ASR6601

Reference Manual

Version: 1.5.0

Date: 2022-08-11

About this document

This document provides detailed and complete information on the IoT LPWAN SoC-ASR6601 for application developers.

Target Audience

This document is mainly intended for the following engineers:

- hardware development engineer
- software engineer
- technical support engineer

Product numbering

Product models corresponding to this document:

Model	Flash	SRAM	Core	Package	Frequency
ASR6601SE	256 KB	64 KB	32-bit 48 MHz Arm China STAR-MC1	QFN68, 8*8 mm	150 ~ 960 MHz
ASR6601CB	128 KB	16 KB	32-bit 48 MHz Arm China STAR-MC1	QFN48, 6*6 mm	150 ~ 960 MHz

Copyright Notice

Trademark Notice

PDF-XChange Editor
Click to BUY NOW!
www.pdf-xchange.com

Disclaimer

ASR Technology Co., Ltd

Address: 9th Floor, Building 10, Zhangjiang Innovation Park, 399 Keyuan Road, Pudong New

District, Shanghai Postal Code: 201203

URL：http://www.asrmicro.com/

Document Revision History

Date	Version	Release Notes
2021.03	V1.0.0	Initial Release.
2021.05	V1.1.0	● Updated Chapter 6 Overview and Table 6-1. ● Updated parts of descriptions in Sections 16.3, 16.9, and 16.14.1. ● Corrected the description of LORAC_SR register in Section 12.4.13.
2021.07	V1.2.0	Updated CPU description.
2022.03	V1.3.0	● Added Chapter 21: DMA and Chapter 22: GPTIMER. ● Corrected several typos.
2022.05	V1.4.0	Modified RCO4M to RCO3.6M due to crystal frequency adjustment.
2022.08	V1.5.0	● Updated descriptions of some register bits in Sections 7.5.3, 8.3.3, 8.3.4, 8.3.7, 8.3.12, and 8.3.13. ● Updated Figure 8-1: Clock network diagram.

PDF-XChange Editor Click to BUY NOW! www.pdf-xchange.com

Table of contents

Click to BUY NOW! PDF-XChange Editor www.pdf-xchange.com

Click to BUY NOW! PDF-XChange Editor www.pdf-xchange.com

Click to BUY NOW! PDF-XChange Editor www.pdf-xchange.com

PDF-XChange Editor Click to BUY NOW! www.pdf-xchange.com

Click to BUY NOW! PDF-XChange Editor www.pdf-xchange.com

List of Tables

List of Figures

1. Overview

ASR6601 is a general LPWAN Wireless Communication SoC chip developed by ASR which supports LoRa modulation. The chip integrates Sub-1G RF transceiver, Arm China STAR-MC1 processor, embedded Flash memory and SRAM, as well as diverse analog modules. ASR6601 is designed for a wide variety of applications, such as smart meters, building automation, smart cities, agricultural sensors, safety and security sensors, supply chain and logistics, etc.

This manual provides detailed and complete information on the IoT LPWAN SoC-ASR6601 for application developers. Together with the API file in SDK, it helps developers solve various problems they may encounter during development. If any further support is needed, please contact us. We will keep this manual updated.

2. ASR6601 Introduction

ASR6601 SoC is a low-power wide area network wireless communication SoC chip that supports LoRa modulation. The ultra-low power transceiver integrated in the ASR6601 chip supports the full frequency band of 150 MHz ~ 960 MHz with the off-chip matching network. In addition to supporting LoRa modulation, it can also support FSK transceiver, MSK transceiver and BPSK transmitter. When powered by 3.3 V power supply, the maximum output power of 22 dBm can be transmitted using the high-power PA. ASR6601 SoC mainly has Run, LpRun, Sleep, LpSleep, Stop0, Stop1, Stop2, Stop3, Standby working modes. Each mode supports different functions, working modules and power consumption. End users can choose the corresponding working mode according to their application scenarios. The two most commonly used low-power modes are Standby mode and Stop3 mode. When powered by 3.3 V, the Standby mode consumes as little as 0.9 uA; the Stop3 mode consumes as little as 1.3 uA (ASR6601CB) and 1.6 uA (ASR6601SE).

ASR6601 SoC uses a 32-bit ARM STAR core with a maximum main frequency of 48 MHz, supports SWD debug interface, supports SysTick, MPU, FPU functions, and supports 37 IRQs with 8 interrupt priorities.

ASR6601 supports UART, I2C, I2S, LPUART, SSP, QSPI and other interfaces. With the peripherals of different types of corresponding interfaces, it can realize rich functions to meet customer needs. In addition to supporting rich number functions, ASR6601 also integrates rich analog functions, including ADC, DAC, OPA and LCD driver.

ASR6601 implements AES encryption through hardware, greatly simplifying the efficiency of encryption and decryption. It also supports national encryption SM2/3/4.

3. Modules and functions

3.1 ASR6601 SoC Diagram

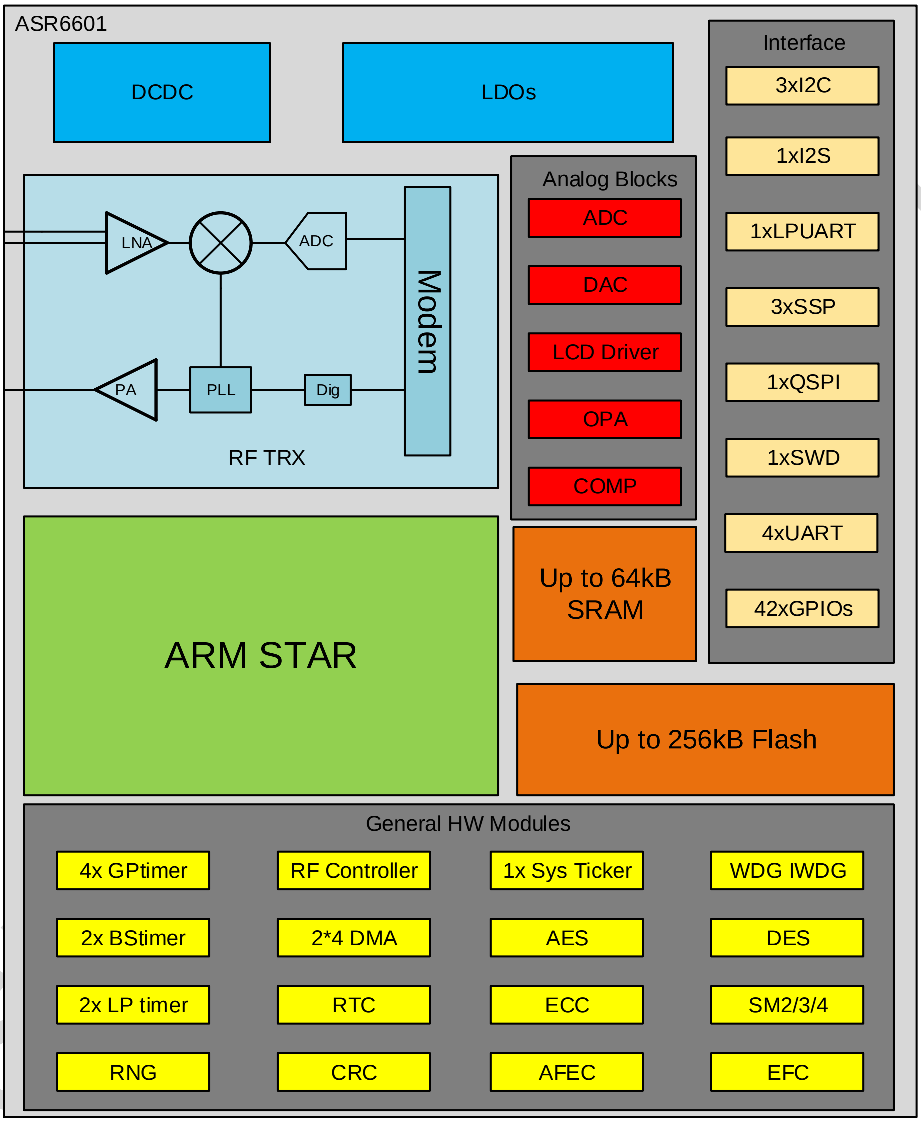

Figure 3-1 ASR6601 SoC Diagram

3.2 ASR6601 function

Table 3-1 ASR6601 functions

Module Name	Functions
RCC	Clock and reset control
SYSCFG	System function registers
PWR	1. Chip low power mode control 2. Interrupt signal generation
SEC	1. Security IP Enable 2. Filtering Security IP alarm signal filtering 3. Alarm signal processing
CPU	1. SWD debug interface 2. Systick function 3. MPU function 4. FPU function 5. 37 IRQs, 8 interrupt priorities
MPU	Access Security control
EFC	1. Power-on chip mode determination 2. Flash info area data loaded at power-on 3. Basic flash operations, including read, program, page erase, mass erase 4. Flash operation key timing control 5. Flash instruction prefetch function 6. Flash program operation supports single and continuous modes 7. Flash info area option bytes operation 8. Interrupt signals generating
I2S	1. Philips I2S serial protocol 2. Master and Slave modes 3. 1 RX channel, 1 TX channel, full duplex 4. Receive FIFO depth is 4 5. Transmit FIFO depth is 4 6. Receiver supports 12, 16, 20, 24, 32-bit resolution 7. Transmitter supports 12, 16, 20, 24, 32-bit resolution 8. Programmable DMA registers 9. Programmable FIFO Threshold 10. 1 interrupt signal generation
UART	1. IrDA, support 3/16 and low-power (1.41-2.23us) modes 2. FIFO, 16x8 bits for transmission, 16x10 bits for reception 3. Transmit and receibing buffers 4. Baud rate generation, using 16 times oversampling, supports 16-bit integer division and 6-bit fractional division, and supports up to interface clok frequency/16

Module Name	Functions
	5. UART data format configuration 6. DMA transfer 7. False start detection 8. Line break sending and detection 9. Hardware flow control CTS and RTS 10. Interrupt signal generation
LPUART	1. Low power wake-up 2. Baud rate generation, does not support oversampling, supports 4-bit fractional frequency division and 12-bit integer frequency division, the minimum integer frequency division is 3 3. UART data format configuration, including 1-2 bits Stop, 0-1 bits parity (odd, even, mark, space, none), 5-8 data bits 4. Hardware flow control CTS and RTS 5. DMA transfer 6. Interrupt signal generation
SSP	1. Master and Slave modes 2. Programmable baud rate and prescaler, Master supports up to 1/2 interface clock frequency, Slave supports up to 1/12 interface clock frequency 3. 8*16 Bit receiving and transmitting FIFO 4. Configurable data length, 4-16 Bit 5. DMA transfer 6. Motorola, Microwire (NS), TI formats 7. Motorola supports 4 polarity phase bit combinations 8. Interrupt signal generation
I2C	1. Master and slave modes, support multi-master arbitration 2. Multi-host arbitration 3. Standard Mode and Fast Mode 4. 7-bit address mode 5. Clock Stretching 6. Interrupt signal generation 7. DMA transfer
AFEC	1. IP status register 2. Simulate IP control register 3. Some registers support Safety lockControl 4. Interrupt signal generation
LORAC	1. LORA IP control register 2. LORA status register 3. LORA IP SPI interface source, supports ssp master control and reg control 4. DMA transfer 5. Interrupt signal generation

PDF-XChange Editor Click to BUY NOW! www.pdf-xchange.com

Module Name	Functions
RTC	1. Calendar counting function in BCD format 2. Ppm adjustment, adjustment step size 0.5ppm, +/-1024 ppm adjustment 3. Low power wake-up 4. Tamper/wakeup IO detection function 5. Cycle counting function, 32-bit counter 6. Alarm clock function, supports two alarm clocks, supports mask selection 7. Clear retention sram on Tamper alarm 8. Some registers support Safety lockControl 9. Internal signal IO output, including alarm0 matching pulse, alarm1 matching pulse, cycle count configuration pulse, seconds signal output 10. Calendar count value reading 11. Sub-second count value reading 12. Cycle count value reading 13. Interrupt signal generation
IWDG	1. Watchdog counting function, down-counting, clock prescaler (4-256) 2. Watchdog exception status occurs when the count reaches 0 (feeding the dog too late) or when the count value when feeding the dog is greater than the counting window value (feeding the dog too early) 3. Interrupt signal generation 4. Dog feeding window configuration 5. Count value reading 6. Low power wake-up
QSPI	1. Master interface only 2. 1-wire, 2-wire, 4-wire modes 3. 3 working modes, including indirect access, status query and Memory-mapping 4. Baud rate division, up to interface clock frequency/2 5. Interrupt signals generation
CRC	1. Configurable polynomial bit width: 7, 8, 16, 32 bits 2. Different hsize accesses, the lower byte is calculated first and can be edited 3. Programmable crc initial value 4. Input data reverse, supports byte, halfword and word 5. Output data reverse, supports word
DMA	1. 1 master interface AHB bus 2. AHB interface only supports little-endian structure 3. Interrupt signal generation 4. Transfer mode supports M2M, P2M, M2P, P2P 5. Software triggering handshake signal 6. 4 sets of hardware handshake signals, including burst and single requests 7. Hardware handshake signal sources, each group supports 64 source selections 9. Channel 0 configutation: ： (1) 8 bytes deep FIFO

PDF-XChange Editor Click to BUY NOW! www.pdf-xchange.com

Module Name	Functions
	(2) Maximum burst length is 8 (3) Maximum transfer length is 2047 (4) Supports dmac flow control only (5) Source address data bit width configurable (6) Destination address data bit width configurable (7) Address increment, decrement, and unchanged (8) Block transfer, including continuous address, automatic loading and linked list (9) Scatter and gather 10. Channel 1-3 configuration: (1) 8 bytes deep FIFO (2) Maximum burst length is 8 (3) Maximum transfer length is 2047 (4) Supports dmac flow control only (5) Source address data bit width configurable (6) Destination address data bit width configurable (7) Address increment, decrement, and unchanged (8) Block transfer, including continuous addresses and automatic loading, but does not support linked lists (9) Scatter and gather not supported
GPIO	1. IO output configuration, push-pull, open drain, output high impedance 2. IO input configuration, floating, input pull-up, input pull-down, analog input 3 IO pull-up configuration, pull-down configuration, drive capability control 4. Interrupt signals generation, including rising edge interrupt, falling edge interrupt, and both edges interrupt 5 Wake-up signals generation, including high level and low level
SAE	1. AES128/192/256 2. DES and 3DES 3. SM2, SM3, SM4 (ASR6601SE) 4. RSA1024/2048 5. ECC224/256/384/512 6. SHA1, SHA-224, SHA256, SHA384, SHA512 7. Random number generation
BSTIMER	1. 32bits counter, supports auto-reload, up-counting, down-counting, center-aligned counting 2. 16-bit counter clock prescaler 3. Supports DMA requests 4. Interrupt signals generation
GPTIMER	1. 32 bits counter, supports auto-reload, up-counting, down-counting, center-aligned counting 2. 16-bit counter clock prescaler 3. gptimer0 and gptimer1 supports 4 channels, gptimer2 and gptimer3 supports 2 channels, each channel can support input capture, output comparison, PWM generation, single pulse output 4. Quadrature decoding 5. Interrupt signals generation 6. Supports DMA requests

