factor for PWM clock. Formula below applies:

$$F_{PWM} = F_{System\ clock} / (PWM_CLKDIV+1)$$

11.3 PWM Waveform, Polarity and Output Inversion

Each PWM channel has independent counter and 2 status including "Count" and "Remaining". Count and Remaining status form a signal frame.

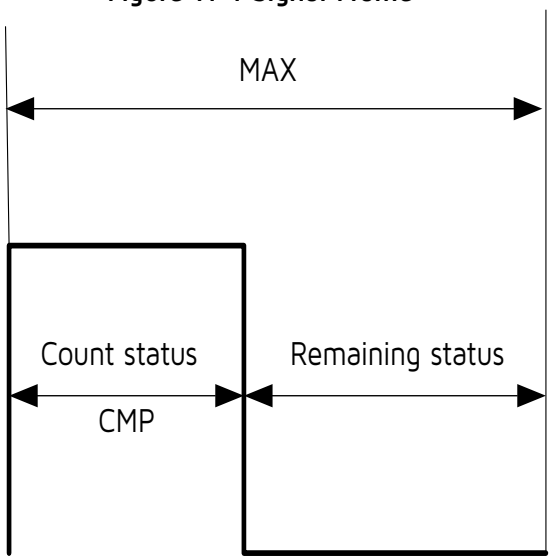
11.3.1 Waveform of Signal Frame

When PWM#n is enabled, first PWM#n enters Count status and outputs High level signal by default. When PWM#n counter reaches cycles set in register PWM_TCMP#n / PWM_TCMP_FSK_L/ PWM_TCMP_FSK_H PWM#n enters Remaining status and outputs Low level till PWM#n cycle time configured in register PWM_TMAX#n / PWM_TMAX_FSK_L/ PWM_TMAX_FSK_H expires.

An interruption will be generated at the end of each signal frame if enabled via register PWM_MASK.

Signal frame is shown as following:

Figure 11-1 Signal Frame



11.3.2 Invert PWM Output

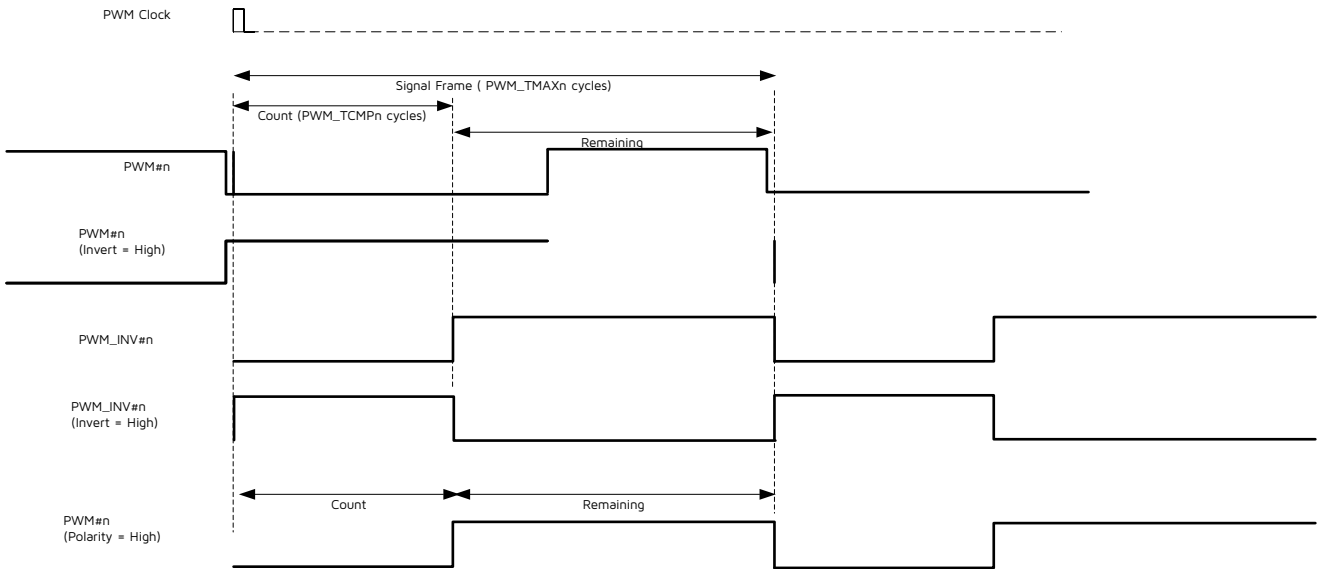
PWM#n and PWM#n_N output could be inverted independently via register PWM_CC0 and PWM_CC1. When the inversion bit is enabled, waveform of the corresponding PWM channel will be inverted completely.

11.3.3 Polarity for Signal Frame

By default, PWM#n outputs High level at Count status and Low level at Remaining status. When the corresponding polarity bit is enabled via register PWM_CC2[5:0], PWM#n will output Low level at Count status and High level at Remaining status.

PWM output waveform is shown as below.

Figure 11-2 PWM Output Waveform Chart



11.4 PWM Mode

11.4.1 Select PWM Modes

PWM0 supports five modes, including Continuous mode (normal mode, default), Counting mode, IR mode, IR FIFO mode, IR DMA FIFO mode.

PWM1~PWM5 only support Continuous mode.

Register PWM_MODE serves to select PWM0 mode.

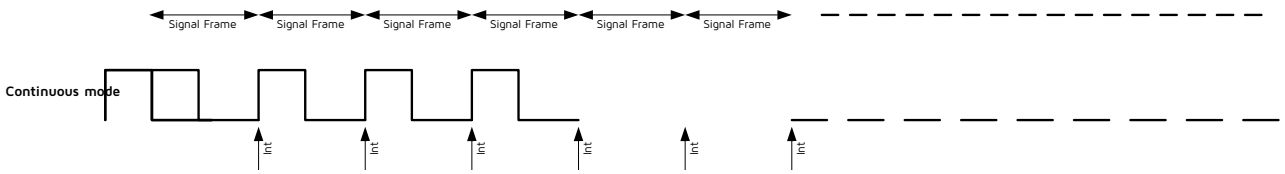
11.4.2 Continuous Mode

PWM0~PWM5 all support Continuous mode. In this mode, PWM#n continuously sends out signal frames. PWM#n should be disabled via PWM_EN/PWN_EN0 to stop it; when stopped, the PWM output will turn low immediately.

During Continuous mode, waveform could be changed freely via PWM_TCMP#n and PWM_TMAX#n. New configuration for PWM_TCMP#n and PWM_TMAX#n will take effect in the next signal frame.

After each signal frame is finished, corresponding PWM cycle done interrupt flag bit (PWM_INT[2:7]) will be automatically set to 1b'1. If the interrupt is enabled by setting PWM_MASK0[2:7] as 1b'1, a frame interruption will be generated. User needs to write 1b'1 to the flag bit to manually clear it.

Figure 11-3 Continuous Mode



11.4.3 Counting Mode

Only PWM0 supports Counting mode. PWM_MODE [2:0] should be set as 4b'0001 to select PWM0 counting mode.

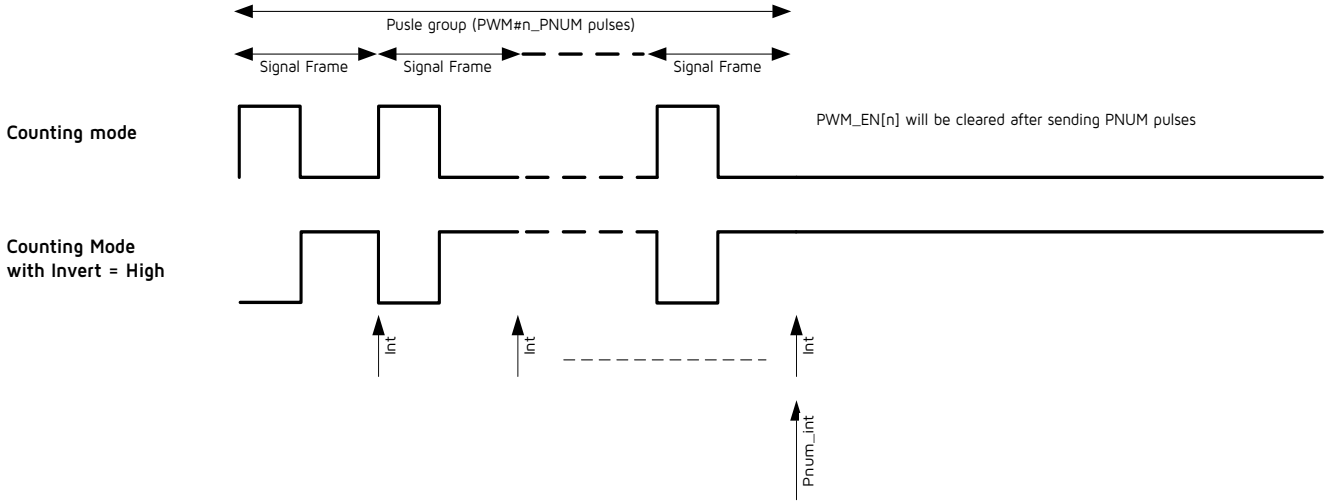
In this mode, PWM0 sends out specified number of signal frames which is defined as a pulse group. The number is configured via register PWM_PNUM.

After each signal frame is finished, PWM0 cycle done interrupt flag bit (PWM_INT[2]) will be automatically set to 1b'1. If the interrupt is enabled by setting PWM_MASK0 [2] as 1b'1, a frame interruption will be generated. User needs to write 1b'1 to the flag bit to manually clear it.

