Datasheet for Telink BLE + IEEE802.15.4 Multi-Standard Wireless SoC TLSR8258F1KET48A

DS- TLSR8258F1KET48A-E7

Ver 1.0.2

2024/1/11

Keyword:

BLE; BLE Mesh; 6LoWPAN; Zigbee; RF4CE; HomeKit; 2.4GHz

Brief:

This datasheet is dedicated for Telink BLE + IEEE802.15.4 multi-standard SoC TLSR8258F1KET48A. In this datasheet, key features, working mode, main modules, electrical specification and application of the TLSR8258F1KET48A are introduced.

Published by
Telink Semiconductor

Bldg 3, 1500 Zuchongzhi Rd,
Zhangjiang Hi-Tech Park, Shanghai, China

© Telink Semiconductor
All Right Reserved

Legal Disclaimer

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:

telinksales@telink-semi.com

telinksupport@telink-semi.com

Revision History

Version	Major Changes	Date	Author
0.8.0	Preliminary release	2018/05	YCQ, SGJ, LY, JJW, SY, LWF, LWT, HZT, Cynthia
0.8.3	Updated the sections below: 1.4 Ordering information, 7.1.1.1 GPIO lookup table, 7.3.4 I2C Master mode, 16.4 AC characteristics, 17 Reference Design.	2018/11	SY, HZF, YCQ, YY, Cynthia
0.8.6	Updated the sections below: 1.1 Block diagram, 1.2 Key features, 1.4 Ordering information, 1.6 Pin layout, 2.1.3 E-Fuse, 7.1.1.1 GPIO lookup table, 9.3.1 Rate Matching, 11.3.2 "Set" state, 17 Reference Design.	2019/05	JJW, YCQ, SY, LWT, HZT, LX, Cynthia
0.8.9	Updated the sections below: 1 Overview: Removed Thread related information 1.4 Ordering information 2.2 Firmware encryption 16.3 DC characteristics 16.4 AC characteristics 16.6 I2C characteristics 17 Reference Design	2020/01	SY, CLH, YH
1.0.0	Removed ANT mode, LDO and E-fuse information, updated chapter 14 low power comparator.	2022/09	LC
1.0.1	For SPI interface, change full-duplex to half-duplex. For 2.4GHz proprietary payload length, change to 1~63 bytes. Updated the note for the reference schematic. Updated frequency range in Table 16-4	2023/06	LC
1.0.2	Deleted temperature sensor; updated 1.6.2 Notes, 2.6.4 VBAT and VANT description	2024/1	LC

1 Table of Contents

2   Table of Figures

3   Table of Tables

1 Overview

The TLSR8258F1KET48A is Telink-developed Bluetooth LE + IEEE802.15.4 multi-standard wireless SoC solution with internal Flash and audio support, which combines the features and functions needed for all 2.4GHz IoT standards into a single SoC. It's completely RoHS-compliant and 100% lead (Pb)-free.

The TLSR8258F1KET48A combines the radio frequency (RF), digital processing, protocols stack software and profiles for multiple standards into a single SoC. The chip supports standards and industrial alliance specifications including Bluetooth Low Energy, BLE Mesh, 6LoWPAN, Zigbee, RF4CE, HomeKit and 2.4GHz proprietary standard. The TLSR8258F1KET48A's embedded FLASH enables dynamic stack and profile configuration, and the final end product functionality is configurable via software, providing ultimate flexibility. The TLSR8258F1KET48A also has hardware OTA upgrades support and multiple boot switching, allowing convenient product feature roll outs and upgrades.

The TLSR8258F1KET48A supports concurrent multi-standards. For some use cases, the TLSR8258F1KET48A can "concurrently" run two standards, for example, stacks such as BLE and 802.15.4 can run concurrently with one application state but dual radio communication channels for interacting with different devices. The end product working in this mode can maintain active Bluetooth Smart connections to smart phones or other BLE devices while control and communicate with 802.15.4 or other 2.4GHz devices at the same time. In this case, it's compatible with Bluetooth standard, supports BLE specification, allows easy connectivity with Bluetooth Smart Ready mobile phones, tablets, laptops, which supports BLE slave and master mode operation, including broadcast, encryption, connection updates, and channel map updates. At the same time, it also supports IEEE 802.15.4 standard and Zigbee-compliant platform, and is perfect for creating interoperable solution for use within the home combined with leading Zigbee/RF4CE software stack. This feature enables products to bridge the smartphone and home automation world with a single chip and no requirement for an external hub.

The TLSR8258F1KET48A integrates hardware acceleration to support the complicated security operations required by HomeKit, and other standards without the requirement for an external DSP, thereby significantly reducing the product eBOM.

The TLSR8258F1KET48A supports single or dual analog microphones or digital microphone, and stereo audio output with enhanced voice performance for voice search and other such applications. The TLSR8258F1KET48A also includes a full range of on-chip peripherals for interfacing with external components such as LEDs, sensors, touch controllers, keyboards, and motors. This makes it an ideal single-chip solution for IoT (Internet of Things) and HID (Human Interface Devices) application such as wearable devices, smart lighting, smart home devices, advanced remote controls, and wireless toys.

1.1 Block diagram

The TLSR8258F1KET48A is designed to offer high integration, ultra-low power application capabilities. The system's block diagram is as shown in Figure 1-1.
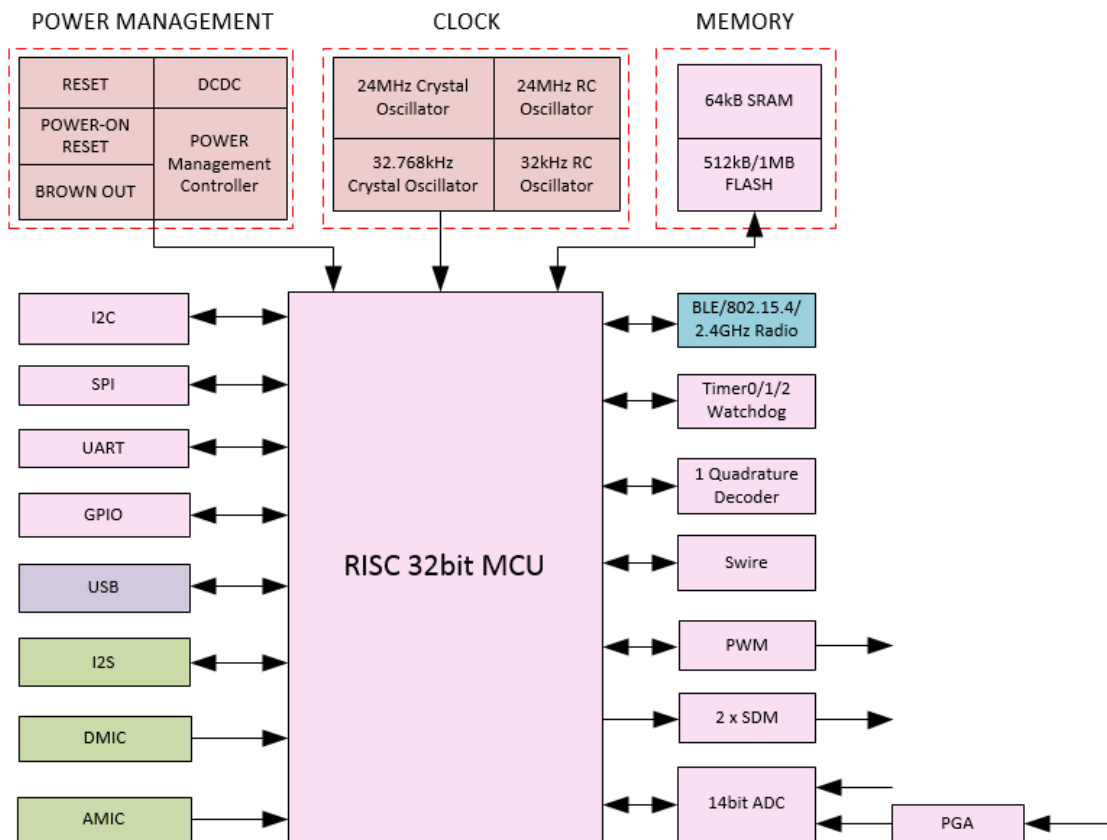
Figure 1- 1   Block diagram of the system

*Note:

1) Modules marked with different colors belong to different power domains. Power state of each power domain can be controlled independent of other power domains, for example, the audio module (including I2S, DMIC, AMIC) can be independently powered on or powered down irrespective of other modules such as power management module, clock, and etc.

2) The BLE/802.15.4/2.4GHz Radio, USB and Audio (I2S, DMIC, AMIC) are powered down by default.

3) The power management module and clock should be always powered on, even in deep sleep.

4) In deep sleep, except for the power management and clock, all other modules should be powered down.

The TLSR8258F1KET48A integrates a power-balanced 32-bit MCU, BLE/802.15.4/2.4GHz Radio, 64kB (16k+8k+8k+32k) SRAM, 1MB internal Flash, 14bit ADC with PGA, analog and digital microphone input, stereo audio output, 6-channel PWM (1-channel IR/IR FIFO/IR DMA FIFO), one quadrature decoder (QDEC), abundant and flexible GPIO interfaces, and nearly all the peripherals needed for IoT (Internet of Things) and HID (Human Interface Devices) application development (e.g. Bluetooth Low Energy and Zigbee/IEEE 802.15.4/RF4CE). The TLSR8258F1KET48A 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 TLSR8258F1KET48A, 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) 4-byte Chip UID (Unique ID).

2) Embedded 32-bit proprietary microcontroller.

   ✧ Better power-balanced performance than ARM M0

   ✧ Instruction cache controller

   ✧ Maximum running speed up to 48MHz

3) Program memory: internal 1MB Flash.

4) Data memory: 64kB on-chip SRAM, including up to 32kB SRAM with retention in deep sleep, and one 32kB SRAM without retention in deep sleep.

5) RTC and other timers:

   ✧ Clock source of 24MHz&32.768kHz Crystal and 32kHz/24MHz embedded RC oscillator

   ✧ Three general 32-bit timers with four selectable modes in active mode

   ✧ Watchdog timer

   ✧ A low-frequency 32kHz timer available in low power mode

6) A rich set of I/Os:

   ✧ Up to 32 GPIOs depending on package option. All digital IOs can be used as GPIOS.

   ✧ DMIC (Digital Mic).

   ✧ AMIC (Analog Mic).

   ✧ I2S.

   ✧ Stereo Audio output.

   ✧ SPI.

   ✧ I2C.

   ✧ UART with hardware flow control and 7816 protocol support.

   ✧ USB.

   ✧ Swire debug Interface.

7) Up to 6 channels of differential PWM:

   ✧ PWM1~PWM5: 5-channel normal PWM output.

   ✧ PWM0: 1 channel with IR/IR FIFO/IR DMA FIFO mode for IR generation.

8) Sensor:

   ✧ 14bit 10-channel (only GPIO input) SAR ADC, with 4-channel differential input PGA

   ✧ Temperature sensor

9) One quadrature decoder.

10) Embedded hardware AES and AES-CCM.

11) Embedded hardware acceleration for Elliptical curve cryptography (ECC)

12) Embedded low power comparator.

13) Operating temperature range: -40℃~+85℃

14) Supports all 2.4GHz IoT standards into a single SoC, including BLE, BLE Mesh, Zigbee, RF4CE, HomeKit, 6LowPAN, and 2.4GHz proprietary technologies without the requirement for an external DSP.

1.2.2   RF Features

RF features include:

1) BLE/802.15.4/2.4GHz RF transceiver embedded, working in worldwide 2.4GHz ISM band.

2) Bluetooth LE 1Mbps and 2Mbps, Long Range 125kbps and 500kbps.

3) IEEE802.15.4 compliant, 250kbps.

4) 2.4GHz proprietary 1Mbps/2Mbps/250kbps/500kbps mode with Adaptive Frequency Hopping feature support.

5) Rx Sensitivity: -96dBm@BLE 1Mbps, -99.5dBm@ IEEE802.15.4 250kbps, -93dBm @ BLE 2Mbps mode, -99dBm @ BLE 500kbps mode, -101dBm @ BLE 125kbps mode.

6) Tx output power: up to +10dBm.

7) Single-pin antenna interface.

8) RSSI monitoring with +/-1dB resolution.

9) Auto acknowledgement, retransmission and flow control.

10) Support full-function BLE location features.

1.2.3   Features of power management module

Features of power management module include:

1) Embedded DCDC.

2) Battery monitor: Supports low battery detection.

3) Power supply: 1.8V~3.6V.

4) Multiple stage power management to minimize power consumption.

5) Low power consumption:

   ✧ Whole Chip RX mode: 5.3mA

   ✧ Whole Chip TX mode: 4.8mA @ 0dBm with DCDC

   ✧ Deep sleep with external wakeup (without SRAM retention): 0.4uA

   ✧ Deep sleep with SRAM retention: 1uA (with 8kB SRAM retention), 1.2uA (with 16kB SRAM retention), 1.4uA (with 32kB SRAM retention)

1.2.4   USB features

USB features include:

1) Compatible with USB2.0 Full speed mode.

2) Supports 9 endpoints including control endpoint 0 and 8 configurable data endpoints.

3) Independent power domain.

4) Supports ISP (In-System Programming) via USB port.

1.2.5   Flash features

The TLSR8258F1KET48A embeds Flash with features below:

1) Total 1M bytes (8Mbits).

2) Flexible architecture: 4kB per Sector, 64kB/32kB per block.

3) Up to 256 Bytes per programmable page.

4) Write protect all or portions of memory.

5) Sector erase (4kB).

6) Block erase (32kB/64kB).

7) Cycle Endurance: 100,000 program/erases.

8) Data Retention: typical 20-year retention.

9) Multi firmware encryption methods for anti-cloning protection.

1.2.6   RF4CE features

RF4CE features include:

1) Based on IEEE 802.15.4 Standard, certified RF4CE platform, with ZRC1.1/ZRC2.0 and MSO profile support.

2) Various transmission options including broadcast.

3) Provides a secured key generation mechanism.

4) Supports a simple pairing mechanism for devices with full application confirmation.

5) Only authorized devices are able to communicate.

6) Various power saving modes are supported for all device classes.

7) Supports AES-128bit encryption and AES-CCM (Counter with the CBC-MAC) mode.

8) Extensible to vendor specific profiles.

9) Telink extended profile with audio support for voice command based searches.

10) Over the air (OTA) firmware upgrade with hardware support.

1.2.7 Zigbee features

Zigbee features include:

1) Based on IEEE 802.15.4 Standard, certified Zigbee Pro and Zigbee 3.0 platform, with ZHA/ZLL profile and Zigbee 3.0 device support.

2) Uses multi-hop mesh networking to eliminate single points of failure and expand the reach of networks.

3) Allow low power operation, even support the Green Power feature.

4) Supports networks of thousands of nodes, providing a networking for the smart home or the smart city.

5) Uses a variety of security mechanisms, such as AES-128 encryption, device and network keys and frame counters.

6) Include all application level functionality of ZigBee Smart Energy.

7) Support seamless interoperability with a wide variety of smart devices.

8) Over the air (OTA) firmware upgrade with hardware support.

1.2.8 6LowPAN features

6LoWPAN features include:

1) Supports 6LowPAN, IPv6 and DHCPv6.

2) Supports UDP and DTLS.

3) Supports networks of 250 nodes or greater.

1.2.9 BLE features

BLE features include:

1) Qualified for Bluetooth 5.3, main features supported include:

   ✧ 1Mbps, 2Mbps, Long Range S2 (500Kbps), S8 (125Kbps)

   ✧ High duty cycle non-connectable ADV

   ✧ Extended ADV

   ✧ LE Channel selection algorithm #2

2) Bluetooth SIG Mesh support

1.2.10 BLE Mesh features

BLE Mesh features include:

1) Compatible with Bluetooth SIG Mesh specification 1.0, with additional features from Telink enhanced design.

2) Support flexible mesh control, e.g. N-to-1 and N-to-M.

3) Supports switch control for over 200 nodes without delay.

4) Supports real time status update for over 200 nodes.

5) Secure and safe control and scalable identification within network.

6) 8/16 groups can be controlled at the same time.

7) 128/256 nodes within mesh network.

8) Configurable to more or fewer hops (e.g. 4 hops) within mesh network, single hop delay less than 15ms.

9) Flexible RF channel usage with both BLE advertising channels and data channels for good anti-interference performance.

1.2.11 Concurrent mode feature

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

Typical combination is Bluetooth LE + 802.15.4 based standard (e.g. Zigbee 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 stack will run alternately during the divided time slots.

1.2.12 HomeKit features

HomeKit features include:

1) Single-chip solution with hardware acceleration for all HomeKit security operations

2) Apple (pre-)certified Software Development Kit reference design

3) Conformant to latest HomeKit specification (HAP v2.0)

4) Tested against Apple HomeKit Accessory Tester and Apple latest-version 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.

1.3 Typical applications

The TLSR8258F1KET48A can be applied to IoT (Internet of Things) and HID (Human Interface Devices) applications, such as BLE smart devices, BLE mesh devices, 6LoWPAN home automation devices, 2.4GHz IEEE 802.15.4, RF4CE remote control /set-top box, and Zigbee systems. Its typical applications include, but are not limited to the following:

◇ Smartphone and tablet accessories;

◇ RF Remote Control;

◇ Sports and fitness tracking;

◇ Wearable devices;

◇ Wireless toys;

$\diamond$ Smart Lighting, Smart Home devices;

$\diamond$ Building Automation;

$\diamond$ Smart Grid;

$\diamond$ Intelligent Logistics/Transportation/City;

$\diamond$ Consumer Electronics；

$\diamond$ Industrial Control;

$\diamond$ Health Care.

1.4 Ordering information

Table 1- 1 Ordering information of the TLSR8258F1KET48A*[1]

Product Series	Package Type	Temperature Range	Product Part No.	Packing Method*[2]	Minimum Order Quantity
TLSR8258F1KET 48A	48-pin TQFN 7x7x0.75mm	-40℃~+85℃	TLSR8258F1KE T48AER	TR	3000

1.5 Package

Package dimensions for the TLSR8258F1KET48A is shown as below.

---

[1] MSL (Moisture Sensitivity Level): The TLSR8258F1KET48A is applicable to MSL3 (Based on JEDEC Standard J-STD-020).

$\diamond$ After the packing opened, the product shall be stored at <30℃/ <60%RH and the product shall be used within 168 hours.

$\diamond$ When the color of the indicator in the packing changed, the product shall be baked before soldering.

$\diamond$ If baking is required, please refer to IPC/JEDEC J-STD-033 for baking procedure.

[2] Packing method "TR" means tape and reel. The tape and reel material DO NOT support baking under high temperature.
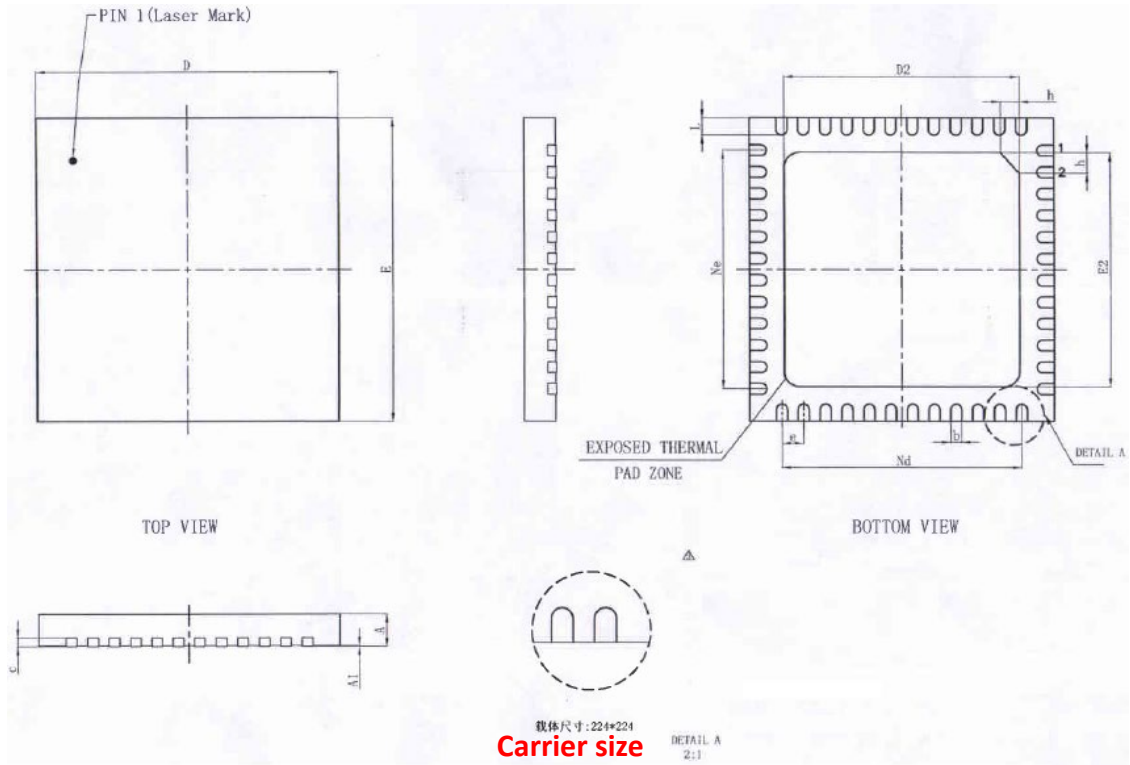
Figure 1- 2   Package drawing for TLSR8258F1KET48A (Unit: mm)

SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
	0.85	0.90	0.95
A1	0	0.02	0.05
b	0.18	0.25	0.30
b1	0.11	0.16	0.21
c	0.18	0.20	0.23
D	6.90	7.00	7.10
D2	5.30	5.40	5.50
e	0.50BSC		
Ne	5.50BSC		
Nd	5.50BSC		
E	6.90	7.00	7.10
E2	5.30	5.40	5.50
L	0.35	0.40	0.45
h	0.30	0.35	0.40
h1	0.03REF		
h2	0.10REF		

L/F carrier size

L/F载体尺寸	SYMBOL	MILLIMETER
224*224	D2	5.40±0.10
	E2	5.40±0.10

Figure 1- 3   Package dimension for TLSR8258F1KET48A (Unit: mm)

1.6   Pin layout

1.6.1   Pin layout for TLSR8258F1KET48A

The figure below shows pin assignment for the TLSR8258F1KET48A.
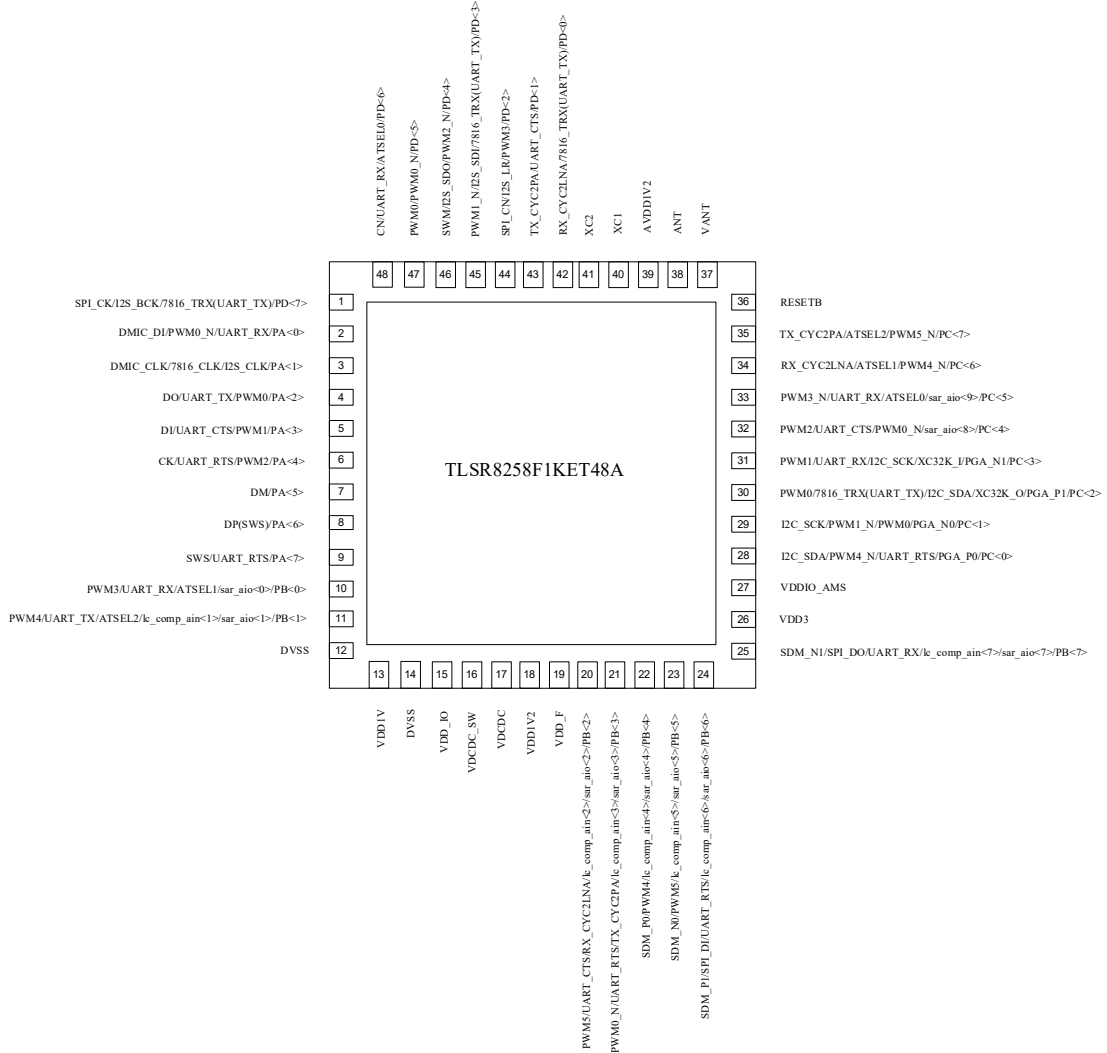
Figure 1- 4   Pin assignment for TLSR8258F1KET48A

Functions of 48 pins for the TLSR8258F1KET48A are described in Table 1- 2:

Table 1- 2   Pin functions for TLSR8258F1KET48A

No.	Pin Name	Type	Description
1	SPI_CK/I2S_BCK/7816_TRX (UART_TX)/PD<7>	Digital I/O	SPI clock (I2C_SCK) / I2S bit clock / UART 7816 TRX (UART_TX) / GPIO PD[7]
2	DMIC_DI/PWM0_N/UART_RX/ PA<0>	Digital I/O	DMIC data input / PWM0 inverting output / UART_RX / GPIO PA[0]
3	DMIC_CLK/7816_CLK/I2S_CLK/ PA<1>	Digital I/O	DMIC clock / UART 7816 clock / I2S clock / GPIO PA[1]
4	DO/UART_TX/PWM0/PA<2>	Digital I/O	SPI data output / UART_TX / PWM0 output / GPIO PA[2]
5	DI/UART_CTS/PWM1/PA<3>	Digital I/O	SPI data input (I2C_SDA) / UART_CTS / PWM1 output / GPIO PA[3]
6	CK/UART_RTS/PWM2/PA<4>	Digital I/O	SPI clock (I2C_SCK) / UART_RTS / PWM2 output / GPIO PA[4]
7	DM/PA<5>	Digital I/O	USB data minus / GPIO PA[5]
8	DP(SWS)/PA<6>	Digital I/O	USB data positive (Single wire slave) / GPIO PA[6]
9	SWS/UART_RTS/PA<7>	Digital I/O	Single wire slave/ UART_RTS / GPIO PA[7]
10	PWM3/UART_RX/ATSEL1/ sar_aio<0>/PB<0>	Digital I/O	PWM3 output / UART_RX / Antenna select pin 1 / SAR ADC input / GPIO PB[0]
11	PWM4/UART_TX/ATSEL2/lc_comp _ain<1>/sar_aio<1>/PB<1>	Digital I/O	PWM4 output / UART_TX / Antenna select pin 2 / Low power comparator input / SAR ADC input / GPIO PB[1]
12	DVSS	GND	Digital LDO ground
13	VDD1V	PWR	Internal LDO generated power supply input for digital core
14	DVSS	GND	Digital LDO ground
15	VDD_IO	PWR	External 3.3V power supply input for IO
16	VDCDC_SW	Analog	Connect with VDCDC via external inductor
17	VDCDC	Analog	Connect with VDCDC_SW via external inductor
18	VDD1V2	PWR	Internal DCDC generated power supply. Connect to GND via external capacitor. Route this 1.2V voltage power supply to AVDD1V2.
19	VDD_F	PWR	Internally generated power supply to flash. Connect to GND via external capacitor.
20	PWM5/UART_CTS/RX_CYC2LNA/ lc_comp_ain<2>/sar_aio<2>/ PB<2>	Digital I/O	PWM5 output / UART_CTS / Control external LNA / Low power comparator input / SAR ADC input / GPIO PB[2]
21	PWM0_N/UART_RTS/TX_CYC2PA/ lc_comp_ain<3>/sar_aio<3>/ PB<3>	Digital I/O	PWM0 inverting output / UART_RTS / Control external PA / Low power comparator input / SAR ADC input / GPIO PB[3]
22	SDM_P0/PWM4/lc_comp_ain<4>/ sar_aio<4>/PB<4>	Digital I/O	SDM positive output 0 / PWM4 output / Low power comparator input / SAR ADC input / GPIO PB[4]

No.	Pin Name	Type	Description
23	SDM_N0/PWM5/lc_comp_ain<5>/ sar_aio<5>/PB<5>	Digital I/O	SDM negative output 0 / PWM5 output / Low power comparator input / SAR ADC input / GPIO PB[5]
24	SDM_P1/SPI_DI/UART_RTS/ lc_comp_ain<6>/sar_aio<6>/ PB<6>	Digital I/O	SDM positive output 1 / SPI data input (I2C_SDA) / UART_RTS / Low power comparator input / SAR ADC input / GPIO PB[6]
25	SDM_N1/SPI_DO/UART_RX/ lc_comp_ain<7>/sar_aio<7>/ PB<7>	Digital I/O	SDM negative output 1 / SPI data output / UART_RX / Low power comparator input / SAR ADC input / GPIO PB[7]
26	VDD3	PWR	Connect to an external 3.3V power supply
27	VDDIO_AMS	PWR	External 3.3V power supply input for analog/mixed signal circuitry.
28	I2C_SDA/PWM4_N/UART_RTS/ PGA_P0/PC<0>	Digital I/O	I2C serial data / PWM4 inverting output / UART_RTS / PGA left channel positive input / GPIO PC[0]
29	I2C_SCK/PWM1_N/PWM0/ PGA_N0/PC<1>	Digital I/O	I2C serial clock / PWM1 inverting output / PWM0 output / PGA left channel negative input / GPIO PC[1]
30	PWM0/7816_TRX(UART_TX)/I2C_ SDA/XC32K_O/PGA_P1/PC<2>	Digital I/O	PWM0 output / UART 7816 TRX (UART_TX) / I2C serial data / (optional) 32kHz crystal output / PGA right channel positive input / GPIO PC[2]
31	PWM1/UART_RX/I2C_SCK/XC32K_ I/PGA_N1/PC<3>	Digital I/O	PWM1 output / UART_RX / I2C serial clock / (optional) 32kHz crystal input / PGA right channel negative input / GPIO PC[3]
32	PWM2/UART_CTS/PWM0_N/ sar_aio<8>/PC<4>	Digital I/O	PWM2 output / UART_CTS / PWM0 inverting output / SAR ADC input / GPIO PC[4]
33	PWM3_N/UART_RX/ATSEL0/ sar_aio<9>/PC<5>	Digital I/O	PWM3 inverting output / UART_RX / Antenna select pin 0 / SAR ADC input / GPIO PC[5]
34	RX_CYC2LNA/ATSEL1/PWM4_N/ PC<6>	Digital I/O	Control external LNA / Antenna select pin 1 / PWM4 inverting output / GPIO PC[6]
35	TX_CYC2PA/ATSEL2/PWM5_N/ PC<7>	Digital I/O	Control external PA / Antenna select pin 2 / PWM5 inverting output / GPIO PC[7]
36	RESETB	RESET	Power on reset, active low
37	VANT	PWR	Internally generated voltage output to bias internal PA. Connect to ANT using an external inductor.
38	ANT	Analog	RF antenna, connect to VANT using external inductor
39	AVDD1V2	PWR	Power supply input for internal RF Modules. Route from VDD1V2. Connect to GND via external capacitor.
40	XC1	Analog	Connect 24MHz crystal
41	XC2	Analog	Connect 24MHz crystal

No.	Pin Name	Type	Description
42	RX_CYC2LNA/7816_TRX (UART_TX)/PD<0>	Digital I/O	Control external LNA / UART 7816 TRX (UART_TX) / GPIO PD[0]
43	TX_CYC2PA/UART_CTS/PD<1>	Digital I/O	Control external PA / UART_CTS / GPIO PD[1]
44	SPI_CN/I2S_LR/PWM3/PD<2>	Digital I/O	SPI chip select (Active low) / I2S left right channel select / PWM3 output / GPIO PD[2]
45	PWM1_N/I2S_SDI/7816_TRX (UART_TX)/PD<3>	Digital I/O	PWM1 inverting output / I2S serial data input / UART 7816 TRX (UART_TX) / GPIO PD[3]
46	SWM/I2S_SDO/PWM2_N/ PD<4>	Digital I/O	Single wire master / I2S serial data output / PWM2 inverting output / GPIO PD[4]
47	PWM0/PWM0_N/PD<5>	Digital I/O	PWM0 output / PWM0 inverting output / GPIO PD[5]
48	CN/UART_RX/ATSEL0/PD<6>	Digital I/O	SPI chip select (Active low) / UART_RX / Antenna select pin 0 / GPIO PD[6]

1.6.2   Notes

1) All digital IOs including PA<0> ~ PD<7> can be used as GPIOs and have configurable pull-up/pull-down resistor.

2) SPI:

   ✧ PD<7>: SPI_CK, PB<6>: SPI_DI, PB<7>: SPI_DO, PD<2>: SPI_CN

   ✧ PA<2>: DO, PA<3>: DI, PA<4>: CK, PD<6>: CN

3) I2C:

   ✧ PC<0> ~ PC<3> can be used as I2C. PC<0>: I2C_SDA, PC<1>: I2C_SCK, PC<2>: I2C_SDA, PC<3>: I2C_SCK

   ✧ I2C can also be multiplexed with SPI interface, i.e. I2C_SDA/I2C_SCK can be multiplexed with SPI_DI (DI)/SPI_CK (CK) respectively.

4) Audio PGA/AMIC input: PC<0>~PC<3>. Please refer to section **12   PGA**. Any GPIO pin can be used as AMIC BIAS.