Module Name	Functions
LPTIMER	1. Supports selecting internal clock and IO clock as counting clock 2. 16 bits counter, up-counting, auto-reload 3. Counter clock prescaler 4. Quadrature decoding support 5. Input capture, output comparison, PWM generation, single pulse output 6. Interrupt signals generation 7. Supports DMA requests
ADC	1. 12 bits sampling resolution 2. Configurable sampling rate up to 1 MHz 3. Single-ended and differential inputs 4. Only right data alignment 5. 8 external channels 6. 7 internal channels, including DAC output, internal Vref, VDD/3 (battery power), Vts (internal temperature sensor), OPA output (3) 7. Trigger mode, supports software trigger and hardware trigger 8. Sequential, continuous, single, and non-continuous sampling modes 9. Analog watchdog function, 3 channels in total, configurable Channel selection and thresholds 10. Supports DMA requests 11. Interrupt signals generation
DAC	1. 10 bits output resolution 2. Configurable output speed up to 1 MHz 3. Right data alignment only 4. Special waveform output, supports triangle wave 5. Software trigger and hardware trigger 6. Supports DMA requests 7. Interrupt signals generation
LCDCTRL	1. Frame rate control 2. Bias control, supports static, 1/2, 1/3, 1/4 3. Duty Control, supports static (1comx27seg), 1/2 (2comx26seg), 1/3 (3comx25seg), 1/4 (4comx24seg), 1/8 (8comx20seg) 4. Dead frame control, supports dead frame of 0-7 shots, used to adjust contrast 5. Blink control, supports the blinking function of 1, 2, 3, 4, 8 or all pixels, configurable blinking frequency 6. High and low current selection control, including state machine dynamic control and register static control. During state machine dynamic control, high current can be configured to maintain the number of beats. 7. Interrupt signal generation

4. Power Management Unit

4.1 Power Supply pins

ASR6601 has several separated power supply pins. Using separated power supply pins, the interference from digital parts of SoC to RF blocks is reduced.

ASRR6601 Power Grid is shown in Figure 4-1:

- **VDD_IN**: Power supply for the PA in the RF transmitter.
- **VBAT_RF**: Power supply for the RF TRX, excluding the PA.
- **VDCC_RF**: Low-power supply for RF TRX, must be connected to VREG pin
- **VBAT_ESD0**: Digital IO power supply.
- **VBAT_ESD1**: Digital IO power supply.
- **VBAT_ESD2**: Digital IO power supply.
- **VBAT_ESD3**: Digital IO power supply.
- **VBAT_DCC**: Dedicated power supply for DCDC in analog circuit.
- **VBAT_ESD_RTC**: Power supply for IOs in RTC domain.
- **VBAT_RTC**: Power supply for analog blocks in RTC domain.
- **VBAT_ANA**: Power supply for analog blocks.

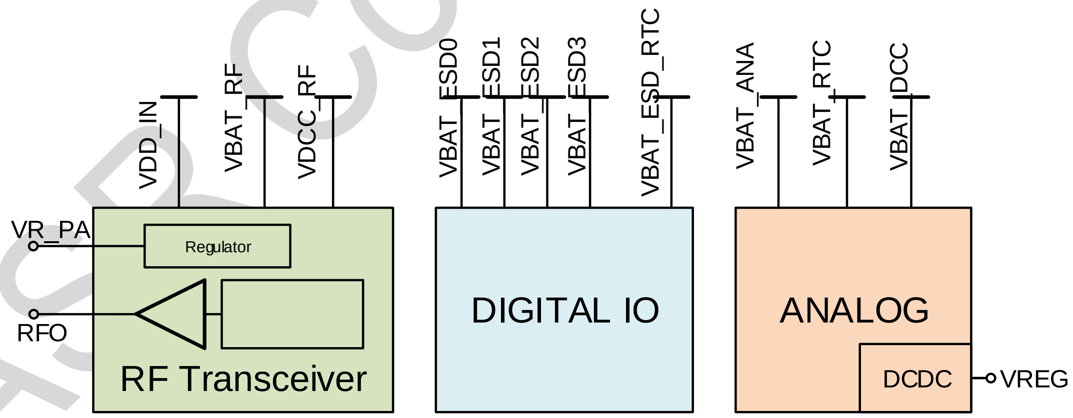

Figure 4-1 ASR6601 Power Grid

4.2 Power Supply Architecture

Internal power domains of the chip are mainly divided into **main** domain, **aon** domain and **aonr** domain. Please note that the power domains are divided according to functions, as shown in Figure 4-2.

1. **main** domain contains most of the digital logic circuits of the SoC chip. In the frequently used low-power modes (Standby and Stop3), the power supply of main domain will be turned off.

2. **aon** (**A**lways **On** domain) means that the power supply for this domain is always available, even in low-power mode. Most blocks in aon domain keep running in all power modes.

3. **aonr** (**A**lways **On** and **R**etention) domain contains the modules that need to keep running in Stop3 mode. These modules will be powered off in Standby mode. When aonr domain modules remain in the current state without power off, the system can quickly recover and continue to execute.

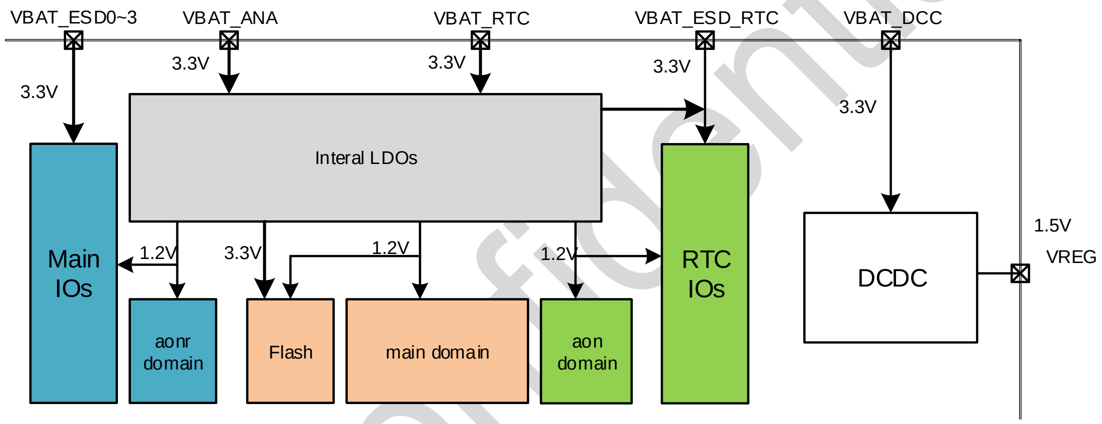

Figure 4-2 ASR6601 Power Supply Architecture

5. Access Control

5.1 Simple Configuration

This section provides customers with commonly used simple configurations to meet their basic security needs.

5.1.1 Recoverable Security Configuration

- **Enable Security**

Configure FlashSecStart to 0 and FlashSecEnd to 0x3F in OPTION1 tab, and set the entire Flash_main area as a secure area.
Consequently, the code in SWD (Serial Wire Debug) and non-secure area cannot read and write data into Flash_main to guarantee security. Please note that code in non-secure SRAM area or non secure DMA will not be able to access Flash_main.

- **Disable Security**

Configure FlashSecStart to 0x3F and FlashSecEnd to 0 in OPTION1 tab, and set the entire Flash_main area as a non-secure area.

The above configurations will erase the entire Flash_main area, and then the program can be re-downloaded.

5.1.2 Unrecoverable Security Configuration

Configure the DebugLevel to 2 in Option0 tab. This operation is irreversible, and the code must be correct and mature.

5.2 Access Control

Based on debug level rules, boot startup mode, exe-only access rules, write-protected access rules, info area access rules, and secure area access rules, access rights from the four main interfaces (cpucode, cpusw, dmac0, and dmac1) are controlled.

5.2.1 Debug Level Rules

Debug level mainly affects the access rights of cpu_code (boot from SRAM and boot from bootloader), cpu_sw, dmac0 and dmac1 to sensitive areas. Sensitive areas include flash_main, otp area of flash_info and retention SRAM.

For details see "*ASR6601 Access Control Description*".

5.2.2 Secure and Non-Secure Operation

- **Secure Operation**

The operations initiated by the code in the secure area include:

- Operations initiated by DMAC0 configured as a secure area
- Operations initiated by flash_main configured as a secure area (CPU_Code)
- Operations initiated by system_sram configured as a secure area (CPU_Code)

- **Non-secure Operation**

The operations initiated by the code in the non-secure area include:

- Operations initiated by DMAC0 configured as a non-secure area
- Operations initiated by DMAC1
- Operations initiated by Debug Port (CPU_SW)
- Operations initiated by Bootloader (CPU_Code)
- Operations initiated by flash_main configured as a non-secure area (CPU_Code)
- Operations initiated by system_sram configured as a non-secure area (CPU_Code)

For details see "*ASR6601 Access Control Description*".

6. Operation Modes

ASR6601 LPWAN SoC supports Run, LpRun, Sleep, LpSleep, Stop0, Stop1, Stop2, Stop3 and Standby modes. Each mode supports different functions with different working modules and power consumption. The user can choose the appropriate operation mode according to specific application scenarios. All modes are described detailedly in the contents below.

In addition, please note the following points:

1. When entering low power mode, peripherals marked as O (excluding GPIO) are turned off by default. Functions required in low power mode need to be turned on before entering low power mode.

2. When entering low power mode, in order to achieve the corresponding design power consumption value, please pay attention to the following points:

(1) Unused GPIOs need to be configured in ANALOG mode (high impedance).

(2) If the GPIO used is in input mode, you need to configure the pull-up and pull-down.

(3) If the peripheral is in output mode, the pull-up and pull-down of the connected peripheral must be configured according to the output level.

3. RCO48M/2 is used to enter and exit low power mode. If a non-RCO48M/2 clock is used before entering low power mode, it is necessary to switch to RCO48M/2. After exiting low power mode, you can switch to the previously used clock.

4. If the analog functions supported by RCO32K/XO32K and other low power modes are needed in low power mode, they need to be enabled before entering low power mode.

5. The clocks other than RCO48M/RCO32K/XO32K and other analog function modules need to be turned off by software before entering low power consumption.

Table 6-1 Status of different modules in each operating mode

	Run	LpRun	Sleep	LpSleep	Stop0	Stop1	Stop2	Stop3	Standby	Stop0-2 Wakeup	Stop3 Wakeup	Standby Wakeup
cpu	Y	Y	NA	NA	NA	NA	NA	NA	NA			
efc	Y	Y	O	O	NA	NA	NA	NA	NA			
sysramc	Y	Y	O	O	NA	NA	NA	NA	NA			
retramc	Y	Y	O	O	NA	NA	NA	NA	NA			
i2s	O	O	O	O	NA	NA	NA	NA	NA			
uart0	O	O	O	O	NA	NA	NA	NA	NA			
uart1	O	O	O	O	NA	NA	NA	NA	NA			
uart2	O	O	O	O	NA	NA	NA	NA	NA			
uart3	O	O	O	O	NA	NA	NA	NA	NA			
ssp0	O	O	O	O	NA	NA	NA	NA	NA			
ssp1	O	O	O	O	NA	NA	NA	NA	NA			
ssp2	O	O	O	O	NA	NA	NA	NA	NA			
qspi	O	O	O	O	NA	NA	NA	NA	NA			
i2c0	O	O	O	O	NA	NA	NA	NA	NA			

	Run	LpRun	Sleep	LpSleep	Stop0	Stop1	Stop2	Stop3	Standby	Stop0~2 Wakeup	Stop3 Wakeup	Standby Wakeup
i2c1	O	O	O	O	NA	NA	NA	NA	NA			
i2c2	O	O	O	O	NA	NA	NA	NA	NA			
adcctrl	O	O	O	O	NA	NA	NA	NA	NA			
dacctrl	O	O	O	O	NA	NA	NA	NA	NA			
gptim0	O	O	O	O	NA	NA	NA	NA	NA			
gptim1	O	O	O	O	NA	NA	NA	NA	NA			
gptim2	O	O	O	O	NA	NA	NA	NA	NA			
gptim3	O	O	O	O	NA	NA	NA	NA	NA			
basictim0	O	O	O	O	NA	NA	NA	NA	NA			
basictim1	O	O	O	O	NA	NA	NA	NA	NA			
wwdg	O	O	O	O	NA	NA	NA	NA	NA			
crc	O	O	O	O	NA	NA	NA	NA	NA			
sec	O	O	O	O	NA	NA	NA	NA	NA			
sac	O	O	O	O	NA	NA	NA	NA	NA			
mpu	O	O	O	O	NA	NA	NA	NA	NA			
dmac0	O	O	O	O	NA	NA	NA	NA	NA			
dmac1	O	O	O	O	NA	NA	NA	NA	NA			
syscfg	O	O	O	O	NA	NA	NA	NA	NA			
afec	O	O	O	O	NA	NA	NA	NA	NA			
lorac	O	O	O	O	NA	NA	NA	NA	NA			
gpio	O	O	O	O	NA	NA	NA	GPIO0~55: Y3 GPIO56~63:Y4	GPIO0~55: NA3 GPIO56~63: Y4	Y	Y	
rcc	Y	Y	Y	Y	Y	Y	Y	Y	Y			
pwr	Y	Y	Y	Y	Y	Y	Y	Y	Y			
lpuart	O	O	O	O	O	O	O	O (RX only)	O (RX only)	Y	Y	Y
lcdctrl	O	O	O	O	O	O	O	O	O			
lptim0	O	O	O	O	O	O	O	O	O	Y	Y	Y
lptim1	O	O	O	O	O	O	O	O	O	Y	Y	Y
iwdg	O	O	O	O	O	O	O	O	O	Y1	Y	Y
rtc	O	O	O	O	O	O	O	O	O	Y	Y	Y
ADC	O	O	O	O	NA	NA	NA	NA	NA			
RCO48M	O	O	O	O	NA	NA	NA	NA	NA			
XO24M	O	O	O	O	NA	NA	NA	NA	NA			
PLL48M	O	O	O	O	NA	NA	NA	NA	NA			
RNG	O	O	O	O	NA	NA	NA	NA	NA			
DAC	O	O	O	O	O3	O3	O3	NA	NA			
OPA	O	O	O	O	O	O	O	NA	NA			
COMP	O	O	O	O	O	O	O	O	O	Y	Y	Y
VD	O	O	O	O	O	O	O	O	O	Y	Y	Y
RCO3.6M	O	O	O	O	O	O	O	O	O			

	Run	LpRun	Sleep	LpSleep	Stop0	Stop1	Stop2	Stop3	Standby	Stop0-2 Wakeup	Stop3 Wakeup	Standby Wakeup
RCO32K	O	O	O	O	O	O	O	O	O			
XO32K	O	O	O	O	O	O	O	O	O			
LCD	O	O	O	O	O	O	O	O	O			
BOR	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y2	Y2	Y2
FLASH	Y	Y	Y	Y	SLM	SLM	SLM	PDM	PDM			
SRAM	Y	Y	Y	Y	NA	NA	NA	NA1	NA2			
IO	Y	Y	Y	Y	Y	Y	Y	Y	Y			
RF	O	O	O	O	O	O	O	O	O	Y	Y	Y

Notes and symbol annotations for the table above：

● **Stop0-2:** all GPIOs can be configured to wake up the CPU; all GPIOs retain the previous state in Stop0-2 mode.