After a pulse group is finished, PWM0 will be disabled automatically, and PWM0 pnum interrupt flag bit (PWM_INT [0]) will be automatically set to 1b'1. If the interrupt is enabled by setting PWM_MASK0 as 1b'1, a Pnum interruption will be generated. User needs to write 1b'1 to the flag bit to manually clear it.

Counting mode also serves to stop IR mode gracefully.

Figure 11-4 Counting Mode (n=0)



11.4.4 IR Mode

Only PWM0 supports IR mode. PWM_MODE[2:0] should be set as 4b'0011 to select PWM0 IR mode.

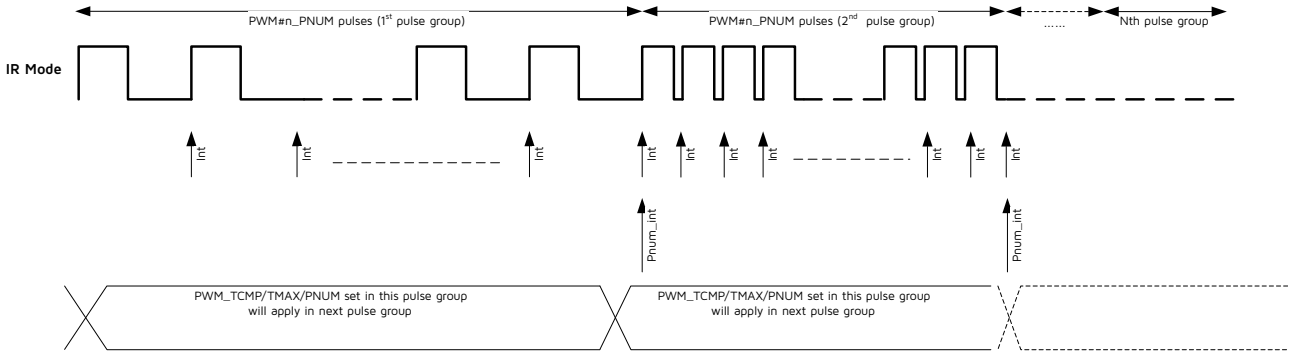
In this mode, specified number of frames is defined as one pulse group. In contrast to Counting mode where PWM0 stops after first pulse group is finished, PWM0 will constantly send pulse groups in IR mode.

During IR mode, PWM0 output waveform could also be changed freely via PWM_TCMP0, PWM_TMAX0 and PWM_PNUM0. New configuration for PWM_TCMP0, PWM_TMAX0 and PWM_PNUM0 will take effect in the next pulse group.

To stop IR mode and complete current pulse group, user can switch PWM0 from IR mode to Counting mode so that PWM0 will stop after current pulse group is finished. If PWM0 is disabled directly via PWM_EN0[0], PWM0 output will turn Low immediately despite of current pulse group.

After each signal frame/pulse group is finished, PWM0 cycle done interrupt flag bit (PWM_INT[2])/PWM0 pnum interrupt flag bit (PWM_INT[0]) will be automatically set to 1b'1. A frame interruption/Pnum interruption will be generated.

Figure 11-5 IR Mode (n=0)



11.4.5 IR FIFO Mode

IR FIFO mode is designed to allow IR transmission of long code patterns without the continued intervention of MCU, and it is designed as a selectable working mode on PWM0. The IR carrier frequency is divided down from the system clock and can be configured as any normal IR frequencies, e.g. 36kHz, 38kHz, 40kHz, or 56kHz.

Only PWM0 supports IR FIFO mode. PWM_MODE[2:0] should be set as 4b'0111 to select PWM0 IR FIFO mode.

An element ("FIFO CFG Data") is defined as basic unit of IR waveform, and written into FIFO. This element consists of 16 bits, including:

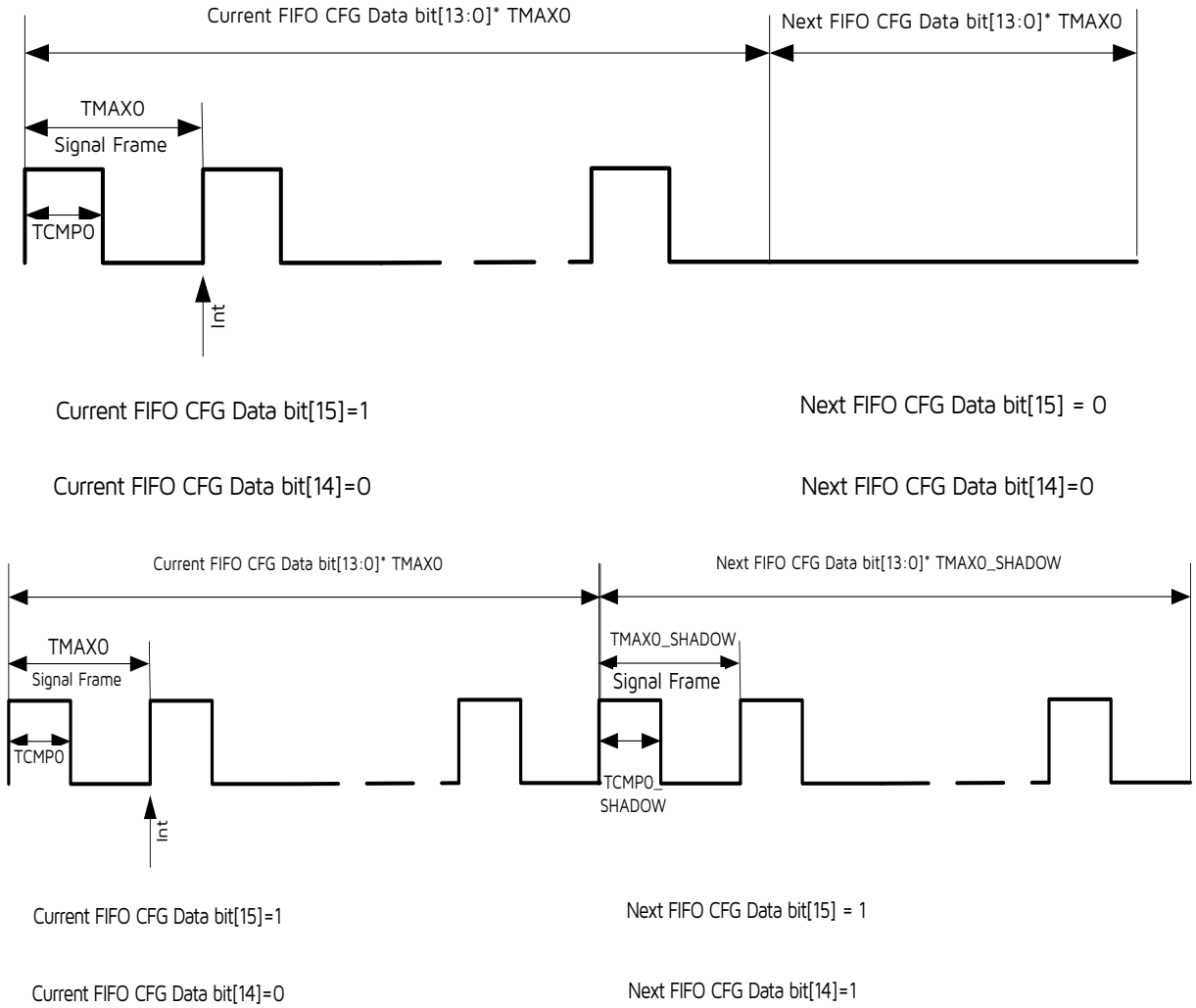
- bit[13:0] defines PWM pulse number of current group.
- bit[14] determines duty cycle and period for current PWM pulse group.
- 0: use configuration of TCMP0 and TMAX0;
- 1: use configuration of PWM_TCMP_FSK_L/PWM_TCMP_FSK_H and PWM_TMAX_FSK_L/ PWM_TMAX_FSK_H.
- bit[15] determines whether current PWM pulse group is used as carrier, i.e. whether PWM will output pulse (1) or low level (0).

User should use PWM_RDAT_L0, PWM_RDAT_H0, PWM_RDAT_L1, PWM_RDAT_H1 in 0x7c8~0x7cb to write the 16-bit "FIFO CFG Data" into FIFO by byte or half word or word.

- To write by byte, user should successively write 0x7c8, 0x7c9, 0x7ca and 0x7cb.
- To write by half word, user should successively write 0x7c8 and 0x7ca.
- To write by word, user should write 0x7c8.

FIFO depth is 8 bytes. User can read the register FIFO_SR in 0x7cd to view FIFO empty/full status and check FIFO data number.

Figure 11-6 IR Format Examples



When "FIFO CFG Data" is configured in FIFO and PWM0 is enabled via PWM_EN0[0], the configured waveforms will be output from PWM0 in sequence. As long as FIFO doesn't overflow, user can continue to add waveforms during IR waveforms sending process, and long IR code that exceeds the FIFO depth can be implemented this way. After all waveforms are sent, FIFO becomes empty, PWM0 will be disabled automatically.

The FIFO_CLR register serves to clear data in FIFO. Writing 1b'1 to this register will clear all data in the FIFO. Note that the FIFO can only be cleared when not in active transmission.

11.4.6 IR DMA FIFO Mode

IR DMA FIFO mode is designed to allow IR transmission of long code patterns without occupation of MCU, and it is designed as a selectable working mode on PWM0. The IR carrier frequency is divided down from the system clock and can be configured as any normal IR frequencies, e.g. 36 kHz, 38 kHz, 40 kHz, or 56 kHz.