5) DMIC:

   ✧ PA<0>: DMIC_DI, PA<1>: DMIC_CLK

6) I2S:

   ✧ PD<7>: I2S_BCK, PA<1>: I2S_CLK, PD<2>: I2S_LR, PD<3>: I2S_SDI, PD<4>: I2S_SDO

7) Stereo audio output:

   ✧ PB<4>: SDM_P0, PB<5>: SDM_N0, PB<6>: SDM_P1, PB<7>: SDM_N1

8) UART:

   ✧ PA<0>: UART_RX, PA<2>: UART_TX, PB<0>: UART_RX, PB<1>: UART_TX, PB<7>: UART_RX, PC<3>: UART_RX, PC<5>: UART_RX, PD<6>: UART_RX

   ✧ UART hardware flow control:

      PA<3>: UART_CTS, PA<4>: UART_RTS, PA<7>: UART_RTS, PB<2>: UART_CTS, PB<3>: UART_RTS, PB<6>: UART_RTS, PC<0>: UART_RTS, PC<4>: UART_CTS, PD<1>: UART_CTS

   ✧ UART 7816:

      PD<7>: 7816_TRX (UART_TX), PA<1>: 7816_CLK, PC<2>: 7816_TRX (UART_TX), PD<0>: 7816_TRX (UART_TX), PD<3>: 7816_TRX (UART_TX)

9) USB:

   ✧ PA<5>: DM, PA<6>: DP

10) Single Wire debug interface:

   ✧ PA<7>: SWS

   ✧ SWS can also be multiplexed with DP.

   ✧ PD<4>: SWM

11) RX_CYC2LNA & TX_CYC2PA:

   ✧ PB<2>: RX_CYC2LNA, PB<3>: TX_CYC2PA, PC<6>: RX_CYC2LNA, PC<7>: TX_CYC2PA, PD<0>: RX_CYC2LNA, PD<1>: TX_CYC2PA

✧ RX_CYC2LNA/TX_CYC2PA serves to control enabling external PA/LNA. Please refer to section **3.1 Block diagram**.

12) ATSEL0~2:

✧ PB<0>: ATSEL1, PB<1>: ATSEL2, PC<5>: ATSEL0, PC<6>: ATSEL1, PC<7>: ATSEL2, PD<6>: ATSEL0

✧ ATSEL0~2 serves to select one of up to eight external antennas connected to the antenna select component. The selected antenna channel is connected to the RF_IO pin.

13) Low power comparator input: PB<1>~ PB<7>. Please refer to chapter 13.

14) ADC GPIO input: PB<0>~ PB<7>, PC<4>~ PC<5>.

15) For register configuration to select pin multiplexed function, please refer to section **7.1.1.2** .

16) For 24MHz crystal, the load capacitor range supported by design is 7.33pF~12.66pF. If the crystal needs load capacitor of 15pF, two external capacitors will be required.

17) Do not use PC<7:5> for dynamic applications. It's highly recommended to use these IOs in DC applications, e.g. as control or detect lines.

18) Pin drive strength:

✧ PA<5:7> and PB<0:3> support drive strength up to 8mA (8mA when "DS"=1, 4mA when "DS"=0); PB<4:7> support drive strength up to 16mA (16mA when "DS"=1, 12mA when "DS"=0); other GPIOs (PA<0:4>, PC<0:7> and PD<0:7>) support drive strength up to 4mA (4mA when "DS"=1, 2mA when "DS"=0).

✧ "DS" configuration will take effect when the pin is used as output. Please refer to section **7.1.1 Basic configuration** for the corresponding "DS" register address and the default setting.

19) When PB[0] and PB[1] are used as wakeup source, they can only act as input IO signals instead of output IO signals when normal working. The PC[7:0] can only wake up the chip at low level voltage.

2 Memory and MCU

2.1 Memory

The TLSR8258F1KET48A embeds 64kB SRAM (including up to 32kB SRAMs with retention in deep sleep, and one 32kB SRAM without retention) as data memory, and 1MB internal FLASH as program memory.

2.1.1 SRAM/Register

SRAM/Register memory map is shown as follows:
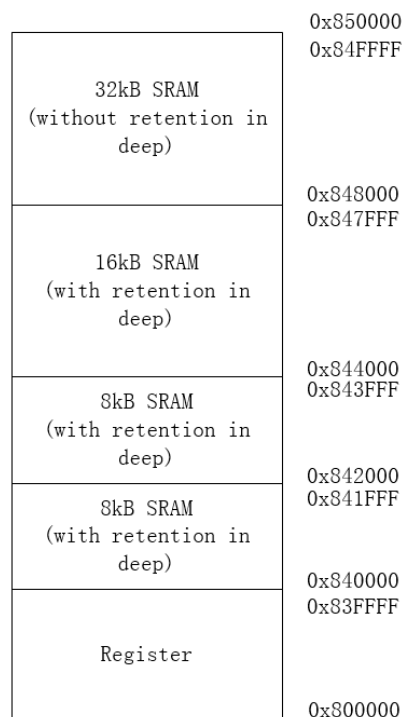
Figure 2- 1   Physical memory map

Register address: 0x800000 ~ 0x83FFFF.

Address for two independent 8kB SRAMs with retention in deep sleep: 0x840000 ~ 0x841FFF, 0x842000 ~ 0x843FFF.

Address for 16kB SRAM with retention in deep sleep: 0x844000 ~ 0x847FFF.

Address for 32kB SRAM without retention in deep sleep: 0x848000 ~ 0x84FFFF.

Both register and SRAM address can be accessed (read or write) via debugging interface (SWS/SWM, SPI/I2C/USB interface).

Register
(Base address: 0x800000)

Register	Address
RSVD	
	0x40000
Modem	
	0x01200
RSVD	
	0x01020
RSVD	
	0x01000
linklayer	
	0x00f00
RSVD	
	0x00d00
dma	
	0x00c00
DMA fifo	
	0x00b00
RSVD	
	0x00800
pwm	
	0x00780
System timer	
	0x00740
RSVD	
	0x00700
MCU	
	0x00600
gpio	
	0x00580
audio	
	0x00560
AES	
	0x00540
RSVD	
	0x00500
Baseband	
	0x00400
RSVD	
	0x00200
usb	
	0x00100
I2C address map	
	0x000e0
qdec	
	0x000d0
RSVD	
	0x000c0
RSVD	
	0x000b8
uart	
	0x000b4
swire	
	0x000b0
RSVD	
	0x000a0
uart	
	0x00090
RSVD	
	0x00080
System control	
	0x00040
RSVD	
	0x00010
RSVD	
	0x0000c
spi	
	0x00008
i2c	
	0x00000

Figure 2- 2   Register space

2.1.2 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.

For chip identification and traceability, the Flash is preloaded with Unique ID (UID). User is not allowed to modify this preloaded UID, but can read the UID via corresponding API interface.

MCU uses the system frequency to load instructions, and adopts flash driver to access (read/write) flash with the speed of half of the system clock.

2.2 Firmware encryption

The TLSR8258F1KET48A supports multiple firmware encryption methods to achieve the anti-cloning protection, including:

✧ UID-based authentication code generation method

During firmware burning (e.g. via specific burning jig), user can use customized key and AES encryption algorithm to encrypt the UID read from the chip flash, generate unique ciphertext and write the ciphertext into Flash section.

During application, an encryption authentication procedure is added. User should use the same key and AES encryption algorithm to encrypt the UID read from the chip flash, and generate new ciphertext. Before running main application firmware, the new ciphertext will be compared with the ciphertext read from the Flash section. Only when the authentication passes, i.e. the comparison result matches, the main firmware will be up and running, otherwise the chip will stop running the main firmware.

✧ Bootloader-based firmware encryption/decryption

The firmware can be encrypted using a customer-provided security key. The customer security key is written into a specific secure register, and becomes unreadable. Any attempt to read the key will only result in either all 1's or all 0's.

The encrypted firmware can be generated based on the plaintext firmware and the customer security key. The customer can burn the security key into the obscured memory area and also the encrypted firmware into Flash.

The firmware is readable by all, but appears as garbled binaries to 3rd party.

2.3 MCU

The TLSR8258F1KET48A integrates a powerful 32-bit MCU developed by Telink. The digital core is based on 32-bit RISC, and the length of instructions is 16 bits; four hardware breakpoints are supported.

2.4 Working modes

The TLSR8258F1KET48A 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 2- 1   Working modes

Mode	Active	Idle	Suspend	Deep sleep with SRAM retention	Deep sleep without SRAM retention	Shutdown
MCU	active	stall	stall	off	off	off
32k Normal SRAM (without retention in deep sleep)	on	on	on	off	off	off
Radio	available	available	off	off	off	off
USB	available	available	off	off	off	off
Audio	available	available	off	off	off	off
Wakeup time to Active mode	——	0us	100us	Shorter than deep sleep without retention, almost same as Suspend	1ms	10ms
(16k+8k+8k) 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	——	——	——	——
Wakeup on reset pin (RESETB)	——	available	available	available	available	on
Current	Please refer to section **15.3 DC characteristics**.					

*Notes:

1) "active": MCU is at working state.

2) "stall": In Idle and Suspend mode, MCU does not work, while its clock is still running.

3) "available" for Modules: It's selectable to be at working state, or stall/be powered down if it does not need to work.

4) "available"/"on" for wakeup: Corresponding wakeup method is supported.

5) "off" for wakeup: Corresponding wakeup method is not supported.

6) "on"/"off"/"full" for SRAMs:

✧ "on": The 32kB SRAM is powered on and works normally (can be accessed) in Active, Idle and Suspend mode.

✧ "full": Full speed. In Active, Idle and Suspend mode, the two 8kB and one 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 32kB SRAM is powered down in two Deep sleep modes and Shutdown mode. The retention SRAMs are powered down in Deep sleep without SRAM retention and Shutdown mode.

7) Current:

✧ In Deep sleep without SRAM retention, only the PM module is active, all digital and analog modules are powered down, thus the power consumption is largely decreased.

✧ In Deep sleep with SRAM retention, the PM module is active, all analog and digital modules except for the retention SRAMs are powered down, thus the power consumption is a little higher than in Deep sleep without SRAM retention, but much lower than in Suspend.

Table 2- 2   Retention analog registers in deep sleep

Address	R/W	Description	Reset Value
0x35	R/W	buffer, watch dog reset clean	0x20
0x36	R/W	buffer, watch dog reset clean	0x00
0x37	R/W	buffer, watch dog reset clean	0x00
0x38	R/W	buffer, watch dog reset clean	0x00
0x39	R/W	buffer, watch dog reset clean	0xff
0x3a	R/W	buffer, power on reset clean	0x00
0x3b	R/W	buffer, power on reset clean	0x00
0x3c	R/W	buffer, power on reset clean	0x00

Analog registers (0x35 ~ 0x3c) as shown in Table 2- 2 are retained in deep sleep mode and can be used to store program state information across deep sleep cycles.

✧ Analog registers 0x3a~0x3c 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 0x35~0x39 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.

For chip software reset, please refer to section **2.5   Reset**.

2.5 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 0x3a~0x3c will be cleared.

3) Software reset: It is also feasible to carry out software reset for the whole chip or some modules.

   ✧ Setting address 0x6f[5] as 1b'1 is to reset the whole chip. Similar to watchdog reset, the retention analog registers 0x3a~0x3c are non-volatile, while other registers including 0x35~0x39 will be cleared by chip software reset.

   ✧ Addresses 0x60~0x62 serve to reset individual modules: if some bit is set to logic "1", the corresponding module is reset.

Table 2- 3 Register configuration for software reset

Address	Mnemonic	Type	Description	Reset Value
0x60	RST0	R/W	Reset control, 1 for reset, 0 for clear [0]: SPI [1]: I2C [2]: UART (rs232) [3]: USB [4]: PWM [5]: QDEC [6]: RSVD [7]: Swire	0x7c
0x61	RST1	R/W	[0] ZB [1] System Timer [2] DMA [3] ALGM [4] AES [5] ADC [6] ALG [7] RSVD	0xff
0x62	RST2	R/W	[0] AIF [1] Audio [2] DFIFO [3] RSVD [4] RISC [5] MCIC [6] RISC1 (R) [7] MCIC1 (R)	0xc7
0x6f	PWDNEN	R/W	[0]: suspend enable (RW) [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

2.6 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.

2.6.1   Power-On-Reset (POR) and Brown-out detect
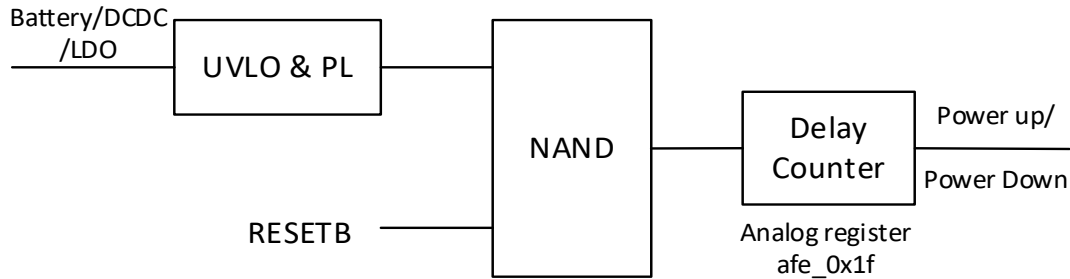
Figure 2- 3   Control logic for power up/down

The whole chip 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_0x1f. Since the content of afe_0x1f is reset to default only after power cycle, watchdog reset, or software reset, the delay change using afe_0x1f is only applicable when the chip has not gone through these reset conditions. For example, after deep sleep wakeup, the setting in afe_0x1f will take effect.

Table 2- 4 Analog register to control delay counters

Address	Name	Type	Description	Default
afe_0x1f	r_dly1	R/W	wait for DCDC ready (16kHz count)	0x40

Initial Power up
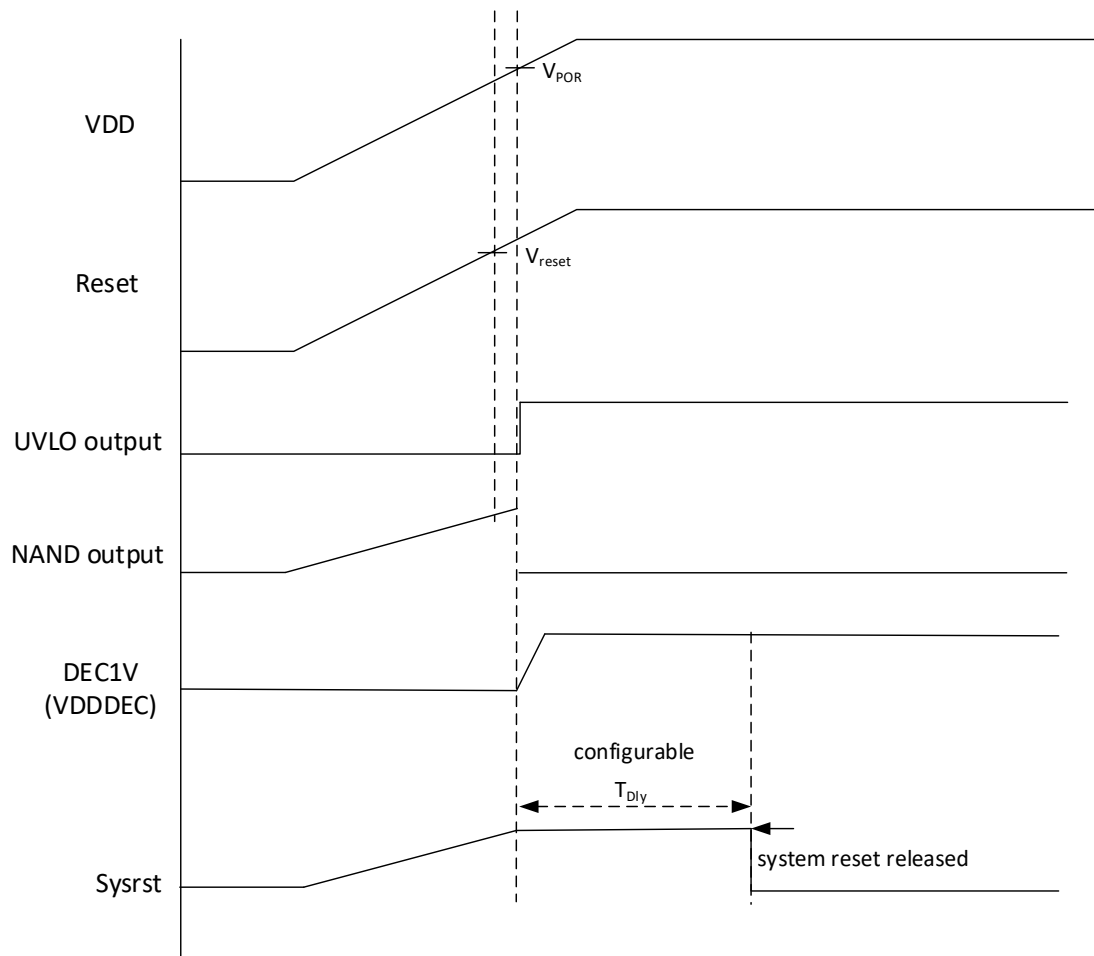
Figure 2- 4   Initial Power-up sequence

Power down
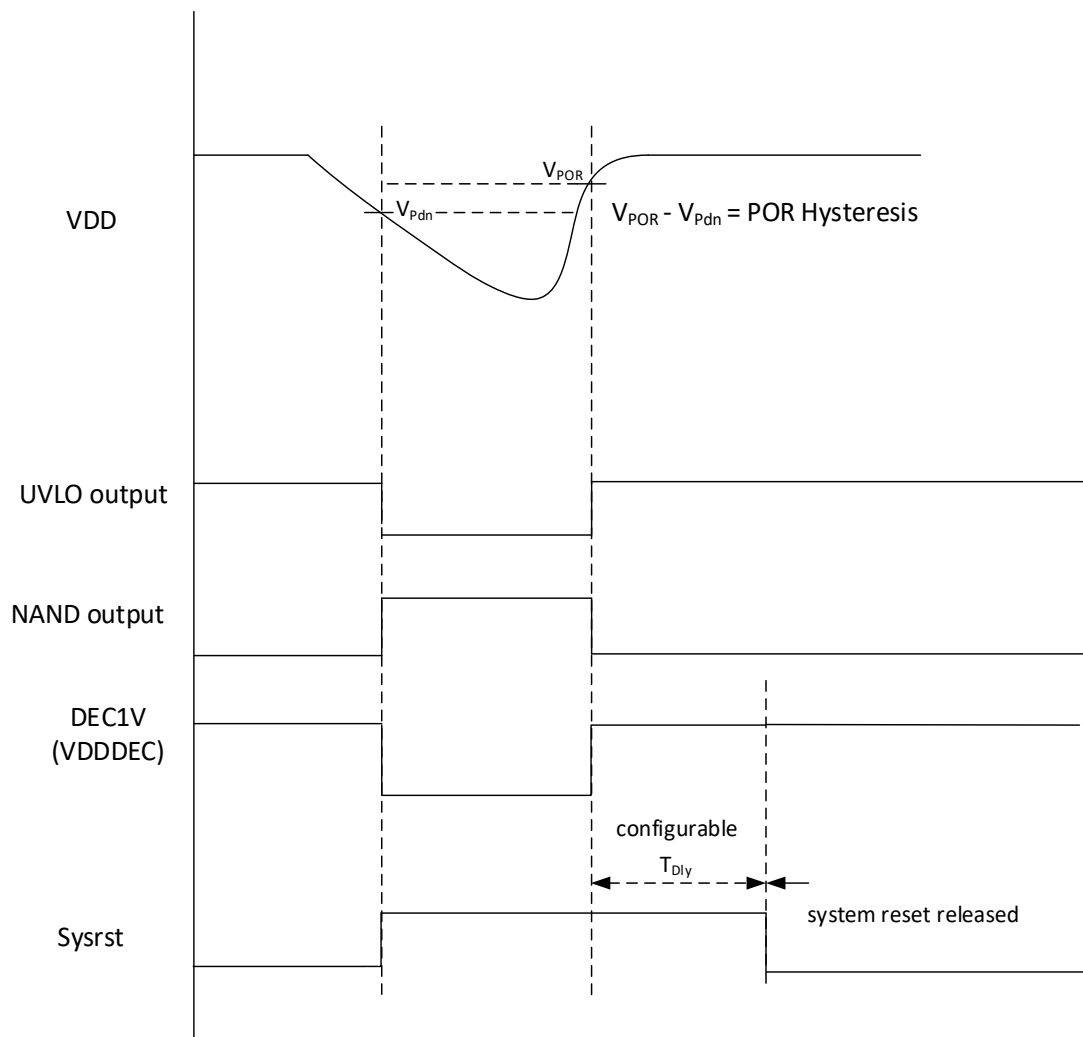
Figure 2- 5   Power-down sequence

Table 2- 5 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		1.62		V
V_{Pdn}	VDD voltage when V_{UVLO} turns to low level		1.55		V
T_{Dly}	Delay counter value	Configurable via analog register afe_0x1f			

2.6.2 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, Audio and 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. (Please refer to Table 2- 1.)

✧ In Suspend mode, MCU stalls, all SRAMs are still accessible, the PM module is active, modules such as RF transceiver, Audio and USB are powered down. The chip can be triggered to Active mode by 32K Timer, IO pin or RESETB pin. It takes 100us 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 two 8kB and one 16kB 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 1ms 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 10ms 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 2- 6 3.3V analog registers for module power up/down control

Address	Local name	Default	Description
afe_0x05<0>	32K_rc_pd	0	Power down 32kHz RC oscillator 1: Power down, 0: Power up
afe_0x05<1>	32k_xtal_pd	1	Power down 32kHz crystal 1: power down, 0: power up
afe_0x05<2>	24M_rc_pd	0	Power down of 24MHz RC oscillator 1: Power down, 0: Power up
afe_0x05<3>	xtal_LDO_pd	0	Power down of 24MHz crystal oscillator 1: Power down, 0: Power up
afe_0x05<4>	pd_pl_all_3v	0	Power down of power logic, 4.2V VBUS_LDO and DCDC 1: Power down, 0: Power up

Address	Local name	Default	Description
afe_0x05<5>	pd_pl_dcdc_3v	0	Power down of DCDC 1: Power down, 0: Power up
afe_0x05<6>	pd_pl_vbus_ldo_3v	0	Power down of VBUS_LDO 1: Power down, 0: Power up
afe_0x05<7>	ana_ldo_pd_3V	0	Power down baseband pll LDO 1: Power down, 0: Power up
afe_0x06	rsvd		
afe_0x07<0>	spd_ldo_pd	1	Power down of spd ldo 1: Power down, 0: power up
afe_0x07<1>	dig_ldo_pd	0	Power down of main digital ldo 1: Power down, 0: power up
afe_0x07<2>	dig_ret_pd	1	Power down of retention ldo 1: Power down, 0: power up
afe_0x07<3>	pd_lc_comp_3v	1	Power down of low current comparator: 1: Power down, 0: Power up
afe_0x07<4>	pd_temp_sensor_3v	1	Power down of temperature sensor: 1: Power down, 0: Power up

2.6.3   DCDC

The chip embedded DCDC can generate 1.8V output voltage and supply power for the internal flash; the DCDC can also generate 1.4V output voltage.

2.6.4   VBAT and VANT power-supply mode

The RF PA module has two power-supply modes including VBAT mode and VANT mode.

✧   In VBAT mode, the RF PA module is supplied by 3.3V voltage regulated from 4.2V lithium battery or directly from two AA/AAA batteries in series. The maximum output power is related to power supply voltage of RF PA, for example, the maximum power is 10dBm at 3.3V power supply.

✧   In VANT mode, the RF PA module is supplied with 1.2V voltage by the embedded DCDC. In this mode, output power won't change with AVDD basically, and the maximum power is 5dBm or so. Corresponding to the VBAT mode, the VANT mode is more power-saving at the same Tx power.

When the chip works in VBAT mode, it can be configured to the maximum output power. However, as the VBAT/VDD supply decreases below 3.0 V, the maximum transmit power of TX is then slightly attenuated. The detailed RF transmit power level refers to the code comments in the corresponding driver SDK, in which the RF transmit power level under VBAT mode is the result tested in 3.3 V VBAT voltage.

2.7 Wakeup sources
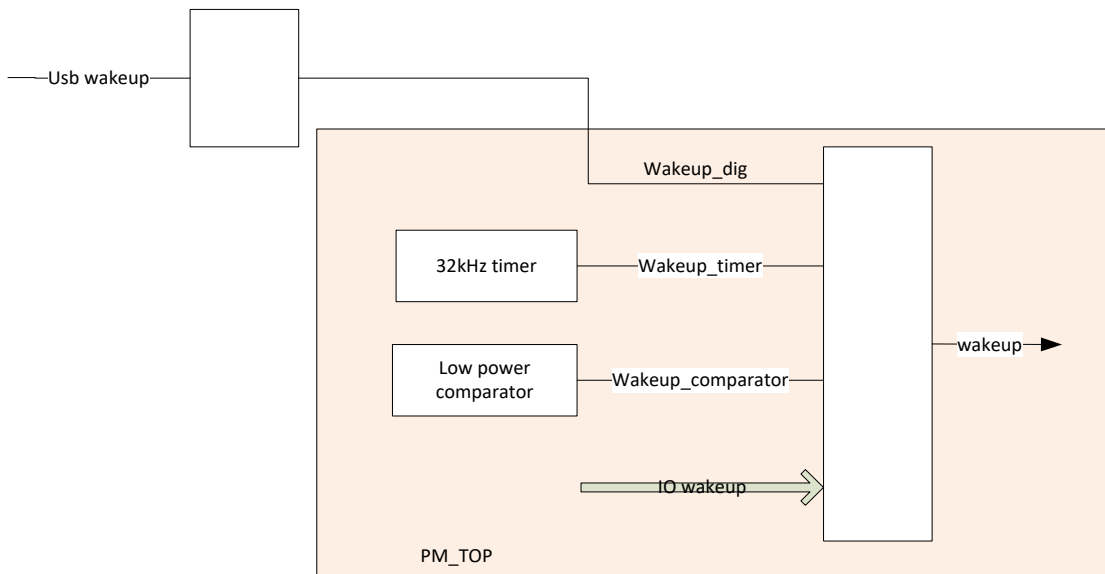
Figure 2- 6 Wakeup sources

2.7.1 Wakeup source - USB

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

First, set the digital register 0x6e bit[2] as 1b'1.

To activate this mode, analog register afe_0x26[5] should also be set as 1b'1.

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

2.7.2 Wakeup source – 32kHz timer

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

To enable the wakeup source from 32kHz timer, analog register afe_0x26[6] should be set as 1b'1.

2.7.3 Wakeup source – low power comparator

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

To enable the wakeup source from low power comparator, analog register 0x26[7] should be set as 1b'1. The low power comparator wakeup is active high.

2.7.4 Wakeup source – IO

This wakeup source is able to wake up the system from suspend mode or two deep sleep modes. And IO wakeup supports high level or low level wakeup which is configurable via polarity control registers.

Analog register afe_0x26[4] should be set as 1b'1 to enable IO wakeup source.

Enabling control analog registers: PA[7:0] enabling control register is afe_0x27[7:0], PB[7:0] enabling control register is afe_0x28[7:0], PC[7:0] enabling control register is afe_0x29[7:0], and PD[7:0]

enabling control register is afe_0x2a[7:0]. Total wakeup pin can be up to 32.

Polarity control registers: PA[7:0] polarity control register is afe_0x21[7:0], PB[7:0] polarity control register is afe_0x22[7:0], PC[7:0] polarity control register is afe_0x23[7:0], and PD[7:0] polarity control register is afe_0x24[7:0].

The corresponding driver is available so that user can directly invoke it to use IO wakeup source.

Analog register 0x44[3:0] indicates the wakeup source which triggers system wakeup. After wakeup, the corresponding wakeup status will be set as 1b'1 automatically, and it's needed to write 1 to manually clean the status.

2.7.5   Register table

Table 2- 7 Analog registers for Wakeup

Address	Name	Type	Description	Default
afe_0x21	PA_POL	R/W	Polarity control registers for IO wakeup	0x00
afe_0x22	PB_POL	R/W		0x00
afe_0x23	PC_POL	R/W		0x00
afe_0x24	PD_POL	R/W		0x00
afe_0x25	rsvd	R		
afe_0x26	wkup_en	R/W	[7] low power comparator wakeup enable [6] 32kHz timer wakeup enable [5] digital core (USB) wakeup enable [4] IO (pad) wake up enable [3] Enable/Mask filter for IO (Pad) wakeup 1: Select 16us filter to filter out jitter on IO PAD input. 0: IO Pad combinational logic output (disable filter)	0x00
afe_0x27	PA wake up enable	R/W	Enabling control registers for IO wakeup	0x00
afe_0x28	PB wake up enable	R/W		0x00
afe_0x29	PC wake up enable	R/W		0x00
afe_0x2a	PD wake up enable	R/W		0x00
afe_0x44	status	R	[7] dcdc_rdy [6] wd_status [5] cal_done_24m [4] cal_done_32k [3] IO (pad) wakeup status [2] digital core (USB) wakeup status [1] 32k timer wakeup status [0] low power comparator wakeup status	

Table 2- 8 Digital register for Wakeup

Address	Mnemonic	Type	Description	Reset Value
0x6e	WAKEUPEN	R/W	Wakeup enable [0]: enable wakeup from I2C host [1]: enable wakeup from SPI host [2]: enable wakeup from USB [3]: enable wakeup from gpio [4]: enable wakeup from I2C synchronous interface System resume control [5]: enable GPIO remote wakeup [6]: if set to1, system will issue USB resume signal on USB bus [7] sleep wakeup reset system enable	0x1f

3   BLE/802.15.4/2.4GHz RF Transceiver

3.1   Block diagram

The TLSR8258F1KET48A integrates an advanced BLE/802.15.4/2.4GHz RF transceiver. The RF transceiver works in the worldwide 2.4GHz ISM (Industrial Scientific Medical) band.

The transceiver consists of a fully integrated RF synthesizer, a Power Amplifier (PA), a Low Noise Amplifier (LNA), a TX filter, a RX filter, a TX DAC, an ADC, a modulator and a demodulator. The transceiver can be configured to work in standard-compliant 1Mbps BLE mode, 2Mbps enhancement BLE mode, 125kbps BLE long range mode (S8), 500kbps BLE long range mode (S2), IEEE 802.15.4 standard-compliant 250kbps mode, and Proprietary 1Mbps, 2Mbps, 250kbps and 500kbps mode.
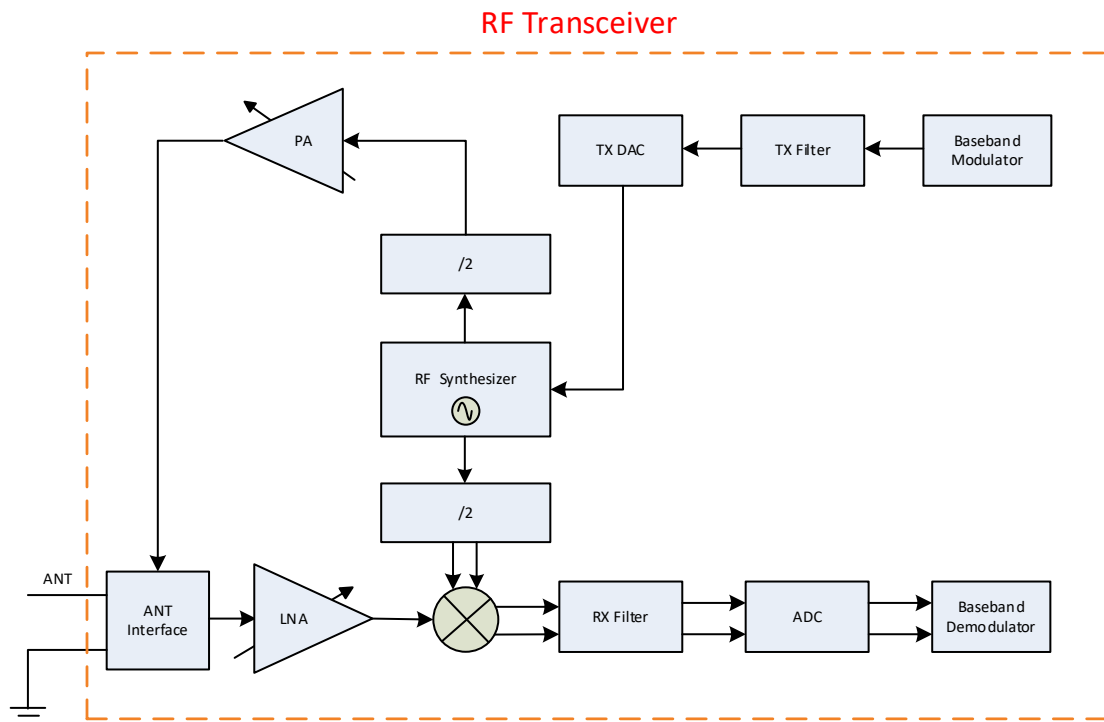
Figure 3- 1   Block diagram of RF transceiver

To control external PA and LNA, first follow the GPIO lookup table (see section 7.1.1.1   GPIO lookup table) to configure the specific two pins as TX_CYC2PA and RX_CYC2LNA function, respectively. Note: To use TX_CYC2PA and RX_CYC2LNA function for the two pins, other functions with higher polarity should be disabled at the same time. After the two pins are configured as TX_CYC2PA and RX_CYC2LNA function, the output function is enabled. Generally the two pins are high active: When both the two pins output low level, the external PA and LNA are disabled; when one of the two pins output high level, the external PA/LNA are enabled correspondingly; the two pins won't output high level simultaneously.

Table 3- 1 External RF transceiver control example

TX_CYC2PA	RX_CYC2LNA	External RF transceiver
L	L	Both LNA and PA OFF
L	H	LNA ON
H	L	PA ON
H	H	N/A

The internal PA can deliver a maximum 10dBm output power, avoiding the need for an external RF PA.

3.2 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.

For the TLSR8258F1KET48A, RF transceiver can operate with frequency ranging from 2.400GHz to 2.4835GHz. The RF channel frequency setting determines the center of the channel.

3.3 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 and 802.15.4 specification.

3.3.1 Packet format

Packet format in standard 1Mbps BLE mode is shown as Table 3- 2:

Table 3- 2 Packet Format in standard 1Mbps BLE mode

LSB   MSB

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

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

Packet format in standard 2Mbps BLE mode is shown as Table 3- 3:

Table 3- 3 Packet format in standard 2Mbps BLE mode

LSB   MSB

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

Packet format in standard 500kbps/125kbps BLE mode is shown as Table 3- 4:

Table 3- 4 Packet format in standard 500kbps/125kbps BLE mode

LSB   MSB

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

Packet format in 250kbps 802.15.4 mode is shown as Table 3- 5:

Table 3- 5   Packet format in 802.15.4 mode

LSB   MSB

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

Packet format in 2.4GHz Proprietary mode is shown as Table 3- 6:

Table 3- 6   Packet format in Proprietary mode

LSB   MSB

Preamble (8 bits)	Address code (configurable 3~5 bytes)	Packet Controller + Payload (1~63 bytes)	CRC (1~2 bytes)

3.3.2   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.

3.3.3   RSSI and frequency offset

The TLSR8258F1KET48A provides accurate RSSI (Receiver Signal Strength Indicator) and frequency offset indication.