● **Stop3:** 56 GPIOs in the main domain can be configured to wake up the CPU; all GPIOs retain the previous state in Stop3 mode.

● **Standby:** 8 GPIOs in the AON domain retain the previous state in Standby mode; 56 GPIOs in the main domain are used as analog functions (such as LCD, COMP) and cannot be used to wake up the CPU. The LPUART only supports RX in Standby/Stop3 mode.

● **Y**: Normal work
● **O**: Optional, configured by software
● **O3**: Data update is not supported, but the output retains current voltage level
● **Y1**: Generate system reset to wake up the system indirectly
● **Y2**: Generate BOR reset to wake up the system indirectly
● **Y3**: Retain the state before entering low-power mode, and can be used to wake up the CPU.
● **Y4**: MUX Function1 of GPIO56~63 is not available and the other alternate functions is available
● **NA1**: Retention and algorithm contents are kept. System content can be configured to be kept or not.
● **NA2**: Retention content is kept
● **NA3**: Analog Output Only

6.1 Run

6.1.1 Enter and Exit

Run mode is the default operation mode after power-on or system reset.

ASR6601 can enter Sleep, LpRun, Stop0, Stop1, Stop2, Stop3 or Standby mode from Run mode.

ASR6601 can return to Run mode from Sleep, LpRun, Stop0, Stop1, Stop2, Stop3 or Standby mode.

For detailed mode switching conditions, please refer to the descriptions of other operation modes.

6.1.2 Wakeup Source

N/A

6.2 LpRun

6.2.1 Enter and Exit

You can enter LpRun from Run. The entry conditions are as follows: the software switches the working state of LDO. Before switching LDO, all high-speed clocks must be turned off and the CPU runs at 32K clock.

LpRun config register is used to switch LDO working state:

1: Set bits[3:3] of the register (address 0x05) to 1, and the other bits remain unchanged.

2: Set bits[21:20] of the register (address 0x06) to 1, and the other bits remain unchanged.

LpRun can return to ***Run***, and the exit conditions are as follows: the software switches the working state of LDO. The high-speed clock can be turned on only after the switch is completed.

Return to ***Run*** mode from ***LpRun*** mode in the following way:

1: Clear bits[21:20] of the register (address 0x06) to 0, and the other bits remain unchanged.

2: Clear bits[3:3] of the register (address 0x05) to 0, and the other bits remain unchanged.

6.2.2 Wakeup Source

N/A

6.3 Sleep

6.3.1 Enter and Exit

Sleep can be entered from Run. The entry conditions are: CPU executes wfi/wfe instruction (SLEEPDEEP=0), or isr returns (SLEEPONEXIT=1 and SLEEPDEEP=0).

It can return to Run from Sleep, and the exit condition is: if wfi is entered, it supports interrupt wake-up, and if wfe is entered, it supports event wake-up.

Note: *Because there is no dedicated event wake-up signal, SVONPEND=1 is used and the corresponding NVIC is turned off to achieve it. At this time, the interrupt signal is used for event wake-up.*

6.3.2 Wakeup Source

Interrupt signals

6.4 LpSleep

6.4.1 Enter and Exit

LpSleep can be entered from LpRun, and the entry conditions are: CPU executes wfi/wfe instruction (SLEEPDEEP=0), or isr returns (SLEEPONEXIT=1 and SLEEPDEEP=0).

LpSleep can return to LpRun, and the exit conditions are: if wfi enters, interrupt wake-up is supported; if wfe enters, event wake-up is supported.

Note: *There is no dedicated event wake-up signal. It is implemented by setting SVONPEND=1 and turning off the corresponding NVIC. In this case, the interrupt signal is used for event wake-up.*

6.4.2 Wakeup Source

Interrupt signals

6.5 Stop0

6.5.1 Enter and Exit

Stop0 can be entered from Run. The entry conditions are: configure lp_mode to 2'b00, CPU executes wfi/wfe instruction (SLEEPDEEP=1), or isr returns (SLEEPONEXIT=1 and SLEEPDEEP=1).

Stop0 can be used to return to Run. The exit conditions are: if wfi is entered, interrupt wake-up is supported; if wfe is entered, event wake-up is supported.

The pwr module summarizes the wakeup source status and outputs the pwr_wakeup_int interrupt signal and the pwr_wakeup_event event signal to the CPU for wakeup.

6.5.2 Wakeup Source

- GPIO00-GPIO63 can be used for wake-up. Four IOs form a group. Each IO in a group has a wake-up enable configuration. A group can generate a wake-up signal. Each IO in a group supports the selection of high-level wake-up or low-level wake-up. The wake-up sources other than GPIO are listed below.

- PVM Alarm
- VD Alarm
- TD Alarm
- LD Alarm
- Comparator
- LPTIM0/1
- FD_32K Alarm
- Wakeup/Tamper IO
- RTC Alarm
- RTC CYC Timer
- LPUART RX Status
- LORA BUSY
- LORA IRQ

6.6 Stop1

6.6.1 Enter and Exit

You can enter Stop1 from Run. The entry conditions are: configure lp_mode to 2'b01, CPU executes wfi/wfe instruction (SLEEPDEEP=1), or isr returns (SLEEPONEXIT=1 and SLEEPDEEP=1);

Stop1 can be used to return to Run. The exit conditions are: if wfi is entered, interrupt wake-up is supported; if wfe is entered, event wake-up is supported.

The pwr module summarizes the wakeup source status and outputs the pwr_wakeup_int interrupt signal and the pwr_wakeup_event event signal to the CPU for wakeup.

6.6.2 Wakeup Source

- GPIO00-GPIO63 can be used for wake-up. Four IOs form a group. Each IO in a group has a wake-up enable configuration. A group can generate a wake-up signal. Each IO in a group supports the selection of high-level wake-up or low-level wake-up. The wake-up sources other than GPIO are listed below.

- PVM Alarm
- VD Alarm
- TD Alarm
- LD Alarm
- Comparator
- LPTIM0/1
- FD_32K Alarm
- Wakeup/Tamper IO
- RTC Alarm
- RTC CYC Timer
- LPUART RX Status
- LORA BUSY
- LORA IRQ

6.7 Stop2

6.7.1 Enter and Exit

You can enter Stop2 from Run. The entry conditions are: configure lp_mode to 2’b10, CPU executes wfi/wfe instruction (SLEEPDEEP=1), or isr returns (SLEEPONEXIT=1 and SLEEPDEEP=1);

Stop2 can be used to return to Run. The exit conditions are: if wfi is entered, interrupt wake-up is supported; if wfe is entered, event wake-up is supported;

The pwr module summarizes the wakeup source status and outputs the pwr_wakeup_int interrupt signal and the pwr_wakeup_event event signal to the CPU for wakeup.

6.7.2 Wakeup Source

- GPIO00-GPIO63 can be used for wake-up. Four IOs form a group. Each IO in a group has a wake-up enable configuration. A group can generate a wake-up signal. Each IO in a group supports the selection of high-level wake-up or low-level wake-up. The wake-up sources other than GPIO are listed below.

- PVM Alarm
- VD Alarm
- TD Alarm
- LD Alarm
- Comparator
- LPTIM0/1
- FD_32K Alarm
- Wakeup/Tamper IO
- RTC Alarm
- RTC CYC Timer
- LPUART RX Status
- LORA BUSY
- LORA IRQ

6.8 Stop3

6.8.1 Enter and Exit

You can enter Stop3 from Run. The entry conditions are: configure lp_mode to 2’b11, lp_mode_ext to 1’b1, CPU executes wfi/wfe instruction (SLEEPDEEP=1), or isr returns (SLEEPONEXIT=1 and SLEEPDEEP=1);

You can return to Run from Stop3 when a Stop3 wake-up event occurs.

6.8.2 Wakeup Source

- GPIO00-GPIO55 can all be used to wake up the CPU, 4 IOs make up a group, and each group can select any of the 4 IOs for wake-up. A group generates a wake-up signal, and any of the IOs can wake up the CPU at high or low level. The wake-up sources other than GPIOs are listed below.

- PVM Alarm
- VD Alarm
- Comparator
- LPTIM0/1
- FD_32K Alarm
- Wakeup/Tamper IO
- RTC Alarm
- RTC CYC Timer
- LPUART RX Status
- LORA BUSY
- LORA IRQ
- IWDG Timeout

6.9 Standby

6.9.1 Enter and Exit

The Standby state can be entered from Run state. The entry conditions are: configure lp_mode to 2’b11, lp_mode_ext to 1’b0, CPU executes wfi/wfe instruction (SLEEPDEEP=1), or isr returns (SLEEPONEXIT=1 and SLEEPDEEP=1);

The system can return to Run from Standby. The exit condition is: a Standby wake-up event occurs.

Note:
1. When the power supply is switched between DCDC and VBAT, the CPU will return to Run mode immediately after entering Standby mode without any wake-up event.
2. When dbg_standby=1, the switch between DCDC and VBAT is disabled.

6.9.2 Wakeup Source

- PVM 报警
- VD 报警
- 比较器
- LPTIM0/1
- FD_32K 报警
- Wakeup/Tamper IO
- RTC Alarm
- RTC CYC Timer
- LPUART 接收状态
- LORA BUSY
- LORA IRQ
- IWDG 超时

7. System configuration

7.1 System Architecture

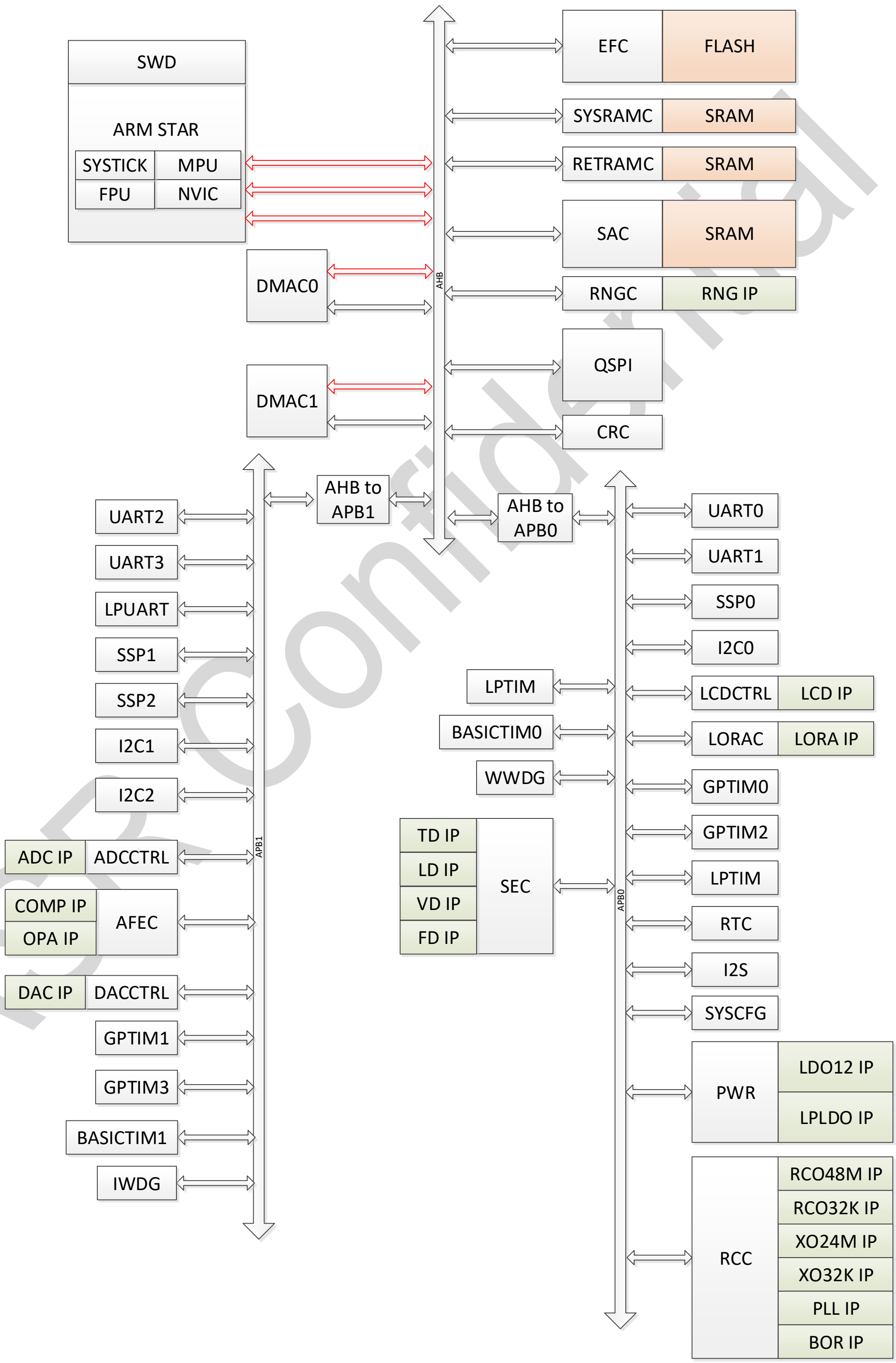

Figure 7-1 System Architecture Diagram

7.1.1 Arm China STAR-MC1 Processor

Arm China STAR-MC1 Processor consists of three master buses, including icode AHB bus, dcode AHB bus and system AHB bus, which are used for program access, data access and register access.

7.1.2 DMAC0

DMAC0 has a master bus, which can assist the CPU to transfer data.

7.1.3 DMAC1

DMAC1 has a master bus, which can assist the CPU to transfer data.

7.1.4 Master

The addresses accessible by each master bus is shown in the table below.