Only PWM0 supports IR DMA FIFO mode.PWM_MODE[3:0] should be set as 4b'1111 to select PWM0 IR DMA FIFO mode.

This mode is similar to IR FIFO mode, except that "FIFO CFG Data" is written into FIFO by DMA instead of MCU. User should write the configuration of "FIFO CFG Data" into RAM, and then enable DMA channel 5. DMA will automatically write the configuration into FIFO.

NOTE: *In this mode, when DMA channel 5 is enabled, PWM will automatically output configured waveform, without the need to manually enable PWM0 via PWM_EN0 (i.e. PWM_EN0[0] will be set as 1b'1 automatically).*

11.5 PWM Interrupt

There are 9 interrupt sources from PWM function.

After each signal frame, PWM#n (n = 0 ~ 5) will generate a frame-done IRQ (Interrupt Request) signal.

In Counting mode and IR mode, PWM0 will generate a Pnum IRQ signal after completing a pulse group.

In IR FIFO mode, PWM0 will generate a FIFO mode count IRQ signal when the FIFO_NUM value is less than the FIFO_NUM_LVL, and will generate a FIFO mode stop IRQ signal after FIFO becomes empty.

In IR DMA FIFO mode, PWM0 will generate an IR waveform send done IRQ signal, after DMA has sent all configuration data, FIFO becomes empty and final waveform is sent.

To enable PWM interrupt, the total enabling bit "irq_pwm" should be set as 1b'1. To enable various PWM interrupt sources, PWM_MASK0 and PWM_MASK1 should be set as 1b'1 correspondingly.

Interrupt status can be cleared via register PWM_INT0 and PWM_INT1.

11.6 Register Description

PWM related registers are listed as following.The base address for below registers is 0x80140400.

Table 11-1 PWM Registers

Offset	R/W	Description	Default Value
0x00	W	PWM_EN pwm[5:1] enable	0x00
0x01	W	PWM_EN0 pwm0 enable	0x00
0x02	RW	PWM_CLKDIV	0x00
0x03	RW	PWM_MODE [0]:crun_o [1]:catch_o [2]:fifio_mode_en	0x00
0x04	RW	PWM_CC0 invert PWM output	0x00

Offset	R/W	Description	Default Value
0x05	RW	PWM_CC1 invert PWM_INV output	0x00
0x06	RW	PWM_CC2 PWM pola	0x00
0x07	RW	MODE32K	0x00
0x14	RW	PWM_TCMP0_L tcmpb0[7:0] bits 7-0 of PWM0's high time or low time	0x00
0x15	RW	PWM_TCMP0_H tcmpb0[15:8] bits 15-8 of PWM0's high time or low time	0x00
0x16	RW	PWM_TMAX0_L tmaxb0[7:0] bits 7-0 of PWM0's cycle time	0x00
0x17	RW	PWM_TMAX0_H tmaxb0[15:8] bits 15-8 of PWM0's cycle time	0x00
0x18	RW	PWM_TCMP1_L tcmpb1_o[7:0] bits 7-0 of PWM1's high time or low time	0x00
0x19	RW	PWM_TCMP1_H tcmpb1_o[15:8] bits 15-8 of PWM1's high time or low time	0x00
0x1a	RW	PWM_TMAX1_L tmaxb1_o[7:0] bits 7-0 of PWM1's cycle time	0x00
0x1b	RW	PWM_TMAX1_H tmaxb1_o[15:8] bits 15-8 of PWM1's cycle time	0x00
0x1c	RW	PWM_TCMP2_L tcmpb2_o[7:0] bits 7-0 of PWM2's high time or low time	0x00
0x1d	RW	PWM_TCMP2_H tcmpb2_o[15:8] bits 15-8 of PWM2's high time or low time	0x00

Offset	R/W	Description	Default Value
0x1e	RW	PWM_TMAX2_L tmaxb2_o[7:0] bits 7-0 of PWM2's cycle time	0x00
0x1f	RW	PWM_TMAX2_H tmaxb2_o[15:8] bits 15-8 of PWM2's cycle time	0x00
0x20	RW	PWM_TCMP3_L tcmpb3_o[7:0] bits 7-0 of PWM3's high time or low time	0x00
0x21	RW	PWM_TCMP3_H tcmpb3_o[15:8] bits 15-8 of PWM3's high time or low time	0x00
0x22	RW	PWM_TMAX3_L tmaxb3_o[7:0] bits 7-0 of PWM3's cycle time	0x00
0x23	RW	PWM_TMAX3_H tmaxb3_o[15:8] bits 15-8 of PWM3's cycle time	0x00
0x24	RW	PWM_TCMP4_L tcmpb4_o[7:0] bits 7-0 of PWM4's high time or low time	0x00
0x25	RW	PWM_TCMP4_H tcmpb4_o[15:8] bits 15-8 of PWM4's high time or low time	0x00
0x26	RW	PWM_TMAX4_L tmaxb4_o[7:0] bits 7-0 of PWM4's cycle time	0x00
0x27	RW	PWM_TMAXB4_H tmaxb4_o[15:8] bits 15-8 of PWM4's cycle time	0x00
0x28	RW	PWM_TCMP5_L tcmpb5_o[7:0] bits 7-0 of PWM5's high time or low time	0x00
0x29	RW	PWM_TCMP5_H tcmpb5_o[15:8] bits 15-8 of PWM5's high time or low time	0x00
0x2a	RW	PWM_TMAX5_L tmaxb5_o[7:0] bits 7-0 of PWM5's cycle time	0x00

Offset	R/W	Description	Default Value
0x2b	RW	PWM_TMAX5_H tmaxb5_o[15:8] bits 15-8 of PWM5's cycle time	0x00
0x2c	RW	PWM_PNUM_L pnumb[7:0]	0x00
0x2d	RW	PWM_PNUM_H pnumb[13:8]	0x00
0x30	RW	PWM_MASK [0]:mask_pwm [1]:mask_fifo [7:2]:mask	0x00
0x31	VOLATILE	PWM_INT [0]:int_pwm,count model interrupt flag [1]:int_fifo_done,int_fifo_done [7:2]:int_flag,w1c too, int_flag[5:0]	0x00
0x32	RW	PWM_MASK_LVL	0x00
0x33	VOLATILE	PWM_INT_LVL	0x00
0x34	VOLATILE	PWM_CNT0_L	0x00
0x35	VOLATILE	PWM_CNT0_H	0x00
0x36	VOLATILE	PWM_CNT1_L	0x00
0x37	VOLATILE	PWM_CNT1_H	0x00
0x38	VOLATILE	PWM_CNT2_L	0x00
0x39	VOLATILE	PWM_CNT2_H	0x00
0x3a	VOLATILE	PWM_CNT3_L	0x00
0x3b	VOLATILE	PWM_CNT3_H	0x00
0x3c	VOLATILE	PWM_CNT4_L	0x00
0x3d	VOLATILE	PWM_CNT4_H	0x00
0x3e	VOLATILE	PWM_CNT5_L	0x00
0x3f	VOLATILE	PWM_CNT5_H	0x00
0x40	R	PWM_NCNT_L	0x00
0x41	R	PWM_NCNT_H	0x00

Offset	R/W	Description	Default Value
0x44	RW	PWM_TCMP_FSK_L	0x00
0x45	RW	PWM_TCMP_FSK_H	0x00
0x46	RW	PWM_TMAX_FSK_L	0x00
0x47	RW	PWM_TMAX_FSK_H	0x00
0x48	R	PWM_RDAT_L0	0x00
0x49	R	PWM_RDAT_H0	0x00
0x4a	R	PWM_RDAT_L1	0x00
0x4b	R	PWM_RDAT_H1	0x00
0x4c	RW	PWM_FIFO_LVL	0x00
0x4d	VOLATILE	PWM_TX_CTRL	0x10
0x4e	W	CLR_TXFIFO	0x00

12 Manchester Decoder(MDEC)

The SoC integrates one Manchester Decoder (MDEC). The MDEC is designed to decode the input Manchester code, data after Manchester coding, into binary data.

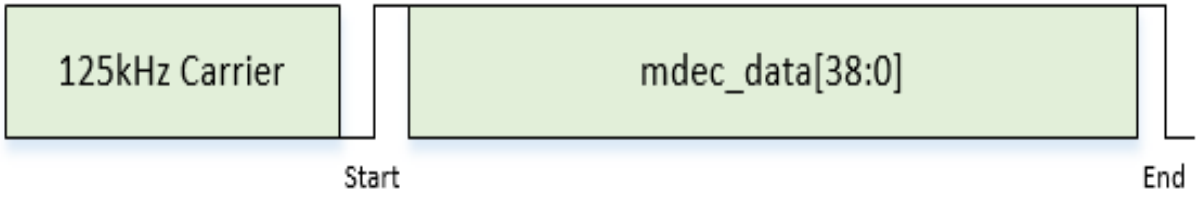
12.1 Frame Format

The MDEC's input sequence includes a Carrier signal, a Start flag, a 39-bit mdec_data filed (mdec_data[38:0]), and an End flag.

- Carrier signal duration should be no less than 3ms.
- Support duty cycle of 50%~90%.
- Period for each bit is 408us.
- The Start flag is Manchester code 1, a positive edge from low level to high level.
- The End flag is Manchester code 0, a negative edge from high level to low level.