✧   RSSI can be read from the 1byte 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 +/-1dB.

✧   Frequency offset can be read from the 2bytes at the tail of the data packet. Valid bits of actual frequency offset may be less than 16bits, and different valid bits correspond to different tolerance range.

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

4 Clock

4.1 Clock sources
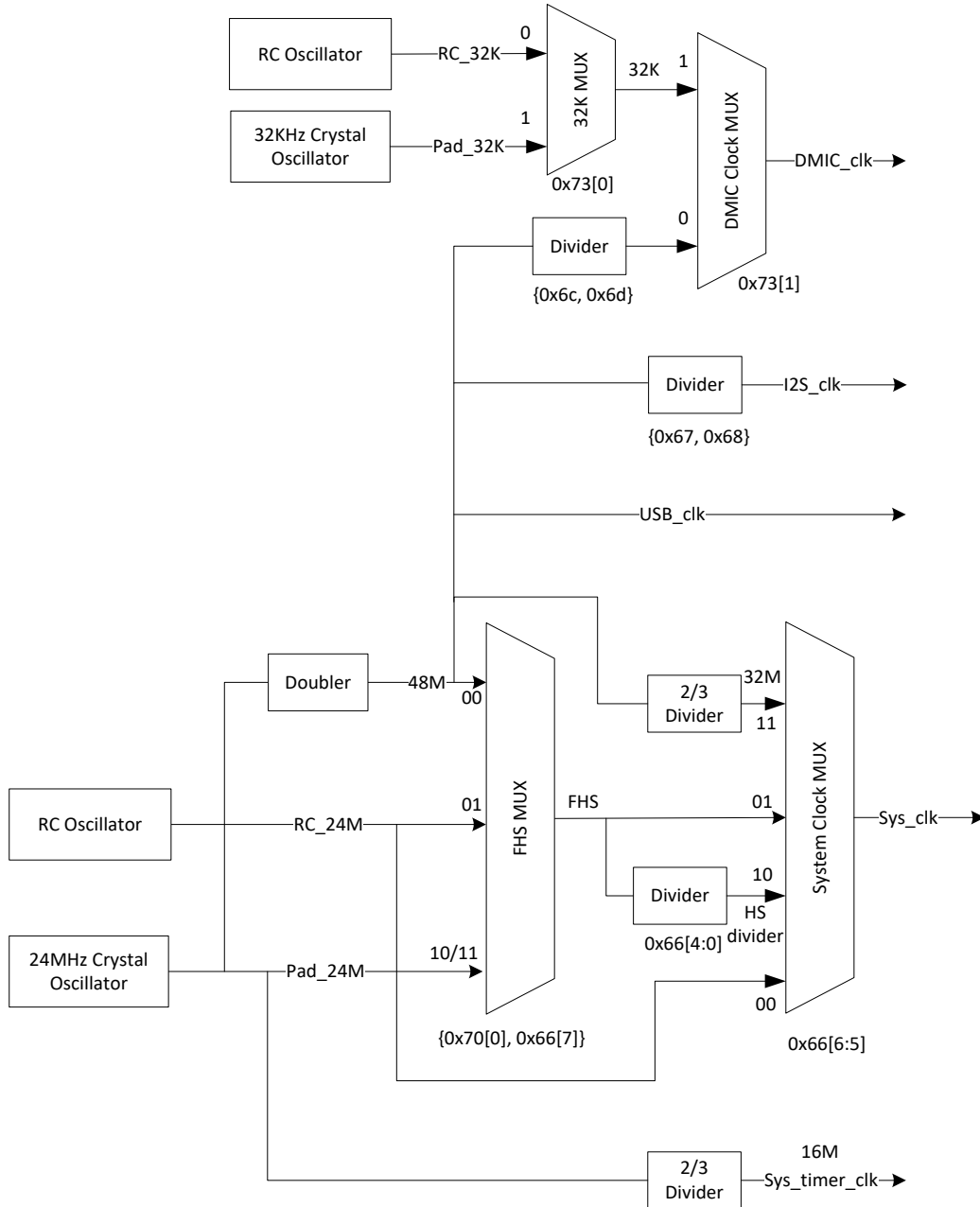
Figure 4- 1   Block diagram of clock

The TLSR8258F1KET48A embeds a 24MHz RC oscillator which can be used as clock source for system, as well as a 32kHz RC oscillator to provide clock source for DMIC and sleep state.

External 24MHz crystal is available via pin XC1 and XC2, which can provide a Pad_24MHz clock source for system and System Timer, and generate a 48M clock via a frequency doubler to provide clock source for DMIC, I2S and USB. External 32K crystal is available via pin PC<2:3>, which can provide a 32kHz clock source for DMIC.

4.2   System clock

There are four selectable clock sources for MCU system clock: **RC_24M** derived from 24MHz RC oscillator, High speed clock "**FHS**", **HS divider clock** (derived from "FHS" via a frequency divider), and **32MHz clock** derived from 48MHz clock via a 2/3 frequency divider (The 48M clock is derived from 24M crystal oscillator via a frequency doubler).

The high speed clock (FHS) is selectable via address {0x70[0], 0x66[7]} from the following sources: **48MHz** clock (derived from 24M crystal oscillator via a frequency doubler), **RC_24M** (derived from 24MHz RC oscillator), and **Pad_24M** (derived from 24M crystal oscillator).

The digital register CLKSEL (address 0x66) serves to set system clock: System clock source is selectable via bit[6:5].

✧   If address 0x66[6:5] is set to 2b'10 to select the HS divider clock, system clock frequency is adjustable via address 0x66[4:0]. The formula is shown as below:

$F_{System\ clock}$ = F_{FHS} / (system clock divider value in address 0x66[4:0]).

Note that address 0x66[4:0] should not be set as 0 or 1.

4.3   Module clock

Registers CLKEN0~CLKEN2 (address 0x63~0x65) are used to enable or disable clock for various modules. By disable the clocks of unused modules, current consumption could be reduced.

4.3.1   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.

4.3.2   USB clock

USB clock is derived from 48M clock. The 48M clock is derived from 24M crystal oscillator via a frequency doubler.

4.3.3   I2S clock

I2S clock is derived from 48M clock via a frequency divider. The 48M clock is derived from 24M crystal oscillator via a frequency doubler.

Address 0x67[7] should be set as 1b'1 to enable I2S clock. I2S clock frequency dividing factor contains step and mod. Address 0x67[6:0] and 0x68 serve to set I2S clock step[6:0] and mod[7:0] respectively, and mod should be no less than 2*step.

I2S clock frequency, $F_{I2S\ clock}$, equals to 48M * I2S_step[6:0] / I2S_mod[7:0].

4.3.4   DMIC clock

Address 0x6c[7] serves to enable DMIC clock.

DMIC clock pin can select 32kHz clock, or derive from 48M clock via a frequency divider. Address 0x73 serves to select DMIC clock source.

✧ In normal DMIC working mode, 0x73[1] needs to be set as 1b'0, DMIC clock divider is selected and frequency dividing factor should be further configured. DMIC clock frequency dividing factor contains step and mod. Address 0x6c[6:0] and 0x6d serve to set DMIC clock step[6:0] and mod[7:0], respectively, and mod should be no less than 2*step.

In this situation, DMIC clock frequency, $F_{DMIC\ clock}$, equals to 48M * DMIC_step[6:0] / DMIC_mod[7:0].

✧ When DMIC is not used, and a 32kHz clock is needed, address 0x73 bit[1] should be set as 1b'1 to select the 32kHz clock, while bit[0] can be configured to select either RC_32K from 32K RC oscillator or Pad_32K from 32K crystal oscillator.

4.4   Register table

Table 4- 1 Register table related to clock

Address	Mnemonic	R/W	Description	Default
0x63	CLKEN0	R/W	Clock enable control: 1 for enable; 0 for disable [0]: SPI [1]: I2C [2]: UART (rs232) [3]: USB [4]: PWM [5]: QDEC [6]: RSVD [7]: Swire	0x83
0x64	CLKEN1	R/W	[0]ZB [1]System Timer [2]DMA [3]ALGM [4]AES [5:7]RSVD	0x00
0x65	CLKEN2	R/W	[0]AIF [1]Audio [2]DFIFO [3]RSVD [4]MC [5]MCIC [6:7]RSVD	0x30
0x66	CLKSEL	R/W	System clock select [4:0]: system clock divider (must exceed 1). If 0x66[6:5] is set as 2b'10, F_{Sysclk} = F_{FHS} / (CLKSEL[4:0]). FHS: refer to 0x70 CLKSEH. [6:5]: select system clock source 2'b00: RC_24M from RC oscillator 2'b01: FHS 2'b10: HS divider (see 0x66[4:0]) 2'b11: 32M clock (48M * 2/3 divider) [7] FHS select (see 0x70[0])	0x06

Address	Mnemonic	R/W	Description	Default
0x67	I2S_STEP	R/W	[7] I2S clock enable [6:0] I2S step	0x00
0x68	I2S_MOD	R/W	I2S mod I2S clock = 48M*I2S_step[6:0]/I2S_mod[7:0], Mod should be larger than or equal to 2*step.	0x02
0x6c	DMIC_STEP	R/W	[7]: Digital MIC clock enable [6:0] DMIC step	0x01
0x6d	DMIC_MOD	R/W	[7:0] DMIC mod DMIC clock = 48M*DMIC_step[6:0]/DMIC_mod Mod should be larger than or equal to 2*step.	0x02
0x70	CLKSEH	R/W	{0x70[0], 0x66[7]} FHS select 2'b00: 48M clock doubled from 24M crystal 2'b01: RC_24M from RC oscillator 2'b1x: Pad_24M from 24M crystal oscillator	0x00
0x73	SEL	R/W	[0]: clk32k select 0: select RC_32k from RC oscillator 1: select Pad_32k from 32K crystal oscillator [1]: DMIC clock select 1: select 32k (see 0x73[0] to select 32K source) 0: select DMIC clk div (see 0x6c~0x6d)	0x04

5   Timers

5.1   Timer0~Timer2

The TLSR8258F1KET48A supports three timers: Timer0~ Timer2. The three timers all 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 0x620) ~ TMR_CTRL1 (address 0x621).

Timer 2 can also be configured as "watchdog" to monitor firmware running.

5.1.1   Register table

Table 5- 1   Register configuration for Timer0~Timer2

Address	Mnemonic	Type	Description	Reset Value
0x72	Wd_status	R/W	[0] watch dog status: verify whether it is power reset (1'b0) or watch dog reset (1'b1), write 1 to clear.	00
0x620	TMR_CTRL0	RW	[0]Timer0 enable [2:1] Timer0 mode. 0: using sclk, 1: using gpio, 2: count width of gpi, 3: tick [3]Timer1 enable [5:4] Timer1 mode. [6]Timer2 enable [7]Bit of timer2 mode	00
0x621	TMR_CTRL1	RW	[0]Bit of timer2 mode [7:1]Low bits of watch dog capture	00
0x622	TMR_CTRL2	RW	[6:0]High bits of watch dog capture. It is compared with [31:18] of timer2 ticker [7]watch dog capture	00
0x623	TMR_STATUS	RW	[0] timer0 status, write 1 to clear [1] timer1 status, write 1 to clear [2] timer2 status, write 1 to clear [3] watch dog status, write 1 to clear (If Watchdog is enabled, need to clear it periodically to avoid triggering watchdog reset)	00
0x624	TMR_CAPT0_0	RW	Byte 0 of timer0 capture	00
0x625	TMR_CAPT0_1	RW	Byte 1 of timer0 capture	00
0x626	TMR_CAPT0_2	RW	Byte 2 of timer0 capture	00
0x627	TMR_CAPT0_3	RW	Byte 3 of timer0 capture	00
0x628	TMR_CAPT1_0	RW	Byte 0 of timer1 capture	00
0x629	TMR_CAPT1_1	RW	Byte 1 of timer1 capture	00
0x62a	TMR_CAPT1_2	RW	Byte 2 of timer1 capture	00
0x62b	TMR_CAPT1_3	RW	Byte 3 of timer1 capture	00
0x62c	TMR_CAPT2_0	RW	Byte 0 of timer2 capture	00
0x62d	TMR_CAPT2_1	RW	Byte 1 of timer2 capture	00

Address	Mnemonic	Type	Description	Reset Value
0x62e	TMR_CAPT2_2	RW	Byte 2 of timer2 capture	00
0x62f	TMR_CAPT2_3	RW	Byte 3 of timer2 capture	00
0x630	TMR_TICK0_0	RW	Byte 0 of timer0 ticker	00
0x631	TMR_TICK0_1	RW	Byte 1 of timer0 ticker	00
0x632	TMR_TICK0_2	RW	Byte 2 of timer0 ticker	00
0x633	TMR_TICK0_3	RW	Byte 3 of timer0 ticker	00
0x634	TMR_TICK1_0	RW	Byte 0 of timer1 ticker	00
0x635	TMR_TICK1_1	RW	Byte 1 of timer1 ticker	00
0x636	TMR_TICK1_2	RW	Byte 2 of timer1 ticker	00
0x637	TMR_TICK1_3	RW	Byte 3 of timer1 ticker	00
0x638	TMR_TICK2_0	RW	Byte 0 of timer2 ticker	00
0x639	TMR_TICK2_1	RW	Byte 1 of timer2 ticker	00
0x63a	TMR_TICK2_2	RW	Byte 2 of timer2 ticker	00
0x63b	TMR_TICK2_3	RW	Byte 3 of timer2 ticker	00

5.1.2 Mode0 (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.

1st: Set initial Tick value of Timer0

Set Initial value of Tick via registers TMR_TICK0_0~TMR_TICK0_3 (address 0x630~0x633). Address 0x630 is lowest byte and 0x633 is highest byte. It's recommended to clear initial Timer Tick value to 0.

2nd: Set Capture value of Timer0

Set registers TMR_CAPT0_0~TMR_CAPT0_3 (address 0x624~0x627). Address 0x624 is lowest byte and 0x627 is highest byte.

3rd: Set Timer0 to Mode 0 and enable Timer0

Set register TMR_CTRL0 (address 0x620) [2:1] to 2b'00 to select Mode 0; Meanwhile set address 0x620[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.

5.1.3 Mode1 (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.

Note: Refer to **Section 7.1.2** for corresponding "**m0**", "**m1**", "**m2**" and "**Polarity**" register address.

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.

1ˢᵗ: Set initial Tick value of Timer1

Set Initial value of Tick via registers TMR_TICK1_0~TMR_TICK1_3 (address 0x634~0x637). Address 0x634 is lowest byte and 0x637 is highest byte. It's recommended to clear initial Timer Tick value to 0.

2ⁿᵈ: Set Capture value of Timer1

Set registers TMR_CAPT1_0~TMR_CAPT1_3 (address 0x628~0x62b). Address 0x628 is lowest byte and 0x62b is highest byte.

3ʳᵈ: 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.

4ᵗʰ: Set Timer1 to Mode 1 and enable Timer1

Set address 0x620[5:4] to 2b'01 to select Mode 1; Meanwhile set address 0x620[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.

5.1.4   Mode2 (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.

Note: Refer to **Section 7.1.2** for corresponding "**m0**", "**m1**", "**m2**" and "**Polarity**" register address.

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 Timer2 for Mode 2 is taken as an example.

1ˢᵗ: Set initial Timer2 Tick value

Set Initial value of Tick via registers TMR_TICK2_0~TMR_TICK2_3 (address 0x638~0x63b). Address 0x638 is lowest byte and 0x63b is highest byte. It's recommended to clear initial Timer Tick value to 0.

2ⁿᵈ: Select GPIO source and edge for Timer2

Select certain GPIO to be the clock source via setting "m2" register.

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

3rd: Set Timer2 to Mode 2 and enable Timer2

Set address 0x620[7:6] to 2b'01 and address 0x621 [0] to 1b'1.

Timer2 Tick is triggered by a positive/negative (specified during the 2nd step) edge of GPIO pulse. Timer2 starts counting upward and Timer2 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 Timer2 tick stops.

4th: Read current Timer2 Tick value to calculate GPIO pulse width

Read current Timer2 Tick value from address 0x638~0x63b.

Then GPIO pulse width is calculated as follows:

$$\text{GPIO pulse width} = \text{System clock period} * (\text{current Timer2 Tick} - \text{intial Timer2 Tick})$$

For initial Timer2 Tick value is set to the recommended value of 0, then:

$$\text{GPIO pulse width} = \text{System clock period} * \text{current Timer2 Tick}.$$

5.1.5 Mode3 (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.

1st: Set initial Tick value of Timer0

Set Initial value of Tick via address 0x630~0x633. Address 0x630 is lowest byte and address 0x633 is highest byte. It's recommended to clear initial Timer Tick value to 0.

2nd: Set Timer0 to Mode 3 and enable Timer0

Set address 0x620[2:1] to 2b'11 to select Mode 3, meanwhile set address 0x620[0] to 1b'1 to enable Timer0. Timer0 Tick starts to roll.

3rd: Read current Timer0 Tick value

Current Timer0 Tick value can be read from address 0x630~0x633.

5.1.6 Watchdog

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

Only Timer2 supports Watchdog.

Timer2 Tick has 32bits. Watchdog Capture has only 14bits, which consists of TMR_CTRL2 (address 0x622) [6:0] as higher bits and TMR_CTRL1 (address 0x621) [7:1] as lower bits. Chip will be reset when the Timer2 Tick[31:18] matches Watch dog capture.

1st: Clear Timer2 Tick value

Clear registers TMR_TICK2_0 ~TMR_TICK2_3 (address 0x638~0x63b). Address 0x638 is lowest byte and 0x63b is highest byte.

2nd: Enable Timer2

Set register TMR_CTRL0 (address 0x620) [6] to 1b'1 to enable Timer2.

3rd: Set 14-bit Watchdog Capture value and enable Watchdog

Set address 0x622[6:0] as higher bits of watchdog capture and 0x621[7:1] as lower bits. Meanwhile set address 0x622[7] to 1b'1 to enable Watchdog.

Then Timer2 Tick starts counting upwards from 0.

If bits[31:18] of Timer2 Tick value read from address 0x638~0x63b reaches watchdog capture, the chip will be reset, and the status bit in address 0x72[0] will be set as 1b'1 automatically. User can read the watchdog status bit after chip reset to check if the reset source is watchdog, and needs to write 1b'1 to this bit to manually clear the flag.

5.2 32K LTIMER

The TLSR8258F1KET48A 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.

5.3 System Timer

The TLSR8258F1KET48A also supports a System Timer. As introduced in section **4.3.1 System Timer clock**, the clock frequency for System Timer is fixed as 16MHz irrespective of system clock.

In suspend mode, both System Timer and Timer0~Timer2 stop counting, and 32K Timer starts counting. When the chip restores to active mode, Timer0~Timer2 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.

Table 5- 2   Register table for System Timer

Address	Mnemonic	R/W	Function	Default Value
0x740	SYS_TIMER0	R/W	[7:3]	0x00
0x741	SYS_TIMER1	R/W	[7:0]	0x00
0x742	SYS_TIMER2	R/W	[7:0]	0x00
0x743	SYS_TIMER3	R/W	[7:0] System timer counter, write to set initial value.	0x00

Address	Mnemonic	R/W	Function	Default Value
			This is the sys timer counter	
0x74c	SYS_TIMER_CTRL 0	R/W	[7]:cal 32k enable (16 cycles 32k, count sys clock cycles) [6]:1:at the pos of 32k clock to set 32k timer value [5]:suspend bypass system_timer module [4]:system timer ss enable [3] manual set 32k timer mode [2]:manual set 32k timer 1:write, 0: read [1]:irq mask 1: enable, 0: disable [0] rsvd	0x90
0x74f	SYS_TIMER_CTRL 1		[1]: timer_en, System Timer busy [2]: rsvd (clk32k_tm) [3]: m_wr_32k_en, busy reading/writing 32k Timer manually [7:4]: rsvd (ss)	0x00

*Note: The lower three bits of address 0x740 is invalid, therefore, the resolution should be 0.5us.

6 Interrupt System

6.1 Interrupt structure

The interrupting function is applied to manage dynamic program sequencing based on real-time events triggered by timers, pins and etc.

For the TLSR8258F1KET48A, there are 24 interrupt sources in all: 16 types are level-triggered interrupt sources (listed in address 0x640~0x641) and 8 types are edge-triggered interrupt sources (listed in address 0x642).

When CPU receives an interrupt request (IRQ) from some interrupt source, it will determine whether to respond to the IRQ. If CPU decides to respond, it pauses current routine and starts to execute interrupt service subroutine. Program will jump to certain code address and execute IRQ commands. After finishing interrupt service subroutine, CPU returns to the breakpoint and continues to execute main function.

6.2 Register configuration

Table 6- 1   Register table for Interrupt system

Address	Mnemonic	Type	Description	Reset Value
0x640	MASK_0	RW	Byte 0 interrupt mask, level-triggered type {irq_mix, irq_uart, irq_dfifo, irq_dma, usb_pwdn, time2, time1, time0} [7] irq_mix, i.e. irq_host_cmd [6] irq_uart [5] irq_dfifo [4] irq_dma [3] usb_pwdn [2] time2 [1] time1 [0] time0	0x00
0x641	MASK_1	RW	Byte 1 interrupt mask, level-triggered type {rsvd, irq_pwm, irq_zb_rt, irq_udc[4:0]} [7] rsvd [6] irq_pwm [5] irq_zb_rt [4] irq_udc[4] [3] irq_udc[3] [2] irq_udc[2] [1] irq_udc[1] [0] irq_udc[0]	0x00
0x642	MASK_2	RW	Byte 2 interrupt mask, edge-triggered type {rsvd, gpio2risc[1:0], irq_stimer, pm_irq, irq_gpio, usb_reset, usb_250us} [7] rsvd [6] gpio2risc[1] [5] gpio2risc[0] [4] irq_stimer [3] pm_irq_tm	0x00

Address	Mnemonic	Type	Description	Reset Value
			[2] irq_gpio [1] usb_reset [0] usb_250us	
0x643	IRQMODE	RW	[0] interrupt enable [1] reserved (Multi-Address enable)	0x00
0x644	PRIO_0	RW	Byte 0 of priority 1: High priority; 0: Low priority	0x00
0x645	PRIO_1	RW	Byte 1 of priority	0x00
0x646	PRIO_2	RW	Byte 2 of priority	0x00
0x648	IRQSRC_0	R	Byte 0 of interrupt source	0x00
0x649	IRQSRC_1	R	Byte 1 of interrupt source	0x00
0x64a	IRQSRC_2	R	Byte 2 of interrupt source	0x00

6.2.1   Enable/Mask interrupt sources

Various interrupt sources could be enabled or masked by registers MASK_0~MASK_2 (address 0x640~0x642).

Interrupt sources of level-triggered type:

✧   irq_mix (0x640[7]): I2C Slave mapping mode or SPI Slave interrupt (irq_host_cmd)

✧   irq_uart (0x640[6]): UART interrupt

✧   irq_dfifo (0x640[5]): DFIFO interrupt

✧   irq_dma (0x640[4]): DMA interrupt

✧   usb_pwdn (0x640[3]): USB Host has sent power down signal

✧   time2, time1, timer0 (0x640[2]~0x640[0]): Timer2~Timer0 interrupt

✧   irq_pwm (0x641[6]): PWM interrupt

✧   irq_zb_rt (0x641[5]): Baseband interrupt

✧   irq_udc[4:0] (0x641[4:0]): USB device interrupt

Interrupt sources of edge-triggered type:

✧   gpio2risc[1:0] (0x642[6]~0x642[5]): gpio2risc[1]~gpio2risc[0] interrupt, please refer to section **7.1.2**.

✧   irq_stimer (0x642[4]): System timer interrupt

✧   pm_irq_tm (0x642[3]): 32kHz timer wakeup interrupt

✧   irq_gpio (0x642[2]): GPIO interrupt, please refer to section **7.1.2**.

✧   usb_reset (0x642[1]): USB Host has sent reset command.

✧   usb_250us (0x642[0]): USB has been in idle for 250us.

6.2.2 Interrupt mode and priority

Interrupt mode is typically-used mode. Register IRQMODE (address 0x643)[0] should be set as 1b'1 to enable interrupt function.

IRQ tasks could be set as High or Low priority via registers PRIO_0~PRIO_2 (address 0x644~0x646). When more than one interrupt sources assert interrupt requests at the same time, CPU will respond depending on respective interrupt priority levels. It's recommended not to modify priority setting.

6.2.3 Interrupt source flag

Three bytes in registers IRQSRC_0~IRQSRC_2 (address 0x648~0x64a) serve to indicate IRQ sources. Once IRQ occurs from certain source, the corresponding IRQ source flag will be set as "1". User could identify IRQ source by reading address 0x648~0x64a.

When handling edge-triggered type interrupt, the corresponding IRQ source flag needs to be cleared via address 0x64a. Take the interrupt source usb_250us for example: First enable the interrupt source by setting address 0x642 bit[0] as 1b'1; then set address 0x643 bit[0] as 1b'1 to enable the interrupt. In interrupt handling function, 24-bit data is read from address 0x648~0x64a to check which IRQ source is valid; if data bit[16] is 1, it means the usb_250us IRQ source is valid. Clear this interrupt source by setting address 0x64a bit[0] as 1b'1.

As for level-type interrupt, IRQ interrupt source status needs to be cleared by setting corresponding module status register. Take Timer0 IRQ interrupt source for example: First enable the interrupt source by setting address 0x640 bit[0] as 1b'1; then set address 0x643 bit[0] as 1b'1 to enable the interrupt. In interrupt handling function, 24-bit data is read from address 0x648~0x64a to check which IRQ source is valid; if data bit[0] is 1, it means the Timer0 IRQ source is valid. Register TMR_STATUS (address 0x623) [0] should be written with 1b'1 to manually clear Timer0 status (refer to section **5.1.1 Register table**).

7 Interface

7.1 GPIO

The TLSR8258F1KET48A support up to 32 GPIOs respectively. All digital IOs can be used as general purpose IOs.

All GPIOs (including PA<0>~PD<7>) have configurable pull-up/pull-down resistor. Please refer to **section 7.1.3 Pull-up/Pull-down resistor** for details.

7.1.1 Basic configuration

7.1.1.1 GPIO lookup table

Table 7- 1GPIO lookup table 1

Pin	Default function	Pad Function Mux				GPIO Setting						Act as GPIO
		Register=2	Register=1	Register=0	Register	Input (R)	IE	OEN	Output	Polarity	DS	
DMIC_DI/ PWM0_N/ UART_RX/ PA<0>	GPIO	UART_RX	PWM0_N	DMIC_DI	0x5a8[1:0]	0x580[0]	0x581[0]	0x582[0]	0x583[0]	0x584[0]	0x585[0]	0x586[0]
DMIC_CLK/ 7816_CLK/ I2S_CLK/ PA<1>	GPIO	I2S_CLK	7816_CLK	DMIC_CLK	0x5a8[3:2]	0x580[1]	0x581[1]	0x582[1]	0x583[1]	0x584[1]	0x585[1]	0x586[1]
DO/ UART_TX/ PWM0/ PA<2>	GPIO	PWM0	UART_TX	DO	0x5a8[5:4]	0x580[2]	0x581[2]	0x582[2]	0x583[2]	0x584[2]	0x585[2]	0x586[2]
DI/ UART_CTS/ PWM1/ PA<3>	GPIO	PWM1	UART_CTS	DI	0x5a8[7:6]	0x580[3]	0x581[3]	0x582[3]	0x583[3]	0x584[3]	0x585[3]	0x586[3]
CK/ UART_RTS/ PWM2/ PA<4>	GPIO	PWM2	UART_RTS	CK	0x5a9[1:0]	0x580[4]	0x581[4]	0x582[4]	0x583[4]	0x584[4]	0x585[4]	0x586[4]
DM/ PA<5>	GPIO	/	/	DM	0x5a9[3:2]	0x580[5]	0x581[5]	0x582[5]	0x583[5]	0x584[5]	0x585[5]	0x586[5]
DP(SWS)/ PA<6>	GPIO	/	/	DP(SWS)	0x5a9[5:4]	0x580[6]	0x581[6]	0x582[6]	0x583[6]	0x584[6]	0x585[6]	0x586[6]
SWS/ UART_RTS/ PA<7>	SWS	/	UART_RTS	SWS	0x5a9[7:6]	0x580[7]	0x581[7]	0x582[7]	0x583[7]	0x584[7]	0x585[7]	0x586[7]
PWM3/ UART_RX/ ATSEL1/ sar_aio<0>/ PB<0>	GPIO	ATSEL1	UART_RX	PWM3	0x5aa[1:0]	0x588[0]	afe_0xbd [0]	0x58a[0]	0x58b[0]	0x58c[0]	afe_0xbf [0]	0x58e[0]

Pin	Default function	Pad Function Mux				GPIO Setting						Act as GPIO
		Register=2	Register=1	Register=0	Register	Input (R)	IE	OEN	Output	Polarity	DS	
PWM4/ UART_TX/ ATSEL2/ lc_comp_ain<1>/ sar_aio<1>/ PB<1>	GPIO	ATSEL2	UART_TX	PWM4	0x5aa[3:2]	0x588[1]	afe_0xbd [1]	0x58a[1]	0x58b[1]	0x58c[1]	afe_0xbf [1]	0x58e[1]
PWM5/ UART_CTS/ RX_CYC2LNA/ lc_comp_ain<2>/ sar_aio<2>/ PB<2>	GPIO	RX_CYC2LNA	UART_CTS	PWM5	0x5aa[5:4]	0x588[2]	afe_0xbd [2]	0x58a[2]	0x58b[2]	0x58c[2]	afe_0xbf [2]	0x58e[2]
PWM0_N/ UART_RTS/ TX_CYC2PA/ lc_comp_ain<3>/ sar_aio<3>/ PB<3>	GPIO	TX_CYC2PA	UART_RTS	PWM0_N	0x5aa[7:6]	0x588[3]	afe_0xbd [3]	0x58a[3]	0x58b[3]	0x58c[3]	afe_0xbf [3]	0x58e[3]
SDM_P0/ PWM4/ lc_comp_ain<4>/ sar_aio<4>/ PB<4>	GPIO	/	PWM4	SDM_P0	0x5ab[1:0]	0x588[4]	afe_0xbd [4]	0x58a[4]	0x58b[4]	0x58c[4]	afe_0xbf [4]	0x58e[4]
SDM_N0/ PWM5/ lc_comp_ain<5>/ sar_aio<5>/ PB<5>	GPIO	/	PWM5	SDM_N0	0x5ab[3:2]	0x588[5]	afe_0xbd [5]	0x58a[5]	0x58b[5]	0x58c[5]	afe_0xbf [5]	0x58e[5]
SDM_P1/ SPI_DI/ UART_RTS/ lc_comp_ain<6>/ sar_aio<6>/ PB<6>	SPI_DI	UART_RTS	SPI_DI	SDM_P1	0x5ab[5:4]	0x588[6]	afe_0xbd [6]	0x58a[6]	0x58b[6]	0x58c[6]	afe_0xbf [6]	0x58e[6]
SDM_N1/ SPI_DO/ UART_RX/ lc_comp_ain<7>/ sar_aio<7>/ PB<7>	SPI_DO	UART_RX	SPI_DO	SDM_N1	0x5ab[7:6]	0x588[7]	afe_0xbd [7]	0x58a[7]	0x58b[7]	0x58c[7]	afe_0xbf [7]	0x58e[7]
I2C_SDA/ PWM4_N/ UART_RTS/ PGA_P0/ PC<0>	GPIO	UART_RTS	PWM4_N	I2C_SDA	0x5ac[1:0]	0x590[0]	afe_0xc0 [0]	0x592[0]	0x593[0]	0x594[0]	afe_0xc2 [0]	0x596[0]

Pin	Default function	Pad Function Mux				GPIO Setting						Act as GPIO
		Register=2	Register=1	Register=0	Register	Input (R)	IE	OEN	Output	Polarity	DS	
I2C_SCK/ PWM1_N/ PWM0/ PGA_N0/ PC<1>	GPIO	PWM0	PWM1_N	I2C_SCK	0x5ac[3:2]	0x590[1]	afe_0xc0 [1]	0x592[1]	0x593[1]	0x594[1]	afe_0xc2 [1]	0x596[1]
PWM0/ 7816_TRX (UART_TX)/ I2C_SDA/ XC32K_O/ PGA_P1/ PC<2>	GPIO	I2C_SDA	7816_TRX (UART_TX)	PWM0	0x5ac[5:4]	0x590[2]	afe_0xc0 [2]	0x592[2]	0x593[2]	0x594[2]	afe_0xc2 [2]	0x596[2]
PWM1/ UART_RX/ I2C_SCK/ XC32K_I/ PGA_N1/ PC<3>	GPIO	I2C_SCK	UART_RX	PWM1	0x5ac[7:6]	0x590[3]	afe_0xc0 [3]	0x592[3]	0x593[3]	0x594[3]	afe_0xc2 [3]	0x596[3]
PWM2/ UART_CTS/ PWM0_N/ sar_aio<8>/ PC<4>	GPIO	PWM0_N	UART_CTS	PWM2	0x5ad[1:0]	0x590[4]	afe_0xc0 [4]	0x592[4]	0x593[4]	0x594[4]	afe_0xc2 [4]	0x596[4]
PWM3_N/ UART_RX/ ATSEL0/ sar_aio<9>/ PC<5>	GPIO	ATSEL0	UART_RX	PWM3_N	0x5ad[3:2]	0x590[5]	afe_0xc0 [5]	0x592[5]	0x593[5]	0x594[5]	afe_0xc2 [5]	0x596[5]
RX_CYC2LNA/ ATSEL1/ PWM4_N/ PC<6>	GPIO	PWM4_N	ATSEL1	RX_CYC2LNA	0x5ad[5:4]	0x590[6]	afe_0xc0 [6]	0x592[6]	0x593[6]	0x594[6]	afe_0xc2 [6]	0x596[6]
TX_CYC2PA/ ATSEL2/ PWM5_N/ PC<7>	GPIO	PWM5_N	ATSEL2	TX_CYC2PA	0x5ad[7:6]	0x590[7]	afe_0xc0 [7]	0x592[7]	0x593[7]	0x594[7]	afe_0xc2 [7]	0x596[7]
RX_CYC2LNA/ 7816_TRX (UART_TX)/ PD<0>	GPIO	7816_TRX (UART_TX)	/	RX_CYC2LNA	0x5ae[1:0]	0x598[0]	0x599[0]	0x59a[0]	0x59b[0]	0x59c[0]	0x59d[0]	0x59e[0]
TX_CYC2PA/ UART_CTS/ PD<1>	GPIO	UART_CTS	/	TX_CYC2PA	0x5ae[3:2]	0x598[1]	0x599[1]	0x59a[1]	0x59b[1]	0x59c[1]	0x59d[1]	0x59e[1]

Pin	Default function	Pad Function Mux				GPIO Setting						
		Register=2	Register=1	Register=0	Register	Input (R)	IE	OEN	Output	Polarity	DS	Act as GPIO
SPI_CN/ I2S_LR/ PWM3/ PD<2>	SPI_CN	PWM3	I2S_LR	SPI_CN	0x5ae[5:4]	0x598[2]	0x599[2]	0x59a[2]	0x59b[2]	0x59c[2]	0x59d[2]	0x59e[2]
PWM1_N/ I2S_SDI/ 7816_TRX (UART_TX)/ PD<3>	GPIO	7816_TRX (UART_TX)	I2S_SDI	PWM1_N	0x5ae[7:6]	0x598[3]	0x599[3]	0x59a[3]	0x59b[3]	0x59c[3]	0x59d[3]	0x59e[3]
SWM/ I2S_SDO/ PWM2_N/ PD<4>	GPIO	PWM2_N	I2S_SDO	SWM	0x5af[1:0]	0x598[4]	0x599[4]	0x59a[4]	0x59b[4]	0x59c[4]	0x59d[4]	0x59e[4]
PWM0/ PWM0_N/ PD<5>	GPIO	PWM0_N	/	PWM0	0x5af[3:2]	0x598[5]	0x599[5]	0x59a[5]	0x59b[5]	0x59c[5]	0x59d[5]	0x59e[5]
CN/ UART_RX/ ATSEL0/ PD<6>	GPIO	ATSEL0	UART_RX	CN	0x5af[5:4]	0x598[6]	0x599[6]	0x59a[6]	0x59b[6]	0x59c[6]	0x59d[6]	0x59e[6]
SPI_CK/ I2S_BCK/ 7816_TRX (UART_TX) /PD<7>	SPI_CK	7816_TRX (UART_TX)	I2S_BCK	SPI_CK	0x5af[7:6]	0x598[7]	0x599[7]	0x59a[7]	0x59b[7]	0x59c[7]	0x59d[7]	0x59e[7]

*Notes:

(1) IE: Input enable, high active. 1: enable input, 0: disable input.