(1) *Only accessible when boot from Bootloader.*

Table 7-1 Master bus access range

起始地址	结束地址	功能描述	可执行	icode 访问	dcode 访问	system 访问	dmac0 访问	dmac1 访问
0xE0100000	0xFFFFFFFFF	Reserved						
0xE0000000	0xE00FFFFF	ARM STAR peripherals						
0xA0000000	0xDFFFFFFF	Reserved						
0x70000000	0x9FFFFFFF	Reserved						
0x60000000	0x6FFFFFFF	Qspi Flash Bank	Y			Y	Y	Y
0x50000000	0x5FFFFFFF	Reserved						
0x40030000	0x4FFFFFFF	AHB1 SFR				Y	Y	Y
0x40020000	0x4002FFFF	AHB0 SFR				Y	Y	Y
0x40010000	0x4001FFFF	APB1 SFR				Y	Y	Y
0x40000000	0x4000FFFF	APB0 SFR				Y	Y	Y
0x30000400	0x3FFFFFFF	Reserved						
0x30000000	0x300003FF	Retention SRAM				Y	Y	Y
0x20010000	0x2FFFFFFF	Reserved						
0x20000000	0x2000FFFF	System SRAM	Y			Y	Y	Y
0x18010000	0x1FFFFFFF	Reserved						
0x18000000	0x1800FFFF	System SRAM	Y	Y	Y			
0x10004000	0x17FFFFFF	Reserved						
0x10003000	0x10003FFF	Option Bytes			Y			
0x10002000	0x10002FFF	Factory Bytes			Y			
0x10001C00	0x10001FFF	OTP			Y			
0x10000000	0x10001BFF	BootLoader		Y(1)	Y(1)			
0x08040000	0x0FFFFFFF	Reserved						
0x08000000	0x0803FFFF	Flash Main	Y	Y	Y		Y	Y
0x00040000	0x07FFFFFF	Reserved						
0x00000000	0x0003FFFF	Flash Main/BootLoader/ System SRAM(1)	Y	Y	Y			

7.2 Memory Mapping

The Memory Mapping table is shown below. The bytes are coded in memory in Little Endian format, i.e. the least significant byte is in the lowest address.

Table 7-2 Memory map

Category	Start Address	End Address	Description	Size
SYSTEM	0xE0100000	0xFFFFFFFFF	Reserved	
PPB	0xE0000000	0xE00FFFFF	ARM STAR peripherals	
EXT PERIPHERAL	0xA0000000	0xDFFFFFFF	Reserved	
EXT SRAM	0x70000000	0x9FFFFFFF	Reserved	
	0x60000000	0x6FFFFFFF	Qspi Flash Bank	256MB
PERIPHERAL	0x50000000	0x5FFFFFFF	Reserved	
	0x40030000	0x4FFFFFFF	AHB1 SFR	
	0x40020000	0x4002FFFF	AHB0 SFR	
	0x40010000	0x4001FFFF	APB1 SFR	
	0x40000000	0x4000FFFF	APB0 SFR	
SRAM	0x30000400	0x3FFFFFFF	Reserved	
	0x30000000	0x300003FF	Retention SRAM	1KB
	0x20010000	0x2FFFFFFF	Reserved	
	0x20000000	0x2000FFFF	System SRAM	64KB
CODE	0x18010000	0x1FFFFFFF	Reserved	
	0x18000000	0x1800FFFF	System SRAM	64KB
	0x10004000	0x17FFFFFF	Reserved	
	0x10003000	0x10003FFF	Option Bytes	4KB
	0x10002000	0x10002FFF	Factory Bytes	4KB
	0x10001C00	0x10001FFF	OTP	1KB
	0x10000000	0x10001BFF	BootLoader	7KB
	0x08040000	0x0FFFFFFF	Reserved	
	0x08000000	0x0803FFFF	Flash Main	256KB
	0x00040000	0x07FFFFFF	Reserved	
	0x00000000	0x0003FFFF	Flash Main/BootLoader/ System SRAM(1)	256KB

(1) The memory corresponding to address 0x00000000 is determined by the boot mode.

7.2.1 AHB0 SFR

See the table below for AHB0 SFR Internal Address Mapping.

Table 7-3 AHB0 SFR address mapping

Start Address	End Address	Description	Size
0x40025000	0x4002FFFF	Reserved	
0x40024000	0x40024FFF	DMAC1	4KB
0x40023000	0x40023FFF	DMAC0	4KB
0x40022000	0x40022FFF	CRC	4KB
0x40021000	0x40021FFF	QSPI	4KB
0x40020000	0x40020FFF	EFC	4KB

7.2.2 AHB1 SFR

See the table below for AHB1 SFR Internal Address Mapping.

Table 7-4 AHB1 SFR address mapping

Start Address	End Address	Description	Size
0x40034000	0x4003FFFF	Reserved	
0x40033000	0x40033FFF	RNGC	4KB
0x40030000	0x40032FFF	SAC	12KB(1)(2)

(1) *Low 8KB is ARAM space, and high 4KB is for registers.* 。

(2) *ARAM space can only be accessed in word.*

7.2.3 APB0 SFR

See the table below for APB0 SFR Internal Address Mapping.

Table 7-5 APB0 SFR address mapping

起始地址	结束地址	功能描述	地址范围
0x4000f000	0x4000FFFF	SEC	4KB
0x4000e000	0x4000EFFF	RTC	4KB
0x4000d800	0x4000DFFF	LPTIM1	2KB
0x4000d000	0x4000D7FF	LPTIM0	2KB
0x4000c000	0x4000CFFF	BASICTIM0	4KB
0x4000b000	0x4000BFFF	GPTIM2	4KB
0x4000a000	0x4000AFFF	GPTIM0	4KB
0x40009000	0x40009FFF	LORAC	4KB
0x40008000	0x40008FFF	AFEC	4KB

起始地址	结束地址	功能描述	地址范围
0x40007000	0x40007FFF	I2C0	4KB
0x40006000	0x40006FFF	SSP0	4KB
0x40005000	0x40005FFF	LPUART	4KB
0x40004000	0x40004FFF	UART1	4KB
0x40003000	0x40003FFF	UART0	4KB
0x40002000	0x40002FFF	I2S	4KB
0x40001800	0x40001FFF	PWR	2KB
0x40001000	0x400017FF	SYSCFG	2KB
0x40000000	0x40000FFF	RCC	4KB

7.2.4 APB1 SFR

See the table below for APB1 SFR Internal Address Mapping.

Table 7-6 APB1 SFR address mapping

起始地址	结束地址	功能描述	地址范围
0x4001fc00	0x4001FFFF	PortD	1KB
0x4001f800	0x4001FBFF	PortC	1KB
0x4001f400	0x4001F7FF	PortB	1KB
0x4001f000	0x4001F3FF	PortA	1KB
0x4001e000	0x4001EFFF	WWDG	4KB
0x4001d000	0x4001DFFF	IWDG	4KB
0x4001c000	0x4001CFFF	BASICTIM1	4KB
0x4001b000	0x4001BFFF	GPTIM3	4KB
0x4001a000	0x4001AFFF	GPTIM1	4KB
0x40019000	0x40019FFF	DACCTRL	4KB
0x40018000	0x40018FFF	LCDCTRL	4KB
0x40017000	0x40017FFF	ADCCTRL	4KB
0x40016000	0x40016FFF	Reserved	4KB
0x40015000	0x40015FFF	I2C2	4KB
0x40014000	0x40014FFF	I2C1	4KB
0x40013000	0x40013FFF	SSP2	4KB
0x40012000	0x40012FFF	SSP1	4KB
0x40011000	0x40011FFF	UART3	4KB
0x40010000	0x40010FFF	UART2	4KB

7.3 SRAM

The SRAM in ASR6601 includes system SRAM, retention SRAM and SAC SRAM. SAC SRAM only supports word access, and system SRAM and retention SRAM support word, halfword, and byte access.

7.4 Boot Modes

The boot mode can be configured by the levels of BOOT0 pin (GPIO02) and the data in the Flash.

Table 7-7 ASR6601 Boot Mode Configuration

DEBUG_LEVEL	USE_FLASH_BOOT0	FLASH_BOOT0	BOOT0 PIN	FLASH_BOOT1	MAIN_FLASH_EMPTY	Boot Config
2	X	X	X	X	X	Boot from Flash Main
<2	0	X	0	X	0	Boot from Flash Main
<2	0	X	0	X	1	Boot from Flash Bootloader
<2	0	X	1	1	X	Boot from Flash Bootloader
<2	0	X	1	0	X	Boot from System SRAM
<2	1	1	X	X	0	Boot from Flash Main
<2	1	1	X	X	1	Boot from Flash Bootloader
<2	1	0	X	1	X	Boot from Flash Bootloader
<2	1	0	X	0	X	Boot from System SRAM

DebugLevel, UseFlashBoot0, FlashBoot0 and FlashBoot1 is the information area of the Flash, they can be modified according to the application. MainFlashEmpty is determined by the data of address 0 in the Flash Main area. If the data in the address 0 of Flash Main area is 0xFFFFFFFF, the value of MainFlashEmpty is 1, otherwise the value of MainFlashEmpty is 0. BOOT0 pin is GPIO02 in the package.

The boot mode is selected according to the configurations when the system is in these status: first powered up, exit the Standby mode or reset.

7.5 SYSCFG Registers

Base Address: 0x40001000

Table 7-8 SYSCFG Registers Summary

Register	Offset	Description
SYSCFG_CR0	0x000	Control Register 0, DMA handshakeSYSCFG_CR1
0x004 Control Register 1, DMA handshake SYSCFG_CR2 0x008 Control Register 2		
SYSCFG_CR3 0x00C	控制寄存器 3，	低功耗 Debug 连接控制
SYSCFG_CR4	0x010	Control Register 4
SYSCFG_CR5 0x014 控制寄存器 5		
SYSCFG_CR6 0x018 控制寄存器 6，安全锁定控制		
SYSCFG_CR7	0x01C 控制寄存器	7，安全锁定控制
SYSCFG_CR8	0x020 控制寄存器	8，QSPI 存储密钥
SYSCFG_CR9 0x024	控制寄存器	9，QSPI REMAP 控制
SYSCFG_CR10 0x028 控制寄存器 10		

7.5.1 SYSCFG_CR0

Offset: 0x000

Reset Value: 0x00000000

31-30	29-24	23-22	21-16
RESERVED	DMAC0_HANDSHAKE0_SEL	RESERVED	DMAC0_HANDSHAKE1_SEL
r	r/w	r	r/w
15-14	**13-8**	**7-6**	**5-0**
RESERVED	DMAC0_HANDSHAKE2_SEL	RESERVED	DMAC0_HANDSHAKE3_SEL
r	r	r	r/w

Bits 31-30 RESERVED: Must be kept, and can't be modified.

Bits 29-24 DMAC0_HANDSHAKE0_SEL: DMAC0 HANDSHAKE0 selection. For details, please refer to *Table7-9 DMA Request MUX.*

Bits 23-22 RESERVED: Must be kept, and can't be modified.

Bits 21-16 DMAC0_HANDSHAKE1_SEL: DMAC0 HANDSHAKE1 selection. For details, please refer to *Table7-9 DMA Request MUX.*

Bits 15-14 RESERVED: Must be kept, and cannot be modified.

Bits 13-8 DMAC0_HANDSHAKE2_SEL: DMAC0 HANDSHAKE2 selection. For details, please refer to *Table7-9 DMA Request MUX.*

Bits 7-6 RESERVED: Must be kept, and cannot be modified.

Bits 5-0 DMAC0_HANDSHAKE3_SEL: DMAC0 HANDSHAKE3 selection. For details, please refer to *Table7-9 DMA Request MUX.*

7.5.2 SYSCFG_CR1

Offset: 0x004

Reset Value: 0x00000000

31-30	29-24	23-22	21-16
RESERVED	DMAC1_HANDSHAKE0_SEL	RESERVED	DMAC1_HANDSHAKE1_SEL
r	r/w	r	r/w
15-14	**13-8**	**7-6**	**5-0**
RESERVED	DMAC1_HANDSHAKE2_SEL	RESERVED	DMAC1_HANDSHAKE3_SEL
r	r	r	r/w

Bits 31-30 RESERVED: Must be kept, and cannot be modified.

Bits 29-24 DMAC1_HANDSHAKE0_SEL: DMAC1 HANDSHAKE0 selection. For details, please refer to *Table7-9 DMA Request MUX.*

Bits 23-22 RESERVED: Must be kept, and cannot be modified.

Bits 21-16 DMAC1_HANDSHAKE1_SEL: DMAC1 HANDSHAKE1 selection. For details, please refer to *Table7-9 DMA Request MUX.*

Bits 15-14 RESERVED: Must be kept, and cannot be modified.

Bits 13-8 DMAC1_HANDSHAKE2_SEL: DMAC1 HANDSHAKE2 selection. For details, please refer to *Table7-9 DMA Request MUX.*

Bits 7-6 RESERVED: Must be kept, and cannot be modified.

Bits 5-0 DMAC1_HANDSHAKE3_SEL: DMAC1 HANDSHAKE3 selection. For details, please refer to *Table7-9 DMA Request MUX.*

7.5.3 SYSCFG_CR2

Offset: 0x008

Reset Value: 0x00000000

31	30	29-28	27
RESERVED	SYSCFG_HALTED_IPTIM1_EN	RESERVED	SYSCFG_HALTED_LPTIM0_EN
r	r/w	r	r/w
26	**25**	**24**	**23**
SYSCFG_HALTED_IWDG_EN	SYSCFG_HALTED_WWDG_EN	SYSCFG_HALTED_GPTIM0_EN	SYSCFG_HALTED_GPTIM1_EN
r/w	r/w	r/w	r/w
22	**21**	**20**	**19**
SYSCFG_HALTED_GPTIM2_EN	SYSCFG_HALTED_GPTIM3_EN	SYSCFG_HALTED_BASICTIM0_EN	SYSCFG_HALTED_BASICTIM1_EN
r/w	r/w	r/w	r/w
18	**17**	**16-12**	
QSPI_MEM_ENCRYPT_EN	QSPI_REMAP_ENABLE	RESERVED	
r/w	r/w	r	
11	**10**	**9-8**	
CPU_STCALIB_SKEW	SYSCFG_DBG_SLEEP	RESERVED	
r/w	r/w	r	
7	**6**	**5**	**4**
UART0_DMA_CLR_SEL	UART1_DMA_CLR_SEL	UART2_DMA_CLR_SEL	UART3_DMA_CLR_SEL
r/w	r/w	r/w	r/w
3	**2**	**1**	**0**
SSP0_DMA_CLR_SEL	SSP1_DMA_CLR_SEL	SSP2_DMA_CLR_SEL	SSP_AFEC_DMA_CLR_SEL
r/w	r/w	r/w	r/w

Bit 31 RESERVED: Must be kept, and cannot be modified.