The frame format is shown as following:

Figure 12-1 Frame Format

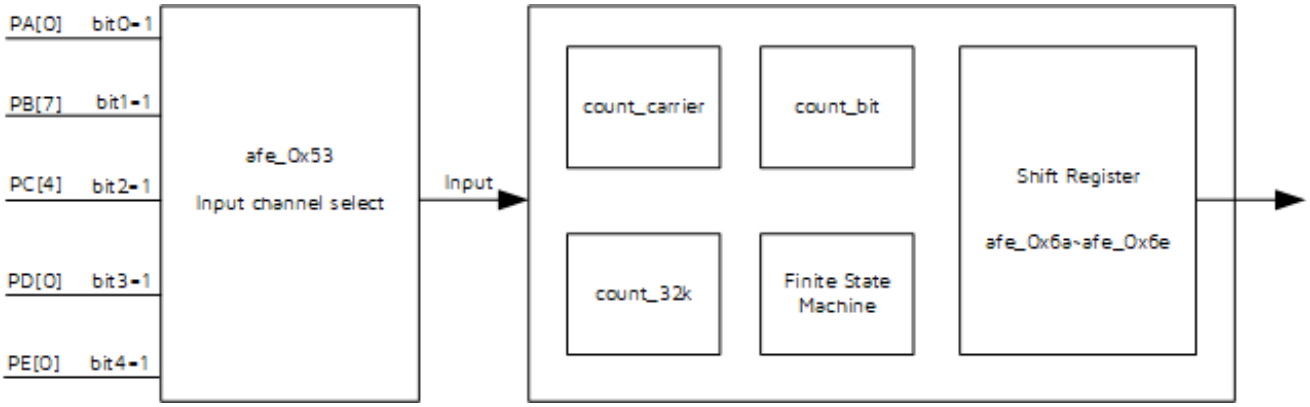


12.2 Function Description

12.2.1 Block Diagram

Function block diagram is shown below:

Figure 12-2 MDEC Function Block Diagram



As shown in figure above, the MDEC uses 32kHz clock, and it mainly embeds a finite State machine, three counters, and a Shift Register to implement its function, including:

- Finite State Machine: It includes Idle state, Carrier state, Start state, Data state, and End state.

- count_carrier: This counter serves to detect carrier signal in Idle state. When a carrier signal is detected, the MDEC's state machine enters Start state.
- count_32k: After entering Start state, this counter serves to calculate the interval between two adjacent positive edges, so as to judge the input data.
- count_bit: This counter serves to record the number of bits that have been decoded, so as to judge whether data decoding
- of a frame is finished. When the bit number reaches 39, it indicates decoding is finished.
- Shift Register: This register serves to store binary data after decoding.

12.2.2 Reset MDEC

The analog register afe_0x53 bit[7] serves to reset the MDEC module. To use the MDEC, it's needed to set this bit as 1b'0.

12.2.3 Select Input Channel

User can input the Manchester code from specific GPIO pin into the MDEC.

The analog register afe_0x53 bits[4:0] serves to select PA[0], PB[7], PC[4], PD[0] and PE[0] as input channel, respectively.

12.2.4 Read Result Data

Data after decoding, mdec_data[38:0], are available in the Shift Register, i.e. the analog registers afe_0x6a~afe_0x6e.

After data decoding of a frame is finished, if the 4-bit mdec_data[38:35] in the analog register afe_0x69[7:4]) is consistent with the mdec_match_value written in the analog register afe_0x54[3:0], a MCU wakeup signal will be generated.

12.3 Register Description

MDEC related register are listed in table below.

Table 12-1 MDEC Related Registers

Address	R/W	Description	Default Value
afe_0x53	R/W	[4:0] mdec_sel [0]:PA[0] as mdec input [1]:PB[7] as mdec input [2]:PC[4] as mdec input [3]:PD[0] as mdec input [4:]:PE[0] as mdec input [7]: Mdec_rst,reset mdec module :1, reset enable,0: reset disable	-

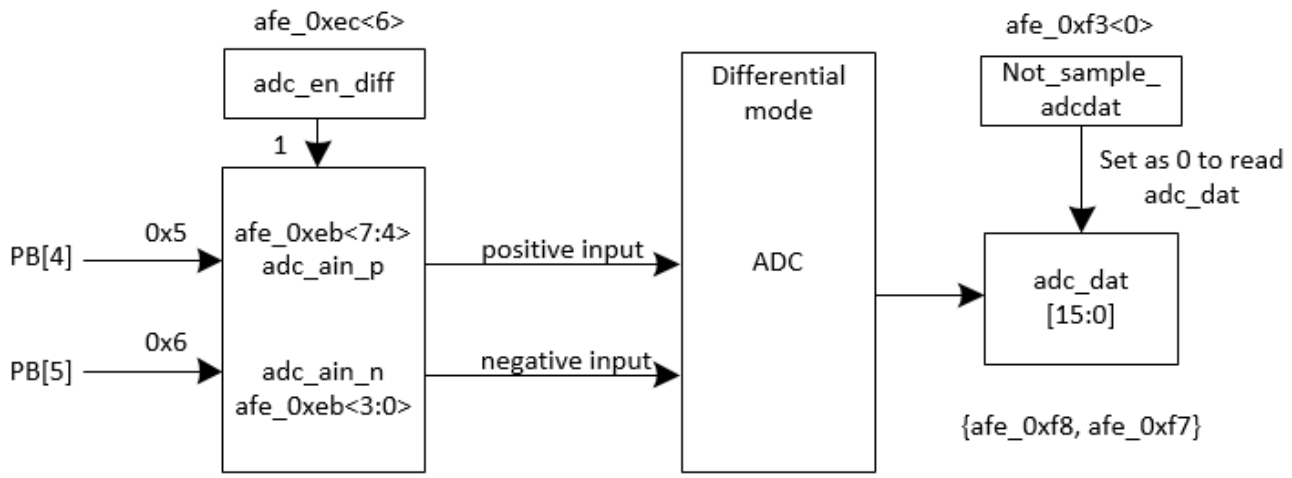
Address	R/W	Description	Default Value
afe_0x54	R/W	MDEC ctrl bit [3:0]: Manchester decoder control bit field value	0x0010
afe_0x6a	R	mdec_rcv_data[38:32] [2:0] mdec_rcv_data[34:32] [3] rsvd [7:4] mdec_rcv_data[38:35]	-
afe_0x6b	R	mdec_rcv_data[31:24] [7:0] mdec_rcv_data[31:24]	-
afe_0x6c	R	mdec_rcv_data[23:16] [7:0] mdec_rcv_data[23:16]	-
afe_0x6d	R	mdec_rcv_data[15:8] [7:0] mdec_rcv_data[15:8]	-
afe_0x6e	R	mdec_rcv_data[7:0] [7:0] mdec_rcv_data[7:0]	-

13 SAR ADC

The SoC integrates one SAR ADC module, which can be used to sample analog input signals such as battery voltage and temperature sensor.

The diagram of SAR ADC module is shown in figure below.

Figure 13-1 Diagram of ADC



13.1 Power On/Down

The SAR ADC is disabled by default. To power on the ADC, the analog register adc_pd (afe_0xfc<5>) should be set as 1b'0.

13.2 ADC Clock

ADC clock is derived from external 24 MHz crystal source, with frequency dividing factor configurable via the analog register adc_clk_div (afe_0xf4<2:0>).

ADC clock frequency (marked as F_{ADC_clk}) = 24MHz/(adc_clk_div+1)

13.3 ADC Control in Auto Mode

13.3.1 Set Max State and Enable Channel

The SAR ADC supports Misc channel which consists of one "Set" state and one "Capture" state.

- The analog register r_max_scnt (afe_0xf2<5:4>) serves to set the max state index. As shown below, the r_max_scnt should be set as 0x02.
- The Misc channel can be enabled via r_en_misc (afe_0xf2<2>).

13.3.2 "Set" State

The length of "Set" state for the Misc channel is configurable via the analog register r_max_s (afe_0xf1<3:0>).

"Set" state duration (marked as T_{sd}) = r_max_s / 24MHz.

Each "Set" state serves to set ADC control signals for the Misc channel via corresponding analog registers, including:

- adc_en_diff: afe_0xec<6>. MUST set as 1b'1 to select differential input mode.
- adc_ain_p: afe_0xeb<7:4>. Select positive input in differential mode.
- adc_ain_n: afe_0xeb<3:0>. Select negative input in differential mode.
- adc_vref: afe_0xe
- a<1:0>. Set reference voltage V_{REF}. ADC maximum input range is determined by the ADC reference voltage.
- adc_sel_ai_scale: afe_0xfa<7:6>. Set scaling factor for ADC analog input as 1 (default), or 1/8.

By setting this scaling factor, ADC maximum input range can be extended based on the V_{REF}.

For example, suppose the V_{REF} is set as 1.2V:

Since the scaling factor is 1 by default, the ADC maximum input range should be 0~1.2V (negative input is GND) / -1.2V~+1.2V (negative input is ADC GPIO pin).

If the scaling factor is set as 1/8, in theory ADC maximum input range should change to 0~9.6V (negative input is GND) / -9.6V~+9.6V (negative input is ADC GPIO pin). But limited by input voltage of the chip's PAD, the actual range is narrower.

- adc_res: afe_0xec<1:0>. Set resolution as 8/10/12/14 bits.

ADC data is always 16-bit format no matter what the resolution is set. For example, 14 bits resolution indicates ADC data consists of 14-bit valid data and 2-bit sign extension bit.

- adc_tsamp: afe_0xee<3:0>. Set sampling time which determines the speed to stabilize input signals.

Sampling time (marked as T_{samp}) = adc_tsamp / F_{ADC_clk}.

The lower sampling cycle, the shorter ADC convert time.

13.3.3 "Capture" State

For the Misc channel, at the beginning of its "Capture" state, a "run" signal is issued automatically to start an ADC sampling and conversion process; at the end of "Capture" state, ADC output data is captured.