(2) OEN: Output enable, low active. 0: enable output, 1: disable output.

(3) Register: Configure multiplexed functions in "Pad Function Mux" column.

(4) Output: configure GPO output.

(5) Input: read GPI input.

(6) DS: Drive strength. Default: 1 (high DS level).

(7) Act as GPIO: enable (1) or disable (0) GPIO function.

(8) Polarity: see section 7.1.2   Connection relationship between GPIO and related modules.

(9) Priority: "Act as GPIO" has the highest priority. To configure as multiplexed function, disable GPIO function first.

(10) afe_0xbd, afe_0xbf, afe_0xc0 and afe_0xc2 marked in red color are analog registers; others are digital registers.

(11) For all unused GPIOs, corresponding "IE" must be set as 0.

(12) When SWS/PA<7> "IE" is set as 1, this pin must be fixed as pull-up/pull-down state (float state is not allowed).

(13) To use PD<6>/CN as SPI Slave CN function, it's needed to enable GPIO function for PD<2>/SPI_CN, because PD<2> has higher priority than PD<6>.

(14) To use SAR ADC/low power comparator/PGA pin function, please refer to corresponding module sections.

(15) The GPIO pins with 7816_TRX function, including PC<2>, PD<0>, PD<3> and PD<7>, can also act as UART_TX, but DO NOT support UART_RX.

7.1.1.2 GPIO Logic Introductioin
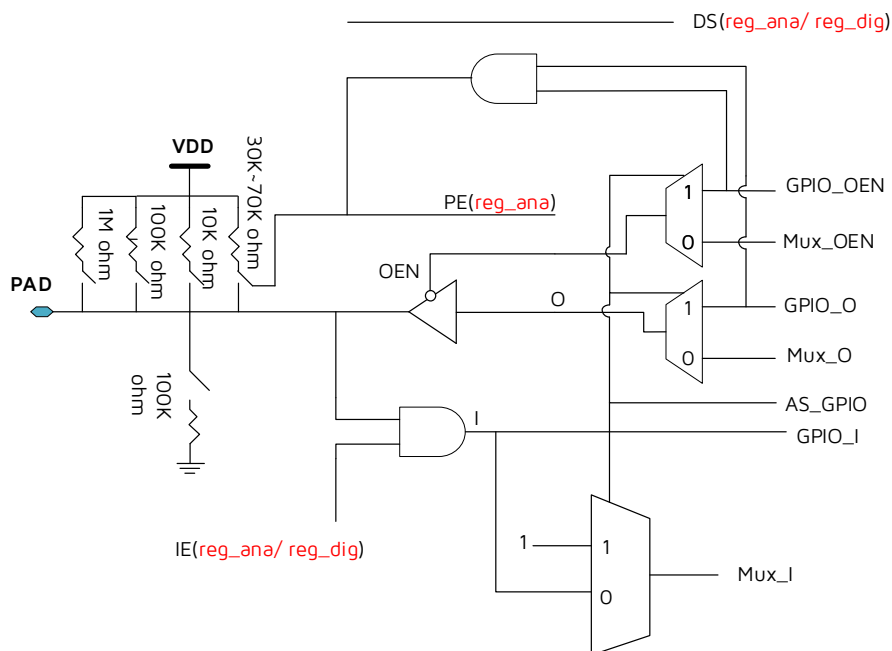
Figure 7- 1   GPIO logic diagram

In the figure above,

(1) DS: drive strength, 1: high drive strength; 0: low drive strength

(2) PE: pull-up enable, 1: pull up; 0: no pull up

(3) OEN: output enable, 1: high Z; 0: output

(4) O: output value, when OEN is 0, output this value

(5) I: input value

(6) IE: input enable, if IE is 0, C is always zero

(7) 1M, 10K, pull up and 100K pull down resistors are controlled by analog 3.3V register controller

*Notes:

(1) When PAD is set as functional IO, no need to configure GPIO_OEN as the functional IO will enable Mux_OEN.

(2) When PAD is input, IE should be enabled regardless of functional IO or GPIO, and output to I, AS_GPIO is 1, Mux_I is 1.

(3) There are two methods to configure digital pull-up of 30k~70k ohm:

✦ PC group and PD group (may vary for different chips), pad can configure analog register PE and enable digital pull-up.

✦ Other group of pad, when GPIO_OEN=1 and GPIO_I=1, it enables digital pull-up.

(4) Analog pull-up has three options: 1M, 100k, 10k ohm; analog pull-down has only 100k ohm. They can be configured via corresponding analog registers.

(5) The GPIO configuration sequence should be: configure the MUX function, and then disable GPIO function. If disable GPIO first and then set function, the default function of the pad may be enabled and will cause false output level.

7.1.1.3 Multiplexed functions

Each pin listed in Table 7-1 acts as the function in the "**Default Function**" column by default.

✦ PA<7> acts as SWS function by default.

✦ PB<6:7>, PD<2,7> act as SPI function by default.

✦ The other digital IOs act as GPIO function by default.

If a pin with multiplexed functions does not act as GPIO function by default, to use it as GPIO, first set the bit in "**Act as GPIO**" column as 1b'1. After GPIO function is enabled, if the pin is used as output, both the bits in "**IE**" and "**OEN**" columns should be set as 1b'0, then set the register value in the "**Output**" column; if the pin is used as input, both the bits in "**IE**" and "**OEN**" columns should be set as 1b'1, and the input data can be read from the register in the "**Input**" column.

To use a pin as certain multiplexed function (neither the default function nor GPIO function), first clear the bit in "**Act as GPIO**" column to disable GPIO function, and then configure "**Register**" in "Pad Function Mux" column to enable multiplexed function correspondingly.

Example 1: DMIC_DI/ PWM0_N/ UART_RX/ PA<0>.

(1) The pin acts as GPIO function by default.

✦ If the pin is used as general output, both address 0x581[0] (IE) and 0x582[0] (OEN) should be set as 1b'0, then configure address 0x583[0] (Output).

✦ If the pin is used as general input, both address 0x581[0] (IE) and 0x582[0] (OEN) should be set as 1b'1, and the input data can be read from address 0x580[0] (Input).

(2) To use the pin as DMIC_DI function, address 0x586[0] (Act as GPIO) should be set as 1b'0, and 0x5a8[1:0] (Register) should be set as 2b'00.

(3) To use the pin as PWM0_N function, address 0x586[0] (Act as GPIO) should be set as 1b'0, and 0x5a8[1:0] (Register) should be set as 2b'01.

(4) To use the pin as UART_RX function, address 0x586[0] (Act as GPIO) should be set as 1b'0, and 0x5a8[1:0] (Register) should be set as 2b'10.

Example 2: SWS/ UART_RTS/ PA<7>.

(1)   The pin acts as SWS function by default.

(2)   To use it as GPIO function, first set address 0x586[7] (Act as GPIO) as 1b'1.

✧   If the pin is used as general output, both address 0x581[7] (IE) and 0x582[7] (OEN) should be set as 1b'0, then configure address 0x583[7] (Output).

✧   If the pin is used as general input, both address 0x581[7] (IE) and 0x582[7] (OEN) should be set to 1b'1, and the input data can be read from address 0x580[7] (Input).

(3)   To use it as UART_RTS function, set address 0x586[7] (Act as GPIO) as 1b'0, and set 0x5a9[7:6] (Register) to 2b'01.

I2C can also be multiplexed with SPI interface, i.e. I2C_SDA/I2C_SCK can be multiplexed with SPI_DI (DI)/SPI_CK (CK) respectively.

To select multiplexed SPI/I2C function, please follow the steps below:

1)   Disable GPIO function by setting corresponding "Act as GPIO" as 1b'0.

2)   Select SPI/I2C function by setting corresponding "Register".

3)   Address 0x5b6[7:4] serve to select SPI or I2C output.

4)   Address 0x5b7[7:0] serve to select SPI input or I2C input.

Table 7- 2 Select multiplexed SPI/I2C

Pin with multiplexed SPI/I2C	Act as GPIO	Register	SPI Input Select	I2C Input Select	SPI/I2C Output Select
DI/UART_CTS/PWM1/PA<3>	0x586[3]=0 Disable GPIO	0x5a8[7:6]=0 Select DI (I2C_SDA)	5b7[0] 1: as SPI input. 0: not as SPI input.	5b7[4] 1: as I2C input. 0: not as I2C input.	0x5b6[4] 1: as SPI/I2C output 0: not as SPI/I2C output
CK/UART_RTS/PWM2/PA<4>	0x586[4]=0 Disable GPIO	0x5a9[1:0]=0 Select CK (I2C_SCK)	5b7[1] 1: as SPI input. 0: not as SPI input.	5b7[5] 1: as I2C input. 0: not as I2C input.	0x5b6[5] 1: as SPI/I2C output 0: not as SPI/I2C output
SDM_P1/SPI_DI/UART_RTS/ lc_comp_ain<6>/sar_aio<6>/PB<6>	0x58e[6]=0 Disable GPIO	0x5ab[5:4]=1 Select SPI_DI (I2C_SDA) (default function)	5b7[2] 1: as SPI input. 0: not as SPI input.	5b7[6] 1: as I2C input. 0: not as I2C input.	0x5b6[6] 1: as SPI/I2C output 0: not as SPI/I2C output
SPI_CK/I2S_BCK/7816_TRX(UART_TX)/PD<7>	0x59e[7] =0 Disable GPIO	0x5af[7:6] =0 Select SPI_CK (I2C_SCK) (default function)	5b7[3] 1: as SPI input. 0: not as SPI input.	5b7[7] 1: as I2C input. 0: not as I2C input.	0x5b6[7] 1: as SPI/I2C output 0: not as SPI/I2C output

7.1.1.4   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.

✧   PA<5:7>, PB<0:3>: maximum=8mA ("DS"=1), minimum=4mA ("DS"=0)

✧   PB<4:7>: maximum=16mA ("DS"=1), minimum=12mA ("DS"=0)

✧   Other GPIOs (PA<0:4>, PC<0:7> and PD<0:7>): maximum=4mA ("DS"=1), minimum=2mA ("DS"=0)

7.1.2   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/m2", and generates counting signal in Mode 1 or control signal in Mode 2 for Timer0/Timer1/Timer2, 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);

Counting (Mode 1) or control (Mode 2) signal for Timer2 = | ((input ^ polarity) & m2);

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

GPIO2RISC[1] interrupt request signal = | ((input ^ polarity) & m1).
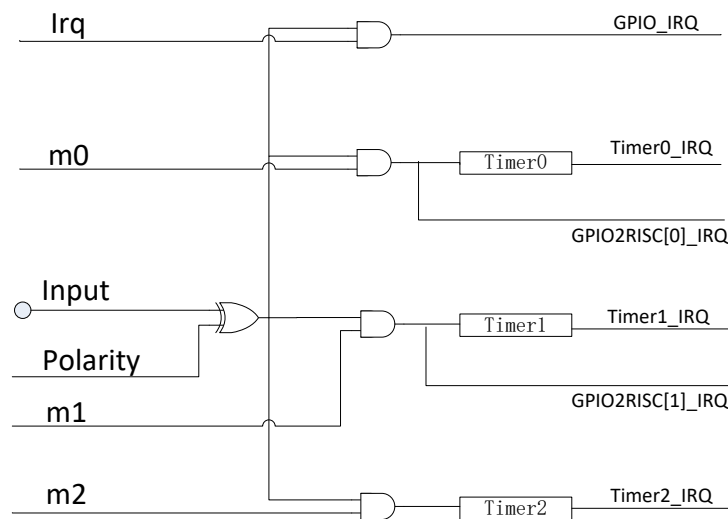
Figure 7- 2   Logic relationship between GPIO and related modules

Please refer to Table 7- 3 and Table 6- 1 to learn how to configure GPIO for interrupt system or Timer/Counter (Mode 1 or Mode 2).

(1)   First enable GPIO function, enable IE and disable OEN. Please see section **7.1.1 Basic configuration**.

(2) GPIO IRQ signal:

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

Then set address 0x5b5[3] (irq_enable) to enable GPIO IRQ.

Finally enable GPIO interrupt (irq_gpio) via address 0x642[2].

User can read addresses 0x5e0 ~ 0x5e3 to see which GPIO asserts GPIO interrupt request signal. Note: 0x5e0[7:0] --> PA<7>~PA<0>, 0x5e1[7:0] --> PB<7>~PB<0>, 0x5e2[7:0] --> PC<7>~PC<0>, 0x5e3[7:0] --> PD<7>~PD<0>.

(3) Timer/Counter counting or control signal:

Configure "**Polarity**". 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.

Then set "**m0/m1/m2**" to specify the GPIO which generates counting signal (Mode 1)/control signal (Mode 2) for Timer0/Timer1/Timer2.

User can read addresses 0x5e8~0x5eb/0x5f0~0x5f3/0x5f8~0x5fb to see which GPIO asserts counting signal (in Mode 1) or control signal (in Mode 2) for Timer0/Timer1/Timer2. Note: Timer0: 0x5e8[7:0] --> PA<7>~PA<0>, 0x5e9[7:0] --> PB<7>~PB<0>, 0x5ea[7:0] --> PC<7>~PC<0>, 0x5eb[7:0] --> PD<7>~PD<0>; Timer1: 0x5f0[7:0] --> PA<7>~PA<0>, 0x5f1[7:0] --> PB<7>~PB<0>, 0x5f2[7:0] --> PC<7>~PC<0>, 0x5f3[7:0] --> PD<7>~PD<0>; Timer2: 0x5f8[7:0] --> PA<7>~PA<0>, 0x5f9[7:0] --> PB<7>~PB<0>, 0x5fa[7:0] --> PC<7>~PC<0>, 0x5fb[7:0] --> PD<7>~PD<0>.

(4) GPIO2RISC IRQ signal:

Select GPIO2RISC interrupt trigger edge (positive edge or negative edge) via configuring "**Polarity**", and set corresponding GPIO enabling bit **"m0"/"m1"**.

Enable GPIO2RISC[0]/GPIO2RISC[1] interrupt, i.e. "gpio2risc[0]" (address 0x642[5]) / "gpio2risc[1]"(address 0x642[6]).

Table 7- 3 GPIO lookup table2

Pin	Input (R)	Polarity 1: active low 0: active high	Irq	m0	m1	m2
PA<0>	0x580[0]	0x584[0]	0x587[0]	0x5b8[0]	0x5c0[0]	0x5c8[0]
PA<1>	0x580[1]	0x584[1]	0x587[1]	0x5b8[1]	0x5c0[1]	0x5c8[1]
PA<2>	0x580[2]	0x584[2]	0x587[2]	0x5b8[2]	0x5c0[2]	0x5c8[2]
PA<3>	0x580[3]	0x584[3]	0x587[3]	0x5b8[3]	0x5c0[3]	0x5c8[3]
PA<4>	0x580[4]	0x584[4]	0x587[4]	0x5b8[4]	0x5c0[4]	0x5c8[4]
PA<5>	0x580[5]	0x584[5]	0x587[5]	0x5b8[5]	0x5c0[5]	0x5c8[5]
PA<6>	0x580[6]	0x584[6]	0x587[6]	0x5b8[6]	0x5c0[6]	0x5c8[6]
PA<7>	0x580[7]	0x584[7]	0x587[7]	0x5b8[7]	0x5c0[7]	0x5c8[7]
PB<0>	0x588[0]	0x58c[0]	0x58f[0]	0x5b9[0]	0x5c1[0]	0x5c9[0]

Pin	Input (R)	Polarity 1: active low 0: active high	Irq	m0	m1	m2
PB<1>	0x588[1]	0x58c[1]	0x58f[1]	0x5b9[1]	0x5c1[1]	0x5c9[1]
PB<2>	0x588[2]	0x58c[2]	0x58f[2]	0x5b9[2]	0x5c1[2]	0x5c9[2]
PB<3>	0x588[3]	0x58c[3]	0x58f[3]	0x5b9[3]	0x5c1[3]	0x5c9[3]
PB<4>	0x588[4]	0x58c[4]	0x58f[4]	0x5b9[4]	0x5c1[4]	0x5c9[4]
PB<5>	0x588[5]	0x58c[5]	0x58f[5]	0x5b9[5]	0x5c1[5]	0x5c9[5]
PB<6>	0x588[6]	0x58c[6]	0x58f[6]	0x5b9[6]	0x5c1[6]	0x5c9[6]
PB<7>	0x588[7]	0x58c[7]	0x58f[7]	0x5b9[7]	0x5c1[7]	0x5c9[7]
PC<0>	0x590[0]	0x594[0]	0x597[0]	0x5ba[0]	0x5c2[0]	0x5ca[0]
PC<1>	0x590[1]	0x594[1]	0x597[1]	0x5ba[1]	0x5c2[1]	0x5ca[1]
PC<2>	0x590[2]	0x594[2]	0x597[2]	0x5ba[2]	0x5c2[2]	0x5ca[2]
PC<3>	0x590[3]	0x594[3]	0x597[3]	0x5ba[3]	0x5c2[3]	0x5ca[3]
PC<4>	0x590[4]	0x594[4]	0x597[4]	0x5ba[4]	0x5c2[4]	0x5ca[4]
PC<5>	0x590[5]	0x594[5]	0x597[5]	0x5ba[5]	0x5c2[5]	0x5ca[5]
PC<6>	0x590[6]	0x594[6]	0x597[6]	0x5ba[6]	0x5c2[6]	0x5ca[6]
PC<7>	0x590[7]	0x594[7]	0x597[7]	0x5ba[7]	0x5c2[7]	0x5ca[7]
PD<0>	0x598[0]	0x59c[0]	0x59f[0]	0x5bb[0]	0x5c3[0]	0x5cb[0]
PD<1>	0x598[1]	0x59c[1]	0x59f[1]	0x5bb[1]	0x5c3[1]	0x5cb[1]
PD<2>	0x598[2]	0x59c[2]	0x59f[2]	0x5bb[2]	0x5c3[2]	0x5cb[2]
PD<3>	0x598[3]	0x59c[3]	0x59f[3]	0x5bb[3]	0x5c3[3]	0x5cb[3]
PD<4>	0x598[4]	0x59c[4]	0x59f[4]	0x5bb[4]	0x5c3[4]	0x5cb[4]
PD<5>	0x598[5]	0x59c[5]	0x59f[5]	0x5bb[5]	0x5c3[5]	0x5cb[5]
PD<6>	0x598[6]	0x59c[6]	0x59f[6]	0x5bb[6]	0x5c3[6]	0x5cb[6]
PD<7>	0x598[7]	0x59c[7]	0x59f[7]	0x5bb[7]	0x5c3[7]	0x5cb[7]

7.1.3   Pull-up/Pull-down resistor

All GPIOs (including PA<0>~PD<7>) 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_0x15<7:0> serve to control the pull-up/pull-down resistor for each GPIO.

The DP pin also supports 1.5kΩ pull-up resistor for USB use. The 1.5kΩ pull up resistor is disabled by default and can be enabled by setting analog register afe_0x0b<7> as 1b'1. For the DP/PA<6> pin, user can only enable either 1.5kΩ pull-up resistor or pull-up resistor of rank x1/x100 / pull-down resistor of rank x10 at the same time. Please refer to Table 7- 4 for details.

Take the PA<0> for example: Setting analog register afe_0x0e<1:0> to 2b'01/2b'11/2b'10 is to respectively enable pull-up resistor of rank x100/pull-up resistor of rank x1/pull-down resistor of rank x10 for PA<0>; Clearing the two bits (default value) disables pull-up and pull-down resistor for PA<0>.

Table 7- 4   Analog registers for pull-up/pull-down resistor control

Address	Mnemonic	Default	Description
afe_0x0b<7>	dp_pullup_res_3v	0	1.5k (Typ.) pull-up resistor for USB DP PAD 0: disable 1: enable

Address	Mnemonic	Default	Description
Rank	Typical value (depend on actual application)		
x1	18kohm		
x10	160kohm		
x100	1Mohm		
afe_0x0e<7:0>	a_sel<7:0>	00000000	PA<3:0> pull up and down select: <7:6>: PA<3> <5:4>: PA<2> <3:2>: PA<1> <1:0>: PA<0> 00: Null 01: x100 pull up 10: x10 pull down 11: x1 pull up
afe_0x0f<7:0>	a_sel<15:8>	00000000	PA<7:4> pull up and down select: <7:6>: PA<7> <5:4>: PA<6> <3:2>: PA<5> <1:0>: PA<4> 00: Null 01: x100 pull up 10: x10 pull down 11: x1 pull up
afe_0x10<7:0>	b_sel<7:0>	00000000	PB<3:0> pull up and down select: 00: Null 01: x100 pull up 10: x10 pull down 11: x1 pull up
afe_0x11<7:0>	b_sel<15:8>	00000000	PB<7:4> pull up and down select: 00: Null 01: x100 pull up 10: x10 pull down 11: x1 pull up
afe_0x12<7:0>	c_sel<7:0>	00000000	PC<3:0> pull up and down select: 00: Null 01: x100 pull up 10: x10 pull down 11: x1 pull up
afe_0x13<7:0>	c_sel<15:8>	00000000	PC<7:4> pull up and down select: 00: Null 01: x100 pull up 10: x10 pull down 11: x1 pull up

Address	Mnemonic	Default	Description
afe_0x14<7:0>	d_sel<7:0>	00000000	PD<3:0> pull up and down select: 00: Null 01: x100 pull up 10: x10 pull down 11: x1 pull up
afe_0x15<7:0>	d_sel<15:8>	00000000	PD<7:4> pull up and down select: 00: Null 01: x100 pull up 10: x10 pull down 11: x1 pull up

7.2   SWM and SWS

The TLSR8258F1KET48A supports Single Wire 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.

7.3   I2C

The TLSR8258F1KET48A 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).

7.3.1   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 3.3kohm 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.
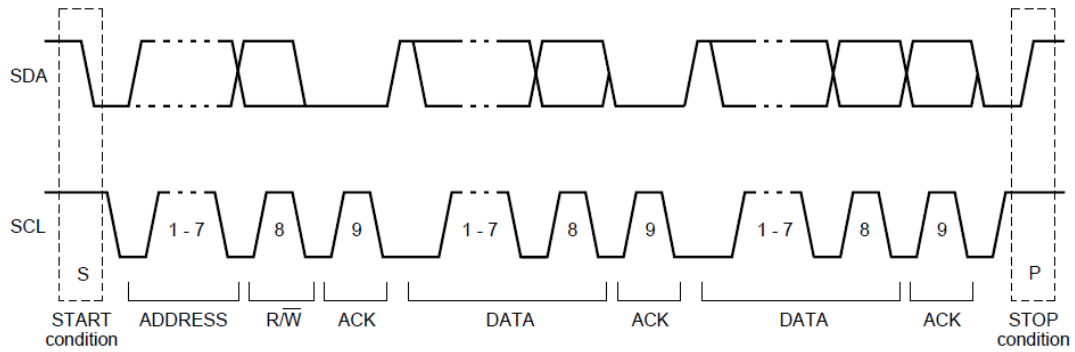
Figure 7- 3   I2C timing chart

7.3.2   Register table

Table 7- 5   Register configuration for I2C

Address	Name	R/W	Description	Reset Value
0x00	I2CSP	RW	I2C master clock speed	0x1f
0x01	I2C_ID	RW	[7:1] I2C ID	0x5c
0x02	I2CMST	RW	[0]: master busy [1]: master packet busy [2]: master received status 0 for ACK; 1 for NAK	0x00
0x03	I2CSCT0	RW	[0]: address auto increase enable [1]: I2C master enable [2]: enable Mapping Mode [3]: r_clk_stretch_en, suspend transmission by pulling SCL down to low level, and continue transmission after SCL is released to high level	0x01
0x04	I2CAD	RW	[7:0] data buffer in master mode	0x5a
0x05	I2CDW	RW	[7:0] Data buffer in master mode	0xf1
0x06	I2CDR	RW	[7:0] Data buffer for Read or Write in master mode	0x00
0x07	I2CSCT1	RW	[0]: launch ID cycle [1]: launch address cycle (send I2CAD data) [2]: launch data write cycle [3]: launch data read cycle For Master Write: 0: I2CAD&I2CDW, 1: I2CAD&I2CDW&I2CDR) To write 3 bytes: bit[3]=1; To write 2 bytes: bit[3]=0. For Master Read: always 1. [4]: launch start cycle [5]: launch stop cycle [6]: enable read ID [7]: enable ACK in read command	0x00

Address	Name	R/W	Description	Reset Value
0xe0	I2CMAP_HADR	R	[6:0] I2C read address	0x00
0xe1	HOSR_ADR_L	RW	Low byte of Mapping mode buffer address	0x80
0xe2	HOSR_ADR_M	RW	Middle byte of Mapping mode buffer address	0xd7
0xe3	HOSR_ADR_H	RW	High byte of Mapping mode buffer address	0x00
0xe4	I2CMAP_HOST	RW	[0]: host_cmd_irq_o, I2C host operation has happened. Write 1 to clear. [1]: host_rd_tag_o, I2C host operation has happened and is read operation. Write 1 to clear.	0x00

7.3.3   I2C Slave mode

I2C module of the TLSR8258F1KET48A acts as Slave mode by default. I2C slave address can be configured via register I2C_ID (address 0x01) [7:1].
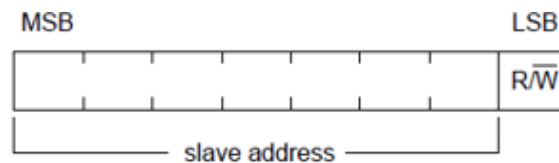
Figure 7- 4   Byte consisted of slave address and R/W flag bit

I2C slave mode supports two sub modes including Direct Memory Access (DMA) mode and Mapping mode, which is selectable via address 0x03[2].

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.

7.3.3.1   DMA mode

In DMA mode, other devices (Master) could access (read/write) designated address in Register and/or SRAM of the TLSR8258F1KET48A according to I2C protocol. I2C module of the TLSR8258F1KET48A will execute the read/write command from I2C master automatically. But user needs to notice that the system clock shall be at least 10x faster than I2C bit rate.

The access address designated by Master is offset by 0x800000. In the TLSR8258F1KET48A, Register address starts from 0x800000 and SRAM address starts from 0x840000. For example, if Addr High (AddrH) is 0x04, Addr Middle (AddrM) is 0x00, and Addr Low (AddrL) is 0xcc, the real address of accessed data is 0x8400cc.

In DMA mode, Master could read/write data byte by byte. The designated access address is initial address and it supports auto increment by setting address 0x03[0] to 1b'1.

Read Format in DMA mode
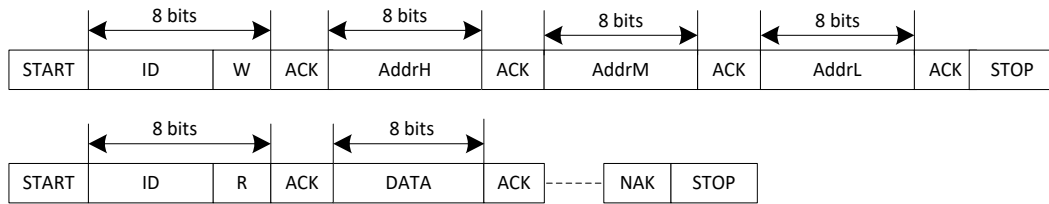
Figure 7- 5   Read format in DMA mode

Write Format in DMA mode
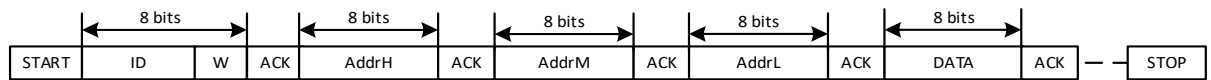
Figure 7- 6   Write format in DMA mode

7.3.3.2   Mapping mode

Mapping mode could be enabled via setting register I2CSCT0 (address 0x03)[2] to 1b'1.

In Mapping mode, data written and read by I2C master will be redirected to specified 128-byte buffer in SRAM. User could specify the initial address of the buffer by configuring registers HOSR_ADR_L (address 0xe1, lower byte), HOSR_ADR_M (address 0xe2, middle byte) and HOSR_ADR_H (address 0xe3, higher byte). The first 64-byte buffer is for written data and following 64-byte buffer is for read data. Every time the data access will start from the beginning of the Write-buffer/Read-buffer after I2C stop condition occurs. The last accessed data address could be checked in register I2CMAP_HADR (address 0xe0) [6:0] which is only updated after I2C STOP occurs.

Read Format in mapping mode
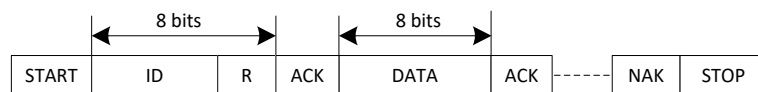
Figure 7- 7   Read format in Mapping mode

Write Format in mapping mode
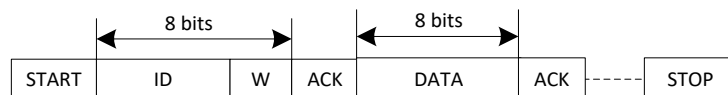
Figure 7- 8   Write format in Mapping mode

7.3.4   I2C Master mode

Address 0x03[1] should be set to 1b'1 to enable I2C master mode for the TLSR8258F1KET48A.

Address 0x00 serves to set I2C Master clock: F_{I2C} = (System Clock / (4 *clock speed configured in address 0x00).

A complete I2C protocol contains START, Slave Address, R/W bit, data, ACK and STOP. Slave address could be configured via address 0x01[7:1].

I2C Master (i.e. I2C module of the TLSR8258F1KET48A) could send START, Slave Address, R/W bit, data and STOP cycle by configuring address 0x07. I2C master will send enabled cycles in the correct sequence.

Address 0x02 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.

7.3.4.1 I2C Master Write transfer

I2C Master has 3-byte buffer for write data, which are I2CAD (0x04), I2CDW (0x05) and I2CDR (0x06). Write transfer will be completed by I2C master module.

For example, to implement an I2C write transfer with 3-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 and STOP, user needs to configure I2C slave address to I2C_ID (0x01) [7:1], 1st byte data to I2CAD, 2nd byte data to I2CDW and 3rd byte to I2CDR. To start I2C write transfer, I2CSCT1 (0x07) is configured to 0x3f (0011 1111). I2C Master will launch START, Slave address, Write bit, load ACK to I2CMST (0x02) [2], send I2CAD data, load ACK to I2CMST[2], send I2CDW data, load ACK to I2CMST[2], send I2CDR data, load ACK to I2CMST[2] and then STOP sequentially.

For I2C write transfer whose data are more than 3 bytes, user could split the cycles according to I2C protocol.

7.3.4.2 I2C Master Read transfer

I2C Master has one byte buffer for read data, which is I2CDR (0x06). Read transfer will be completed by I2C Master.

For example, to implement an I2C read transfer with 1 byte data, which contains START, Slave Address, Read bit, Ack from Slave, 1st byte from Slave, Ack by master and STOP, user needs to configure I2C slave address to I2C_ID (0x01) [7:1]. To start I2C read transfer, I2CSCT1 (0x07) is configured to 0xf9 (1111 1001). I2C Master will launch START, Slave address, Read bit, load ACK to I2CMST (0x02) [2], load data to I2CDR, reply ACK and then STOP sequentially.

For I2C read transfer whose data are more than 1 byte, user could split the cycles according to I2C protocol.

7.3.5 I2C and SPI Usage

I2C hardware and SPI hardware modules in the chip share part of the hardware, as a result, when both hardware interfaces are used, the restrictions listed within this section need to be taken into consideration.

I2C and SPI hardware cannot be used as Slave at the same time.

The other cases are supported, including:

✧ I2C Slave and SPI Master can be used at the same time.

✧ I2C Master and SPI Slave can be used at the same time.

✧ I2C and SPI can be used as Master at the same time.

Please refer to corresponding SDK instructions for details.

7.4 SPI

The TLSR8258F1KET48A embeds SPI (Serial Peripheral interface), which could act as Master mode or Slave mode. SPI is a high-speed, half-duplex and synchronous communication bus requiring 4 bus lines including a chip select (CS) line, a data input (DI) line, a data output (DO) line and a clock (CK) line.

7.4.1 Register table

Table 7- 6   Register configuration for SPI

Address	Name	R/W	Description	Reset Value
0x08	SPIDAT	RW	[7:0]: SPI data access	0x00
0x09	SPICT	RW	[0]: mst_csn, control SPI_CSN output when SPI acts as Master [1]: enable master mode [2]: spi data output disable [3]: 1 for read command; 0 for write command [4]: address auto increase [5]: share_mode [6]: busy status	0x11
0x0a	SPISP	RW	[6:0]: SPI clock speed [7]: SPI function mode, p_csn, p_scl, p_sda and p_sdo function as SPI if 1	0x05
0x0b	SPIMODE	RW	[0]: inverse SPI clock output [1]: data delay half clk	0x00

7.4.2 SPI Master mode

SPI for the TLSR8258F1KET48A supports both master mode and slave mode and acts as slave mode by default. Address 0x09 bit[1] should be set to 1b'1 to enable SPI Master mode. Register SPISP is to configure SPI pin and clock: setting address 0x0a bit[7] to 1 is to enable SPI function mode, and corresponding pins can be used as SPI pins; SPI clock = system clock/((clock speed configured in address 0x0a bit[6:0] +1)*2).

Address 0x08 serves as the data register. One reading/writing operation of 0x08 enables the SPI_CK pin to generate 8 SPI clock cycles.