Bit 30 SYSCFG_HALTED_LPTIM1_EN: Stop LPTIM1 counter if the core is halted

- 0: LPTIM1 counter continues to work normally when the core is halted

- 1: LPTIM1 counter is stopped when the core is halted

Bits 29-28 RESERVED: Must be kept, and cannot be modified.

Bit 27 SYSCFG_HALTED_LPTIM0_EN: Stop LPTIM0 counter if the core is halted

- 0: LPTIM0 counter continues to work normally when the core is halted
- 1: LPTIM0 counter is stopped when the core is halted

Bit 26 SYSCFG_HALTED_IWDG_EN: Stop independent watchdog counter if the core is halted

- 0: the independent watchdog counter continues to work normally when the core is halted
- 1: the independent watchdog counter is stopped when the core is halted

Bit 25 SYSCFG_HALTED_WWDG_EN: Stop window watchdog counter if the core is halted

- 0: The window watchdog counter continues to work normally when the core is halted
- 1: The window watchdog counter is stopped when the core is halted

Bit 24 SYSCFG_HALTED_GPTIM0_EN: Stop GPTIM0 counter if the core is halted

- 0: GPTIM0 counter continues to work normally when the core is halted
- 1: GPTIM0 counter is stopped when the core is halted

Bit 23 SYSCFG_HALTED_GPTIM1_EN: Stop GPTIM1 counter if the core is halted

- 0: GPTIM1 counter continues to work normally when the core is halted
- 1: GPTIM1 counter is stopped when the core is halted

Bit 22 SYSCFG_HALTED_GPTIM2_EN: Stop GPTIM2 counter if the core is halted

- 0: GPTIM2 counter continues to work normally when the core is halted
- 1: GPTIM2 counter is stopped when the core is halted

Bit 21 SYSCFG_HALTED_GPTIM3_EN: Stop GPTIM3 counter if the core is halted

- 0: GPTIM3 counter continues to work normally when the core is halted
- 1: GPTIM3 counter is stopped when the core is halted

Bit 20 SYSCFG_HALTED_BASICTIM0_EN: Stop BASICTIM0 counter if the core is halted

- 0: BASICTIM0 counter continues to work normally when the core is halted
- 1: BASICTIM0 counter is stopped when the core is halted

Bit 19 SYSCFG_HALTED_BASICTIM1_EN: Stop BASICTIM1 counter if the core is halted

- 0: BASICTIM1 counter continues to work normally when the core is halted
- 1: BASICTIM1 counter is stopped when the core is halted

Bit 18 QSPI_MEM_ENCRYPT_EN: QSPI memory encryption enable

- 0: disabled
- 1: enabled

Bit 17 QSPI_REMAP_ENABLE: QSPI remap function enable

- 0: disabled
- 1: enabled

Bits 16-12 RESERVED: Must be kept, and cannot be modified.

Bit 11 CPU_STCALIB_SKEW: CPU SysTick skew configuration. Affects STCALIB[24] bit.

- 0: disable
- 1: enable

Bit 10 SYSCFG_DBG_SLEEP: Allow debug connection in Deepsleep mode

It is only used in debug mode and it will affect the Deepsleep mode.

- 0: not allowed
- 1: allowed

Bits 9-8 RESERVED: Must be kept, and cannot be modified.

Bit 7 UART0_DMA_CLR_SEL: UART0 DMA_CLR signal selection

It is recommended to set this bit to improve DMAC transfer efficiency. UART module uses the synchronized DMA_CLR signal by default.

- 0: use the DMA_CLR signal after 2 cycles
- 1: directly use the DMA_CLR signal output by DMAC

Bit 6 UART1_DMA_CLR_SEL: UART1 DMA_CLR signal selection

It is recommended to set this bit to improve DMAC transfer efficiency. UART module uses the synchronized DMA_CLR signal by default.

- 0: use the DMA_CLR signal after 2 cycles
- 1: directly use the DMA_CLR signal output by DMAC

Bit 5 UART2_DMA_CLR_SEL: UART2 DMA_CLR signal selection

It is recommended to set this bit to improve DMAC transfer efficiency. UART module uses the synchronized DMA_CLR signal by default.

- 0: use the DMA_CLR signal after 2 cycles
- 1: directly use the DMA_CLR signal output by DMAC

Bit 4 UART3_DMA_CLR_SEL: UART3 DMA_CLR signal selection

It is recommended to set this bit to improve DMAC transfer efficiency. UART module uses the synchronized DMA_CLR signal by default.

- 0: use the DMA_CLR signal after 2 cycles
- 1: directly use the DMA_CLR signal output by DMAC

Bit 3 SSP0_DMA_CLR_SEL: SSP0 DMA_CLR signal selection

It is recommended to set this bit to improve DMAC transfer efficiency. SSP module uses the synchronized DMA_CLR signal by default.

- 0: use the DMA_CLR signal after 2 cycles
- 1: directly use the DMA_CLR signal output by DMAC

Bit 2 SSP1_DMA_CLR_SEL: SSP1 DMA_CLR signal selection

It is recommended to set this bit to improve DMAC transfer efficiency. SSP module uses the synchronized DMA_CLR signal by default.

- 0: use the DMA_CLR signal after 2 cycles
- 1: directly use the DMA_CLR signal output by DMAC

Bit 1 SSP2_DMA_CLR_SEL: SSP2 DMA_CLR signal selection

It is recommended to set this bit to improve DMAC transfer efficiency. SSP module uses the synchronized DMA_CLR signal by default.

- 0: use the DMA_CLR signal after 2 cycles
- 1: directly use the DMA_CLR signal output by DMAC

Bit 0 SSP_AFEC_DMA_CLR_SEL: SSP (for afec) DMA_CLR signal selection

It is recommended to set this bit to improve DMAC transfer efficiency. SSP module uses the synchronized DMA_CLR signal by default.

- 0: use the DMA_CLR signal after 2 cycles
- 1: directly use the DMA_CLR signal output by DMAC

7.5.4 SYSCFG-CR3

Offset: 0x00C

Reset Value: 0x00000000

This register is in the AON domain.

31-2	1	0
RESERVED	SYSCFG_DBG_STOP	SYSCFG_DBG_STANDBY
r	r/w	r/w

Bits 31-2 RESERVED: Must be kept, and cannot be modified.

Bit 1 SYSCFG_DBG_STOP: Allow a debug connection in Stop mode. It is only used in debug and it will affect Stop mode implementation.

- 0: not allowed
- 1: allowed

Bit 0 SYSCFG_DBG_STANDBY: Allow a debug connection in Standby mode. It is only used in debug and it will affect Standby mode implementation.

- 0: not allowed
- 1: allowed

7.5.5 SYSCFG_CR4

Offset: 0x010

Reset Value: 0x00000000

This register is in the AON domain.

31	30-0
SYSCFG_CR4_REG	USER-DEFINED
r/w	r/w

Bit 31 SYSCFG_CR4_REG: LPTIM1 IN2 remapping enable

- 0: disabled, LPTIM1_IN2 is determined by GPIO AFR
- 1: enabled, LPTIM1_IN2 is derived from LPTIM0_IN1

Bits 30-0 USER-DEFINED: These bits are user-defined and can be used to store a small amount of data by software.

7.5.6 SYSCFG_CR5

Offset: 0x014

Reset Value: 0x00000000

This register is in the AON domain.

31-0
SYSCFG_CR5_REG
r/w

Bits 31-0 SYSCFG_CR5_REG: These bits are user-defined and can be used to store a small amount of data by software.

7.5.7 SYSCFG_CR6

Offset: 0x018

Reset Value: 0x00000000

31-16	15	14-5	4
RESERVED	RNGC_SECURE_LOCK	ANALOG_MAIN_SECURE_LOCK	RESERVED
r	r/w	r/w	r
3	**2**	**1**	**0**
SEC_SECURE_LOCK	SAC_SECURE_LOCK	DMAC0_SLAVE_SECURE_LOCK	DMAC0_MASTER_SECURE_LOCK
r/w	r/w	r/w	r/w

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bit 15 RNGC_SECURE_LOCK: RNGC security lock

- 0: security lock disabled
- 1: security lock enabled

Bits 14-5 ANALOG_MAIN_SECURE_LOCK: Security lock for main domain configuration of AFEC

[5] Correspond to VD

- 0: security lock disabled
- 1: security lock enabled

[6] Correspond to TD

- 0: security lock disabled
- 1: security lock enabled

[7] Correspond to LD

- 0: security lock disabled
- 1: security lock enabled

[8] Correspond to FD24M

- 0: security lock disabled
- 1: security lock enabled

[9] Correspond to FD32M

- 0: security lock disabled
- 1: security lock enabled

[10] Correspond to RNG

- 0: security lock disabled
- 1: security lock enabled

[11] Correspond to TEST

- 0: security lock disabled
- 1: security lock enabled

[14:12]: Unused

- 0: security lock disabled
- 1: security lock enabled

Bit 4 RESERVED: Must be kept, and cannot be modified.

Bit 3 SEC_SECURE_LOCK: SEC security lock

- 0: security lock disabled
- 1: security lock enabled

Bit 3 SAC_SECURE_LOCK: SAC security lock

- 0: security lock disabled
- 1: security lock enabled

Bit 1 DMAC0_SLAVE_SECURE_LOCK: DMAC0 slave interface security lock

- 0: security lock disabled
- 1: security lock enabled

Bit 0 DMAC0_MASTER_SECURE_LOCK: DMAC0 master interface security lock

- 0: security lock disabled
- 1: security lock enabled

7.5.8 SYSCFG_CR7

Offset: 0x01C

Reset Value: 0x00000000

This register is in the AON domain.

31-15	14-5	4
RESERVED	ANALOG_AON_SECURE_LOCK	RTC_CALENDAR_SECURE_LOCK
r	r/w	r/w

3	2	1	0
RTC_WAKEUP2_SECURE_LOCK	RTC_WAKEUP1_SECURE_LOCK	RTC_WAKEUP0_SECURE_LOCK	RTC_TAMPER_SECURE_LOCK
r/w	r/w	r/w	r/w

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 14-5 ANALOG_AON_SECURE_LOCK: Security lock for AON domain configuration of AFEC

[5] Correspond to LPLDO

- 0: security lock disabled
- 1: security lock enabled

[6] Correspond to RCO3.6M

- 0: security lock disabled
- 1: security lock enabled

[7] Correspond to PWRSW

- 0: security lock disabled
- 1: security lock enabled

[8] Correspond to RCO32K

- 0: security lock disabled
- 1: security lock enabled

[9] Correspond to XO32K

- 0: security lock disabled
- 1: security lock enabled

[10] Correspond to LDO12

- 0: security lock disabled
- 1: security lock enabled

[11] Correspond to FD32K

- 0: security lock disabled
- 1: security lock enabled

[14:12] Unused

- 0: security lock disabled
- 1: security lock enabled

Bit 4 RTC_CALENDAR_SECURE_LOCK: Calendar configuration security lock in RTC

- 0: security lock disabled
- 1: security lock enabled

Bit 3 RTC_WAKEUP2_SECURE_LOCK: Wakeup2 configuration security lock in RTC

- 0: security lock disabled
- 1: security lock enabled

Bit 2 RTC_WAKEUP1_SECURE_LOCK: Wakeup1 configuration security lock in RTC

- 0: security lock disabled
- 1: security lock enabled

Bit 1 RTC_WAKEUP0_SECURE_LOCK: Wakeup0 configuration security lock in RTC

- 0: security lock disabled
- 1: security lock enabled

Bit 0 RTC_TAMPER_SECURE_LOCK: Tamper configuration security lock in RTC

- 0: security lock disabled
- 1: security lock enabled

7.5.9 SYSCFG_CR8

Offset: 0x020

Reset Value: 0x00000000

31-0
QSPI_MEM_ENCRYPT_KEY
r/w

Bits 31-0 QSPI_MEM_ENCRYPT_KEY: Encryption key for QSPI memory

7.5.10 SYSCFG_CR9

Offset:0x024

Reset Value: 0x00000000

31-28	27-14	13-0
RESERVED	QSPI_REMAP_SRC_ADDR	QSPI_REMAP_DST_ADDR
r	r/w	r/w

Bits 31-28 RESERVED: Must be kept, and cannot be modified.

Bits 27-14 QSPI_REMAP_SRC_ADDR: QSPI remap source address, aligned in 1KB

Bits 13-0 QSPI_REMAP_DST_ADDR: QSPI remap destination address, aligned in 1KB

7.5.11 SYSCFG_CR10

Offset: 0x028

Reset Value: 0x00000000

31-24	23	22	21-15	14	13-0
RESERVED	I2S_WS_SEL	I2S_WS_EN	I2S_WS_LEN	I2S_MODE_SEL	QSPI_REMAP_SIZE
r	r/w	r/w	r/w	r/w	r/w

Bits 31-24 RESERVED: Must be kept, and cannot be modified.

Bit 23 I2S_WS_SEL: I2S WS output delay enable

- 0: output delay disabled
- 1: output delay enabled

Note: *This bit can only be configured when the I2S acts as master interface. When enabled, the WS signal is output one cycle later than the data transmission.*

Bit 22 I2S_WS_EN: I2S WS enable

- 0: disabled
- 1: enabled

Note: *This bit can only be configured when the I2S acts as master interface. When enabled, the WS signal is generated based on the I2S_WS_LEN configuration.*

Bits 21-15 I2S_WS_LEN: I2S main interface resolution configuration

N: WS frequency=I2S interface clock frequency/[(N+1)*2]

The I2S interface clock frequency is jointly determined by the I2S_CLK_DIV and I2S_CLK_SEL bits in the RCC_CR3 and RCC_CR2 registers.

Bit 14 I2S_MODE_SEL: I2S works in master or slave mode

- 0: slave mode
- 1: master mode

Note: *In addition to this register, it is also necessary to configure the I2S_CLK_DIV and I2S_CLK_SEL bits in the RCC_CR3 and RCC_CR2 registers, as well as the alternate functions of GPIOs.*

Bits 13-0 QSPI_REMAP_SIZE: Address space for QSPI remapping, aligned in 1KB

7.6 DMA Request MUX

Table 7-9 DMA Request MUX

No.	Source
63	
62	
61	
60	
59	
58	
57	
56	
55	
54	
53	basictim0_up
52	basictim1_up
51	gptim3_up
50	gptim3_trg
49	gptim3_ch0
48	gptim3_ch1
47	gptim2_up
46	gptim2_trg
45	gptim2_ch0
44	gptim2_ch1
43	gptim1_up
42	gptim1_trg
41	gptim1_ch0
40	gptim1_ch1
39	gptim1_ch2
38	gptim1_ch3
37	gptim0_up
36	gptim0_trg
35	gptim0_ch0
34	gptim0_ch1
33	gptim0_ch2
32	gptim0_ch3
31	uart0_rx
30	uart0_tx
29	uart1_rx
28	uart1_tx
27	uart2_rx

No.	Source
26	uart2_tx
25	uart3_rx
24	uart3_tx
23	lpuart_rx
22	lpuart_tx
21	ssp0_rx
20	ssp0_tx
19	ssp1_rx
18	ssp1_tx
17	ssp2_rx
16	ssp2_tx
15	i2c0_rx
14	i2c0_tx
13	i2c1_rx
12	i2c1_tx
11	i2c2_rx
10	i2c2_tx
9	
8	
7	adcctrl
6	dacctrl
5	lorac_rx
4	lorac_tx
3	
2	
1	
0	

8. Reset and Clock Control (RCC)

8.1 Reset

There are four types of reset: external reset, power reset, system reset and low-power reset.