- The length of "Capture" state is configurable via the analog register r_max_mc[9:0] (afe_0xf1<7:6>, afe_0xef<7:0>).

  "Capture" state duration for Misc channel (marked as T_{cd}) = r_max_mc / 24MHz.
- The "VLD" bit (afe_0xf6<0>) will be set as 1b'1 at the end of "Capture" state to indicate the ADC data is valid, and this flag bit will be cleared automatically.
- The 16-bit ADC output data can be read from the analog register adc_dat[15:0] (afe_0xf8<7:0>, afe_0xf7<7:0>) while the
- afe_0xf3<0> is set as 1b'0 (default). If the afe_0xf3<0> is set as 1b'1, the data in the afe_0xf8 and afe_0xf7 won't be updated.

NOTE: *The total duration "T_{td}", which is the sum of the length of "Set" state and "Capture" state, determines the sampling rate. Sampling frequency (marked as F_s) = 1 / T_{td}*

13.3.4 Usage Case with Detailed Register Setting

This case introduces the register setting details for Misc channel sampling.

In this case, afe_0xf2<2> should be set as 1b'1, so as to enable the Misc channel, while the max state index should be set as "2" by setting afe_0xf2<5:4> as 0x2.

The total duration (marked as T_{td}) = (1*r_max_s+1*r_max_mc) / 24MHz.

Table 13-1 Overall Register Setting

Function	Register Setting
Power on the ADC	afe_0xfc<5> = 1b'0
Set F_{ADC_clk} (ADC clock frequency) as 4MHz	afe_0xf4<2:0> = 5 F_{ADC_clk} = 24MHz/(5+1) = 4 MHz
Enable the Misc channel	afe_0xf2<2> = 1b'1
Set the max state index as "2"	afe_0xf2<5:4> = 2b'10
Set T_{sd} ("Set" state duration)	afe_0xf1<3:0> = 10 T_{sd} = r_max_s/24 MHz = 10/24 MHz = 0.417 μs
Set T_{cd} ("Capture" state duration)	afe_0xf1<7:6> = 1, afe_0xef<7:0> = 0xea T_{cd} = r_max_mc[9:0]/24 MHz = 490/24 MHz = 20.417 μs
T_{td} (total duration)	T_{td} = (1*r_max_s+1*r_max_mc) / 24 MHz = 500/24 MHz = 20.83 μs
F_s (Sampling frequency)	F_s = 1 / T_{td} = 24 MHz/500 = 48 kHz
Set differential input	afe_0xec<6> = 1
Set input channel	afe_0xeb = 0x56 Select PB[4] as positive input and PB[5] as negative input
Set reference voltage V_{REF}	afe_0xea<1:0> = 2 V_{REF} =1.2V
Set scaling factor for ADC analog input	afe_0xfa<7:6> = 0 scaling factor: 1 ADC maximum input range: -1.2V ~ +1.2V
Set resolution	afe_0xec<1:0> = 3 resolution: 14 bits

Function	Register Setting
Set T_{samp} (determines the speed to stabilize input before sampling)	afe_0xee<3:0> = 3 T_{samp} = adc_tsamp / F_{ADC_clk} = 12/4 MHz = 3 μs

13.4 Battery Voltage Sampling

For battery voltage sampling, set register afe_0xeb<7:4> and afe_0xeb<3:0> to 0xf.

13.5 Register Table

Table 13-2 SAR ADC Registers

Address	Default Value	Description
afe_0xea<1:0>	00	Select V_{REF} for Misc channel 0x0: rsvd 0x1: 0.9V 0x2: 1.2V 0x3: rsvd
afe_0xea<7:2>		rsvd
afe_0xeb<3:0>	0000	Select negative input for Misc channel: 0x0: No input 0x1: B<0> 0x2: B<1> … 0x8: B<7> 0x9: C<4> 0xa: C<5> 0xb: rsvd 0xc: rsvd 0xd: old tempsensor_n (Temperature sensor negative output) 0xe:new tempsensor_n (Temperature sensor negative output) 0xf: Ground

Address	Default Value	Description
afe_0xeb<7:4>	0000	Select positive input for Misc channel: 0x0: No input 0x1: B<0> 0x2: B<1> ... 0x8: B<7> 0x9: C<4> 0xa: C<5> 0xb: rsvd 0xc: rsvd 0xd:old tempsensor_p (Temperature sensor positive output) 0xe: new tempsensor_n (Temperature sensor negative output) 0xf: vbat
afe_0xec<1:0>	11	Set resolution for Misc channel 0x0: 8bits 0x1: 10bits 0x2: 12bits 0x3: 14bits
afe_0xec<5:2>	-	rsvd
afe_0xec<6>	0	Select input mode for Misc channel. 0: rsvd 1: differential mode
afe_0xec<7>	-	rsvd
afe_0xee<3:0>	0000	Number of ADC clock cycles in sampling phase for Misc channel to stabilize the input before sampling: 0x0: 3 cycles 0x1: 6 cycles 0x2: 9 cycles 0x3: 12 cycles ... 0xf: 48 cycles

Address	Default Value	Description
afe_0xef<7:0>	-	r_max_mc[9:0]serves to set length of "capture" state for Misc channel. r_max_s serves to set length of "set" state for Misc channel. Note: State length indicates number of 24M clock cycles occupied by the state.
afe_0xf0<7:0>	-	
afe_0xf1<3:0>	-	
afe_0xf1<5:4>	-	
afe_0xf1<7:6>	-	
afe_0xf2<0>	-	rsvd
afe_0xf2<1>	-	rsvd
afe_0xf2<2>	-	Enable Misc channel sampling. 1: enable
afe_0xf2<3>	0	0: enable write to core 1: disable write to core
afe_0xf2<5:4>	00	Set total length for sampling state machine (i.e. max state index)
afe_0xf2<7>	-	rsvd
afe_0xf3<0>	0	0: sample ADC data to afe_0xf8 and afe_0xf7 1: not sample ADC data to afe_0xf8 and afe_0xf7
afe_0xf3<7:2>	-	rsvd
afe_0xf4<2:0>	011	ADC clock (derive from external 24M crystal) ADC clock frequency = 24M/(adc_clk_div+1)
afe_0xf4<7:3>-	-	rsvd
afe_0xf5<7:0>	-	rsvd
afe_0xf6<0>	-	[0]: vld, ADC data valid status bit (This bit will be set as 1 at the end of capture state to indicate the ADC data is valid, and will be cleared when set state starts.)
afe_0xf6<7:1>	-	rsvd
afe_0xf7<7:0>		Read only [7:0]: Misc adc dat[7:0]
afe_0xf8<7:0>		Read only [7:0]: Misc adc_dat[15:8]
afe_0xf9<1:0>	-	rsvd

Address	Default Value	Description
afe_0xf9<3:2>	0	Vbat divider select sel_vbatdiv[1:0] Vbatdiv 0x0 OFF 0x1 VBAT/4 0x2 VBAT/3 0x3 rsvd
afe_0xf9<5:4>	00	Analog Test Bus (ATB) select sel_atb[1:0] ATB 0x0 NONE 0x1 atb<0> 0x2 atb<1> 0x3 atb<2>
afe_0xf9<7:6>	-	rsvd
afe_0xfa<1:0>	0	Comparator preamp bias current trimming itrim_preamp[1:0] Ibias 0x0 75% 0x1 100% 0x2 125% 0x3 150%
afe_0xfa<3:2>	0	Vref buffer bias current trimming of itrim_vrefbuf[1:0] Ibias 0x0 75% 0x1 100% 0x2 125% 0x3 150%
afe_0xfa<5:4>	0	Vref buffer bias current trimming of itrim_vcmbuf[1:0] Ibias 0x0 75% 0x1 100% 0x2 125% 0x3 150%

Address	Default Value	Description
afe_0xfa<7:6>	0	Analog input pre-scaling select sel_ai_scale[1:0]: scaling factor 0x0: 1 0x1: rsvd 0x2: rsvd 0x3: 1/8
afe_0xfc<4>	0	rsvd
afe_0xfc<5>	1	Power down ADC 1: Power down 0: Power up

14 Temperature Sensor

The SoC integrates a temperature sensor and it's used in combination with the SAR ADC to detect real-time temperature.

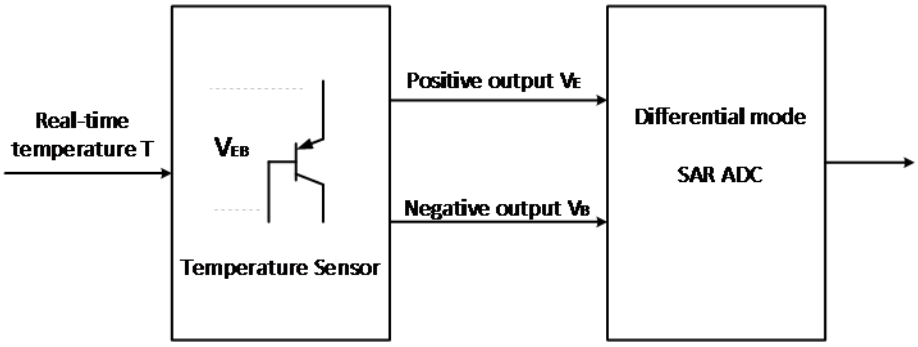
The temperature sensor is disabled by default. The analog register afe_0x06<2> should be set as 1b'0 to enable the temperature sensor.