Telink SPI supports four standard working modes: Mode 0~Mode 3. Register SPIMODE (address 0x0b) serves to select one of the four SPI modes:

Table 7- 7   SPI Master mode

SPI mode	CPOL/CPHA	SPIMODE register (Address 0x0b)
Mode 0	CPOL=0, CPHA=0	bit[0]=0, bit[1]=0
Mode 1	CPOL=0, CPHA=1	bit[0]=0, bit[1]=1
Mode 2	CPOL=1, CPHA=0	bit[0]=1, bit[1]=0
Mode 3	CPOL=1, CPHA=1	bit[0]=1, bit[1]=1
CPOL: Clock Polarity When CPOL=0, SPI_CLK keeps low level in idle state; When CPOL=1, SPI_CLK keeps high level in idle state. CPHA: Clock Phase When CPHA=0, data is sampled at the first edge of clock period When CPHA=1, data is sampled at the latter edge of clock period		

Address 0x09 bit[0] is to control the CS line: when the bit is set to 1, the CS level is high; when the bit is cleared, the CS level is low.

Address 0x09 bit[2] is the disabling bit for SPI Master output. When the bit is cleared, MCU writes data into address 0x08, then the SPI_DO pin outputs the data bit by bit during the 8 clock cycles generated by the SPI_CK pin. When the bit is set to 1b'1, SPI_DO output is disabled.

Address 0x09 bit[3] is the enabling bit for SPI Master reading data function. When the bit is set to 1b'1, MCU reads the data from address 0x08, then the input data from the SPI_DI pin is shifted into address 0x08 during the 8 clock cycles generated by the SPI_CK pin. When the bit is cleared, SPI Master reading function is disabled.

Address 0x09[5] is the enabling bit for share mode, i.e. whether SPI_DI and SPI_DO share one common line.

Users can read address 0x09 bit[6] to get SPI busy status, i.e. whether the 8 clock pulses have been sent.

7.4.3 SPI Slave mode

SPI for the TLSR8258F1KET48A acts as slave mode by default. SPI Slave mode supports DMA. User could access registers of the TLSR8258F1KET48A by SPI interface. It's noted that system clock of TLSR8258F1KET48A shall be at least 5x faster than SPI clock for reliable connection. Address 0x0a should be written with data 0xa5 by the SPI host to activate SPI slave mode. SPI salve only supports Mode0 and Mode3.

Table 7- 8SPI Slave mode

SPI slave mode	CPOL/CPHA
Mode 0	CPOL=0, CPHA=0
Mode 3	CPOL=1, CPHA=1
Receive data at positive edge of SPI MCLK clock. Send data at negative edge of SPI MCLK clock.	

Address 0x09[4] is dedicated for SPI Slave mode and indicates address auto increment. SPI write command format and read command format are illustrated in Figure 7-8:

SPI Write Format

| SPIDI | Addr(High) | Addr(Middle) | Addr(Low) | CMD(Write) 0x00 | Data0 | Data1 | Data.... |

SPIDO ‒

SPI Read Format

| SPIDI | Addr(High) | Addr(Middle) | Addr(Low) | CMD(Read) 0x80 | ‒ ‒ ‒ ‒ ‒ ‒ ‒ |

| SPIDO ‒ ‒ ‒ ‒ ‒ ‒ ‒ ‒ | Data0 | Data1 | Data.... |
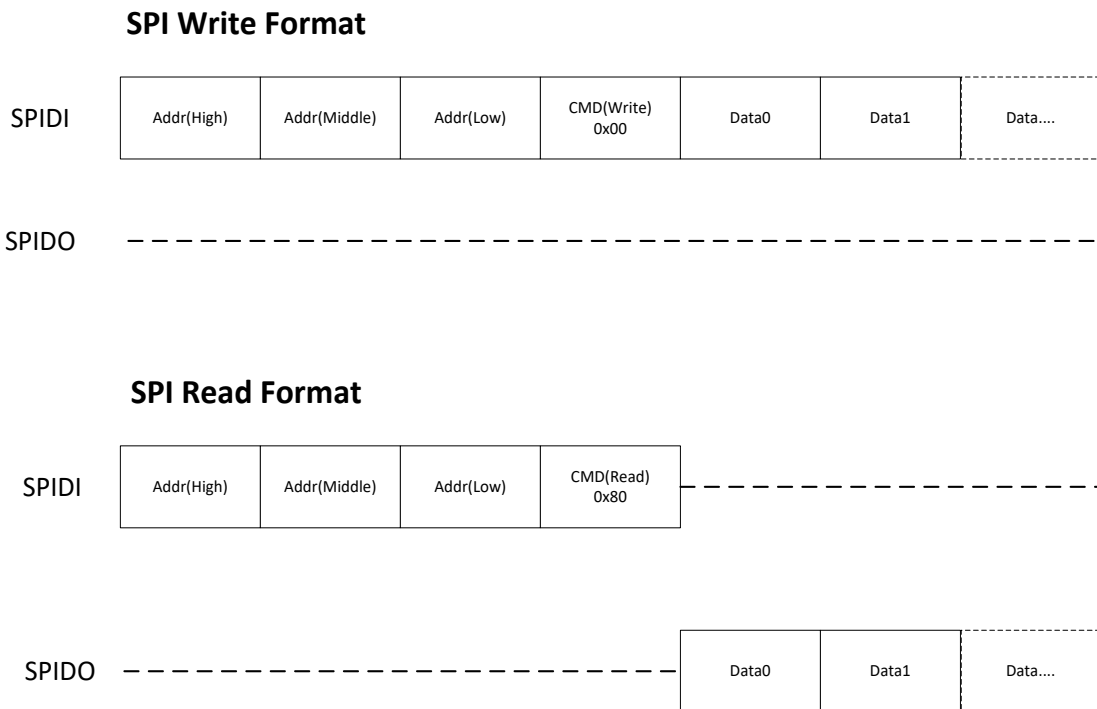
Figure 7- 9 SPI write/read command format

7.4.4   I2C and SPI Usage

I2C hardware and SPI hardware modules in the chip share part of the hardware, as a result, when both hardware interfaces are used, certain restrictions apply.

See **Section 7.3.5   I2C and SPI Usage** for detailed instructions.

7.5 UART

The TLSR8258F1KET48A 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.

Hardware flow control is supported via RTS and CTS.

The UART module also supports ISO7816 protocol to enable communication with ISO/IEC 7816 integrated circuit card, especially smart card. In this mode, half-duplex communication (transmission or reception) is supported via the shared 7816_TRX interface.
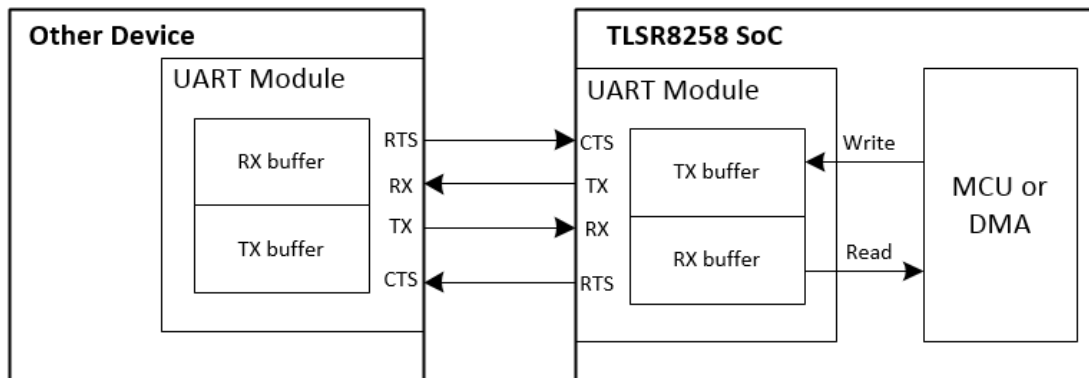
Figure 7- 10   UART communication

As shown in Figure 7-9, 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 TLSR8258F1KET48A UART is close to full, the TLSR8258F1KET48A 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 TLSR8258F1KET48A receives a signal from pin CTS, it indicates that RX buffer of other device is close to full and the TLSR8258F1KET48A should stop sending data.

Table 7- 9   Register configuration for UART

Address	Name	R/W	Description	Reset Value
0x90	UART_DATA_BUF0	RW	write/read buffer[7:0]	0x00
0x91	UART_DATA_BUF1	RW	Write/read buffer[15:8]	0x00
0x92	UART_DATA_BUF2	RW	Write/read buffer[23:16]	0x00
0x93	UART_DATA_BUF3	RW	Write/read buffer[31:24]	0x00
0x94	UART_CLK_DIV_L	RW	[7:0]: uart clk div register uart_clk_div[7:0]	0xff
0x95	UART_CLK_DIV_H	RW	[6:0]: uart clk div register uart_clk_div[14:8] uart_sclk = sclk/(uart_clk_div[14:0]+1) [7]: uart_clk_div[15] 1: enable clock divider, 0: disable.	0x0f

Address	Name	R/W	Description	Reset Value
0x96	UART_CTRL0	RW	[3:0] bwpc, bit width, should be larger than 2 Baudrate = uart_sclk/(bwpc+1) [4] rx dma enable [5] tx dma enable [6] rx interrupt enable [7]tx interrupt enable	0x0f
0x97	UART_CTRL1	RW	[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 [7]: uart tx, rx loopback	0x0e
0x98	UART_CTRL2	RW	[3:0] rts trig level [4] rts Parity [5] rts manual value [6] rts manual enable [7] rts enable	0xa5
0x99	UART_CTRL3	RW	[3:0]: rx_irq_trig level [7:4] tx_irq_trig level	0x44
0x9a	UART_RXTIMEOUT_O_L	RW	[7:0]: R_rxtimeout_o[7:0] The setting is transfer one bytes need cycles base on uart_clk. For example, if transfer one bytes (1 start bit+8bits data+1 priority bit+2 stop bits) total 12 bits, this register setting should be (bwpc+1)*12.	0xc0
0x9b	UART_RXTIMEOUT_O_H	RW	[1:0]: R_rxtimeout_o[9:8] 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. [5]: p7816_en_o [6]: mask_txdone [7]: mask_err	0x01
0x9c	UART_BUFCNT	R	[3:0]: rx_buf_cnt [7:4]: tx_buf_cnt	0x00
0x9d	UART_STATUS	R	[2:0] rbcnt	0x00

Address	Name	R/W	Description	Reset Value
			[3] irq_o [6:4]wbcnt [6] write 1 to clear rx [7] rx_err, write 1 to clear tx	
0x9e	UART_TXRX_STATUS	R	[0] txdone [1] tx_buf_irq [2] rxdone [3] rx_buf_irq	0x00
0x9f	UART_STATE	R	[2:0] tstate_i [7:4] rstate_i	0x00

Addresses 0x90~0x93 serve to write data into TX buffer or read data from RX buffer.

Addresses 0x94~0x95 serve to configure UART clock.

Address 0x96 serves to set baud rate (bit[3:0]), enable RX/TX DMA mode (bit[4:5]), and enable RX/TX interrupt (bit[6:7]).

Address 0x97 mainly serves to configure CTS. Bit[1] should be set to 1b'1 to enable CTS. Bit[0] serves to configure CTS signal level. Bit[2:3] serve to enable parity bit and select even/odd parity. Bit[5:4] serve to select 1/1.5/2 bits for stop bit. Bit[6] serves to configure whether RX/TX level should be inverted.

Address 0x98 serves to configure RTS. Bit[7] and Bit[3:0] serve to enable RTS and configure RTS signal level.

Address 0x99 serves to configure the number of bytes in RX/TX buffer to trigger interrupt.

The number of bytes in RX/TX buffer can be read from address 0x9c.

7.6   USB

The TLSR8258F1KET48A 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, which can be enabled via analog register afe_0x0b<7>.

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.

8 PWM

The TLSR8258F1KET48A supports 6-channel PWM (Pulse-Width-Modulation) output. Each PWM#n (n=0~5) has its corresponding inverted output at PWM#n_N pin.

8.1 Register table

Table 8- 1 Register table for PWM

Address	Mnemonic	Type	Description	Reset Value
0x780	PWM_EN	R/W	[1]: 0--disable PWM1, 1--enable PWM1 [2]: 0--disable PWM2, 1--enable PWM2 [3]: 0--disable PWM3, 1--enable PWM3 [4]: 0--disable PWM4, 1--enable PWM4 [5]: 0--disable PWM5, 1--enable PWM5	0x00
0x781	PWM_EN0	R/W	[0]: 0--disable PWM0, 1--enable PWM0	0x00
0x782	PWM_CLKDIV	R/W	Set PWM_clk: (PWM_CLKDIV+1)*sys_clk	0x00
0x783	PWM_MODE	R/W	[3:0]: PWM0 mode select 0000-pwm0 normal mode 0001-pwm0 count mode 0011-pwm0 IR mode 0111-pwm0 IR FIFO mode 1111-pwm0 IR DMA FIFO mode	0x00
0x784	PWM_CC0	R/W	[5:0]:1'b1 invert PWM output	0x00
0x785	PWM_CC1	R/W	[5:0]:1'b1 invert PWM_INV output	0x00
0x786	PWM_CC2	R/W	[5:0]: Signal frame polarity of PWM5~PWM0 1b'0-high level first 1b'1-low level first	0x00
0x788~ 0x793	reserved			
0x794	PWM_TCMP0	R/W	[7:0] bits 7-0 of PWM0's high time or low time(if pola[0]=1)	0x00
0x795	PWM_TCMP0	R/W	[15:8] bits 15-8 of PWM0's high time or low time	0x00
0x796	PWM_TMAX0	R/W	[7:0] bits 7-0 of PWM0's cycle time	0x00
0x797	PWM_TMAX0	R/W	[15:8] bits 15-8 of PWM0's cycle time	0x00
0x798	PWM_TCMP1	R/W	[7:0] bits 7-0 of PWM1's high time	0x00

Address	Mnemonic	Type	Description	Reset Value
			or low time(if pola[1]=1)	
0x799	PWM_TCMP1	R/W	[15:8] bits 15-8 of PWM1's high time or low time	0x00
0x79a	PWM_TMAX1	R/W	[7:0] bits 7-0 of PWM1's cycle time	0x00
0x79b	PWM_TMAX1	R/W	[15:8] bits 15-8 of PWM1's cycle time	0x00
0x79c	PWM_TCMP2	R/W	[7:0] bits 7-0 of PWM2's high time or low time(if pola[2]=1)	0x00
0x79d	PWM_TCMP2	R/W	[15:8] bits 15-8 of PWM2's high time or low time	0x00
0x79e	PWM_TMAX2	R/W	[7:0] bits 7-0 of PWM2's cycle time	0x00
0x79f	PWM_TMAX2	R/W	[15:8] bits 15-8 of PWM2's cycle time	0x00
0x7a0	PWM_TCMP3	R/W	[7:0] bits 7-0 of PWM3's high time or low time(if pola[3]=1)	0x00
0x7a1	PWM_TCMP3	R/W	[15:8] bits 15-8 of PWM3's high time or low time	0x00
0x7a2	PWM_TMAX3	R/W	[7:0] bits 7-0 of PWM3's cycle time	0x00
0x7a3	PWM_TMAX3	R/W	[15:8] bits 15-8 of PWM3's cycle time	0x00
0x7a4	PWM_TCMP4	R/W	[7:0] bits 7-0 of PWM4's high time or low time(if pola[4]=1)	0x00
0x7a5	PWM_TCMP4	R/W	[15:8] bits 15-8 of PWM4's high time or low time	0x00
0x7a6	PWM_TMAX4	R/W	[7:0] bits 7-0 of PWM4's cycle time	0x00
0x7a7	PWM_TMAX4	R/W	[15:8] bits 15-8 of PWM4's cycle time	0x00
0x7a8	PWM_TCMP5	R/W	[7:0] bits 7-0 of PWM5's high time or low time(if pola[5]=1)	0x00
0x7a9	PWM_TCMP5	R/W	[15:8] bits 15-8 of PWM5's high time or low time	0x00
0x7aa	PWM_TMAX5	R/W	[7:0] bits 7-0 of PWM5's cycle time	0x00
0x7ab	PWM_TMAX5	R/W	[15:8] bits 15-8 of PWM5's cycle time	0x00
0x7ac	PWM_PNUM0	R/W	[7:0] bits 7-0 of PWM0 Pulse number in count mode and IR mode	0x00
0x7ad	PWM_PNUM0	R/W	[13:8] bits 13-8 of PWM0 Pulse number in count mode and IR mode	0x00
0x7ae~ 0x7af	reserved			
0x7b0	PWM_MASK0	R/W	INT mask [0] PWM0 Pnum int 0: disable 1: Enable [1] PWM0 ir dma fifo mode int	0x00

Address	Mnemonic	Type	Description	Reset Value
			0: disable 1: Enable [2] PWM0 frame int 0: disable 1: Enable [3] PWM1 frame int 0: disable 1: Enable [4] PWM2 frame int 0: disable 1: Enable [5] PWM3 frame int 0: disable 1: Enable [6] PWM4 frame int 0: disable 1: Enable [7] PWM5 frame int 0: disable 1: Enable	
0x7b1	PWM_INT0	R/W	INT status, write 1 to clear [0]: PWM0 pnum int (have sent PNUM pulses, PWM_NCNT==PWM_PNUM) [1]:PWM0 ir dma fifo mode int(pnum int &fifo empty in ir dma fifo mode) [2]: PWM0 cycle done int (PWM_CNT==PWM_TMAX) [3]: PWM1 cycle done int (PWM_CNT==PWM_TMAX) [4]: PWM2 cycle done int (PWM_CNT==PWM_TMAX) [5]: PWM3 cycle done int (PWM_CNT==PWM_TMAX) [6]: PWM4 cycle done int (PWM_CNT==PWM_TMAX) [7]: PWM5 cycle done int (PWM_CNT==PWM_TMAX)	0x00
0x7b2	PWM_MASK1	R/W	[0]: PWM0 fifo mode fifo cnt int mask 0: disable, 1: Enable	0x00
0x7b3	PWM_INT1	R/W	INT status, write 1 to clear [0]: fifo mode cnt int, when FIFO_NUM (0x7cd[3:0]) is less than FIFO_NUM_LVL (0x7cc[3:0])	0x00
0x7b4	PWM_CNT0	R	[7:0]PWM0 cnt value	0x00
0x7b5	PWM_CNT0		[15:8]PWM0 cnt value	0x00
0x7b6	PWM_CNT1	R	[7:0]PWM1 cnt value	0x00
0x7b7	PWM_CNT1		[15:8]PWM1 cnt value	0x00
0x7b8	PWM_CNT2	R	[7:0]PWM2 cnt value	0x00
0x7b9	PWM_CNT2		[15:8]PWM2 cnt value	0x00
0x7ba	PWM_CNT3	R	[7:0]PWM3 cnt value	0x00

Address	Mnemonic	Type	Description	Reset Value
0x7bb	PWM_CNT3		[15:8]PWM3 cnt value	0x00
0x7bc	PWM_CNT4	R	[7:0]PWM4 cnt value	0x00
0x7bd	PWM_CNT4		[15:8]PWM4 cnt value	0x00
0x7be	PWM_CNT5	R	[7:0]PWM5 cnt value	0x00
0x7bf	PWM_CNT5		[15:8]PWM5 cnt value	0x00
0x7c0	PWM_NCNT0	R	[7:0]PWM0 pluse_cnt value	0x00
0x7c1	PWM_NCNT0		[15:8]PWM0 pluse_cnt value	0x00
0x7c2 ~ 0x7c3	reserved			
0x7c4	PWM_TCMP0_SHADOW	R/W	[7:0] bits 7-0 of PWM0's high time or low time(if pola[0]=1),if shadow bit(fifo data[14]) is 1'b1 in ir fifo mode or dma fifo mode	0x55
0x7c5	PWM_TCMP0_SHADOW	R/W	[15:8] bits 15-8 of PWM0's high time or low time ,if shadow bit(fifo data[14]) is 1'b1 in ir fifo mode or dma fifo mode	0x55
0x7c6	PWM_TMAX0_SHADOW	R/W	[7:0] bits 7-0 of PWM0's cycle time, if shadow bit(fifo data[14]) is 1'b1 in ir fifo mode or dma fifo mode	0x00
0x7c7	PWM_TMAX0_SHADOW	R/W	[15:8] bits 15-8 of PWM0's cycle time, if shadow bit(fifo frame[14]) is 1'b1 in ir fifo mode or dma fifo mode	0x00
0x7c8	FIFO_DAT0_ENTRY	R/W	Use in ir fifo mode	0x00
0x7c9	FIFO_DAT1_ENTRY	R/W	Use in ir fifo mode	0x00
0x7ca	FIFO_DAT2_ENTRY	R/W	Use in ir fifo mode	0x00
0x7cb	FIFO_DAT3_ENTRY	R/W	Use in ir fifo mode	0x00
0x7cc	FIFO_NUM_LVL	R/W	FIFO num int trigger level	0x00
0x7cd	FIFO_SR	R	[3:0]:FIFO DATA NUM(byte) [4]:FIFO EMPTY [5]:FIFO FULL	0x10
0x7ce	FIFO_CLR	W1	[0]: write 1 to clear data in FIFO	0x00

8.2 Enable PWM

Register PWM_EN (address 0x780)[5:1] and PWM_EN0 (address 0x781)[0] serves to enable PWM5~PWM0 respectively via writing "1" for the corresponding bits.

8.3 Set PWM clock

PWM clock derives from system clock. Register PWM_CLKDIV (address 0x782) serves to set the frequency dividing factor for PWM clock. Formula below applies:

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

8.4 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.

8.4.1 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 (address 0x794~0x795, 0x798~0x799, 0x79c~0x79d, 0x7a0~0x7a1, 0x7a4~0x7a5, 0x7a8~0x7a9) / PWM_TCMP0_SHADOW (0x7c4~0x7c5), PWM#n enters Remaining status and outputs Low level till PWM#n cycle time configured in register PWM_TMAX#n (address 0x796~0x797, 0x79a~0x79b, 0x79e~0x79f, 0x7a2~0x7a3, 0x7a6~0x7a7, 0x7aa~0x7ab) / PWM_TMAX0_SHADOW (0x7c6~0x7c7) expires.
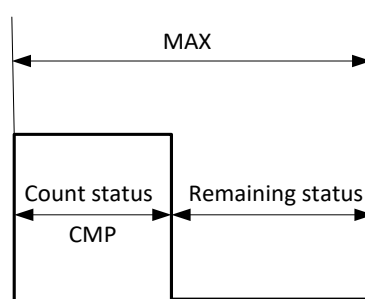
Figure 8- 1   A signal frame

An interruption will be generated at the end of each signal frame if enabled via register PWM_MASK (address 0x7b0[2:7]).

8.4.2 Invert PWM output

PWM#n and PWM#n_N output could be inverted independently via register PWM_CC0 (address 0x784) and PWM_CC1 (address 0x785). When the inversion bit is enabled, waveform of the corresponding PWM channel will be inverted completely.

8.4.3 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 (address 0x786[5:0]), PWM#n will output Low level at Count status and High level at Remaining status.
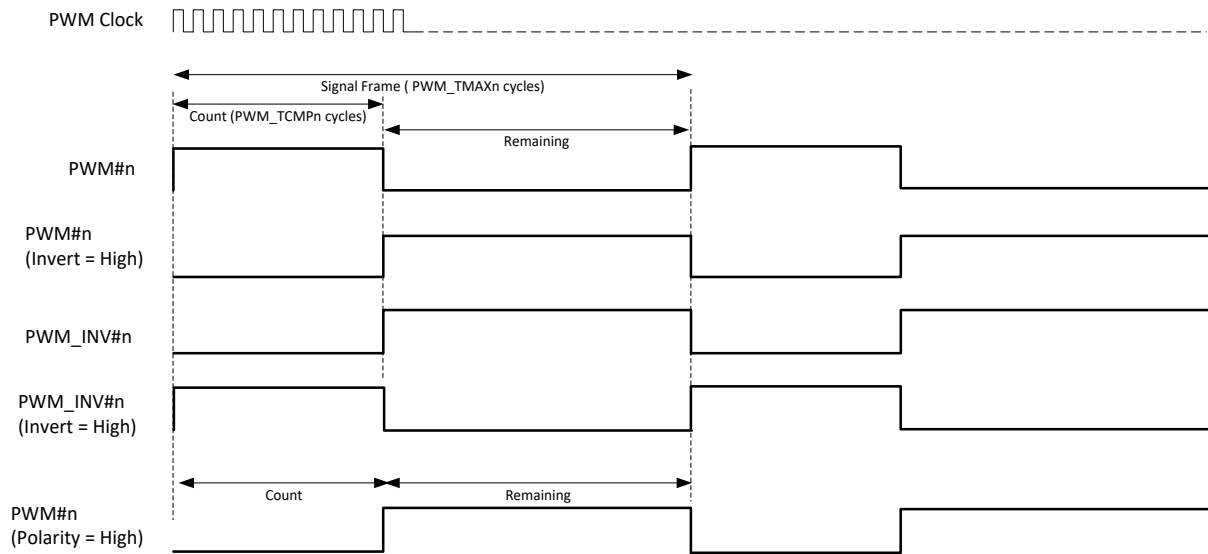
Figure 8- 2   PWM output waveform chart

8.5 PWM mode

8.5.1   Select PWM mode

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 (address 0x783) serves to select PWM0 mode.

8.5.2   Continuous mode

PWM0~PWM5 all support Continuous mode. In this mode, PWM#n continuously sends out signal frames. PWM#n should be disabled via address 0x780/0x781 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 (0x7b1[2:7]) will be automatically set to 1b'1. If the interrupt is enabled by setting PWM_MASK0 (address 0x7b0[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.
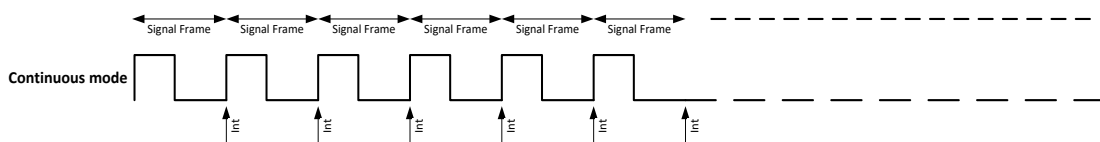
Figure 8-3   Continuous mode

8.5.3 Counting mode

Only PWM0 supports Counting mode. Address 0x783[3:0] 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_PNUM0 (address 0x7ac~0x7ad).

After each signal frame is finished, PWM0 cycle done interrupt flag bit (0x7b1[2]) will be automatically set to 1b'1. If the interrupt is enabled by setting PWM_MASK0 (address 0x7b0[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 (0x7b1[0]) will be automatically set to 1b'1. If the interrupt is enabled by setting PWM_MASK0 (address 0x7b0[0]) as 1b'1, a Pnum interruption will be generated. User needs to write 1b'1 to the flag bit to manually clear it.
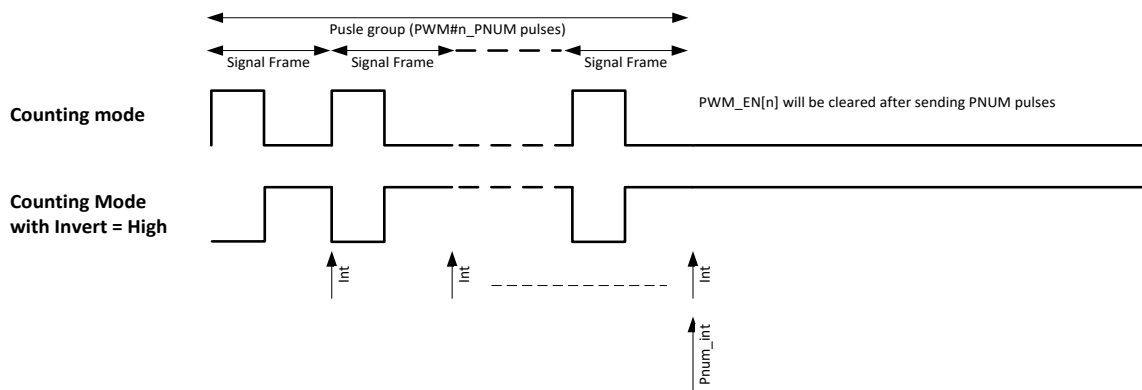
Figure 8-4   Counting mode (n=0)

Counting mode also serves to stop IR mode gracefully. Refer to **section 8.5.4** for details.

8.5.4 IR mode

Only PWM0 supports IR mode. Address 0x783[3:0] 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 WM_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 (0x781[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

(0x7b1[2])/PWM0 pnum interrupt flag bit (0x7b1[0]) will be automatically set to 1b'1. A frame interruption/Pnum interruption will be generated (if enabled by setting address 0x7b0[2]/0x7b0[0] as 1b'1).
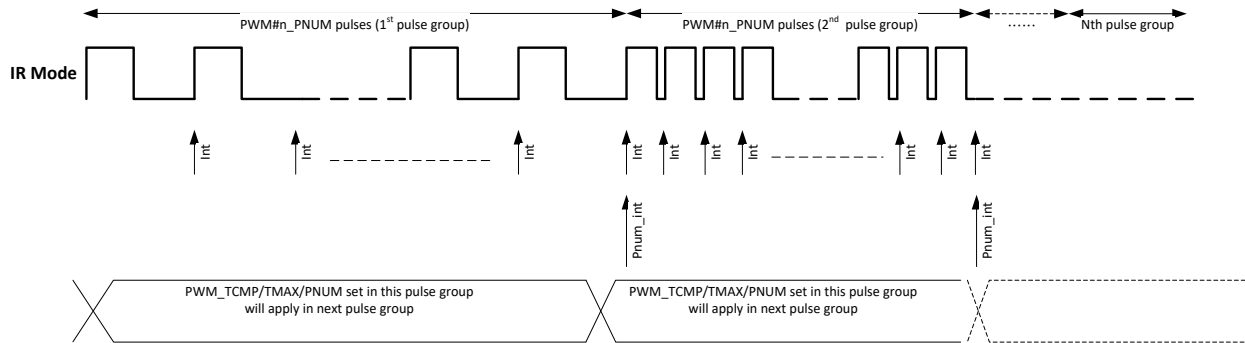
Figure 8-5   IR mode (n=0)

8.5.5 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. Address 0x783[3:0] 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 in 0x794~0x797;

    1: use configuration of TCMP0_SHADOW and TMAX0_SHADOW in 0x7c4~0x7c7.

  ✧ 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 FIFO_DATA_ENTRY 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.

Current FIFO CFG Data bit[15]=1

Current FIFO CFG Data bit[14]=0

Next FIFO CFG Data bit[15] = 0

Next FIFO CFG Data bit[14]=0

Current FIFO CFG Data bit[15]=1

Current FIFO CFG Data bit[14]=0

Next FIFO CFG Data bit[15] = 1

Next FIFO CFG Data bit[14]=1
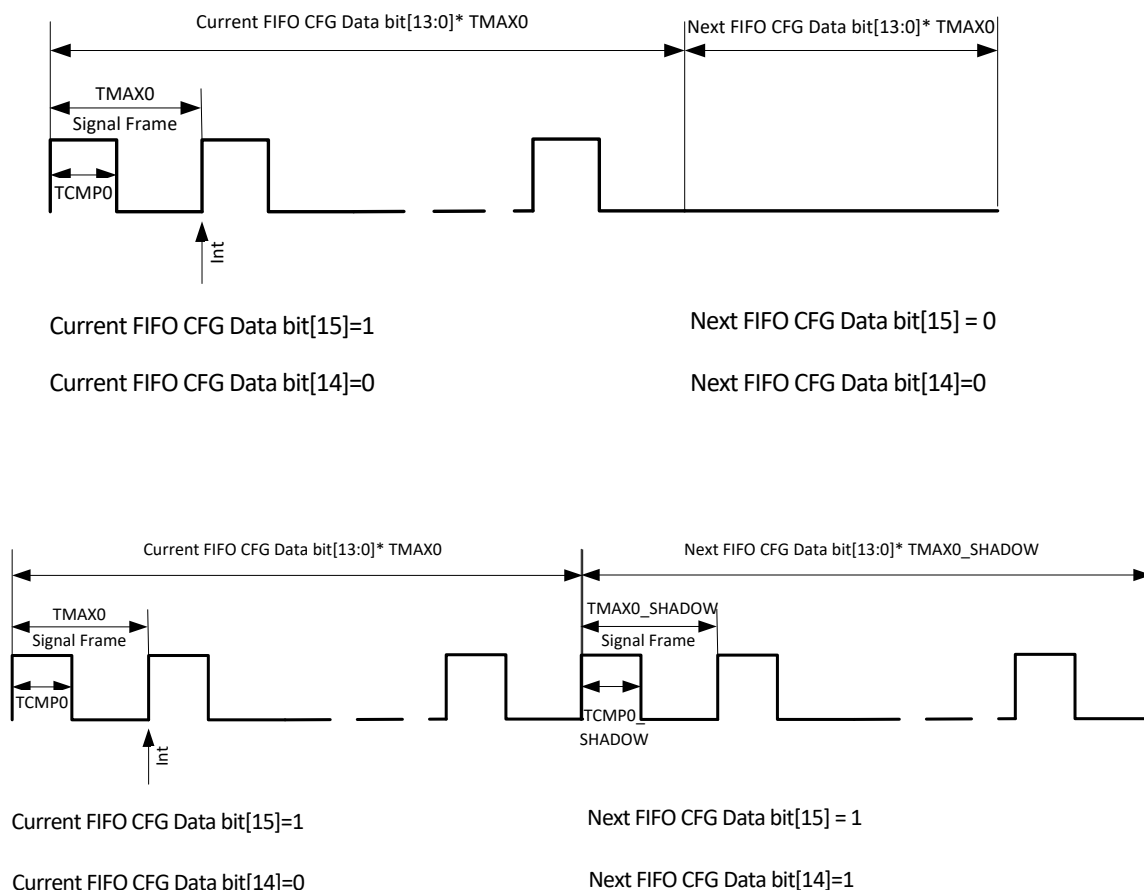
Figure 8- 6   IR format examples

When "FIFO CFG Data" is configured in FIFO and PWM0 is enabled via PWM_EN0 (address 0x781[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 (address 0x7ce[0]) 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.

8.5.6   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. 36kHz, 38kHz, 40kHz, or 56kHz.

Only PWM0 supports IR DMA FIFO mode. Address 0x783[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 0x781[0] (i.e. 0x781[0] will be set as 1b'1 automatically).

Example 1:

Suppose Mark carrier (pulse) frequency1(F1) = 40kHz, duty cycle 1/3

Mark carrier (pulse) frequency2(F2) = 50kHz, duty cycle 1/2

Space carrier (low level) frequency(F3) = 40kHz

If user wants to make PWM send waveforms in following format (PWM CLK =24MHz):

Burst(20[F1]), i.e. 20 F1 pulses

Burst(30[F2]),

Burst(50[F1]) ,

Burst(50[F2]),

Burst(20[F1],10[F3]),

Burst(30[F2],10[F3])

Step1: Set carrier F1 frequency as 40kHz, set duty cycle as 1/3.