8.1.1 External Reset

The external reset is triggered by RSTN IO input (active at low level).
The external reset is used to reset all digital logic.

8.1.2 Power-on Reset

The power-on reset is generated by the BOR (Brownout reset) circuitry. The BOR circuitry monitors VBAT to ensure that the internal reset is released when the voltage is greater than 1.8V.

Power-on reset is used to reset all digital logic.

8.1.3 System Reset

System reset sources include IWDG Reset, WWDG Reset, Option Byte Load Reset, Software Reset, SEC Reset, Power-on Reset, and External Reset.

- IWDG Reset: generated by the IWDG module for exception recovery.
- WWDG Reset: generated by the WWDG module for exception recovery.
- Option Byte Load Reset: generated by the EFC module and used to start option byte reloading.
- Software Reset: generated by the CPU.
- SEC Reset: generated by the SEC module and used for system reset after security alarm.

System reset is used to reset most of the data logic in the Main domain, but does not affect the reset source status register, which is used to determine which system reset source generates this reset.

8.1.4 Low-power Reset

The low-power reset is generated by the low-power state machine and is used to reset the logic of the main domain when the CPU exits Standby or Stop3 mode.

8.2 Clock

System clock structure:

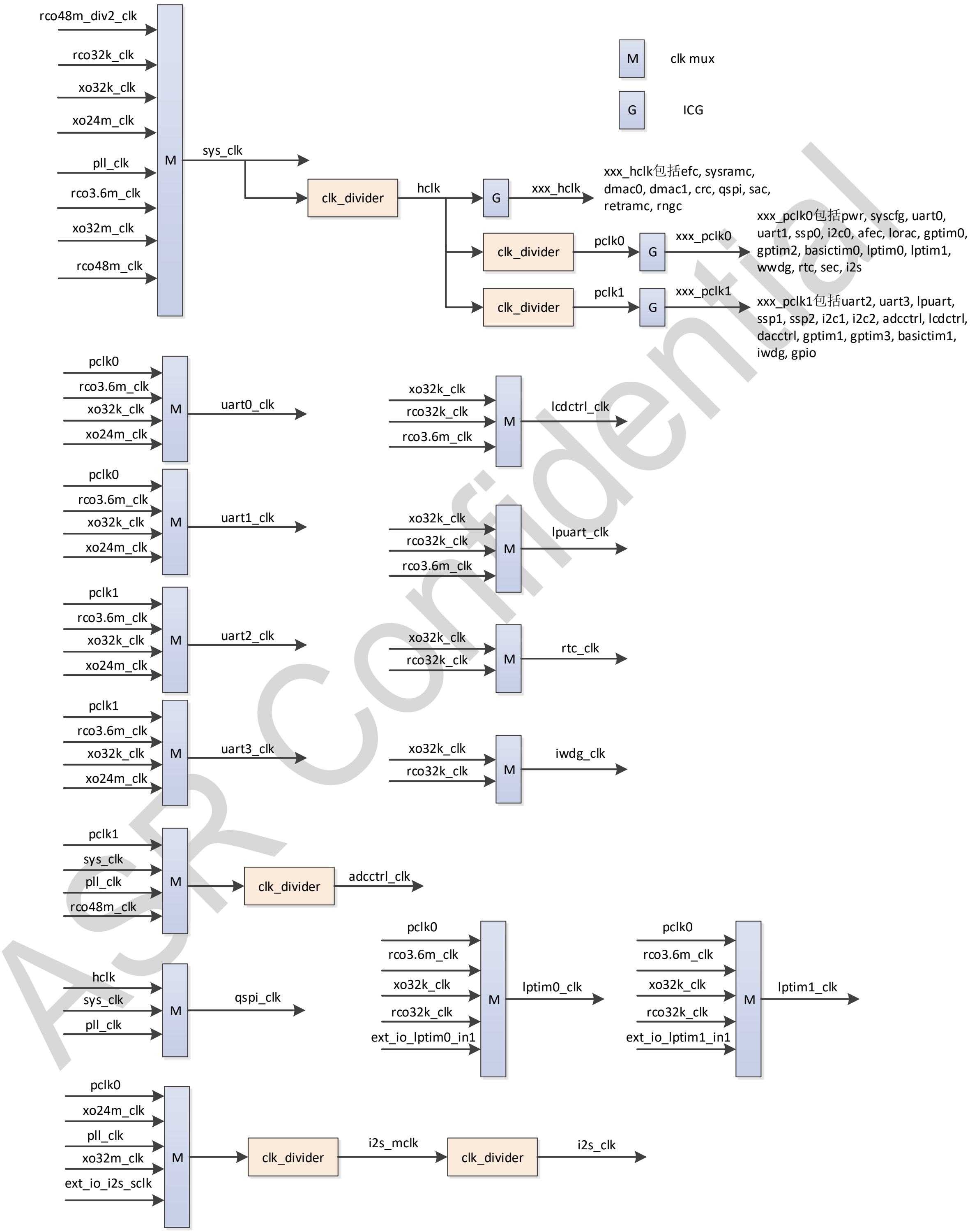

Figure 8-1 Clock Tree

8.2.1 System clock SYS_CLK

The sources of system clock SYS_CLK include RCO48M divided by 2, RCO32K, XO32K, PLL, XO24M, XO32M, RCO3.6M, RCO48M. The default is RCO48M divided by 2.

- RCO48M (48MHz) is generated from the internal clock circuit.
- RCO32K (32kHz) is generated from the internal clock circuit.
- RCO3.6M (3.6MHz) is generated from the internal clock circuit.
- XO32K (32.768kHz) is generated from an external crystal oscillator.
- XO32M (32MHz) is generated from an external crystal oscillator.
- XO24M (24MHz) is generated from an external crystal oscillator.
- PLL is an internal clock circuit, RCO48M, XO32M, XO24M or RCO3.6M can be selected as PLL clock source, and the PLL clock output supports up to 48MHz.

AHB bus clock HCLK is generated from SYS_CLK divided by 2^N (N ranges from 0 to 9).
The system includes two APB buses, the APB bus clock PCLK1 and PCLK2 are generated from HCLK divided by 2^M (M in range from 0 to 4). The clock division factor for the two APB buses can be configured independently.

8.2.2 Clocks for the Modules

The clocks for the modules consist of bus clocks and interface clocks.

The bus clock is generated by HCLK or PCLK gating and is used for modules to access bus.

In addition to a bus clock, some modules also have an independent interface clock, which is different from the bus clock, and is used to realize the function of the module.

The interface clock source for each module is selectable by software:

- LPTIM: PCLK0, RCO3.6M, XO32K, RCO32K, IO input clock;
- LCDCTRL: XO32K, RCO32K, RCO3.6M;
- LPUART: XO32K, RCO32K, RCO3.6M;
- RTC: XO32K, RCO32K;
- IWDG: XO32K, RCO32K;
- UART: PCLK0/PCLK1, RCO3.6M, XO32K, XO24M;
- ADCCTRL: PCLK1, SYS_CLK, PLL, RCO48M;
- I2S: PCLK0, XO24M, PLL, XO32M, input clock IO;
- QSPI: HCLK, SYS_CLK, PLL;

ADCCTRL and I2S also support interface clock division, which is used to generate low frequency interface clocks.
LPTIM, LCDCTRL, LPUART, RTC and IWDG in AON domain and those in Main domain can be enabled or disabled independently.

8.2.3 MCO Clock output

The microcontroller clock output (MCO) capability allows the internal clock to be output by IO.

MCO clock source can be RCO32K, XO32K, RCO3.6M, XO24M, XO32M, RCO48M, PLL or SYS_CLK.

The clock can be output with a frequency divided by software configuration.

8.3 RCC Registers

Base Address: 0x40000000

Table 8-1 RCC Registers Summary

寄存器	偏移量	描述
RCC_CR0	0x000	控制寄存器 0
RCC_CR1	0x004	控制寄存器 1，接口时钟来源选择
RCC_CR2	0x008	控制寄存器 2，接口时钟来源选择
RCC_CGR0	0x00C	模块时钟门控寄存器 0
RCC_CGR1	0x010	模块时钟门控寄存器 1
RCC_CGR2	0x014	模块时钟门控寄存器 2
RCC_RST0	0x018	模块复位控制寄存器 0
RCC_RST1	0x01C	模块复位控制寄存器 1
RCC_RST_SR	0x020	系统复位源状态寄存器
RCC_RST_CR	0x024	系统复位源使能寄存器
RCC_SR	0x028	状态寄存器，配置完成状态
RCC_SR1	0x02C	状态寄存器 1，模块时钟门控状态
RCC_CR3	0x030	控制寄存器 3，接口时钟分频控制

8.3.1 RCC_CR0

Offset: 0x000

Reset Value: 0x00000000

31-26	25	24-22	21-19	18
RESERVED	STCLKEN_SEL	MCO_CLK_DIV_NUM	MCO_CLK_SEL	MCO_CLK_OUT_EN
r	r/w	r/w	r/w	r/w
17-15	**14-12**	**11-8**	**7-5**	**4-0**
PCLK1_DIV	SYS_CLK_SEL	HCLK_DIV	PCLK0_DIV	RESERVED
r/w	r/w	r/w	r/w	r

Bits 31-26 RESERVED: Must be kept, and cannot be modified.

Bit 25 STCLKEN_SEL: CPU SysTick clock source selection.

- 0: XO32K
- 1: RCO32K

Bits 24-22 MCO_CLK_DIV_NUM: MCO division factor.

- <4: division factor 1
- 4: division factor 2
- 5: division factor 4
- 6: division factor 8
- 7: division factor 16

Note: *Make sure to configure this bit when MCO_CLK_OUT_EN=0. If the MCO_CLK_OUT_EN bit is enabled, users must disable it by software first, wait for at least 2 current clock cycles or query the RCC_SR1 register, and then configure the MCO division factor.*

Bits 21-19 MCO_CLK_SEL: MCO clock source selection.

- 0: RCO32K
- 1: XO32K
- 2: RCO3.6M
- 3: XO24M
- 4: XO32M
- 5: RCO48M
- 6: PLL
- 7: SYS_CLK

Note: *Make sure to configure this bit when MCO_CLK_OUT_EN=0. If the MCO_CLK_OUT_EN bit is enabled, users must disable it by software first, wait for at least 2 current clock cycles or query the RCC_SR1 register, and then configure the MCO clock source.*

Bit 18 MCO_CLK_OUT_EN: MCO output enable

- 0: disabled
- 1: enabled

Bits 17-15 PCLK1_DIV: PCLK1 division factor.

- 0: PCLK1 clock frequency = HCLK clock frequency
- 1: PCLK1 clock frequency = 1/2 HCLK clock frequency
- 2: PCLK1 clock frequency = 1/4 HCLK clock frequency
- 3: PCLK1 clock frequency = 1/8 HCLK clock frequency
- >3: PCLK1 clock frequency = 1/16 HCLK clock frequency

Bits 14-12 SYS_CLK_SEL: SYS_CLK clock source selection=

- 0: RCO48M divided by 2
- 1: RCO32K
- 2: XO32K
- 3: PLL
- 4: XO24M
- 5: XO32M
- 6: RCO3.6M
- 7: RCO48M

Bits 11-8 HCLK_DIV: HCLK division factor

- 0: HCLK clock frequency = SYS_CLK clock frequency
- 1: HCLK clock frequency = 1/2 SYS_CLK clock frequency
- 2: HCLK clock frequency = 1/4 SYS_CLK clock frequency
- 3: HCLK clock frequency = 1/8 SYS_CLK clock frequency
- 4: HCLK clock frequency = 1/16 SYS_CLK clock frequency
- 5: HCLK clock frequency = 1/32 SYS_CLK clock frequency
- 6: HCLK clock frequency = 1/64 SYS_CLK clock frequency
- 7: HCLK clock frequency = 1/128 SYS_CLK clock frequency
- 8: HCLK clock frequency = 1/256 SYS_CLK clock frequency
- >8: HCLK clock frequency = 1/512 SYS_CLK clock frequency

Bits 7-5 PCLK0_DIV: PCLK0 division factor

- 0: PCLK0 clock frequency = HCLK clock frequency
- 1: PCLK0 clock frequency = 1/2 HCLK clock frequency
- 2: PCLK0 clock frequency = 1/4 HCLK clock frequency
- 3: PCLK0 clock frequency = 1/8 HCLK clock frequency
- >3: PCLK0 clock frequency = 1/16 HCLK clock frequency

Bits 4-0 RESERVED: Must be kept, and cannot be modified.

8.3.2 RCC_CR1

Offset: 0x004

Reset Value: 0x00000000

This register is in the AON domain.

31-12	11	10	9-8	
RESERVED	LPTIM1_EXT_CLK_SEL	LPTIM0_EXT_CLK_SEL	LPTIM1_CLK_SEL	
r	r/w	r/w	r/w	
7-6	**5-4**	**3-2**	**1**	**0**
LPTIM0_CLK_SEL	LCDCTRL_CLK_SEL	LPUART_CLK_SEL	RTC_CLK_SEL	IWDG_CLK_SEL
r/w	r/w	r/w	r/w	r/w

Bits 31-12 RESERVED: Must be kept, and cannot be modified.

Bit 11 LPTIM1_EXT_CLK_SEL: LPTIM1 interface clock source selection.

- 0: decided by the LPTIM1_CLK_SEL bit
- 1: use external clock from IN1

Notes:

1. Make sure to configure this bit when LPTIM1_CLK_EN=0. If the LPTIM1_CLK_EN bit is enabled, the user must disable it by software first, wait for at least 2 current clock cycles or query the RCC_SR1 register, and then configure the LPTIM1 interface clock source.

2. This bit and the LPTIM1_CLK_SEL bit jointly determine the LPTIM1 interface clock source.

Bit 10 LPTIM0_EXT_CLK_SEL: LPTIM0 interface clock source selection.