Table 14-1 Analog Register for Temperature Sensor

Address	R/W	Description	Default Value
afe_0x06	R/W	[0]: Power down of temp sensor: 1: Power down 0: Power up	0x1

The temperature sensor embeds a pnp transistor. It takes the real-time temperature (T) as input, and outputs voltage drop (V_{EB}) signals of pnp transistor as positive and negative output respectively.

Figure 14-1 Block Diagram of Temperature Sensor



The voltage drop V_{EB} signals is determined by the real-time temperature T, as shown below:

$$V_{EB} = 884mV - 1.4286mV/°C * (T - (-40°C))$$
$$= 884mV - 1.4286mV/°C * (T + 40°C)$$

In this formula, "884mV" indicates the value of V_{EB} at the temperature of -40 .

To detect the temperature, the positive and negative output of the temperature sensor should be enabled as the input channels of the SAR ADC. The ADC will convert the V_{EB} signals into digital signal.

The ADC should be configured as differential mode, and the positive and negative output of the temperature sensor should be configured as differential input of the ADC. The ADC should initiate one operation and obtain one output signal (ADCOUT); therefore,

$$V_{EB} = \frac{ADCOUT}{2^N - 1} * V_{REF}.$$

In the formula, "N" and "V_{REF}" indicate the selected resolution and reference voltage of the SAR ADC.

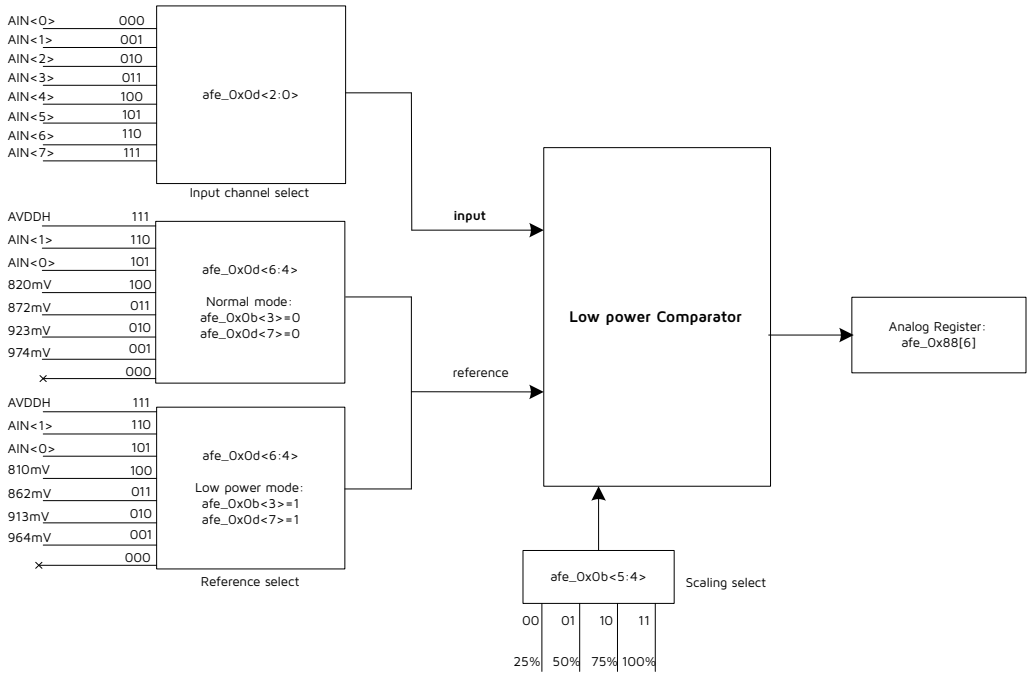
Then the real-time temperature T can be calculated according to the V_{EB}.

15 Low Power Comparator

The SoC embeds a low power comparator. This comparator takes two inputs: input derived from external PortB (PB[1]~PB[7]), and reference input derived from internal reference, PB[0], PB[3], AVDD3 or float.

By comparing the input voltage multiplied by selected scaling coefficient with reference input voltage, the low power comparator will output high or low level accordingly.

Figure 15-1 Block Diagram of Low Power Comparator



15.1 Power On/Down

The low power comparator is powered down by default.

The analog register afe_0x07<3> serves to control power state of the low power comparator: By clearing this bit, this comparator will be powered on; by setting this bit to 1b'1, this comparator will be powered down.

To use the low power comparator, first set afe_0x07<3> as 1b'0, then the 32K RC clock source is enabled as the comparator clock.

15.2 Select Input Channel

Input channel is selectable from the PortB (PB<1> ~ PB<7>) via the analog register afe_0x0d<2:0>.

15.3 Select Mode and Input Channel for Reference

Generally, it's needed to clear both the afe_0x0b<3> and afe_0x0d<7> to select the normal mode. In normal mode, the internal reference is derived from UVLO and has higher accuracy, but current bias is larger (10 μA);

reference voltage input channel is selectable from internal reference of 972 mV, 921 mV, 870 mV and 819 mV, as well as PB<0>, PB<3>, AVDD3 and float.

To select the low power mode, both the afe_0x0b<3> and afe_0x0d<7> should be set as 1b'1. In low power mode, the internal reference is derived from Bandgap and has lower accuracy, but current bias is decreased to 50 nA; reference voltage input channel is selectable from internal reference of 964 mV, 913 mV, 862 mV and 810 mV, as well as PB<0>, PB<3>, AVDD3 and float.

15.4 Select Scaling Coefficient

Equivalent reference voltage equals the selected reference input voltage divided by scaling coefficient.

The analog register afe_0x0b<5:4> serves to select one of the four scaling options: 25%, 50%, 75% and 100%.

15.5 Low Power Comparator Output

The low power comparator output is determined by the comparison result of the value of [input voltage *scaling] and reference voltage input. The comparison principle is shown as below:

- If the value of [input voltage *scaling] is larger than reference voltage input, the output will be low ("0").
- If the value of [input voltage *scaling] is lower than reference voltage input, the output will be high ("1").
- If the value of [input voltage *scaling] equals reference voltage input, or input channel is selected as float, the output will be uncertain.

User can read the output of the low power comparator via the analog register afe_0x88[6].

The output of the low power comparator can be used as signal to wakeup system from low power modes.

15.6 Register Description

Table 15-1 Analog Register Related to Low Power Comparator

Address	R/W	Description	Default Value
afe_0x06	R/W	[1]: Power down of low current comparator: 1: Power down 0: Power up	0x1

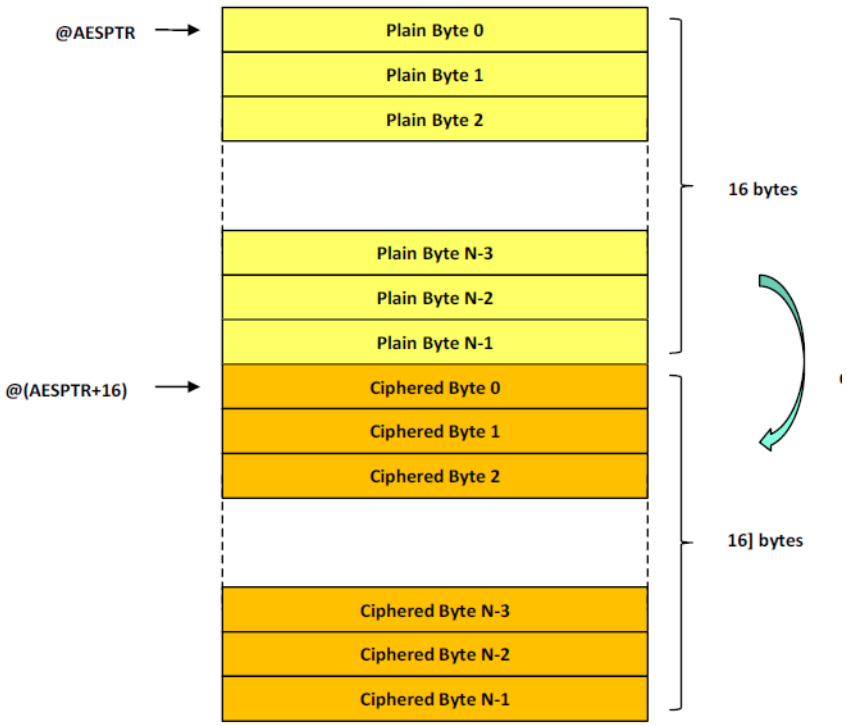
Address	R/W	Description	Default Value
afe_0x0b	R/W	[3]: comp_ref_mod_sel reference mode select: 1: ref from BG 0: ref from UVLO [5:4]: comp_refscale<1:0> reference scale select: 11: 1 10: 3/4 01: 1/2 00: 1/4	-
afe_0x0d	R/W	[2:0]: lc_comp_chsel<2:0> channel select of lc comparator: 0x000 -> AIN<0> ... 0x111 -> AIN<7> [3]: lc_comp_vbus_inen inner detect point enable: 1: enable 0: disable [6:4]: lc_comp_refsel<2:0> channel select of lc comparator: 0x000 -> float 0x001 -> 974mV 0x010 -> 923mV 0x011 -> 872mV 0x100 -> 820mV 0x101 -> AIN_in<0> 0x110 -> AIN_in<1> 0x111 -> avddh [7]: lc_comp_pd_10u power down of 10u current to voltage reference 1: power down 0: active	-

16 AES

The SoC embeds AES module with encryption and decryption function. The input 128-bit plaintext in combination of key is converted into the final output ciphertext via encryption; the 128-bit ciphertext in combination of key can also be converted into 128-bit plaintext via decryption.