Set **PWM_TMAX0** as 0x258 (i.e. 24MHz/40kHz=600=0x258).

Since duty cycle is 1/3, set **PWM_TCMP0** as 0xc8 (i.e. 600/3=200=0xc8).

Set carrier F2 frequency as 50kHz, set duty cycle as 1/2.

Set **PWM_TMAX0_SHADOW** as 0x1e0 (i.e. 24MHz/50kHz=480=0x1e0).

Since duty cycle is 1/2, set **PWM_TCMP0_SHADOW** as 0xf0 (i.e. 480/2=240=0xf0).

Step2: Generate "FIFO CFG Data" sequence.

Burst(20[F1]): {[15]: 1'b1, [14]: 1'b0, [13:0]: 'd20}=0x8014.

Burst(30[F2]): {[15]: 1'b1, [14]: 1'b1, [13:0]: 'd30}=0xc01e.

Burst(50[F1]) : {[15]: 1'b1, [14]: 1'b0, [13:0]: 'd50}=0x8032.

Burst(50[F2]): {[15]: 1'b1, [14]: 1'b1, [13:0]:'d50}=0xc032.

Burst(20[F1],10[F3]): {[15]: 1'b1, [14]: 1'b0, [13:0]: 'd20}=0x8014,

{[15]: 1'b0, [14]: 1'b0, [13:0]: 'd10}=0x000a.

Burst(30[F2],10[F3]): {[15]: 1'b1, [14]: 1'b1, [13:0]: 'd30}=0xc01e,

{[15]:1'b0, [14]: 1'b0, [13:0]: 'd10}=0x000a.

Step3: Write "FIFO CFG Data" into SRAM in DMA format.

DMA SOURCE ADDRESS+0x00: 0x0000_0010 (dma transfer-length: 16byte)

DMA SOURCE ADDRESS+0x04: 0xc01e_8014 (LITTLE ENDIAN)

DMA SOURCE ADDRESS+0x08: 0xc032_8032

DMA SOURCE ADDRESS+0x0c: 0x000a_8014

DMA SOURCE ADDRESS+0x10: 0x000a_c01e

Step4: Enable DMA channel 5 to send PWM waveforms.

Write 1'b1 to address 0x524[5] to enable DMA channel 5.

After all waveforms are sent, FIFO becomes empty, PWM0 will be disabled automatically (address 0x781[0] is automatically cleared). The FIFO mode stop interrupt flag bit (address 0x7b3[0]) will be automatically set as 1b'1. If the interrupt is enabled by setting PWM_MASK1 (address 0x7b2[0]) as 1b'1, a FIFO mode stop interrupt will be generated. User needs to write 1b'1 to the flag bit to manually clear it.

Example 2:

Suppose carrier frequency is 38kHz, system clock frequency is 24MHz, duty cycle is 1/3, and the format of IR code to be sent is shown as below:

1) Preamble waveform: 9ms carrier + 4.5ms low level.

2) Data 1 waveform: 0.56ms carrier + 0.56ms low level.

3) Data 0 waveform: 0.56ms carrier + 1.69ms low level.

4) Repeat waveform: 9ms carrier + 2.25ms low level + 0.56ms carrier. Repeat waveform duration is 11.81ms, interval between two adjacent repeat waveforms is 108ms.

5) End waveform: 0.56ms carrier.

User can follow the steps below to configure related registers:

Step1: Set carrier frequency as 38kHz, set duty cycle as 1/3.

Set **PWM_TMAX0** as 0x277 (i.e. 24MHz/38kHz=631=0x277).

Since duty cycle is 1/3, set **PWM_TCMP0** as 0xd2 (i.e. 631/3=210=0xd2).

Step2: Generate "FIFO CFG Data" sequence.

Preamble waveform:

9ms carrier: {[15]:1'b1, [14]:1'b0, [13:0]: 9*38='d 342=14'h 156}=0x8156

4.5ms low level: {[15]:1'b0, [14]:1'b0, [13:0]: 4.5*38='d 171=14'h ab}=0x00ab

Data 1 waveform:

0.56ms carrier: {[15]:1'b1, [14]:1'b0, [13:0]: 0.56*38='d 21=14'h 15}=0x8015

0.56ms low level: {[15]:1'b0, [14]:1'b0, [13:0]: 0.56*38='d 21=14'h 15}=0x0015

Data 0 waveform:

0.56ms carrier: {[15]:1'b1, [14]:1'b0, [13:0]: 0.56*38='d 21=14'h 15}=0x8015

1.69ms low level: {[15]:1'b0, [14]:1'b0, [13:0]: 1.69*38='d 64=14'h 40}=0x0040

Repeat waveform:

9ms carrier: {[15]:1'b1, [14]:1'b0, [13:0]: 9*38='d 342=14'h 156}=0x8156

2.25ms low level: {[15]:1'b0, [14]:1'b0, [13:0]: 2.25*38='d 86=14'h 56}=0x0056

0.56ms carrier: {[15]:1'b1, [14]:1'b0, [13:0]: 0.56*38='d 21=14'h 15}=0x8015

108ms -11.81ms =96.19ms low level:

{[15]:1'b0, [14]:1'b0, [13:0]: 96.19*38='d 3655=14'h e47}=0x0e47

End waveform:

0.56ms carrier: {[15]:1'b1, [14]:1'b0, [13:0]: 0.56*38='d 21=14'h 15}=0x8015

Step3: Write "IR CFG Data" into SRAM in DMA format.

If user want PWM0 to send IR waveform in following format:

Preamble+0x5a+Repeat+End

Preamble: 0x8156, 0x00ab

0x5a=8'b01011010

Data 0: 0x8015, 0x0040

Data 1: 0x8015, 0x0015

Data 0: 0x8015, 0x0040

Data 1: 0x8015, 0x0015

Data 1: 0x8015, 0x0015

Data 0: 0x8015, 0x0040

Data 1: 0x8015, 0x0015

Data 0: 0x8015, 0x0040

Repeat: 0x8156, 0x0056, 0x8015, 0x0e47

End: 0x8015.

User needs to write the configuration information above into source address of DMA channel 5, as shown below:

DMA SOURCE ADDRESS+0x00: 0x0000_002e (dma transfer-length: 46byte)

DMA SOURCE ADDRESS+0x04: 0x00ab_8156 (Preamble) (LITTLE ENDIAN)

DMA SOURCE ADDRESS+0x08: 0x0040_8015 (Data 0)

DMA SOURCE ADDRESS+0x0c: 0x0015_8015 (Data 1)

DMA SOURCE ADDRESS+0x10: 0x0040_8015 (Data 0)

DMA SOURCE ADDRESS+0x14: 0x0015_8015 (Data 1)

DMA SOURCE ADDRESS+0x18: 0x0015_8015 (Data 1)

DMA SOURCE ADDRESS+0x1c: 0x0040_8015 (Data 0)

DMA SOURCE ADDRESS+0x20: 0x0015_8015 (Data 1)

DMA SOURCE ADDRESS+0x24: 0x0040_8015 (Data 0)

DMA SOURCE ADDRESS+0x28: 0x0056_8156 (Repeat)

DMA SOURCE ADDRESS+0x2c: 0x0e47_8015 (Repeat)

DMA SOURCE ADDRESS+0x30: 0x8015 (End)

Step4: Enable DMA channel 5 to send PWM waveforms.

   Write 1'b1 to address 0x524[5] to enable DMA channel 5.

After all waveforms are sent, FIFO becomes empty, PWM0 will be disabled automatically (address 0x781[0] is automatically cleared). The FIFO mode stop interrupt flag bit (address 0x7b3[0]) will be automatically set as 1b'1. If the interrupt is enabled by setting PWM_MASK1 (address 0x7b2[0]) as 1b'1, a FIFO mode stop interrupt will be generated. User needs to write 1b'1 to the flag bit to manually clear it.

8.6 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" (address 0x641[6], see **section 6 Interrupt**) should be set as 1b'1. To enable various PWM interrupt sources, PWM_MASK0 (address 0x7b0[7:0]) and PWM_MASK1 (address 0x7b2[0]) should be set as 1b'1 correspondingly.

Interrupt status can be cleared via register PWM_INT0 (address 0x7b1[7:0]) and PWM_INT1 (address 0x7b3[0]).

9   Audio

9.1   Audio input path

There are four types of audio input path: digital microphone (DMIC), Codec (I2S), USB and analog input channel (AMIC), which is selectable by writing address 0xb11[3:2].

Address 0xb11[4] should be set as 1b'1/1b'0 to select mono/stereo input for audio input processing module.
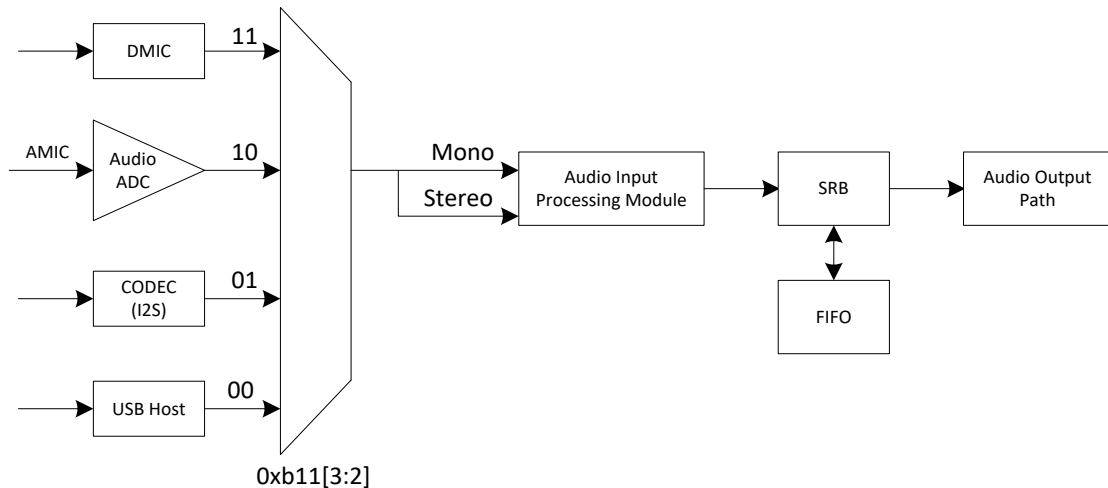
Figure 9- 1   Audio input path

Table 9- 1 Audio data flow direction

Data Path		Target SRAM		
		FIFO0	FIFO1	FIFO2
DMIC	Decimation filter /LPF/Down Sample/ HPF/ALC	√	√	×
I2S		√	√	×
USB		√	√	×
ANALOG CH0		√	√	×
ANALOG CH1		√	√	×
ANALOG CH2		×	×	√

9.1.1   AMIC input

Address 0xb11[3:2] should be set as 2b'10 to select AMIC as audio input.

A programmable stereo PGA (Programmable Gain Amplifier) with adjustable gain is built in for AMIC. AMIC input channel can carry out signal amplification via the PGA.

After implementing AD conversion for selected AMIC input signal, data of 3 analog channels (CH0~CH2) will be generated. Data of CH0/CH1 are sent to the audio input processing module, while data of CH2 are directly written into FIFO.

9.1.2   DMIC input

Stereo digital microphone (DMIC) interface is also supported in the TLSR8258F1KET48A. Address 0xb11[3:2] should be set as 2b'11 to select DMIC as audio input.

DMIC interface includes one configurable clock line and one data line. Address 0xb11[1:0] serves to set the rising/falling edge of clock signal at which to sample data of DMIC, and it should not be set as 2b'11. Address 0xb11[7:6] serves to enable/mask the DMIC channel which samples data at rising/falling edge. Generally, address 0xb11 bit[0] is set as 1b'0 and bit[7:6] is set as 2b'01 to enable DMIC0 sampling at rising edge of clock.

After data sampling of DMIC interface, sign extension and audio input processing, the signal can be written into FIFO.

9.1.3   I2S input

Address 0xb11[3:2] should be set as 2b'01 to select I2S as audio input.

Digital I2S audio interface supports Master mode only, 16-bit data width, and variable sampling rate: 8K/16K/22.05K/24K/32K/44.1K/48K. The sampling rate is determined by I2S clock.

For I2S clock configuration, please refer to section **4.3.3   I2S clock**.

Address 0x560[5]/[4]/[1] should be set to "1" to enable I2S interface, I2S Recorder and I2S Player, respectively. I2S interface includes one configurable clock line, one data line and one channel selection line. Data generated by the audio codec will be written into FIFO after implementing conversion via I2S Recorder and audio input processing.

9.1.4   USB Host input

Address 0xb11[3:2] should be set as 2b'00 to select USB as audio input.

Packet transmitted by USB Host will be written into FIFO after implementing conversion via USB Interface and audio input processing.

Address 0x560[3] should be set to 1b'1 to enable ISO player.

9.1.5   DFIFO

As shown in Table 9-1, for any type of audio input path, the data will be finally written into DFIFO (DMA FIFO) 0, 1 or 2.

Address 0xb10[0]/[1]/[2] should be set as 1b'1 to enable audio input of DFIFO 0~2.

DFIFO supports auto mode and manual mode. It's highly recommended to clear address 0xb2c[0] to select auto mode.

Take DFIFO0 as an example:

✧   Address 0xb00, 0xb01 and 0xb03 serve to set base address for DFIFO0, i.e. starting address to write/read data into/from DFIFO0.

✧   Address 0xb02 serves to set depth (i.e. the maximum data number) for DFIFO0. Suppose address 0xb02 is set as 0x01, then the DFIFO0 depth is 4 words, i.e. 16 bytes.

✧ Current data number (difference value of write-pointer and read-pointer) in DFIFO0 can be read from address 0xb20 and 0xb21.

✧ User can check current DFIFO0 read pointer/write pointer location by reading address 0xb14~0xb15/0xb16~0xb17.

✧ When current data number in DFIFO0 is less than the underflow threshold set in address 0xb0c, address 0xb13 bit[0] and bit[4] will be set as 1b'1 successively, and a FIFO0 low interrupt will be generated if enabled via 0xb10[4].

Address 0xb13[4] will be automatically cleared when the data number in DFIFO0 is no less than the threshold; address 0xb13[0] needs to be cleared manually.

✧ When current data number in DFIFO0 is more than the overflow threshold set in address 0xb0d, address 0xb13 bit[1] and bit[5] will be set as 1b'1 successively, and a FIFO0 high interrupt will be generated if enabled via 0xb10[5].

Address 0xb13[5] will be automatically cleared when the data number in DFIFO0 is no more than the threshold; address 0xb13[1] needs to be cleared manually.

✧ When current data number in DFIFO1 is more than the overflow threshold set in address 0xb0e, address 0xb13 bit[2] and bit[6] will be set as 1b'1 successively, and a FIFO0 high interrupt will be generated if enabled via 0xb10[6].

Address 0xb13[6] will be automatically cleared when the data number in DFIFO0 is no more than the threshold; address 0xb13[2] needs to be cleared manually.

✧ When current data number in DFIFO2 is more than the overflow threshold set in address 0xb0f, address 0xb13 bit[3] and bit[7] will be set as 1b'1 successively, and a FIFO0 high interrupt will be generated if enabled via 0xb10[7].

Address 0xb13[7] will be automatically cleared when the data number in DFIFO0 is no more than the threshold; address 0xb13[3] needs to be cleared manually.

9.2 Audio input processing

Audio input processing mainly includes configurable decimation filter, LPF (Low Pass Filter), Down-sample module, HPF (High Pass Filter) and ALC (Automatic Level Control). The decimation filter, LPF, Down-sample module, HPF and ALC can be enabled or bypassed via address 0xb11[5] and 0xb40 [6]/[7]/[4]/[5].

Decimation Filter → LPF → Down Sample → HPF → ALC
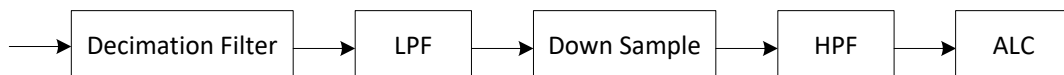
Figure 9- 2   Audio input processing

9.2.1 Decimation filter

Address 0xb11[5] should be cleared to enable decimation filter.

The decimation filter serves to down-sample the mono or stereo input (e.g. DMIC) data to required audio data playback rate (e.g. 48K or 32K).

Down-sampling rate is configurable as 1~8, 16, 32, 64, 128 or 256 by writing address 0xb12[3:0].

Address 0xb12[7:4] serves to adjust decimation filter output by right shift, so that the data after down-sampling won't exceed data bit width.

9.2.2 LPF

Address 0xb40[6] should be cleared to enable the LPF. The LPF serves to conduct frequency compensation.

9.2.3 Down-sample

If the Down-sample module is enabled by setting address 0xb40[7] to 1b'1, it will down-sample the data from LPF with fixed ratio of 2.

9.2.4 HPF

Address 0xb40[4] should be cleared to enable the HPF.

The HPF serves to eliminate internal DC offset to ensure audio amplification range.

The HPF output is adjustable via setting the parameter in address 0xb40[3:0].

9.2.5 ALC

The ALC supports analog mode and digital mode, and it mainly serves to regulate input volume level automatically or manually in each mode.

The analog mode is designed only for AMIC input, while the digital mode applies to all audio input types.

In analog mode, input volume level is regulated via PGA; while in digital mode, input volume level is regulated via Multiplier/Divider.

9.2.5.1 Auto regulate in analog mode

In this case, address 0xb54[0] should be set as 1b'1, and 0xb41[7]/0xb42[7] should be set as 1b'1 to enable auto regulation mode for left/right channel.

The result of (AMIC input * current PGA gain) is compared with high volume target (ALC_VOL_THH) and low volume target (ALC_VOL_THL); meanwhile, it will be compared with volume noise level (ALC_VOL_THN) to judge noise signal and help to regulate the PGA gain. The PGA gain will be automatically adjusted according to the comparison results, and it should be within the range from minimum PGA gain (ALC_VOL_L/ALC_VOL_R) to maximum PGA gain (ALC_VOL_H).

✧ Address 0xb44[6:1] serve to set integer part of high volume target in unit of dB; while address 0xb44[0] serve to set fractional part of high volume target in unit of dB.

✧ Address 0xb46[6:1] serve to set integer part of low volume target in unit of dB; while address 0xb46[0] serve to set fractional part of low volume target in unit of dB.

✧ Address 0xb48[6:1] serve to set integer part of volume noise level in unit of dB; while address 0xb48[0] serve to set fractional part of volume noise level in unit of dB.

✧ Address 0xb43[6:0] (ALC_VOL_H) serves to set the maximum PGA gain, while 0xb41[6:0]/0xb42[6:0] (ALC_VOL_L/ALC_VOL_R) serves to set the minimum PGA gain in left/right channel.

✧ User can check current PGA gain in left/right channel by reading address 0xb5e/0xb5f.

9.2.5.2   Manual regulate in analog mode

In this case, PGA gain can be adjusted by either of the following two manual modes.

✧ Manual mode 1: Address 0xb63[7] should be set as 1b'1 to select manual mode 1. In this mode, the PGA consists of two stages of amplifiers including pre-amplifier and post-amplifier, and each stage has configurable gain. Address 0xb63[6] serves to set gain for the pre-amplifier (Boost-stage) as 18dB (1b'0, default) or 38dB (1b'1); while address 0xb63[5:0] serves to set gain for the post-amplifier (Gain-stage) as -10dB (0x0, default) ~ 14dB (0x30) with step of 0.5dB.

✧ Manual mode 2: Address 0xb63[7] and 0xb54[0] should be cleared to select manual mode 2. In this mode, address 0xb61[6:0]/0xb65[6:0] serves to set target gain value for left/right channel, while 0xb60 serves to set the speed for PGA gain to reach the target gain value. Current PGA gain can be read from address 0xb62[6:0]/0xb66[6:0] which changes until the target gain value is reached. When PGA gain reaches the target value, address 0xb61[7]/0xb65[7] will be set as 1b'1.

9.2.5.3   Auto regulate in digital mode

In this case, address 0xb54[0] should be set as 1b'0, and 0xb41[7]/0xb42[7] should be set as 1b'1 to enable auto regulation mode for left/right channel.

The result of (Audio input * current digital gain) is compared with high volume target (ALC_VOL_THH), low volume target (ALC_VOL_THL); meanwhile, it will be compared with volume noise level (ALC_VOL_THN) to judge noise signal and help to regulate the digital gain. The digital gain will be automatically adjusted according to the comparison results, and it should be within the range from minimum digital gain (ALC_VOL_L/ALC_VOL_R) to maximum digital gain (ALC_VOL_H).

✧ Address 0xb44[6:1] serve to set integer part of high volume target in unit of dB; while address 0xb44[0] serve to set fractional part of high volume target in unit of dB.

✧ Address 0xb46[6:1] serve to set integer part of low volume target in unit of dB; while address 0xb46[0] serve to set fractional part of low volume target in unit of dB.

✧ Address 0xb48[6:1] serve to set integer part of volume noise level in unit of dB; while address 0xb48[0] serve to set fractional part of volume noise level in unit of dB.

✧ Address 0xb43[6:0] (ALC_VOL_H) serves to set the maximum digital gain, while 0xb41[6:0]/0xb42[6:0] (ALC_VOL_L/ALC_VOL_R) serves to set the minimum digital gain in left/right channel.

User can check current digital gain in left/right channel by reading address 0xb4d/0xb4e.

9.2.5.4　Manual regulate in digital mode

In this case, address 0xb54[0] should be set as 1b'0, and 0xb41[7]/0xb42[7] should be set as 1b'0 to enable manual regulation mode for left/right channel.

The coefficient of multiplier/divider is configurable via the register ALC_VOL_L (address 0xb41[5:0] / ALC_VOL_R (address 0xb42[5:0]) for left/right channel.

In manual mode (either analog mode or digital mode), the volume of the signal sent to the ALC module can be read via the address 0xb4d/0xb4e for left/right channel.

9.2.6　Register configuration

Table 9- 2　　Register configuration related to audio input processing

Address	Mnemonic	Type	Description	Reset value
0x67	I2S step	R/W	[7] I2S clock enable [6:0] i2s step	0x00
0x68	I2S Mod	R/W	I2s mod I2S clock = 48M*I2S_step[6:0]/I2S_mod[7:0], Mod should be larger than or equal to 2*step.	0x02
0xb00	DFIFO0_BADR0	RW	FIFO0 base address [7:0]	0x00
0xb01	DFIFO0 BADR1	RW	FIFO0 base address [15:8]	0x40
0xb02	DFIFO0_DEPTH	RW	FIFO depth=FIFO0_DEPTH*4words	0x7f
0xb03	DFIFO0_BADR2	RW	FIFO0 base address [18:16]	0x04
0xb04	DFIFO1_BADR0	RW	FIFO1 base address [7:0]	0x00
0xb05	DFIFO1 BADR1	RW	FIFO1 base address [15:8]	0x48
0xb06	DFIFO1_DEPTH	RW	FIFO depth=FIFO1_DEPTH*4words	0x7f
0xb07	DFIFO1_BADR2	RW	FIFO1 base address [18:16]	0x04
0xb08	DFIFO2_BADR0	RW	FIFO2 base address [7:0]	0x00
0xb09	DFIFO2 BADR1	RW	FIFO2 base address [15:8]	0x3c
0xb0a	DFIFO2_DEPTH	RW	FIFO depth=FIFO2_DEPTH*4words	0x3f
0xb0b	DFIFO2_BADR2	RW	FIFO2 base address [18:16]	0x04
0xb0c	DFIFO0_LLEV	RW	FIFO0 low level	0x20
0xb0d	DFIFO0_HLEV	RW	FIFO0 high level	0x60
0xb0e	DFIFO1_HLEV	RW	FIFO1 high level	0x20
0xb0f	DFIFO2_HLEV	RW	FIFO2 high level	0x20
0xb10	DFIFOMODE	RW	DFIFO enable [0]: enable audio input of FIFO0	F9

Address	Mnemonic	Type	Description	Reset value
			[1]: enable audio input of FIFO1 [2]: enable audio input of FIFO2 [3]: enable audio output of FIFO0 [4]: FIFO0 low interrupt enable [5]: FIFO0 high interrupt enable [6]: FIFO1 high interrupt enable [7]: FIFO2 high interrupt enable	
0xb11	DFIFOAIN	RW	[0]: 0: D-MIC0 uses data at rising edge of clock 1: at falling edge [1]: 0: D-MIC1 uses data at rising edge of clock 1: at falling edge [3:2]: audio input select 0: USB; 1: I2S; 2: ADC; 3: D-MIC [4]: mono (1)/stereo (0) input [5]: enable (0) or bypass (1) decimation filter input [6]:dmic_raise_chn_not_en [7]:dmic_fall_chn_not_en (generally disabled)	21
0xb12	DFIFODEC	RW	[3:0]: CIC down convert ratio 0~7: [3:0] + 1 8: 16; 9: 32; 10: 64; 11: 128: else: 256 [7:4]: CIC shift select (0 ~ 7)	5b
0xb13	DFIFOIRQST	R	[0]: FIFO0 low interrupt flag. Write 1 to clear. [1]: FIFO0 high interrupt flag. Write 1 to clear. [2]: FIFO1 high interrupt flag. Write 1 to clear. [3]: FIFO2 high interrupt flag. Write 1 to clear. [4]: FIFO0 low. [5]: FIFO0 high. [6]: FIFO1 high. [7]: FIFO2 high.	00
0xb14	DFIFO0_RPTRL	RO	FIFO READ PTR low byte	00
0xb15	DFIFO0_RPTRH	RO	[3:0]:FIFO READ PTR high byte	00
0xb16	DFIFO0_WPTRL	RO	FIFO WRITE PTR low byte	00
0xb17	DFIFO0_WPTRH	RO	[3:0]:FIFO WRITE PTR high byte	00
0xb18	DFIFO1_RPTRL	RO	FIFO READ PTR low byte	00
0xb19	DFIFO1_RPTRH	RO	[3:0]:FIFO READ PTR high byte	00

Address	Mnemonic	Type	Description	Reset value
0xb1a	DFIFO1_WPTRL	RO	FIFO WRITE PTR low byte	00
0xb1b	DFIFO1_WPTRH	RO	[3:0]:FIFO WRITE PTR high byte	00
0xb1c	DFIFO2_RPTRL	RO	FIFO READ PTR low byte	00
0xb1d	DFIFO2_RPTRH	RO	[3:0]:FIFO READ PTR high byte	00
0xb1e	DFIFO2_WPTRL	RO	FIFO WRITE PTR low byte	00
0xb1f	DFIFO2_WPTRH	RO	[3:0]:FIFO WRITE PTR high byte	00
0xb20	DFIFO0_NUML	RO	FIFO DATA NUMBER low byte	00
0xb21	DFIFO0_NUMH	RO	FIFO DATA NUMBER high byte	00
0xb22	reserved			
0xb23	reserved			
0xb24	DFIFO1_NUML	RO	FIFO DATA NUMBER low byte	00
0xb25	DFIFO1_NUMH	RO	FIFO DATA NUMBER high byte	00
0xb26	reserved			
0xb27	reserved			
0xb28	DFIFO2_NUML	RO	FIFO DATA NUMBER low byte	00
0xb29	DFIFO2_NUMH	RO	FIFO DATA NUMBER high byte	00
0xb2a	reserved			
0xb2b	reserved			
0xb2c	DFIFO_MANUAL	RW	[0]: 0: fifo auto mode 1: enable fifo manual mode	00
0xb2d	reserved			
0xb2e	reserved			
0xb2f	reserved			
0xb30	DFIFO_MAN_DAT0	W	Fifo manual mode data in[7:0]	00
0xb31	DFIFO_MAN_DAT1	W	Fifo manual mode data in[15:8]	00
0xb32	DFIFO_MAN_DAT2	W	Fifo manual mode data in[23:16]	00
0xb33	DFIFO_MAN_DAT3	W	Fifo manual mode data in[31:24]	00
0xb34	reserved			
0xb35	AUDIO_CFG	RW	[0]: adc lnr valid select [1]: adc 64/63 option [2]: adc_bsign [3]: bypass adc trim [4]: audio input (USB/AMIC/DMIC/I2S) left/right channel swap	09
0xb36	ADC_MUL	RW	Adc cal mul	00
0xb37	ADC_BIAS	RW	Adc cal bias	00
0xb38-0xb3f	reserved			
0xb40	ALC_HPF_LPF_BYPASS	RW	[3:0]: Parameter to adjust HPF output. [4]: bypass HPF 1: bypass HPF, 0: use HPF [5]: bypass ALC 1: bypass ALC, 0: use ALC [6]: bypass LPF 1: bypass LPF, 0: use LPF [7]: enable double down sample	fb

Address	Mnemonic	Type	Description	Reset value
			1: enable, 0: disable	
0xb41	ALC_VOL_L	RW	ALC left channel setting [5:0]: digital gain in manual mode [6:0]:minimum gain limit in auto mode [7]: select auto or manual mode 1: auto mode, 0: manual mode	20
0xb42	ALC_VOL_R	RW	ALC right channel setting [5:0]: digital gain in manual mode [6:0]:minimum gain limit in auto mode [7]: select auto or manual mode 1: auto mode, 0: manual mode	20
0xb43	ALC_VOL_H	RW	[6:0]:maximum pga gain limit in auto mode	33
0xb44	ALC_VOL_THH	RW	[0]: high volume target db fra in auto mode [6:1]:high volume target db int[5:0] in auto mode	60
0xb45	reserved			
0xb46	ALC_VOL_THL	RW	[0]: low volume target db fra in auto mode [6:1]: low volume target db int[5:0] in auto mode	40
0xb47	reserved			
0xb48	ALC_VOL_THN	RW	[0]: volume noise level db fra in auto mode [6:1]:volume noise level db int[5:0] in auto mode	02
0xb49~0xb4c	reserved			
0xb4d	ALC_VOL_L_R	R	L channel: Current gain in auto mode volume in manual mode	
0xb4e	ALC_VOL_R_R	R	R channel: Current gain in auto mode volume in manual mode	
0xb4f	reserved			
0xb50	ALC_PEAK_TICK_L	RW	[7:0]:low byte of tick	00
0xb51	ALC_PEAK_TICK_H	RW	[7:0]:high byte of tick	02
0xb52~0xb53	reserved			
0xb54	ALC_CFG	RW	[0]: analog mode en [1]: vad iir en [2]: ana_iir_en [3]: iir tick clear en [4]: vad max en	0a
0xb55	ALC_COEF_IIR	RW	Iir coef	fa
0xb56	ALC_DAT_MASK	RW	[3:0]:In auto mode when gain changes, configure number of data be masked	00

Address	Mnemonic	Type	Description	Reset value
0xb57	ALC_INC_SPD	RW	In auto mode, pga gain increase speed	0a
0xb58	ALC_INC_MAX	RW	[6:0]:In auto mode, pga gain increase max in a peak tick cycle	03
0xb59	ALC_DEC_SPD	RW	In auto mode, pga gain decrease speed	08
0xb5a	ALC_DEC_MAX	RW	[6:0]:In auto mode, pga gain decrease max in a peak tick cycle	06
0xb5b	ALC_NOI_SPD	RW	In auto mode, pga gain decrease speed when as noise	06
0xb5c	ALC_NOI_MAX	RW	[6:0]:In auto mode, pga gain decrease max in a peak tick cycle when as noise	06
0xb5d~x0b5c	reserved			
0xb5e	CUR_PGA_GAIN_L	R	Current pga gain in auto analog mode	
0xb5f	CUR_PGA_GAIN_R	R	Current pga gain in auto analog mode	
0xb60	PGA_MAN_SPEED	RW	Pga manual mode judge speed	40
0xb61	PGA_MAN_TARGET_L	RW	Left channel [6:0]: pga manual mode target [7]: pga judge done	
0xb62	PGA_VALUE_L	R	Left channel [6:0]: PGA value in manual mode [7]: Mute status (R). Enable (0)/Mute (1) PGA output (W).	
0xb63	PGA_FIX_VALUE	RW	[6:0]: Pga gain fix value [7]: pga gain fix enable	0x80
0xb64	PGA_R_L	RW	[0]: change PGA L R CHANNEL	0x00
0xb65	PGA_MAN_TARGET_R	RW	Right channel [6:0]: pga manual mode target [7]: pga judge done	
0xb66	PGA_VALUE_R	R	Right channel [6:0]: PGA value in manual mode [7]: Mute status (R). Enable (0)/Mute (1) PGA output (W).	

9.3 Audio output path

Audio output path mainly includes Rate Matching module, SDMDAC (Sigma-Delta Modulation DAC) and I2S Player. The audio data fetched from SRAM is processed by the Rate Matching module, then transferred to the SDM/I2S Player as the input signal.
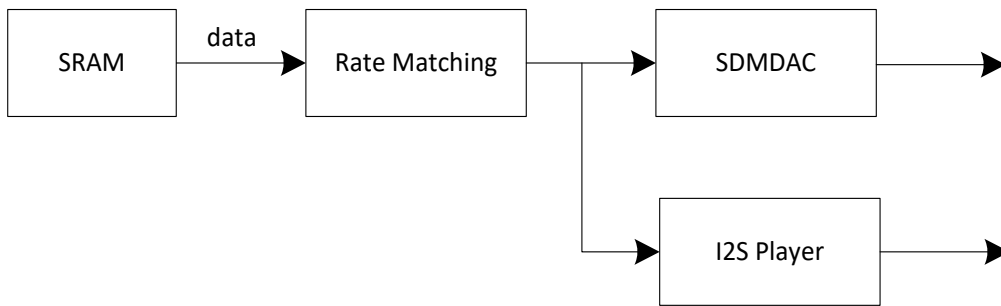
Figure 9- 3   Audio output path

9.3.1 Rate Matching

The rate matching block performs clock rate conversion and data synchronization between two domains: the input audio data is fetched from SRAM which works in system clock domain with 24Mhz/32Mhz/48Mhz clocks and the SDM/I2S which works between 4Mhz and 8Mhz.

When needed, the audio data from SRAM is interpolated to the SDM/I2S input rate. If the audio sampling rate is ClkUsbIn (e.g. 48kHz), and the working clock of SDM/I2S is aclk_i, then the interpolation ratio is given as follows:

$$\frac{ClkUsbIn}{aclk_i} = \frac{step_i}{0x80000}$$

Where step_i[19:0] is configured in addresses 0x567~0x565.

Linear interpolation or delay interpolation is used as shown below.
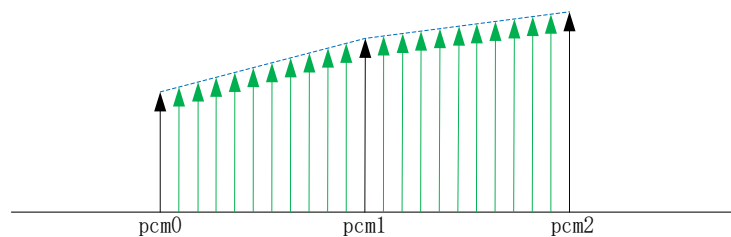
Figure 9- 4   Linear interpolation
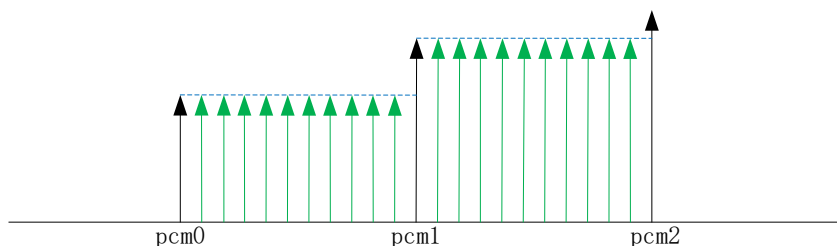
Figure 9- 5   Delay interpolation

9.3.2 SDM

The SDM takes 16bits audio data from SRAM and provides 1bit modulated output. Only a simple passive filter network is needed to drive audio device directly.