- 0: decided by the LPTIM0_CLK_SEL bit
- 1: use external clock from IN1

Notes:

1. Make sure to configure this bit when LPTIM0_CLK_EN=0. If the LPTIM0_CLK_EN bit is enabled, the user must disable it by software first, wait for at least 2 current clock cycles or query the RCC_SR1 register, and then configure the LPTIM0 interface clock source.

2. This bit and the LPTIM0_CLK_SEL bit jointly determine the LPTIM0 interface clock source.

Bits 9-8 LPTIM1_CLK_SEL: LPTIM1 interface clock source selection.

- 0: PCLK0
- 1: RCO3.6M
- 2: XO32K
- 3: RCO32K

Notes:

1. Make sure to configure this bit when LPTIM1_CLK_EN=0. If the LPTIM1_CLK_EN bit is enabled, the user must disable it by software first, wait for at least 2 current clock cycles or query the RCC_SR1 register, and then configure the LPTIM1 interface clock source.

2. This bit and the LPTIM1_EXT_CLK_SEL bit jointly determine the LPTIM1 interface clock source.

3. To select PCLK0 as clock source, the LPTIM1_INF_CLK_EN bit in the RCC_CGR1 register must be enabled.

Bits 7-6 LPTIM0_CLK_SEL: LPTIM0 interface clock source selection.

- 0: PCLK0
- 1: RCO3.6M
- 2: XO32K
- 3: RCO32K

Notes:

1. Make sure to configure this bit when LPTIM0_CLK_EN=0. If the LPTIM0_CLK_EN bit is enabled, the user must disable it by software first, wait for at least 2 current clock cycles or query the RCC_SR1 register, and then configure the LPTIM0 interface clock source.

2. This bit and the LPTIM0_EXT_CLK_SEL bit jointly determine the LPTIM0 interface clock source.

3. To select PCLK0 as clock source, the LPTIM0_INF_CLK_EN bit in the RCC_CGR1 register must be enabled.

Bits 5-4 LCDCTRL_CLK_SEL: LCDCTRL interface clock source selection.

- 0: XO32K
- 1: RCO32K
- >1: RCO3.6M

Bits 3-2 LPUART_CLK_SEL: LPUART interface clock source selection.

- 0: XO32K
- 1: RCO32K
- >1: RCO3.6M

Bit 1 RTC_CLK_SEL: RTC interface clock source selection

- 0: XO32K
- 1: RCO32K

Bit 0 IWDG_CLK_SEL: IWDG interface clock source selection

- 0: XO32K
- 1: RCO32K

8.3.3 RCC_CR2

Offset: 0x008

Reset Value: 0x00000000

31-17	16-15	14-13	12-11	
RESERVED	UART0_CLK_SEL	UART1_CLK_SEL	UART2_CLK_SEL	
r	r/w	r/w	r/w	
10-9	**8-7**	**6-5**	**4-2**	**1-0**
UART3_CLK_SEL	RESERVED	ADCCTRL_CLK_SEL	I2S_CLK_SEL	QSPI_CLK_SEL
r/w	r	r/w	r/w	r/w

Bits 31-17 RESERVED: Must be kept, and cannot be modified.

Bits 16-15 UART0_CLK_SEL: UART0 interface clock source selection.

- 0: PCLK0
- 1: RCO3.6M
- 2: XO32K
- 3: XO24M

Note: *Make sure to configure this bit when UART0_CLK_EN=0. If the UART0_CLK_EN bit is enabled, the user must disable it by software first, wait for at least 2 current clock cycles or query the RCC_SR1 register, and then configure the UART0 interface clock source.*

Bits 14-13 UART1_CLK_SEL: UART1 interface clock source selection.

- 0: PCLK0
- 1: RCO3.6M
- 2: XO32K
- 3: XO24M

Note: *Make sure to configure this bit when UART1_CLK_EN=0. If the UART1_CLK_EN bit is enabled, the user must disable it by software first, wait for at least 2 current clock cycles or query the RCC_SR1 register, and then configure the UART1 interface clock source.*

Bits 12-11 UART2_CLK_SEL: UART2 interface clock source selection.

- 0: PCLK1
- 1: RCO3.6M
- 2: XO32K
- 3: XO24M

Note: *Make sure to configure this bit when UART2_CLK_EN=0. If the UART2_CLK_EN bit is enabled, the user must disable it by software first, wait for at least 2 current clock cycles or query the RCC_SR1 register, and then configure the UART2 interface clock source.*

Bits 10-9 UART3_CLK_SEL: UART3 interface clock source selection.

- 0: PCLK1
- 1: RCO3.6M
- 2: XO32K
- 3: XO24M

Note: *Make sure to configure this bit when UART3_CLK_EN=0. If the UART3_CLK_EN bit is enabled, the user must disable it by software first, wait for at least 2 current clock cycles or query the RCC_SR1 register, and then configure the UART3 interface clock source.*

Bits 8-7 RESERVED: Must be kept, and cannot be modified.

Bits 6-5 ADCCTRL_CLK_SEL: ADCCTRL interface clock source selection.

- 0: PCLK1
- 1: SYS_CLK
- 2: PLL
- 3: RCO48M

Note: *Make sure to configure this bit when ADCCTRL_CLK_EN=0. If the ADCCTRL_CLK_EN bit is enabled, the user must disable it by software first, wait for at least 2 current clock cycles or query the RCC_SR1 register, and then configure the ADCCTRL interface clock source.*

Bits 4-2 I2S_CLK_SEL: I2S interface clock source selection

- 0: PCLK0
- 1: XO24M
- 2: PLL
- 3： XO32M
- 3: XO32M

Notes:

1. Make sure to configure this bit when I2S_CLK_EN=0. If the I2S_CLK_EN bit is enabled, the user must disable it by software first, wait for at least 2 current clock cycles or query the RCC_SR1 register, and then configure the I2S interface clock source.

2. When I2S acts as a slave, the clock source must be configured to external IOM_I2S_CLK; when I2S acts as a master, the clock source is selected according to functional requirements.

Bits 1-0 QSPI_CLK_SEL: QSPI interface clock source selection

- 0: HCLK
- 1: SYS_CLK
- >1: PLL

Note: *Make sure to configure this bit when QSPI_CLK_EN=0. If the QSPI_CLK_EN bit is enabled, the user must disable it by software first, wait for at least 2 current clock cycles or query the RCC_SR1 register, and then configure the QSPI interface clock source.*

8.3.4 RCC_CGR0

Offset: 0x00C

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
PWR_CLK_EN	DMAC0_CLK_EN	DMAC1_CLK_EN	CRC_CLK_EN	BASICTIM0_CLK_EN	BASICTIM1_CLK_EN	IOM0_CLK_EN	IOM1_CLK_EN
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
IOM2_CLK_EN	IOM3_CLK_EN	SYSCFG_CLK_EN	UART0_CLK_EN	UART1_CLK_EN	UART2_CLK_EN	UART3_CLK_EN	LPUART_CLK_EN
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SSP0_CLK_EN	SSP1_CLK_EN	SSP2_CLK_EN	I2C0_CLK_EN	I2C1_CLK_EN	I2C2_CLK_EN	RESERVED	ADCCTRL_CLK_EN
r/w	r/w	r/w	r/w	r/w	r/w	r	r/w
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AFEC_CLK_EN	LCDCTRL_CLK_EN	DACCTRL_CLK_EN	LORAC_CLK_EN	GPTIM0_CLK_EN	GPTIM1_CLK_EN	GPTIM2_CLK_EN	GPTIM3_CLK_EN
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bit 31 PWR_CLK_EN: PWR clock enable.

- 0: disabled
- 1: enabled

Bit 30 DMAC0_CLK_EN: DMAC0 clock enable.

- 0: disabled
- 1: enabled

Bit 29 DMAC1_CLK_EN: DMAC1 clock enable.

- 0: disabled
- 1: enabled

Bit 28 CRC_CLK_EN: CRC clock enable.

- 0: disabled
- 1: enabled

Bit 27 BASICTIM0_CLK_EN: BASICTIM0 clock enable.

- 0: disabled
- 1: enabled

Bit 26 BASICTIM1_CLK_EN: BASICTIM1 clock enable

- 0: disabled
- 1: enabled

Bit 25 IOM0_CLK_EN: IOM0 clock enable.

- 0: disabled
- 1: enabled

Bit 24 IOM1_CLK_EN: IOM1 clock enable.

- 0: disabled
- 1: enabled

Bit 23 IOM2_CLK_EN: IOM2 clock enable.

- 0: disabled
- 1: enabled

Bit 22 IOM3_CLK_EN: IOM3 clock enable.

- 0: disabled
- 1: enabled

Bit 21 SYSCFG_CLK_EN: SYSCFG clock enable.

- 0: disabled
- 1: enabled

Bit 20 UART0_CLK_EN: UART0 clock enable.

- 0: disabled
- 1: enabled

Bit 19 UART1_CLK_EN: UART1 clock enable.

- 0: disabled
- 1: enabled

Bit 18 UART2_CLK_EN: UART2 clock enable.

- 0: disabled
- 1: enabled

Bit 17 UART3_CLK_EN: UART3 clock enable.

- 0: disabled
- 1: enabled

Bit 16 LPUART_CLK_EN: LPUART clock enable.

- 0: disabled
- 1: enabled

Bit 15 SSP0_CLK_EN: SSP0 clock enable.

- 0: disabled
- 1: enabled

Bit 14 SSP1_CLK_EN: SSP1 clock enable.

- 0: disabled
- 1: enabled

Bit 13 SSP2_CLK_EN: SSP2 clock enable.

- 0: disabled
- 1: enabled

Bit 12 I2C0_CLK_EN: I2C0 clock enable.

- 0: disabled
- 1: enabled

Bit 11 I2C1_CLK_EN: I2C1 clock enable.

- 0: disabled
- 1: enabled

Bit 10 I2C2_CLK_EN: I2C2 clock enable.

- 0: disabled
- 1: enable

Bit 9 RESERVED: Must be kept, and cannot be modified.

Bit 8 ADCCTRL_CLK_EN: ADCCTRL clock enable.

- 0: disabled
- 1: enabled

Bit 7 AFEC_CLK_EN: AFEC clock enable.

- 0: disabled
- 1: enabled

Bit 6 LCDCTRL_CLK_EN: LCDCTRL clock enable.

- 0: disabled
- 1: enabled

Bit 5 DACCTRL_CLK_EN: DACCTRL clock enable.

- 0: disabled
- 1: enabled

Bit 4 LORAC_CLK_EN: LORAC clock enable.

- 0: disabled
- 1: enabled

Bit 3 GPTIM0_CLK_EN: GPTIM0 clock enable.

- 0: disabled
- 1: enabled

Bit 2 GPTIM1_CLK_EN: GPTIM1 clock enable.

- 0: disabled
- 1: enabled

Bit 1 GPTIM2_CLK_EN: GPTIM2 clock enable.

- 0: disabled
- 1: enabled

Bit 0 GPTIM3_CLK_EN: GPTIM3 clock enable.

- 0: disabled
- 1: enabled

8.3.5 RCC_CGR1

Offset: 0x010

Reset Value: 0x00000000

31-13	12	11	10	9	8	7
RESERVED	LPTIM1_INF_CLK_EN	LPTIM1_CLK_EN	RNGC_CLK_EN	LPTIM0_INF_CLK_EN	I2S_CLK_EN	SAC_CLK_EN
r	r/w	r/w	r/w	r/w	r/w	r/w
6	**5**	**4**	**3**	**2**	**1**	**0**
WWDG_CNT_CLK_EN	QSPI_CLK_EN	LPTIM0_CLK_EN	IWDG_CLK_EN	WWDG_CLK_EN	RTC_CLK_EN	SEC_CLK_EN
r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bits 31-13 RESERVED: Must be kept, and cannot be modified.

Bit 12 LPTIM1_INF_CLK_EN: LPTIM1 interface PCLK0 clock enable.

- 0: disabled
- 1: enabled

Bit 11 LPTIM1_CLK_EN: LPTIM1 clock enable.

- 0: disabled
- 1: enabled

Note: *If PCLK0 is selected as the clock source, the LPTIM1_INF_CLK_EN bit must be enabled before enabling the LPTIM1 clock, while it must be disabled after the LPTIM1 clock is disabled.*

Bit 10 RNGC_CLK_EN: RNGC clock enable.

- 0: disabled
- 1: enabled

Bit 9 LPTIM0_INF_CLK_EN: LPTIM0 interface PCLK0 clock enable.

- 0: disabled
- 1: enabled

Bit 8 I2S_CLK_EN: I2S clock enable.

- 0: disabled
- 1: enabled

Bit 7 SAC_CLK_EN: SAC clock enable.

- 0: disabled
- 1: enabled

Bit 6 WWDG_CNT_CLK_EN: WWDG counter clock enable.

- 0: disabled
- 1: enabled

Bit 5 QSPI_CLK_EN: QSPI clock enable.

- 0: disabled
- 1: enabled

Bit 4 LPTIM0_CLK_EN: LPTIM0 clock enable.

- 0: disabled
- 1: enabled

Note: If PCLK0 is selected as the clock source, the LPTIM0_INF_CLK_EN bit must be enabled before enabling the LPTIM0 clock, while it must be disabled after the LPTIM0 clock is disabled.

Bit 3 IWDG_CLK_EN: IWDG clock enable.

- 0: disabled
- 1: enabled

Bit 2 WWDG_CLK_EN: WWDG clock enable.

- 0: disabled
- 1: enabled

Bit 1 RTC_CLK_EN: RTC clock enable.

- 0: disabled
- 1: enabled

Bit 0 SEC_CLK_EN: SEC clock enable.

- 0: disabled
- 1: enabled

8.3.6 RCC_CGR2

Offset:0x014

Reset Value: 0x00000000

This register is in the AON power domain. Read the *RCC_SR* register before configuring this register. When the corresponding bit is set in the *RCC_SR* register, this register can be read; when all the bits are set in the *RCC_SR* register, this register can be written.

31-6	5	4	3	2	1	0
RESERVED	LPTIM1_AON_CLK_EN	LPTIM_AON_CLK_EN	LCDCTRL_AON_CLK_EN	LPUART_AON_CLK_EN	RTC_AON_CLK_EN	IWDG_AON_CLK_EN
r	r/w	r/w	r/w	r/w	r/w	r/w

Bits 31-6 RESERVED: Must be kept, and cannot be modified.

Bit 5 LPTIM1_AON_CLK_EN: Enable the LPTIM1 interface clock in AON domain.

- 0: disabled
- 1: enabled

Bit 4 LPTIM_AON_CLK_EN: Enable the LPTIM interface clock in AON domain.

- 0: disabled
- 1: enabled

Bit 3 LCDCTRL_AON_CLK_EN: Enable the LCDCTRL interface clock in AON domain.