Software stores the block to encrypt, at AESPTR address in the SRAM. The block size to encrypt is always considered to equal 128-bit. Software defines the input key setting its value in AESKEY<> registers. Once the settings done, Software starts AES-128 use by writing a 1 in AES_START. On normal termination, the Software receives a crypt_irq. When the process ends the Software can find encrypted data at AESPTR+16 address as shown in figure below (considering byte address memory).

Figure 16-1 AES Address



AES related registers are listed as following:

Table 16-1 AES Registers

Address	Type	Description	Reset Value
0xb0	R/W	Bit[0] AES_START	0x00
0xb4	R/W	[31:0] KEY0	0x00
0xb8	R/W	[31:0] KEY1	0x00
0xbc	R/W	[31:0] KEY2	0x00
0xc0	R/W	[31:0] KEY3	0x00
0xc4	R/W	[15:0] AESPTR	0x00

17 Public Key Engine (PKE)

The SoC embeds Public Key Engine Standard Performance acceleration module and this section describes its function and use.

17.1 Calculation Model Overview

Public Key Engine (PKE) is specifically designed to accelerate large digital-to-analog operations in public key cryptographic operations. PKE SP-ECC is a version optimized for the elliptic curve algorithm. In this version, the following features are available.

- Support different bit width ECC (prime field): 192, 256 bits
- Support curve parameters: NIST p192, NIST p256, X25519, EdDSA

17.2 Function Description

17.2.1 Module Description

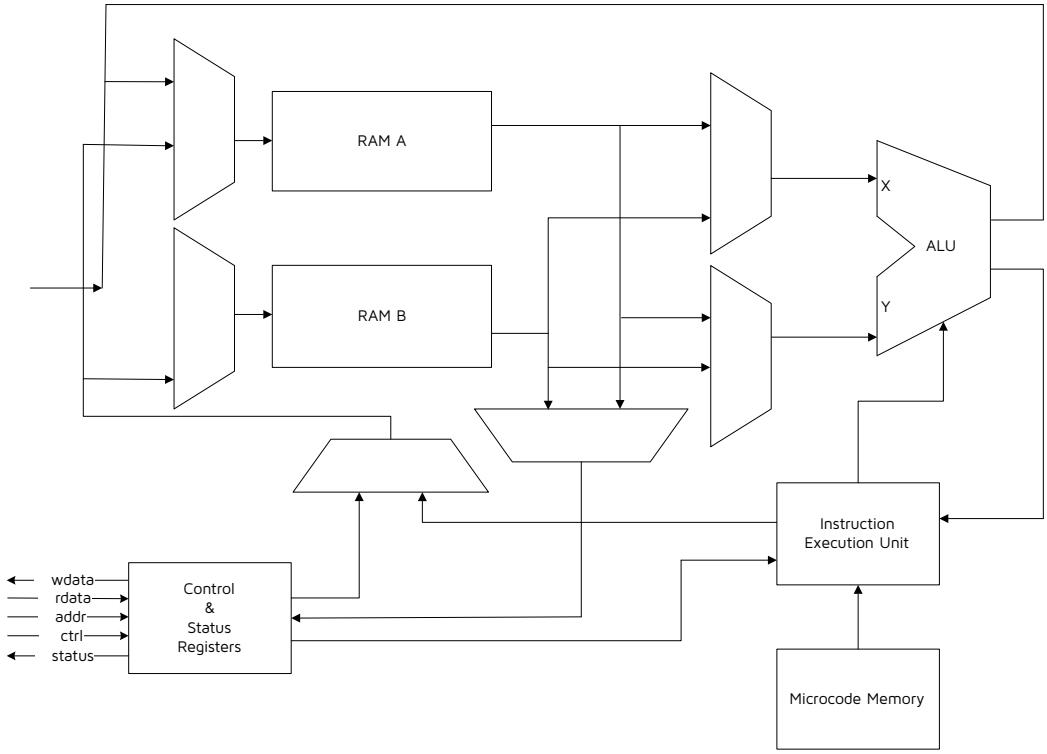
There are a large number of large digital-to-analog operations in public key cryptographic operations. PKE is designed to accelerate large digital-to-analog operations involved in RSA and Elliptic Curve Cryptography (ECC) operations in public key cryptography. Recently PKE can directly complete modular exponentiation in RSA and point multiplication in ECC. The CPU can query the operation of the PKE by polling or interrupting. The PKE includes one program memory unit (ROM), one instruction arithmetic unit (IEU), one 32-bit arithmetic unit (ALU), two pseudo-double-ended data RAMs, one register combination with interface module.

According to different register configurations, PKE can perform the following operations with different precisions:

- ECC (Prime field): 192 and 256 bits

In addition, the calculation of the PKE is finished in the form of Microcode and the Microcode is stored in the program storage unit. Therefore, different kind of public key cryptographic calculations can be implemented by pouring different microcode into the program storage unit. For instance, a high security public key algorithm instruction can be injected into a program storage unit in the PKE module in a SoC with high security requirements. Certainly these arithmetic instructions can be written to the ROM with a large program memory unit capacity. The CPU makes real-time calls according to different usage scenarios. The full microcode size is approximately 2 KB.

Figure 17-1 Block Diagram of PKE SP Module



17.2.2 Software Interface (Programming Model)

The interfaces of the PKE SP are all mapped into the 7KB address space. The block of address mapping space mainly contains all the operands that the CPU can access. These operands contain modulus, power exponents, partial intermediate variables, and so on. In addition to this, the address map also contains control and status registers. The CPU can configure and monitor the PKE module through these control and status registers.

In the operations supported by PKE, the operands are also 192 bits at minimum. Therefore, it will encounter the problem of big-endian and little-endian when putting data into data RAM in the CPU or DMA. In the PKE module, words are arranged following an order of little-endian.

In PKE, the smallest operand is 32 bits (1 word), because the current ALU bit width input is 32 bits. If the operand is not word aligned, the high bit needs to be filled as 0.

After the PKE receives the start command, it starts the operation. During the operation, the host computer can query the current running state through the status register, or interrupt the current operation through the control register. In addition, the result of partial intermediate operations can be obtained by accessing the data RAM address.

The host computer can obtain the result of target operation finish by PKE through polling or interrupting. Data RAM supports word aligned and does not support byte alignment.

Table 17-1 Dual Port Ram Address Map

Operand First Address	ECC		
	256 bits	**512 bits**	**1024 bits**
A0	0x0400	0x0400	0x0400
A1	0x0424	0x0444	0x0484
A2	0x0448	0x0488	0x0508
A3	0x046C	0x046C	0x058C
A4	0x0490	0x0510	0x0610
A5	0x04B4	0x0554	0x0694
A6	0x04D8	0x0598	0x0718
A7	0x04FC	0x05DC	0x079C
A8	0x0520	0x0620	0x0820
A9	0x0544	0x0664	0x08A4
B0	0x1000	0x1000	0x1000
B1	0x1024	0x1044	0x1084
B2	0x1048	0x1088	0x1108
B3	0x106C	0x10CC	0x118C
B4	0x1090	0x1110	0x1210
B5	0x10B4	0x1154	0x1294
B6	0x10D8	0x1198	0x1318
B7	0x10FC	0x11DC	0x139C
B8	0x1120	0x1220	0x1420
B9	0x1144	0x1264	0x14A4

The above table shows the address assignment of two RAMs in ECC mode. The operand registers are distributed in two blocks of data RAM, using the prefixes A and B to distinguish the two blocks of RAM. The addresses listed in the table are all CPU addressable addresses, RAM A has an address offset of 0x400, and RAM B has an address offset of 0x1000. The actual space used by RAM will be larger than the space listed in the table and some intermediate variable storage is not open to the CPU.

Data will be stored in the mode of little-endian in RAM.

17.3 Register Description

The following registers' base address is 0x110000.

Table 17-2 PKE Related Registers

Offset	R/W	Description	Default Value
0x00	W1S	PKE_CR [0] Go Start signal. When write 1 to the byte, the PKE will start running in the next clock cycle. The operation of the PKE is based on the configuration of the control registers and data registers for that clock cycle written as 1. [7:1] Rsvd	0x00
0x02	W1S	PKE_CR2 [16] Stop Stop signal. When write 1 to the byte, PKE will stop in the next clock cycle. [23:17] Rsvd	0x00
0x05	RW	PKE_CFG1 [8] IRQEN Interrupt enable. When the bit is set as 1, the o_irq interface is valid. Regardless of whether the bit is set as 1, the STAT register is not affected by it. [15:9] Rsvd	0x00
0x06	RW	PKE_CFG2 [23:16] Partial_Radix Select part of BASE_RADIX to determine the bit width that the operation really needs to use during the operation. The value of this field indicates the number of words, and the bit width of the operand is PARTICAL_RADIX*32 bits. For example, if BASE_RADIX=2, PARTIAL_RADIX=6, then the bit width of the operand is (6/ (256/32))*256=192. If the operations of ECC-192 need to be processed, BASE_RADIX and PARTIAL_RADIX should be configured as shown in this example. When using operands of other bit widths, configure BASE_RADIX and PARTIAL_RADIX according to the above formula.	0x00

Offset	R/W	Description	Default Value
0x07	RW	PKE_CFG3 [31:27] Rsvd [26:24] Base_Radix This field indicates the bit width cardinality at which the operation is performed. At the same time, the cardinality also represents the space required for the operand to be stored in the data RAM. For ECC point operations, the value of this field should be 2. 2: 256 bits Others: Reserved	0x02
0x10	RW	MC_PTR0 [7:0] ADDR This field indicates the address of the next instruction to be executed by the PKE. This register can only be rewritten when the PKE is not working. Any write operation while the PKE is operating will be ignored. This field is also updated in real time when running the PKE and always pointing to the address of the instruction that will be executed next. Therefore, this register can also be combined with CTRL.STOP for debugging. It should be noted that the instructions are all word aligned. Therefore, the lowest 2 bits of the field are 0. When writing an instruction address to this field, it is limited to the address range of 0x00~0x2F. The written address will proceed "And" Operation with a mask, therefore ignoring the upper 6 bits.	0x00
0x11	RW	MC_PTR1 [11:8] ADDR See above description for [7:0] [15:12] Rsvd	0x00