Dither control can be added to the SDM to avoid spurs in output data. There are three dithering options: PN sequence, PN sequence with Shapping, and DC constant; only one type of input is allowed any time.
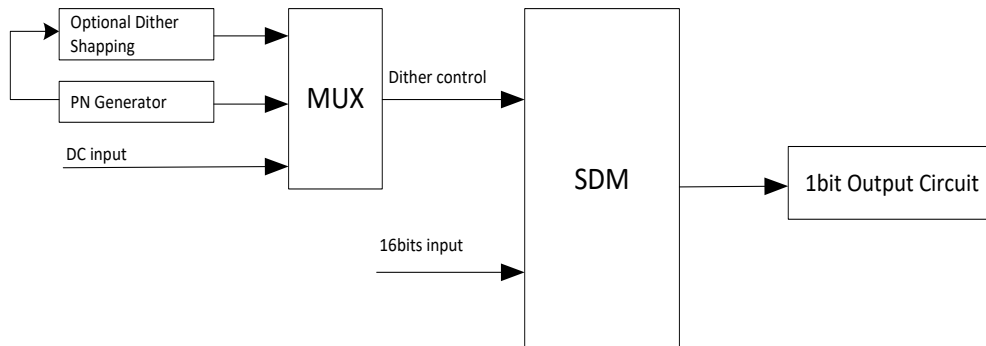
Figure 9- 6   Block diagram of SDM

9.3.3   Register configuration

Address 0x560[4:1] should be set to "1" to enable I2S recorder/ISO player/SDM player/I2S player, while bit[0] is to select either mono or stereo audio output. Address 0x560[7] should be set to "1" to enable the HPF in audio output path.

Register VOL_CTRL (address 0x562) serves to adjust volume level.

Address 0x563[2] serves to select either linear interpolation or delay interpolation for the rate matching block: Setting bit[2] to "1" is to select linear interpolation, while clearing the bit is to select delay interpolation.

Input for SDM Dither control is selectable via addresses 0x56b[6:5]), 0x563[6:5] and 0x568~0x569.

For the **left channel**:

1. Address 0x56b[5] should be set to 1b'1 to select constant DC input. When DC input is used, addresses 0x56c~0x56d serve to configure the input constant value.

2. Address 0x56b[5] should be set to 1b'0 to use PN generator. Address 0x563[5] serves to enable/mask dither shapping module. There are two PN generators to generate random dithering sequence; address 0x568 bit[6]/bit[5] is enabling bit of the two PN generators.

    1) To select PN sequence as input, address 0x56b[5] and 0x563[5] should be set to 0, 0x568[6]/[5]/[6:5] should be set to 1.

    2) To select PN sequence with Shapping as input, address 0x56b[5] should be set to 0, 0x563[5] and 0x568[6]/[5]/[6:5] should be set to 1.

    When PN sequence or PN with Shapping is used, address 0x568[4:0]/0x569[4:0] determines the number of bits (ranging from 0 to 16) used in PN1/PN2 generator.

For the **right channel**:

1. Address 0x56b[6] should be set to 1b'1 to select constant DC input. When DC input is enabled, addresses 0x56e~0x56f serve to configure the input constant value.

2. Address 0x56b[6] should be set to 1b'0 to use PN generator. Address 0x563[6] serves to enable/mask dither shapping module. There are two PN generators to generate random dithering sequence; address 0x569 bit[6]/bit[5] is enabling bit of the two PN generators.

   1) To select PN sequence as input, address 0x56b[6] and 0x563[6] should be set to 0, 0x569[6]/[5]/[6:5] should be set to 1.

   2) To select PN sequence with Shapping as input, address 0x56b[6] should be set to 0, 0x563[6] and 0x569[6]/[5]/[6:5] should be set to 1.

   When PN sequence or PN with Shapping is used, address 0x56a[4:0]/0x56b[4:0] determines the number of bits (ranging from 0 to 16) used in PN1/PN2 generator.

Address 0x567, 0x566 and 0x565[7:4] are to set step_i[19:0] for the rate matching block, while address 0x564 is to tune the step_i value. The step_i should be in accordance with the aclk_i provided by SDM/I2S clock.

<p align="center">Table 9- 3   Register configuration related to audio output path</p>

Address	Mnemonic	Type	Description	Reset value
0x560	AUDIO_CTRL	RW	[0]1--mono mode audio output, 0--stereo mode audio output [1]1--enable I2S player, 0--disable I2S player [2]1--enable SDM player, 0--disable SDM player [3]1--enable ISO player, 0--disable ISO player [4]1--enable I2S recorder , 0--disable I2s recorder [5]1--enable interface of I2S, 0--disable interface of I2S [6]1--enable GRP,0--disable GRP [7]1--enable HPF,0--disable HPF	04
0x561	MIDGRP	RW	[7:0]-- Middle of GRP	40
0x562	VOL_CTRL	RW	[0]--Add a quarter [1]--Add a half [6:2]--shift left [7]1--mute, 0--normal	40
0x563	PWM_CTRL	RW	[0]1--not multiply 2 when PWM, 0--mutiply2 [1]1--PWM, 0--not PWM [2]1-linear interpolate, 0-delay interpolate [4:3] reserved	64

Address	Mnemonic	Type	Description	Reset value
			[5]1-left Shapping used, 0-left Shapping not used [6]1-right Shapping used, 0-right Shapping not used [7]: I2S input left/right channel swap	
0x564	ASCL_TUNE	RW	[7:0] tune step_i for rate matching block	01
0x565	I2S_FCT	RW	[3:0]factor to generate I2S clock [7:4]low 4 bits of rate matching block step_i[3:0]	90
0x566	ASCL_STEP0	RW	[7:0] middle byte of rate matching block step_i[11:4]	c4
0x567	ASCL_STEP1	RW	[7:0]high byte of rate matching block step_i[19:12]	00
0x568	PN1_LEFT	RW	[4:0]bits used in pn1 of left channel, range from 0 to 16 [5]1-pn2 of left enable, 0-pn2 of left disable [6]1-pn1 of left enable, 0-pn1 of left disable	50
0x569	PN2_LEFT	RW	[4:0]bits used in pn2 of left channel, range from 0 to 16 [5]1-pn2 of right enable, 0-pn2 of right disable [6]1-pn1 of right enable, 0-pn1 of right disable	40
0x56a	PN1_RIGHT	RW	[4:0]bits used in pn1 of right channel, range from 0 to 16 [5]1-exchange data in between SDMs, 0-not exchanged	10
0x56b	PN2_RIGHT	RW	[4:0]bits used in pn2 of right channel, range from 0 to 16 [5]1-left channel use const value, 0-left channel use pn [6]1-right channel use const value, 0-right channel use pn	00
0x56c	CONST_LEFT0	RW	[7:0]low byte of left channel const, i.e, const_l[7:0]	00
0x56d	CONST_LEFT1	RW	[7:0]high byte of left channel const, i.e.const_l[15:8]	00
0x56e	CONST_RIGHT0	RW	[7:0]low byte of right channel const, i.e const_r[7:0]	00
0x56f	CONST_RIGHT1	RW	[7:0]high byte of right channel const, i.e const_r[15:8]	00

10 Quadrature Decoder

The TLSR8258F1KET48A embeds one quadrature decoder (QDEC) which is designed mainly for applications such as wheel. The QDEC implements debounce function to filter out jitter on the two phase inputs, and generates smooth square waves for the two phase.

10.1 Input pin selection

The QDEC supports two phase input; each input is selectable from the 8 pins of PortD, PortC, PortB and PortA via setting address 0xd2[2:0] (for channel a)/0xd3[2:0] (for channel b).

Table 10- 1   Input pin selection

Address 0xd2[2:0]/0xd3[2:0]	Pin
0	PA<2>
1	PA<3>
2	PB<6>
3	PB<7>
4	PC<2>
5	PC<3>
6	PD<6>
7	PD<7>

Note: To use corresponding IO as QDEC input pin, it's needed first to enable GPIO function, enable "IE" (1) and disable "OEN" (1) for this IO.

10.2 Common mode and double accuracy mode

The QDEC embeds an internal hardware counter, which is not connected with bus.

Address 0xd7[0] serves to select common mode or double accuracy mode.

For each wheel rolling step, two pulse edges (rising edge or falling edge) are generated.

If address 0xd7[0] is cleared to select common mode, the QDEC Counter value (real time counting value) is increased/decreased by 1 only when the same rising/falling edges are detected from the two phase signals.

One wheel rolling   Another wheel rolling

COUNT0 value
increased by 1

COUNT0 value
increased by 1

Another wheel rolling

One wheel rolling

COUNT0 value
decreased by 1

COUNT0 value
decreased by 1
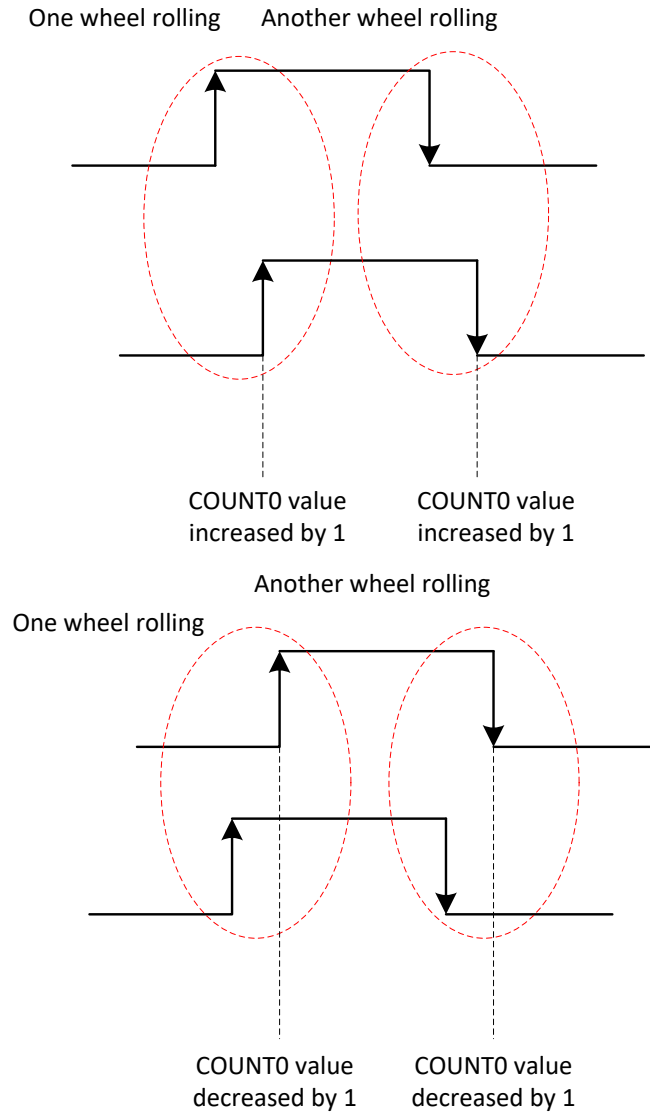
Figure 10- 1   Common mode

If address 0xd7[0] is set to 1b'1 to select double accuracy mode, the QDEC Counter value (real time counting value) is increased/decreased by 1 on each rising/falling edge of the two phase signals; the COUNT0 will be increased/decreased by 2 for one wheel rolling.

One wheel rolling   Another wheel rolling

COUNT0 value
increased by 1

COUNT0 value
increased by 1

COUNT0 value
increased by 1

COUNT0 value
increased by 1

Another wheel rolling

One wheel rolling

COUNT0 value
decreased by 1

COUNT0 value
decreased by 1

COUNT0 value
decreased by 1

COUNT0 value
decreased by 1
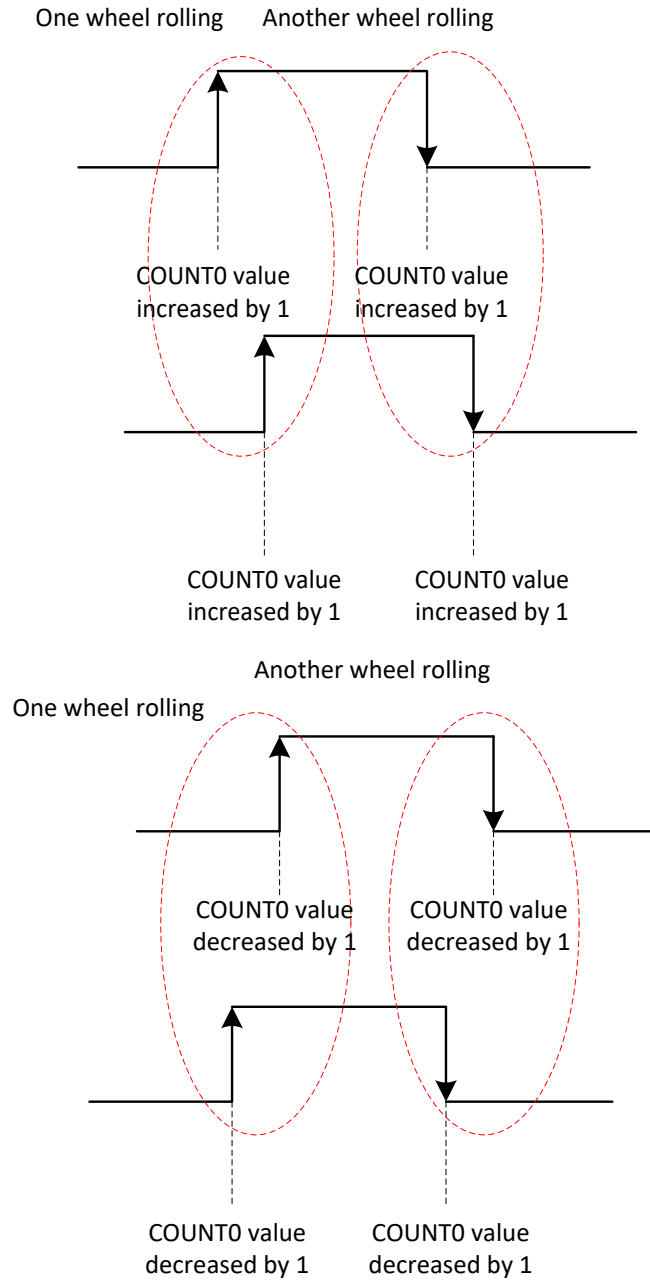
Figure 10- 2   Double accuracy mode

10.3 Read real time counting value

Neither can Hardware Counter value be read directly via software, nor can the counting value in address 0xd0 be updated automatically.

To read real time counting value, first write address 0xd8[0] with 1b'1 to load Hardware Counter data into the QDEC_COUNT register, then read address 0xd0.
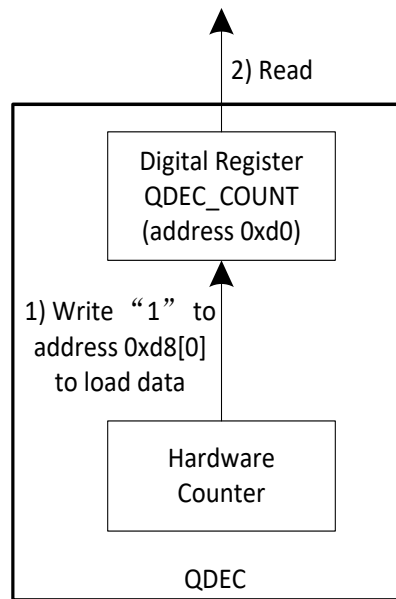
Figure 10- 3   Read real time counting value

10.4 QDEC reset

Address 0x60[5] serves to reset the QDEC. The QDEC Counter value is cleared to zero.

10.5 Other configuration

The QDEC supports hardware debouncing. Address 0xd1[2:0] serves to set filtering window duration. All jitter with period less than the value will be filtered out and thus does not trigger count change.

Address 0xd1[4] serves to set input signal initial polarity.

Address 0xd1[5] serves to enable shuttle mode. Shuttle mode allows non-overlapping two phase signals as shown in the following figure.
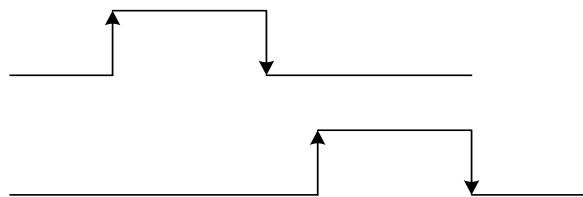
Figure 10- 4   Shuttle mode

10.6 Timing sequence
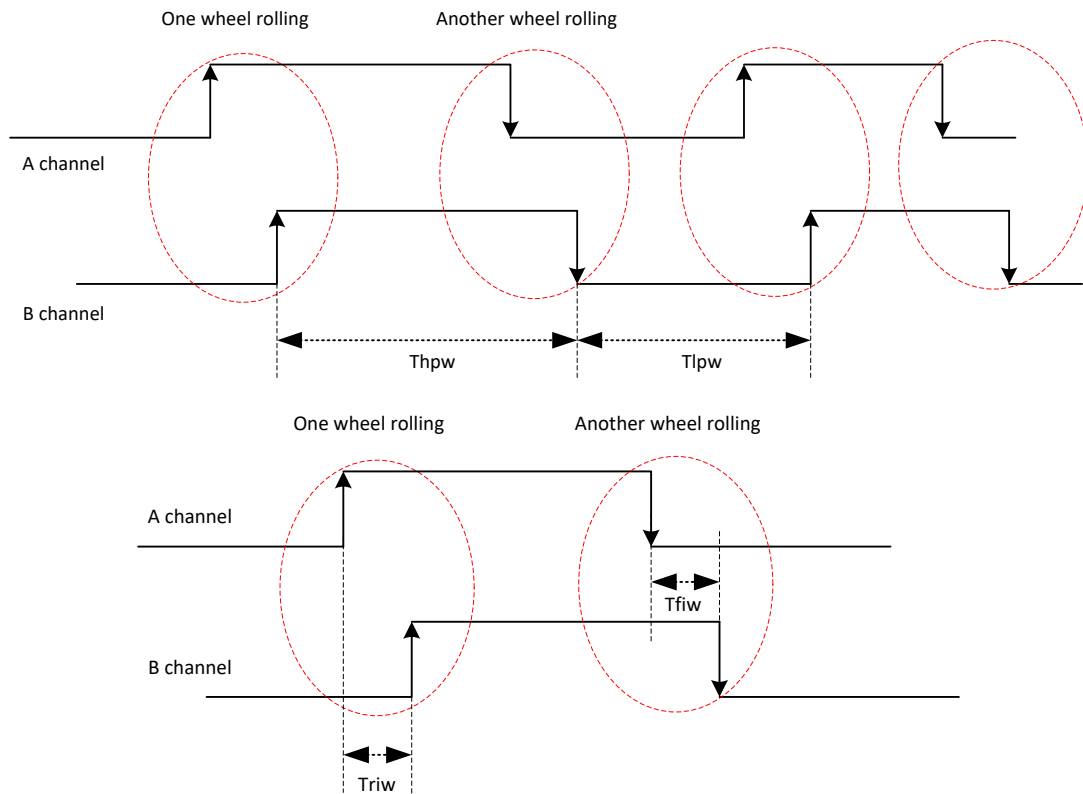
Figure 10- 5   Timing sequence chart

Table 10- 2   Timing

Time interval	Min Value
Thpw (High-level pulse width)	$2^{(n+1)} *clk\_32kHz *3$ (n=0xd1[2:0])
Tlpw (Low-level pulse width)	$2^{(n+1)} *clk\_32kHz *3$ (n=0xd1[2:0])
Triw (Interval width between two rising edges)	$2^{(n+1)} *clk\_32kHz$ (n=0xd1[2:0])
Tfiw (Interval width between two falling edges)	$2^{(n+1)} *clk\_32kHz$ (n=0xd1[2:0])

QDEC module works based on 32kHz clock to ensure it can work in suspend mode. QDEC module supports debouncing function, and any signal with width lower than the threshold (i.e. "$2^{(n+1)}$ *clk_32kHz *3 (n=0xd1[2:0])) will be regarded as jitter. Therefore, effective signals input from Channel A and B should contain high/low level with width Thpw/Tlpw more than the threshold. The 2^n *clk_32kHz clock is used to synchronize input signal of QDEC module, so the interval between two adjacent rising/falling edges from Channel A and B, which are marked as Triw and Tfiw, should exceed "$2^{(n+1)}$ *clk_32kHz".

Only when the timing requirements above are met, can QDEC module recognize wheel rolling times correctly.

10.7 Register table

Table 10- 3   Register table for QDEC

Address	Mnemonic	Type	Description	Reset value
0xd0	QDEC_COUNT	R	QDEC Counting value (read to clear): Pulse edge number	0x00
0xd1	QDEC_CC	R/W	[2:0] : filter time (can filter 2^n *clk_32k*2 width deglitch) [4]: pola, input signal pola 0: no signal is low, 1: no signal is high [5]:shuttle mode 1 to enable shuttle mode	0x00
0xd2	QDEC_CHNA	R/W	[2:0] QDEC input pin select for channel a choose 1 of 8 pins for input channel a 7~0: {pd[7:6],pc[3:2],pb[7:6],pa[3:2]}	0x00
0xd3	QDEC_CHNB	R/W	[2:0] QDEC input pin select for channel b choose 1 of 8 pins for input channel b 7~0: {pd[7:6],pc[3:2],pb[7:6],pa[3:2]}	0x01
0xd6	QDEC_RST	R/W	[0]RSVD	0x00
0xd7	QDEC_DOUBLE	R/W	[0]Enable double accuracy mode	0x01
0xd8	DATA_LOAD	R/W	[0]write 1 to load data when load completes it will be 0	0x00

11 SAR ADC

The TLSR8258F1KET48A integrates one SAR ADC module, which can be used to sample analog input signals such as battery voltage, mono or stereo audio signals.
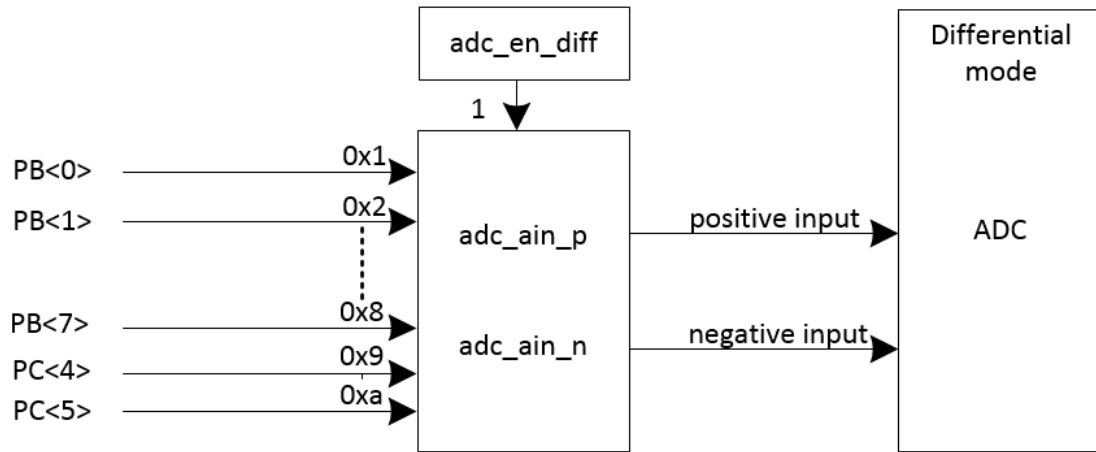
Figure 11- 1   Block diagram of ADC

11.1 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.

11.2 ADC clock

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

$$\text{ADC clock frequency (marked as } F_{ADC\_clk}) = 24\text{MHz}/(adc\_clk\_div+1)$$

11.3 ADC control in auto mode

11.3.1 Set max state and enable channel

The SAR ADC supports up to three channels including left channel, right channel and Misc channel. The left, right and Misc channels all consist of one "Set" state and one "Capture" state.

✧ The analog register r_max_scnt (afe_0xf2<6:4>) serves to set the max state index. As shown in the example below, the r_max_scnt should be set as 0x06.

1	2	3	4	5	6
Set	Capture	Set	Capture	Set	Capture

✧ The left/Misc channel can be enabled independently via r_en_left (afe_0xf2<0>), r_en_misc (afe_0xf2<2>).

✧ Only when the left channel is enabled, can the right channel be enabled via r_en_right (afe_0xf2<1>).

✧ To sample mono audio signals, the left channel should be enabled. To sample stereo audio signals, both the left channel and the right channel should be enabled.

11.3.2 "Set" state

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

$$\text{"Set" state duration (marked as } T_{sd}) = r\_max\_s / 24\text{MHz}.$$

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

✧ adc_en_diff: afe_0xec<4> (left channel), afe_0xec<5> (right channel), afe_0xec<6> (Misc channel). MUST set as 1b'1 to select differential input mode.

✧ adc_ain_p: afe_0xe8<7:4> (Misc channel), afe_0xe9<7:4> (left channel), afe_0xea<7:4> (right channel). Select positive input in differential mode.

✧ adc_ain_n: afe_0xe8<3:0> (Misc channel), afe_0xe9<3:0> (left channel), afe_0xea<3:0> (right channel). Select negative input in differential mode.

✧ adc_vref: afe_0xe7<1:0> (left channel), afe_0xe7<3:2> (right channel), afe_0xe7<5:4> (Misc channel). Set reference voltage V_{REF}. ADC maximum input range is the 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_0xeb<1:0> (left channel), afe_0xeb<5:4> (right channel), afe_0xec<1:0> (Misc channel). Set resolution as 8/10/12/14 bits.

   ADC data is always 15-bit format no matter what the resolution is set. For example, 14 bits resolution indicates ADC data consists of 14-bit valid data and 1-bit sign extension bit.

✧ adc_tsamp: afe_0xed<3:0> (left channel), afe_0xed<7:4> (right channel), afe_0xee<3:0> (Misc channel). Set sampling time which determines the speed to stabilize input signals.

$$\text{Sampling time (marked as } T_{samp}) = adc\_tsamp / F_{ADC\_clk}.$$

The lower sampling cycle, the shorter ADC convert time.

✧ pga_boost, pga_gain: Set PGA gain in Boost stage and Gain stage. See PGA section.

11.3.3   "Capture" state

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

✧ The length of "Capture" state for Misc channel 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 length of "Capture" state for left and right channel is configurable via the analog register r_max_c[9:0] (afe_0xf1<5:4>, afe_0xf0<7:0>).

   "Capture" state duration for left & right channel (marked as T_{cd}) = r_max_c / 24MHz.

✧ The "VLD" bit (afe_0xf8<7>) 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 15-bit ADC output data for Misc channel can be read from the analog register adc_dat[14:0] (afe_0xf8<6:0>, afe_0xf7<7:0>).

Note: The total duration "T_{td}", which is the sum of the length of "Set" state and "Capture" state for all channels available, determines the sampling rate.

$$\text{Sampling frequency (marked as } F_s) = 1 / T_{td}$$

11.3.4   Usage cases

11.3.4.1   Case 1: 3-channel sampling for stereo audio and Misc

In this case, afe_0xf2<3:0> should be set as 0x7, so as to enable the left, right and Misc channels, the max state index should be set as "6" by setting afe_0xf2<6:4> as 0x6.

The total duration (marked as T_{td}) = (1*r_max_mc+3*r_max_s+2*r_max_c) / 24MHz.

r_max_s	r_max_c	r_max_s	r_max_c	r_max_s	r_max_mc
1	2	3	4	5	6
Set	Capture	Set	Capture	Set	Capture
←—Left—→		←—Right—→		←—Misc—→	
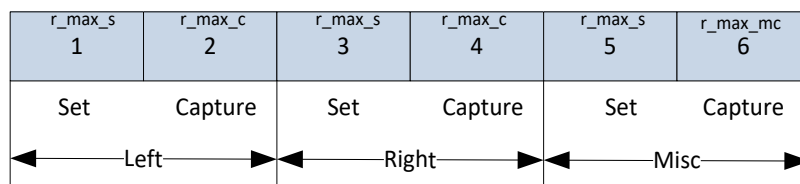
11.3.4.2   Case 2: 2-channel sampling for mono audio and Misc

In this case, afe_0xf2<3:0> should be set as 0x5, so as to enable the left and Misc channels and disable the right channel, the max state index should be set as "4" by setting afe_0xf2<6:4> as 0x4.

The total duration (marked as T_{td}) = (1*r_max_mc+2*r_max_s+1*r_max_c) / 24MHz.

r_max_s	r_max_c	r_max_s	r_max_mc
1	2	3	4
Set	Capture	Set	Capture
←—Left—→		←—Misc—→	
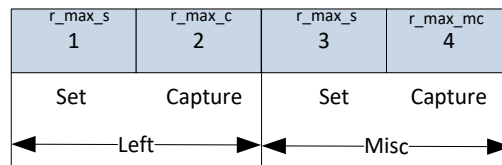
11.3.4.3   Case 3: 2-channel sampling for stereo audio

In this case, afe_0xf2<3:0> should be set as 0x3, so as to enable the left and right channels and disable the Misc channel, the max state index should be set as "4" by setting afe_0xf2<6:4> as 0x4.

The total duration (marked as T_{td}) = (2*r_max_s+2*r_max_c) / 24MHz.

r_max_s	r_max_c	r_max_s	r_max_c
1	2	3	4
Set	Capture	Set	Capture
←—Left—→		←—Right—→	
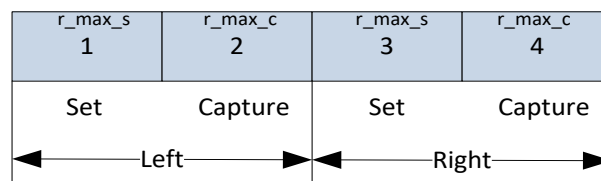
11.3.4.4   Case 4: 1-channel sampling for mono audio

In this case, afe_0xf2<3:0> should be set as 0x1, so as to enable the left channel and disable the

right and Misc channels, the max state index should be set as "2" by setting afe_0xf2<6:4> as 0x2.

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

r_max_s 1	r_max_c 2
Set	Capture
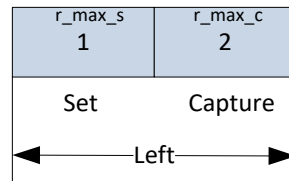
←————Left————→

11.3.4.5 Case 5: 1-channel sampling for Misc

In this case, afe_0xf2<3:0> should be set as 0x4, so as to enable the Misc channel and disable the left and right channels, the max state index should be set as "2" by setting afe_0xf2<6:4> as 0x2.

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

r_max_s 1	r_max_mc 2
Set	Capture
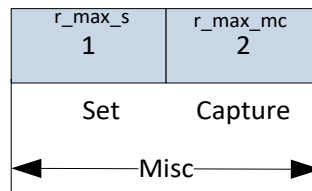
←————Misc————→

11.3.4.6 Case 6 with detailed register setting

This case introduces the register setting details for 3-channel sampling of left, right and Misc channels.

r_max_s 1	r_max_c 2	r_max_s 3	r_max_c 4	r_max_s 5	r_max_mc 6
Set	Capture	Set	Capture	Set	Capture
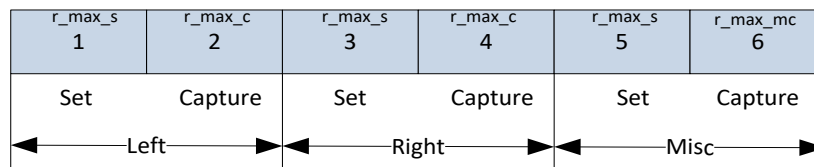
←———Left———→ ←———Right———→ ←———Misc———→

Table 11- 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)=4MHz
Enable the left, right and Misc channels	afe_0xf2<3:0> = 0x7
Set the max state index as "6"	afe_0xf2<6:4> = 0x6

Table 11- 2   Register setting for L/R/M channel

Function	Register setting		
	Left	**Right**	**Misc**
Set T_{sd} ("Set" state duration)	afe_0xf1<3:0> = 10 T_{sd} = r_max_s/24MHz = 10/24MHz = 0.417us		
Set T_{cd} ("Capture" state duration)	afe_0xf1<5:4>=0, afe_0xf0<7:0>=170 T_{cd} = r_max_c[9:0]/24MHz = 170/24MHz = 7.1us		afe_0xf1<7:6>=0, afe_0xef<7:0>=130 T_{cd} = r_max_mc[9:0]/24MHz =130/24MHz = 5.4us
T_{td} (total duration)	T_{td} = (1*r_max_mc+3*r_max_s+2*r_max_c) / 24MHz = 500/24MHz =20.83us		
F_s (Sampling frequency)	F_s = 1 / T_{td} = 24MHz/500 = 48kHz		
Select differential input	afe_0xec<4>=1 differential input	afe_0xec<5>=1 differential input	afe_0xec<6>=1 differential input
Set input channel	afe_0xe9=0x12 Select B<0> and B<1> as positive input and negative input	afe_0xea=0x34 Select B<2> and B<3> as positive input and negative input	afe_0xe8=0xaf Select C<5> as positive input, select GND as negative input
Set reference voltage V_{REF}	afe_0xe7<1:0>=0 V_{REF} = 0.6V	afe_0xe7<3:2>=1 V_{REF} = 0.9V	afe_0xe7<5:4>=2 V_{REF} =1.2V
Set scaling factor for ADC analog input	afe_0xfa<7:6>=0 scaling factor: 1		
	ADC maximum input range: -0.6~+0.6V	ADC maximum input range: -0.9~+0.9V	ADC maximum input range: 0 ~ +1.2V
Set resolution	afe_0xeb<1:0>=1 resolution: 10bits	afe_0xeb<5:4>=2 resolution: 12bits	afe_0xec<1:0>=3 resolution: 14bits
Set T_{samp} (determines the speed to stabilize input before sampling)	afe_0xed<3:0>=1 T_{samp} = adc_tsamp / F_{ADC_clk} = 6/4MHz=1.5us	afe_0xed<7:4>=2 T_{samp} = adc_tsamp / F_{ADC_clk} = 9/4MHz=2.25us	afe_0xee<3:0>=3 T_{samp} = adc_tsamp / F_{ADC_clk} = 12/4MHz=3us

11.4 Register table

Table 11- 3   Register table related to SAR ADC

Address	Mnemonic	Default value	Description
afe_0xe7<1:0>	adc_vrefl	00	Select V_{REF} for left channel 0x0: 0.6V 0x1: 0.9V 0x2: 1.2V 0x3: rsvd
afe_0xe7<3:2>	adc_vrefr	00	Select V_{REF} for right channel 0x0: 0.6V 0x1: 0.9V 0x2: 1.2V 0x3: rsvd
afe_0xe7<5:4>	adc_vrefm	00	Select V_{REF} for Misc channel 0x0: 0.6V 0x1: 0.9V 0x2: 1.2V 0x3: rsvd
afe_0xe7<7:6>	RSVD		
afe_0xe8<3:0>	adc_ain_m_n	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: pga_n<0> (PGA left-channel negative output) 0xc: pga_n<1> (PGA right-channel negative output) 0xd: tempsensor_n (Temperature sensor negative output) 0xe: Ground 0xf: Ground
afe_0xe8<7:4>	adc_ain_m_p	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: pga_p<0> (PGA left-channel positive output) 0xc: pga_p<1> (PGA right-channel positive output)

Address	Mnemonic	Default value	Description
			0xd: tempsensor_p (Temperature sensor positive output) 0xe: rsvd 0xf: rsvd
afe_0xe9<3:0>	adc_ain_l_n	00	Select negative input for left channel 0x0: No input 0x1: B<0> 0x2: B<1> ... 0x8: B<7> 0x9: C<4> 0xa: C<5> 0xb: pga_n<0> (PGA left-channel negative output) 0xc: pga_n<1> (PGA right-channel negative output) 0xd: tempsensor_n (Temperature sensor negative output) 0xe: Ground 0xf: Ground
afe_0xe9<7:4>	adc_ain_l_p	00	Select positive input for left channel: 0x0: No input 0x1: B<0> 0x2: B<1> ... 0x8: B<7> 0x9: C<4> 0xa: C<5> 0xb: pga_p<0> (PGA left-channel positive output) 0xc: pga_p<1> (PGA right-channel positive output) 0xd: tempsensor_p (Temperature sensor positive output) 0xe: rsvd 0xf: rsvd
afe_0xea<3:0>	adc_ain_r_n	00	Select negative input for right channel: 0x0: No input 0x1: B<0> 0x2: B<1> ... 0x8: B<7> 0x9: C<4> 0xa: C<5> 0xb: pga_n<0> (PGA left-channel negative output) 0xc: pga_n<1> (PGA right-channel negative output)

Address	Mnemonic	Default value	Description
			0xd: tempsensor_n (Temperature sensor negative output) 0xe: Ground 0xf: Ground
afe_0xea<7:4>	adc_ain_r_p	0000	Select positive input for right channel: 0x0: No input 0x1: B<0> 0x2: B<1> ... 0x8: B<7> 0x9: C<4> 0xa: C<5> 0xb: pga_p<0> (PGA left-channel positive output) 0xc: pga_p<1> (PGA right-channel positive output) 0xd: tempsensor_p (Temperature sensor positive output) 0xe: rsvd 0xf: rsvd
afe_0xeb<1:0>	adc_resl	11	Set resolution for left channel 0x0: 8bits 0x1: 10bits 0x2: 12bits 0x3: 14bits
afe_0xeb<3:2>	RSVD		
afe_0xeb<5:4>	adc_resr	11	Set resolution for right channel 0x0: 8bits 0x1: 10bits 0x2: 12bits 0x3: 14bits
afe_0xeb<7:6>	RSVD		
afe_0xec<1:0>	adc_resm	11	Set resolution for Misc channel 0x0: 8bits 0x1: 10bits 0x2: 12bits 0x3: 14bits
afe_0xec<3:2>	RSVD		
afe_0xec<4>	adc_en_diffl	0	Select input mode for left channel. 0: rsvd 1: differential mode
afe_0xec<5>	adc_en_diffr	0	Select input mode for right channel. 0: rsvd 1: differential mode

Address	Mnemonic	Default value	Description
afe_0xec<6>	adc_en_diffm	0	Select input mode for Misc channel. 0: rsvd 1: differential mode
afe_0xec<7>	RSVD		
afe_0xed<3:0>	adc_tsampl	0000	Number of ADC clock cycles in sampling phase for left channel to stabilize the input before sampling: 0x0: 3 cycles 0x1: 6 cycles 0x2: 9 cycles 0x3: 12 cycles … 0xf: 48 cycles
afe_0xed<7:4>	adc_tsampr	0000	Number of ADC clock cycles in sampling phase for right channel to stabilize the input before sampling: 0x0: 3 cycles 0x1: 6 cycles 0x2: 9 cycles 0x3: 12 cycles … 0xf: 48 cycles
afe_0xee<3:0>	adc_tsampm	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
afe_0xef<7:0>	r_max_mc[7:0]		r_max_mc[9:0]serves to set length of "capture" state for Misc channel. r_max_c[9:0] serves to set length of "capture" state for left and right channel. r_max_s serves to set length of "set" state for left, right and Misc channel. Note: State length indicates number of 24M clock cycles occupied by the state.
afe_0xf0<7:0>	r_max_c[7:0]		
afe_0xf1<3:0>	r_max_s		
afe_0xf1<5:4>	r_max_c[9:8]		
afe_0xf1<7:6>	r_max_mc[9:8]		
afe_0xf2<0>	r_en_left	0	Enable left channel. 1: enable
afe_0xf2<1>	r_en_right	0	Enable right channel. 1: enable
afe_0xf2<2>	r_en_misc		Enable Misc channel sampling. 1: enable
afe_0xf2<3>	rsvd	0	rsvd
afe_0xf2<6:4>	r_max_scnt	00	Set total length for sampling state machine (i.e. max state index)

Address	Mnemonic	Default value	Description
afe_0xf2<7>	rsvd		
afe_0xf3<7:0>	rsvd		
afe_0xf4<2:0>	adc_clk_div	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		rsvd
afe_0xf6<7:0>	rsvd		rsvd
afe_0xf7<7:0>	adc_dat[7:0]		Read only, Misc adc dat[7:0]
afe_0xf8<7:0>	adc_dat[15:8]		Read only [7]: 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.) [6:0]: Misc adc_dat[14:8]
afe_0xf9<3:2>	rsvd	00	rsvd
afe_0xfa<7:6>	adc_sel_ai_scale	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>	rsvd	0	rsvd
afe_0xfc<5>	adc_pd	1	Power down ADC 1: Power down 0: Power up

12 PGA

The TLSR8258F1KET48A integrates a PGA (Programmable Gain Amplifier) module.

The PGA supports two channels including left channel and right channel, and each channel consists of Boost stage pre-amplifier and Gain stage post-amplifier.

The PGA is used in combination with the audio and ADC module: By adjusting the gain of pre-amplifier and post-amplifier, the PGA can amplify mono or stereo differential input audio signals from specific AMIC pins before ADC sampling.
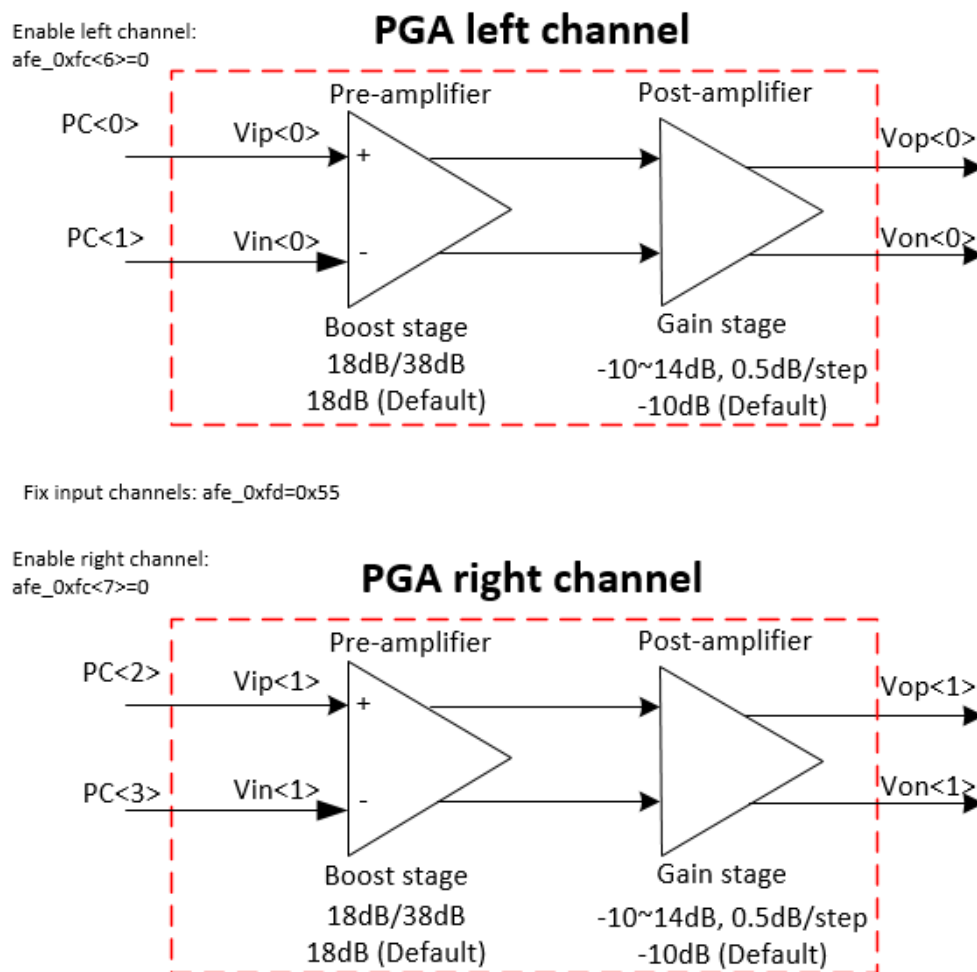
Figure 12- 1   Block diagram of PGA

*Note:

   Vip<0>, Vin<0>: Positive/Negative input of PGA left channel;

   Vop<0>, Von<0>: Positive/Negative output of PGA left channel;

   Vip<1>, Vin<1>: Positive/Negative input of PGA right channel;

   Vop<1>, Von<1>: Positive/Negative output of PGA right channel.

12.1 Power on/down

Both PGA left channel and right channel are disabled by default.

To power on PGA left/right channel, the analog register pga_pd_l (afe_0xfc<6>) / pga_pd_r (afe_0xfc<7>) should be set as 1b'0.

12.2 Input channel

The analog register afe_0xfd must be set as 0x55, i.e. PGA input channel is fixed as PC<0>~PC<3>.

✧ PC<0>: positive input for PGA left channel.

✧ PC<1>: negative input for PGA left channel.

✧ PC<2>: positive input for PGA right channel.

✧ PC<3>: negative input for PGA right channel.

12.3 Adjust gain

*NOTE: Since the ADC right channel cannot be used independently, gain of the PGA right channel is only adjustable when enabling both the ADC left channel and right channel.

To adjust the PGA gain, the ALC of the audio module should be configured as analog mode, and the following three cases can apply:

1) Auto regulate

If auto regulate function is enabled, user can set an initial PGA gain. The (AMIC input * current PGA gain) is compared with the pre-configured high volume target, low volume target and volume noise level, and the PGA gain will be automatically adjusted within the pre-configured range accordingly.

Please refer to section **9.2.5.1   Auto regulate in analog mode** for details.

2) Manual mode 1

If manual mode 1 is enabled, the PGA gain is directly adjustable via digital register 0xb63: Address 0xb63[6] serves to set gain for the pre-amplifier as 18dB (1b'0, default) or 38dB (1b'1); while address 0xb63[5:0] serves to set gain for the post-amplifier as -10dB (0x0, default) ~ 14dB (0x30) with step of 0.5dB. The total PGA gain should be the sum of the two gain values.

Please refer to section **9.2.5.2   Manual regulate in analog mode** for details.

3) Manual mode 2

If manual mode 2 is enabled, the PGA gain will be automatically adjusted to the pre-configured target gain value with the pre-configured speed.

Please refer to section **9.2.5.2   Manual regulate in analog mode** for details.

12.4 Register table

Table 12- 1   Analog register table related to PGA

Address	Mnemonic	Default	Description
afe_0xfc<6>	pga_pd_l	1	Power down left channel PGA 1: Power down, 0: Power up
afe_0xfc<7>	pga_pd_r	1	Power down right channel PGA 1: Power down, 0: Power up
afe_0xfd<1:0>	pga_sel_vin_l_p	00	Select left channel positive input source. Gate off all input with pga_pd_l. 0x1: C<0> 0x0: No input 0x2: No input 0x3: No input
afe_0xfd<3:2>	pga_sel_vin_l_n	00	Select left channel negative input source. Gate off all input with pga_pd_l. 0x1: C<1> 0x0: No input 0x2: No input 0x3: No input
afe_0xfd<5:4>	pga_sel_vin_r_p	00	Select right channel positive input source. Gate off all input with pga_pd_r. 0x1: C<2> 0x0: No input 0x2: No input 0x3: No input
afe_0xfd<7:6>	pga_sel_vin_r_n	00	Select right channel negative input source. Gate off all input with pga_pd_r. 0x1: C<3> 0x0: No input 0x2: No input 0x3: No input

For digital registers related to the PGA, please refer to section **9.2.5** and **9.2.6**.

13 Low Power Comparator

The TLSR8258F1KET48A 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>, 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.
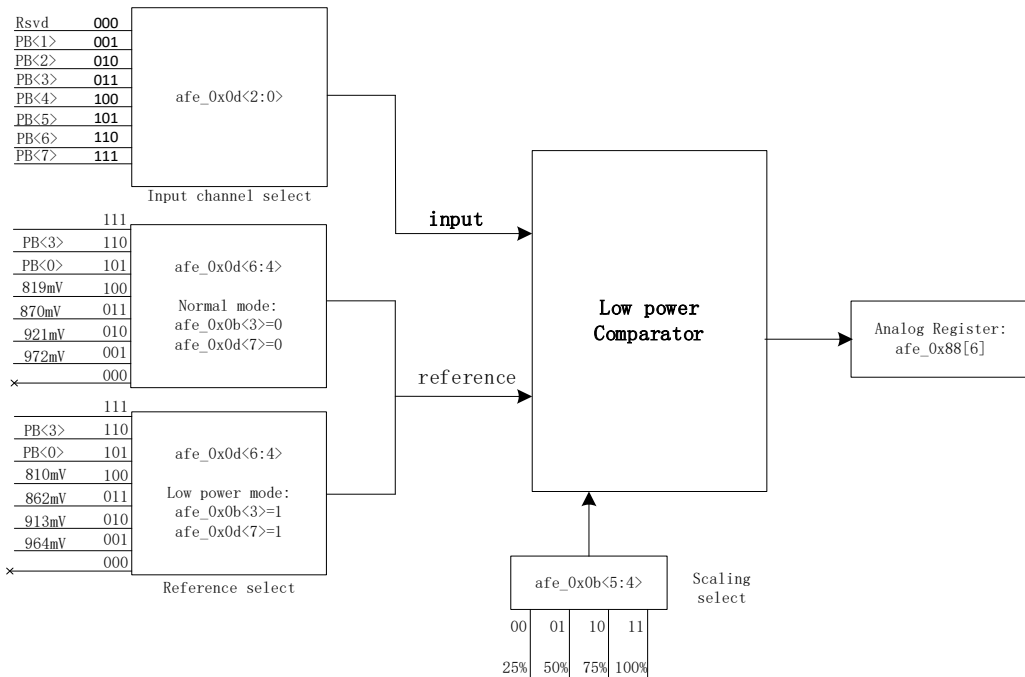
Figure 13- 1   Block diagram of low power comparator

13.1 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.

13.2 Select input channel

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

13.3 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 (10uA); reference voltage input channel is selectable from internal reference of 972mV, 921mV, 870mV and 819mV, as well as PB<0>, PB<3>, 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 50nA; reference voltage input channel is selectable from internal reference of 964mV, 913mV, 862mV and 810mV, as well as PB<0>, PB<3>, and float.

13.4 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%.

13.5 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.

13.6 Register table

Table 13- 1   Analog register table related to low power comparator

Address	Name	Description	Default Value
afe_0x07<3>	pd_lc_comp_3V	Power on/down low power comparator: 0: Power up 1: Power down	1
afe_0x0b<3>	ref_mode_sel	Reference mode select: 0: normal mode (current bias 10uA) 1: low power mode (current bias 50nA) See afe_0x0d<7>.	1
afe_0x0b<5:4>	comp_refscale<1:0>	Reference voltage scaling: 00: 25% 01: 50% 10: 75% 11: 100%	01

Address	Name	Description	Default Value
afe_0x0d<2:0>	comp_chsel<2:0>	Input Channel select: 000: rsvd 001: B<1> 010: B<2> 011: B<3> 100: B<4> 101: B<5> 110: B<6> 111: B<7>	000
afe_0x0d<3>	vbus_inen	rsvd	0
afe_0x0d<6:4>	comp_refesel<2:0>	Reference select: normal mode low power mode 000: Float 000: Float 001: 972mV 001: 964mV 010: 921mV 010: 913mV 011: 870mV 011: 862mV 100: 819mV 100: 810mV 101: B<0> 101: B<0> 110: B<3> 110: B<3> 111: float 111: float	000
afe_0x0d<7>	pd_l10u	Enable or disable 10uA current bias: 0: Enable 10uA current bias 1: Disable 10uA current bias	1

14 AES

The TLSR8258F1KET48A embeds AES module with encryption and decryption function. The input 128bit plaintext in combination of key is converted into the final output ciphertext via encryption; the 128bit ciphertext in combination of key can also be converted into 128bit plaintext via decryption.

The AES hardware accelerator provides automatic encryption and decryption. It only takes (1000*system clock cycles) to implement AES encryption/decryption. Suppose system clock is 20MHz, the time needed for AES encryption/decryption is 50us.

Both RISC mode and DMA mode are supported for AES operation.

14.1 RISC mode

For RISC mode, configuration of related registers is as follows:

1) Set the value of key via writing registers AES_KEY0~ AES_KEY15 (address 0x550~0x55f).

2) Set operation method of AES module via register AES_CTRL: set address 0x540[0] as 1b'1 for decryption method, while clear this bit for encryption method.

3) For encryption method, write registers AES-DAT0~ AES-DAT3 (address 0x548~0x54b) for four times to set the 128bit plaintext. After encryption, the 128bit ciphertext can be obtained by reading address 0x548~0x54b for four times.

4) For decryption method, write registers AES-DAT0~ AES-DAT3 (address 0x548~0x54b) for four times to set the 128bit ciphertext. After decryption, the 128bit plaintext can be obtained by reading address 0x548~0x54b for four times.

5) Address 0x540 bit[1] and bit[2] are read only bits: bit[1] will be cleared automatically after quartic writing of address 0x548~0x54b; bit[2] will be set as 1 automatically after encryption/decryption, and then cleared automatically after quartic reading of address 0x548~0x54b.

14.2 DMA mode

As for DMA mode, it is only needed to configure the value of key and encryption/decryption method for AES module. Please refer to point 1) ~ 2) in section **15.1**.

14.3 AES-CCM

The AES-CCM (Counter with the CBC-MAC) mode is disabled by default. AES output is directly determined by current encryption and decryption, irrespective of previous encryption and decryption result.

If 0x540[7] is set as 1b'1 to enable AES-CCM mode, AES output will also take previous encryption and decryption result into consideration.

14.4 Register table

Table 14- 1　Register table related to AES

Address	Mnemonic	Type	Description	Reset Value
0x540	AES_CTRL	R/W	[0] Select decrypt/encrypt. 1: decrypt, 0: encrypt [1] Read-only. 1: input data needed, 0: input data ready. [2] Read-only. 0: output data not ready, 1: output data ready. [7] 1: enable AES-CCM mode.	0x02
0x548	AES-DAT0		Input/Output Data byte 0	0x00
0x549	AES-DAT1		Input/Output Data byte 1	0x00
0x54a	AES-DAT2		Input/Output Data byte 2	0x00
0x54b	AES-DAT3		Input/Output Data byte 3	0x00
0x550	AES_KEY0	R/W	[7:0] KEY0	0x00
0x551	AES_KEY1	R/W	[7:0] KEY1	0x00
0x552	AES_KEY2	R/W	[7:0] KEY2	0x00
0x553	AES_KEY3	R/W	[7:0] KEY3	0x00
0x554	AES_KEY4	R/W	[7:0] KEY4	0x00
0x555	AES_KEY5	R/W	[7:0] KEY5	0x00
0x556	AES_KEY6	R/W	[7:0] KEY6	0x00
0x557	AES_KEY7	R/W	[7:0] KEY7	0x00
0x558	AES_KEY8	R/W	[7:0] KEY8	0x00
0x559	AES_KEY9	R/W	[7:0] KEY9	0x00
0x55a	AES_KEY10	R/W	[7:0] KEY10	0x00
0x55b	AES_KEY11	R/W	[7:0] KEY11	0x00
0x55c	AES_KEY12	R/W	[7:0] KEY12	0x00
0x55d	AES_KEY13	R/W	[7:0] KEY13	0x00
0x55e	AES_KEY14	R/W	[7:0] KEY14	0x00
0x55f	AES_KEY15	R/W	[7:0] KEY15	0x00

15 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.

15.1 Absolute maximum ratings

Table 15- 1   Absolute Maximum Ratings

Characteristics	Sym.	Min.	Max	Unit	Test Condition
Supply Voltage	VDD	-0.3	3.6	V	All AVDD, DVDD and VDD_IO pin must have the same voltage
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	ºC	
Soldering Temperature	T_{Sld}		260	ºC	

CAUTION: 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.

15.2 Recommended operating condition

Table 15- 2   Recommended operation condition

Item	Sym.	Min	Typ.	Max	Unit	Condition
Power-supply voltage	VDD	1.8	3.3	3.6	V	All AVDD, DVDD and VDD_IO pin must have the same voltage
Supply rise time (from 1.6V to 1.8V)	t_R			10	ms	
Operating Temperature Range	T_{Opr}	-40		85	ºC	ET version
		-40		125	ºC	AT version

15.3 DC characteristics

<p align="center">Table 15- 3   DC characteristics (VDD=3.3V, T=25℃)</p>

Item	Sym.	Min	Typ.	Max	Unit	Condition
RX current	I$_{Rx}$		5.3		mA	Whole Chip
TX current	I$_{Tx}$		4.8		mA	whole chip @ 0dBm with DCDC
Deep sleep with 8kB SRAM retention	I$_{Deep1}$		1	3.1	uA	Without 32K RC
Deep sleep with 16kB SRAM retention			1.2	3.3	uA	Without 32K RC
Deep sleep with 32kB SRAM retention			1.4	3.5	uA	Without 32K RC
Deep sleep without SRAM retention	I$_{Deep2}$		0.4		uA	Without 32K RC

15.4 AC characteristics

<p align="center">Table 15- 4   AC Characteristics (VDD=3.3V, T=25℃)</p>

Item	Sym.	Min	Typ.	Max	Unit	Condition
Digital inputs/outputs						
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	
USB characteristics						
USB Output Signal Cross-over Voltage	V$_{Crs}$	1.3	-	2.0	V	
RF performance						
Item		**Min**	**Typ**	**Max**	**Unit**	
RF frequency range		2400		2483.5	MHz	Programmable in 1MHz step
Data rate	BLE/2.4G Proprietary 1Mbps, ±250kHz deviation BLE/2.4G Proprietary 2Mbps, ±500kHz deviation BLE 125kbps, ±250kHz deviation BLE 500kbps, ±250kHz deviation 802.15.4 250kbps, ±500kHz deviation 2.4G Proprietary 500kbps, ±125kHz deviation 2.4G Proprietary 250kbps, ±62.5kHz deviation					

Item	Sym.	Min	Typ.	Max	Unit	Condition
BLE 1Mbps RF_Rx performance (±250kHz deviation)*³						
Sensitivity	1Mbps		-96		dBm	
Frequency Offset Tolerance		-250		+300	kHz	
Co-channel rejection			11		dB	Wanted signal at -67dBm
In-band blocking rejection（Equal Modulation Interference）	+1/-1 MHz offset		-1/-3		dB	Wanted signal at -67dBm
	+2/-2 MHz offset		-37/-39		dB	
	>=3MHz offset		-42		dB	
Image rejection			-37		dB	Wanted signal at -67dBm
BLE 1Mbps RF_Tx performance						
Output power, maximum setting			10	12	dBm	
Output power, minimum setting			-45		dBm	
Programmable output power range			55		dB	
Modulation 20dB bandwidth			1.4		MHz	
IEEE802.15.4 (250kbps) RF_Rx performance (±500kHz deviation)*⁴						
Sensitivity	250kbps		-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
Error vector magnitude	EVM			2%		Max (10dBm) power output
IEEE802.15.4 (250kbps) RF_Tx performance						
Output power, maximum setting			10	12	dBm	
Output power, minimum setting			-45		dBm	
Programmable output			55		dB	

³ For actual sensitivity level of BLE 1Mbps mode, please refer to Bluetooth specification.
⁴ For actual sensitivity level of IEEE802.15.4 mode, please refer to 802.15.4 specification.

Item	Sym.	Min	Typ.	Max	Unit	Condition
power range						
Modulation 20dB bandwidth			2.7		MHz	
BLE 2Mbps RF_Rx performance (±500kHz deviation)*5						
Sensitivity	2Mbps		-93		dBm	
Frequency Offset Tolerance		-300		+200	kHz	
Co-channel rejection			10		dB	Wanted signal at -67dBm
In-band blocking rejection	+2/-2 MHz offset		-6/-6		dB	Wanted signal at -67dBm
	+4/-4 MHz offset		-39/-38		dB	
	>4MHz offset		-42		dB	
Image rejection			-25		dB	Wanted signal at -67dBm
BLE 2Mbps RF_Tx performance						
Output power, maximum setting			10	12	dBm	
Output power, minimum setting			-45		dBm	
Programmable output power range			55		dB	
Modulation 20dB bandwidth			2.5		MHz	
BLE 500kbps RF_Rx performance (±250kHz deviation)*6						
Sensitivity	500kbps		-99		dBm	
Frequency Offset Tolerance		-150		+50	kHz	
Co-channel rejection			1		dB	Wanted signal at -67dBm
In-band blocking rejection （Equal Modulation Interference）	+1/-1 MHz offset		-34/-36		dB	Wanted signal at -67dBm
	+2/-2 MHz offset		-42/-42		dB	
	>=3MHz offset		-42		dB	
Image rejection			-42		dB	Wanted signal at -67dBm

---

5 For actual sensitivity level of BLE 2Mbps mode, please refer to Bluetooth specification.
6 For actual sensitivity level of BLE 500kbps mode, please refer to Bluetooth specification.

Item	Sym.	Min	Typ.	Max	Unit	Condition
BLE 500kbps RF_Tx performance						
Output power, maximum setting			10	12	dBm	
Output power, minimum setting			-45		dBm	
Programmable output power range			55		dB	
Modulation 20dB bandwidth			1.4		MHz	
BLE 125kbps RF_Rx performance (±250kHz deviation)*[7]						
Sensitivity	125kbps		-101		dBm	
Frequency Offset Tolerance		-150		+50	kHz	
Co-channel rejection			3		dB	Wanted signal at -67dBm
In-band blocking rejection （Equal Modulation Interference）	+1/-1 MHz offset		-32/-34		dB	Wanted signal at -67dBm
	+2/-2 MHz offset		-42/-42		dB	
	>=3MHz offset		-42		dB	
Image rejection			-42		dB	Wanted signal at -67dBm
BLE 125kbps RF_Tx performance						
Output power, maximum setting			10	12	dBm	
Output power, minimum setting			-45		dBm	
Programmable output power range			55		dB	
Modulation 20dB bandwidth			1.4		MHz	
RSSI						
RSSI range		-100		10	dBm	
Resolution			1		dB	
24MHz crystal						
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

[7] For actual sensitivity level of BLE 125kbps mode, please refer to Bluetooth specification.

Item	Sym.	Min	Typ.	Max	Unit	Condition
Equivalent series resistance	ESR		50	100	ohm	
32.768kHz crystal						
Nominal frequency (parallel resonant)	f_{NOM}		32.768		kHz	
Frequency tolerance	f_{TOL}	-100		+100	ppm	
Load capacitance	C_L	6	9	12.5	pF	Programmable on chip load cap
Equivalent series resistance	ESR		50	80	kohm	
24MHz RC oscillator						
Nominal frequency	f_{NOM}		24		MHz	
Frequency tolerance	f_{TOL}			1	%	On chip calibration
32kHz RC oscillator						
Nominal frequency	f_{NOM}		32		kHz	
Frequency tolerance	f_{TOL}		0.03		%	On chip calibration
Calibration time			3		ms	
ADC						
Differential nonlinearity	DNL			1	LSB	11.5bit resolution mode
Integral nonlinearity	INL			2	LSB	11.5bit resolution mode
Signal-to-noise and distortion ratio	SINAD		70		dB	fin=1kHz, fS=16kHz
Signal-to-noise	SNR		73		dB	
Effective Number of Bits	ENOB		11.5		bits	
Sampling frequency	Fs			200	ksps	

15.5 SPI characteristics

Table 15- 5   SPI characteristics

(over process, voltage 1.9~3.6V, and T=-40~+85℃)

Item	Sym.	Min	Typ.	Max	Unit	Condition
CK frequency	F_{CK}			4	MHz	Slave
CK duty cycle clock			50		%	Master
DI setup time		30			ns	Slave
		90			ns	Master
DI hold time		10			ns	Slave
		90			ns	Master
CK low to DO valid time				30	ns	Slave
				120	ns	Master
CN setup time		60			ns	Master/Slave
CN high to DI tri-state*[8]					ns	Master

---

[8] Note: Master actively stops reading during transmission, and Slave releases its driver DO and turns to tri-state.

15.6 I2C characteristics

Table 15- 6 I2C characteristics

(over process, voltage 1.9~3.6V, and T=-40~+85℃)

Item	Sym.	Standard mode		Fast mode		Unit	Condition
		Min	Max	Min	Max		
SCL frequency	F_{SCL}		100		400	kHz	5kohm pull-up resistor for standard mode; 2kohm pull-up resistor for fast mode
Rise time of SDA and SCL signals	T_R		1000		300	ns	
Fall time of SDA and SCL signals	T_F		300		300	ns	
START condition hold time	$T_{HD;STA}$	4		0.6		us	
Data hold time	$T_{HD;DAT}$	0	3.45		0.9	us	
Data setup time	$T_{SU;DAT}$	250		100		ns	
STOP condition setup time	$T_{SU;STO}$	4		0.6		us	

15.7 Flash characteristics

Table 15- 7   Flash memory characteristics

(T= -40℃~85℃)

Item	Sym.	Min	Typ.	Max	Unit	Condition
Retention period		20			year	
Number of erase cycles		100k			cycle	
VDD for programming		1.65		2.0	V	Note this refers to the SoC supply
Sector size			4		kB	
Page programming time	TPP		1.6	6	ms	
Sector erase time	TSE		150	500	ms	
Block erase time (32kB/64kB)	TBE		0.5/0.8	2.0/3.0	s	
Program current	I_P			10	mA	
Erase current	I_E			10	mA	

16 Reference Design

16.1 Reference Schematic for TLSR8258F1KET48A
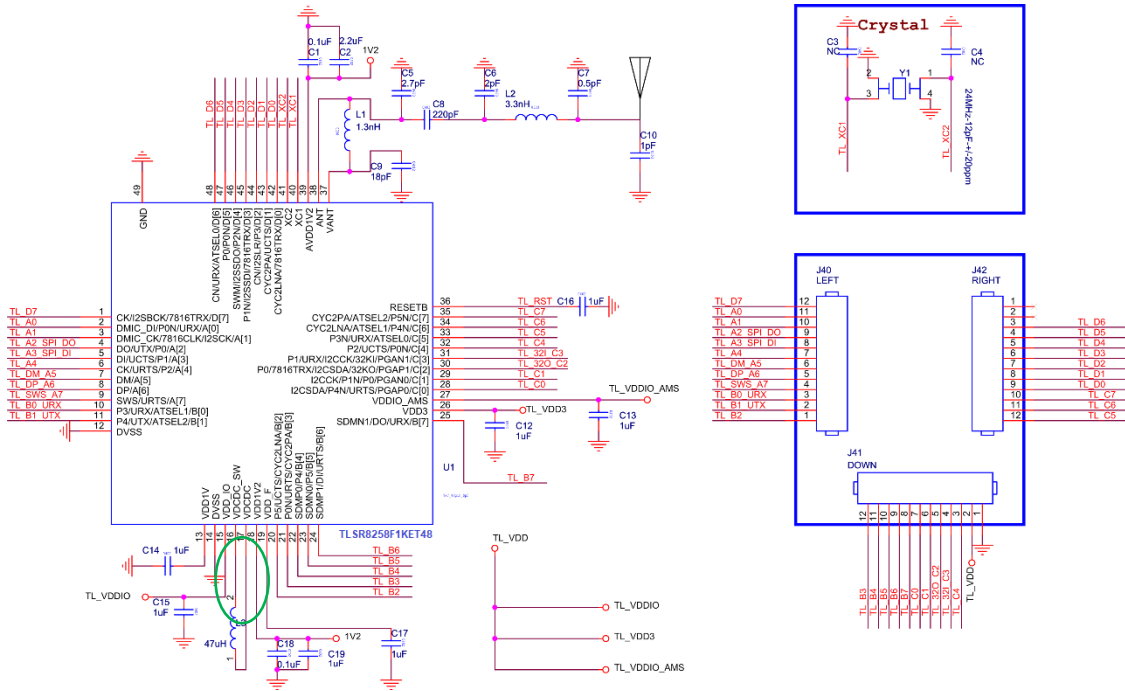

Figure 16- 1   Schematic

*Note:

In the reference design, the L3 adopts 47uH inductor, which can ensure sensitivity index given in this datasheet.

16.2 BOM (Bill of Material)

Table 16- 1   BOM table

Quantity	Reference	Value	Description	PCB Footprint
2	C1,C18	0.1uF	Capacitance,X5R,±10%	0402
1	C2	2.2uF	Capacitance,X5R,±10%	0402
2	C3,C4	N.C.	Not mounted	0402
1	C5	2.7pF	Capacitance,C0G,±0.1pF	0402
1	C6	2pF	Capacitance,C0G,±0.1pF	0402
1	C7	0.5pF	Capacitance,C0G,±0.05pF	0402
1	C8	220pF	Capacitance,X7R,±10%	0402
1	C9	18pF	Capacitance,C0G,±5%	0402
1	C10	1pF	Capacitance,C0G,±0.25pF	0402
6	C12,C13,C14, C15,C16,C17	1uF	Capacitance,X5R,±10%	0402
1	C19	1uF	Capacitance,X5R,±10%	0402
1	J40	LEFT	Pin headers	hdr254f-1x12x850
1	J41	DOWN	Pin headers	hdr254f-1x12x850
1	J42	RIGHT	Pin headers	hdr254f-1x12x850
1	L1	1.3nH	High frequency chip inductor,±0.3nH	0402
1	L2	3.3nH	High frequency chip inductor,±0.3nH	0402
1	L3	47uH	High frequency chip inductor,±20%	0805
1	U1	TLSR8258F 1KET48A	Multi-Standard Wireless on chip	QFN-48
1	Y1	24MHz	XTAL SMD 3225,24 MHz,Cl=12pF,total tol.±20ppm	XTAL_3225