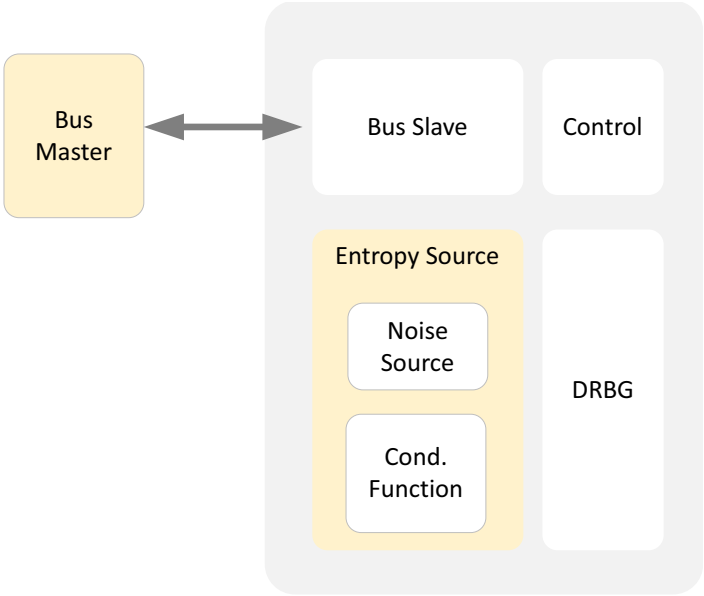
Offset	R/W	Description	Default Value
0x20	W1C	STAT [0] Done When the bit is set to 1, it indicates that the operation ends. When this bit is set as 1 from external, the bit is cleared. In addition, this bit also acts as a clear bit for the external interrupt. When the bit is high as CTRL.IRQEN is active, the external interrupt signal is also pulled high. To write 1 from external, the external interrupt is also cleared. [7:1] Rsvd	0x00
0x24	R	RT_CODE This field is used to indicate the reason for PKE stop. If PKE is stopped because the operation is completed, then the value of this field is 0. If the value of this field is non-zero, the operation of PKE has not been completed and some exceptions have been encountered, which require external processing and the results are not available. [0]: Normal stop [1]: Termination request received (CTRL.STOP is high) [2]: No valid modular inverse result [3]: Point is not on the curve (CTRL.CMD: PVER) [4]: Invalid Microcodeothers: Reserved	0x00
0x50	RW	EXE_CONF0	0x2a
0x51	R	EXE_CONF1	0x00
0x80	R	PKE_RBG_VERSION0 [3:0]:Sub version number. [7:4]:Main version number	0x00
0x82	R	PKE_RBG_VERSION2 PROJECT number low	0x00
0x83	R	PKE_RBG_VERSION3 PROJECT number high	0x00

18 True Random Number Generator (TRNG)

18.1 Model Overview

True random number generator module contains entropy source and post processing (Deterministic Random Bit Generators, DRBG). The entropy source is designed using RO. The top block diagram of the random number generator is shown below.

Figure 18-1 Module Boundary



18.2 Interrupt Description

The Random Bit Generator (RBG) module has the following interrupt sources:

- CPU reads RBG_DR without data
- Data valid

The above interrupts can be set by RBG_CR. By default, the data valid interrupt is enabled.

When the RBGEN of RBG_CR is low, the interrupt signal will not be cleared. Therefore, before enabling RBGEN, it is necessary to ensure that there is no previous interrupt signal, otherwise it will affect the next interrupt.

18.2.1 CPU Reads RBG_DR without Data

In order to prevent the CPU from reading the invalid data, the RBG can remind the CPU to read in such a situation when there is no valid random number. In order to avoid the CPU reading the empty data, it is recommended to read the RBG_FIFO_SR first every time to get the random number before the CPU gets data in the current FIFO to avoid invalid data.

The CPU can clear the interrupt by writing 1 to ERERR in RBG_SR. If the write is successful, the interrupt will be cleared. When the above situation occurs again, the interrupt will be valid again.

18.2.2 Data Valid

RBG provides two ways to output data. When the interrupt is enabled, the random number can be read by the way of interrupting. In this design, the data in the corresponding FIFO will only be pulled up after the threshold is reached, thus the CPU can obtain multiple data at once. The threshold can be set by RBG_FIFO_CR. The CPU can clear the interrupt by writing 1 to DRDY of RBG_SR. If the write is successful, the interrupt will be pulled down. The interrupt is pulled high again when the data in the FIFO reaches the threshold again.

It is important to note that the interrupt will only be pulled up when the amount of data in the FIFO reaches the threshold. Therefore, the data in the FIFO exceeds the threshold firstly and then RBG module pulls up the interrupt. When the CPU doesn't obtain data or have obtained data but the amount of data remaining in the FIFO is still larger than the threshold, then clear the interrupt. Although the data in the FIFO is still larger than the threshold, it will not be interrupted.

In addition, the CPU can use the RBG_FIFO_SR register to view the remaining data in the FIFO. It can also use this method to obtain a random number. Check the RBG_FIFO_SR register when the random number is needed and the number of random numbers indicated by the register can be fetched at one time. If the rate at which the CPU handles random numbers is slower than the rate at which RBG random numbers are generated, it is generally not recommended to use interrupt to obtain random numbers.

18.3 Usage Procedure

18.3.1 Normal Operation

Turn off the RBG module first after the CPU works normally, that is to set RBGEN of the RBG_CR to 0. Then it can be configured and write 1 to RBGEN after the configuration is complete to make it work normally.

The CPU can configure RBG module by configuring RBG_CR, RBG_FIFO_CR and other optional configuration registers.

When writing 1 to RBGEN in RBG_CR, the modification of the value of the above register will not affect the RBG. Therefore, when configuring, set the RBGEN in the RBG_CR register after configuring other registers to enable the OSR_RBG module.

TRBG and DRBG can be switched by modifying RBG_RTCR during the operation to meet different usage environments.

18.3.2 Entropy Source

In this design, the random number generator module uses RO RNG as the entropy source. RO RNG contains modules such as random source and post-processing. RO RNG has four independent RO entropy sources. Each entropy source can choose to use its own RO CLK as the sampling clock or select the system clock as the sampling clock. The selection is determined by the input of I_rbg_sclk_sel, which is high for the system clock and low for the internal RO CLK. All RO enable signals are open at the same time and some of the ROs can be turned on or off by controlling the register.

18.4 Register Description

TRNG related registers are listed in the following table. The base address for the following registers is 0x101800.

Table 18-1 Register Description

Offset	R/W	Description	Default Value
0x00	RW	TRNG_CR0 [0]: Random bit generator enable. [1]: Each bit states enable for one RO SOURCE0 [2]: Each bit states enable for one RO SOURCE1 [3]: Each bit states enable for one RO SOURCE2 [4]: Each bit states enable for one RO SOURCE3	0x1f
0x04	RW	TRNG_RTCR [0]: Mode select.0: TRBG without post-processing.1: TRBG with post-processing	0x00
0x08	R	RBG_SR [0]: Data ready.	0x00
0x0c	R	RBG_DR0 rbg data	-
0x0d	R	RBG_DR1 rbg data	-
0x0e	R	RBG_DR2 rbg data	-
0x0f	R	RBG_DR3 rbg data	-
0x10	R	RBG_VERSION0 [3:0]: Sub version number. [7:4]: Main version number	0x01
0x12	RO	RBG_VERSION2 PROJECT number low	0x3a
0x13	RO	RBG_VERSION3 PROJECT number high	0xef
0x80	RW	RO_CR1_0 RO enable of RO SOURCE1.Each bit controls one RO. In total, there are 16 ROs in RW RO SOURCE 1.	0xff
0x81	RW	RO_CR1_1 RO enable of RO SOURCE1.Each bit controls one RO. In total, there are 16 ROs in RW RO SOURCE 1.	0xff

Offset	R/W	Description	Default Value
0x82	RW	RO_CR0_0 RO enable of RO SOURCE0.Each bit controls one RO. In total, there are 16 ROs in RW RO SOURCE 0.	0xff
0x83	RW	RO_CR0_1 RO enable of RO SOURCE0.Each bit controls one RO. In total, there are 16 ROs in RW RO SOURCE 0.	0xff
0x84	RW	RO_CR3_0 RO enable of RO SOURCE3.Each bit controls one RO. In total, there are 16 ROs in RW RO SOURCE 3.	0xff
0x85	RW	RO_CR3_1 RO enable of RO SOURCE3.Each bit controls one RO. In total, there are 16 ROs in RW RO SOURCE 3.	0xff
0x86	RW	RO_CR2_0 RO enable of RO SOURCE2.Each bit controls one RO. In total, there are 16 ROs in RW RO SOURCE 2.	0xff
0x87	RW	RO_CR2_1 RO enable of RO SOURCE2.Each bit controls one RO. In total, there are 16 ROs in RW RO SOURCE 2.	0xff
0x88	RW	FSEL [1:0]:RO sampling clock frequency division selection. 00: 4 frequency division, 01: 8 frequency division, 10: 16 frequency division, 11: 32 frequency division	0x03