- 0: disabled
- 1: enabled

Bit 2 LPUART_AON_CLK_EN: Enable the LPUART interface clock in AON domain.

- 0: disabled
- 1: enabled

Bit 1 RTC_AON_CLK_EN: Enable the RTC interface clock in AON domain.

- 0: disabled
- 1: enabled

Bit 0 IWDG_AON_CLK_EN: Enable the IWDG interface clock in AON domain.

- 0: disabled
- 1: enabled

8.3.7 RCC_RST0

Offset: 0x018

Reset Value: 0xffffffff

31	30	29	28	27	26	25	24
UART0_RST_N	UART1_RST_N	UART2_RST_N	UART3_RST_N	LPUART_RST_N	SSP0_RST_N	SSP1_RST_N	SSP2_RST_N
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
QSPI_RST_N	I2C0_RST_N	I2C1_RST_N	I2C2_RST_N	RESERVED	ADCCTRL_RST_N	AFEC_RST_N	LCDCTRL_RST_N
r/w	r/w	r/w	r/w	r	r/w	r/w	r/w
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DACCTRL_RST_N	LORAC_RST_N	IOM_RST_N	GPTIM0_RST_N	GPTIM1_RST_N	GPTIM2_RST_N	GPTIM3_RST_N	BASICTIM0_RST_N
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BASICTIM1_RST_N	LPTIM_RST_N	IWDG_RST_N	WWDG_RST_N	RTC_RST_N	CRC_RST_N	SEC_RST_N	SAC_RST_N
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bit 31 UART0_RST_N: UART0 reset control.

- 0: reset
- 1: no action

Bit 30 UART1_RST_N: UART1 reset control.

- 0: reset
- 1: no action

Bit 29 UART2_RST_N: UART2 reset control.

- 0: reset
- 1: no action

Bit 28 UART3_RST_N: UART3 reset control.

- 0: reset
- 1: no action

Bit 27 LPUART_RST_N: LPUART reset control.

- 0: reset
- 1: no action

Bit 26 SSP0_RST_N: SSP0 reset control.

- 0: reset
- 1: no action

Bit 25 SSP1_RST_N: SSP1 reset control.

- 0: reset
- 1: no action

Bit 24 SSP2_RST_N: SSP2 reset control.

- 0: reset
- 1: no action

Bit 23 QSPI_RST_N: QSPI reset control.

- 0: reset
- 1: no action

Bit 22 I2C0_RST_N: I2C0 reset control.

- 0: reset
- 1: no action

Bit 21 I2C1_RST_N: I2C1 reset control.

- 0: reset
- 1: no action

Bit 20 I2C2_RST_N: I2C2 reset control.

- 0: reset
- 1: no action

Bit 19 RESERVED: Must be kept, and cannot be modified.

Bit 18 ADCCTRL_RST_N: ADCCTRL reset control.

- 0: reset
- 1: no action

Bit 17 AFEC_RST_N: AFEC reset control.

- 0: reset
- 1: no action

Bit 16 LCDCTRL_RST_N: LCDCTRL reset control.

- 0: reset
- 1: no action

Bit 15 DACCTRL_RST_N: DACCTRL reset control.

- 0: reset

- 1: no action

Bit 14 LORAC_RST_N: LORAC reset control.

- 0: reset
- 1: no action

Bit 13 IOM_RST_N: IOM reset control.

- 0: reset
- 1: no action

Bit 12 GPTIM0_RST_N: GPTIM0 reset control.

- 0: reset
- 1: no action

Bit 11 GPTIM1_RST_N: GPTIM1 reset control.

- 0: reset
- 1: no action

Bit 10 GPTIM2_RST_N: GPTIM2 reset control.

- 0: reset
- 1: no action

Bit 9 GPTIM3_RST_N: GPTIM3 reset control.

- 0: reset
- 1: no action

Bit 8 BASICTIM0_RST_N: BASICTIM0 reset control.

- 0: reset
- 1: no action

Bit 7 BASICTIM1_RST_N: BASICTIM1 reset control.

- 0: reset
- 1: no action

Bit 6 LPTIM0_RST_N: LPTIM0 reset control.

- 0: reset
- 1: no action

Bit 5 IWDG_RST_N: IWDG reset control.

- 0: reset
- 1: no action

Bit 4 WWDG_RST_N: WWDG reset control.

- 0: reset
- 1: no action

Bit 3 RTC_RST_N: RTC reset control.

- 0: reset
- 1: no action

Bit 2 CRC_RST_N: CRC reset control.

- 0: reset

PDF-XChange Editor
Click to BUY NOW!
www.pdf-xchange.com

- 1: no action

Bit 1 SEC_RST_N: SEC reset control.

- 0: reset
- 1: no action

Bit 0 SAC_RST_N: SAC reset control.

- 0: reset
- 1: no action

8.3.8 RCC_RST1

Offset: 0x01C

Reset Value: 0x0000001f

31-5	4	3	2	1	0
RESERVED	LPTIM1_RST_N	RNGC_RST_N	I2S_RST_N	DMAC0_RST_N	DMAC1_RST_N
r	r/w	r/w	r/w	r/w	r/w

Bits 31-5 RESERVED: Must be kept, and cannot be modified.

Bit 4 LPTIM1_RST_N: LPTIM1 reset control.

- 0: reset
- 1: no action

Bit 3 RNGC_RST_N: RNGC reset control.

- 0: reset
- 1: no action

Bit 2 I2S_RST_N: I2S reset control.

- 0: reset
- 1: no action

Bit 1 DMAC0_RST_N: DMAC0 reset control.

- 0: reset
- 1: no action

Bit 0 DMAC1_RST_N: DMAC1 reset control.

- 0: reset
- 1: no action

8.3.9 RCC_RST_SR

Offset: 0x020

Reset Value: 0x00000040

Note: *The BOR_RESET_SR and STANDBY_RESET_SR are in the AON domain.*

31-7	6	5	4	3	2	1	0
RESERVED	BOR_RESET_SR	IWDG_RESET_SR	WWDG_RESET_SR	EFC_RESET_SR	CPU_RESET_SR	SEC_RESET_SR	STANDBY_RESET_SR
r	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bits 31-7 RESERVED: Must be kept, and cannot be modified.

Bit 6 BOR_RESET_SR: BOR reset status. Set by hardware and cleared by software by writing 1.

- 0: no BOR reset
- 1: BOR reset occurred

Bit 5 IWDG_RESET_SR: IWDG reset status. Set by hardware and cleared by software by writing 1.

- 0: no IWDG reset
- 1: IWDG reset occurred

Bit 4 WWDG_RESET_SR: WWDG reset status. Set by hardware and cleared by software by writing 1.

- 0: no WWDG reset
- 1: WWDG reset occurred

Bit 3 EFC_RESET_SR: EFC reset status. Set by hardware and cleared by software by writing 1.

- 0: no EFC reset
- 1: EFC reset occurred

Bit 2 CPU_RESET_SR: CPU reset status. Set by hardware and cleared by software by writing 1.

- 0: no CPU reset
- 1: CPU reset occurred

Bit 1 SEC_RESET_SR: SEC reset status. Set by hardware and cleared by software by writing 1.

- 0: no SEC reset
- 1: SEC reset occurred

Bit 0 STANDBY_RESET_SR: Standby reset status. Set by hardware and cleared by software by writing 1.

- 0: no MPU reset
- 1: MPU reset occurred

8.3.10 RCC_RST_CR

Offset: 0x024

Reset Value: 0x00000004

31-6	5	4	3	2	1	0
RESERVED	IWDG_RESET_REQ_EN	WWDG_RESET_REQ_EN	EFC_RESET_REQ_EN	CPU_RESET_REQ_EN	SEC_RESET_REQ_EN	RESERVED
r	r/w	r/w	r/w	r/w	r/w	r

Bits 31-6 RESERVED: Must be kept, and cannot be modified.

Bit 5 IWDG_RESET_REQ_EN: IWDG reset enable.

- 0: disabled
- 1: enabled

Bit 4 WWDG_RESET_REQ_EN: WWDG reset enable.

- 0: disabled
- 1: enabled

Bit 3 EFC_RESET_REQ_EN: EFC reset enable.

- 0: disabled
- 1: enabled

Bit 2 CPU_RESET_REQ_EN: CPU reset enable.

- 0: disabled
- 1: enabled

Bit 1 SEC_RESET_REQ_EN: SEC reset enable.

- 0: disabled
- 1: enabled

Bit 0 RESERVED: Must be kept, and cannot be modified.

8.3.11 RCC_SR

Offset: 0x028

Reset Value: 0x0000003f

31-6		5	4
RESERVED		SET_LPTIM1_AON_CLK_EN_DONE	SET_LPTIM_AON_CLK_EN_DONE
r		r	r
3	**2**	**1**	**0**
SET_LCDCTRL_AON_CLK_EN_DONE	SET_LPUART_AON_CLK_EN_DONE	SET_RTC_AON_CLK_EN_DONE	SET_IWDG_AON_CLK_EN_DONE
r	r	r	r

Bits 31-6 RESERVED: Must be kept, and cannot be modified.

Bit 5 SET_LPTIM1_AON_CLK_EN_DONE: LPTIM1_AON_CLK_EN configuration status. This bit is set and cleared by hardware.

- 0: configuration in progress
- 1: configuration completed

Bit 4 SET_LPTIM0_AON_CLK_EN_DONE: LPTIM0_AON_CLK_EN configuration status. This bit is set and cleared by hardware.

- 0: configuration in progress
- 1: configuration completed

Bit 3 SET_LCDCTRL_AON_CLK_EN_DONE: LCDCTRL_AON_CLK_EN configuration status. This bit is set and cleared by hardware.

- 0: configuration in progress
- 1: configuration completed

Bit 2 SET_LPUART_AON_CLK_EN_DONE: LPUART_AON_CLK_EN configuration status. This bit is set and cleared by hardware.

- 0: configuration in progress
- 1: configuration completed

Bit 1 SET_RTC_AON_CLK_EN_DONE: RTC_AON_CLK_EN configuration status. This bit is set and cleared by hardware.

- 0: configuration in progress
- 1: configuration completed

Bit 0 SET_IWDG_AON_CLK_EN_DONE: IWDG_AON_CLK_EN configuration status. This bit is set and cleared by hardware.

- 0: configuration in progress
- 1: configuration completed

8.3.12 RCC_SR1

Offset: 0x02C

Reset Value: 0x00000000

The clock should be disabled before the clock source is switched or the frequency division changes to avoid glitches. This register is used to determine enable status of the clock.

31-21	20	19	18	17	16
RESERVED	LPTIM1_CLK_EN_SYNC	LPTIM1_AON_CLK_EN_SYNC	UART0_CLK_EN_SYNC	UART1_CLK_EN_SYNC	UART2_CLK_EN_SYNC
r	r	r	r	r	r
15	**14**	**13**	**12**	**11**	**10**
UART3_CLK_EN_SYNC	RESERVED	ADCCTRL_CLK_EN_SYNC	LPTIM_CLK_EN_SYNC	QSPI_CLK_EN_SYNC	LPUART_CLK_EN_SYNC
r	r	r	r	r	r
9	**8**	**7**	**6**	**5**	**4**
LCDCTRL_CLK_EN_SYNC	IWDG_CLK_EN_SYNC	RTC_CLK_EN_SYNC	MCO_CLK_EN_SYNC	I2S_CLK_EN_SYNC	LPTIM_AON_CLK_EN_SYNC
r	r	r	r	r	r

3	2	1	0
LCDCTRL_AON_CLK_EN_SYNC	LPUART_AON_CLK_EN_SYNC	RTC_AON_CLK_EN_SYNC	IWDG_AON_CLK_EN_SYNC
r	r	r	r

Bits 31-21 RESERVED: Must be kept, and cannot be modified.

Bit 20 LPTIM1_CLK_EN_SYNC: LPTIM1_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 19 LPTIM1_AON_CLK_EN_SYNC: LPTIM1_AON_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 18 UART0_CLK_EN_SYNC: UART0_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 17 UART1_CLK_EN_SYNC: UART1_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 16 UART2_CLK_EN_SYNC: UART2_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 15 UART3_CLK_EN_SYNC: UART3_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 14 RESERVED: Must be kept, and cannot be modified.

Bit 13 ADCCTRL_CLK_EN_SYNC: ADCCTRL_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 12 LPTIM0_CLK_EN_SYNC: LPTIM0_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 11 QSPI_CLK_EN_SYNC: QSPI_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 10 LPUART_CLK_EN_SYNC: Indicate LPUART_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 9 LCDCTRL_CLK_EN_SYNC: LCDCTRL_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 8 IWDG_CLK_EN_SYNC: IWDG_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 7 RTC_CLK_EN_SYNC: RTC_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 6 MCO_CLK_EN_SYNC: MCO_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 5 I2S_CLK_EN_SYNC: I2S_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 4 LPTIM0_AON_CLK_EN_SYNC: LPTIM0_AON_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 3 LCDCTRL_AON_CLK_EN_SYNC: LCDCTRL_AON_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 2 LPUART_AON_CLK_EN_SYNC: LPUART_AON_CLK_EN actual status.

- 0: disabled

- 1: enabled

Bit 1 RTC_AON_CLK_EN_SYNC: Indicate RTC_AON_CLK_EN actual status.

- 0: disabled
- 1: enabled

Bit 0 IWDG_AON_CLK_EN_SYNC: IWDG_AON_CLK_EN actual status.

- 0: disabled
- 1: enabled

8.3.13 RCC_CR3

Offset: 0x030

Reset Value: 0x00000000

31-16	15-8	7-0
RESERVED	I2S_MCLK_DIV	I2S_SCLK_DIV
r	r/w	r/w

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-8 I2S_MCLK_DIV: I2S interface clock MCLK frequency division.

- 0: not divided
- 0: not divided
- 2: divided by 2
- 3: divided by 3
- N: divided by N

Notes:

1. Make sure to configure I2S_MCLK_DIV when I2S_CLK_EN=0. If the I2S_CLK_EN bit is enabled, the user must disable it by software first, wait for at least 2 current clock cycles or query the RCC_SR1 register, and then configure I2S_MCLK_DIV.

2. When I2S acts as a slave, this bit must be configured to 0 or 1; when I2S acts as a master, this bit is configured according to functional requirements.

3. The duty cycle of the output clock is 50%.

Bits 7-0 I2S_SCLK_DIV: I2S interface clock SCLK frequency division.

- 0: not divided
- 0: not divided
- 2: divided by 2
- 3: divided by 3
- N: divided by N

Notes:

1. Make sure to configure I2S_SCLK_DIV when I2S_CLK_EN=0. If the I2S_CLK_EN bit is enabled, the user must disable it by software first, wait for at least 2 current clock cycles or query the RCC_SR1 register, and then configure I2S_SCLK_DIV.

2. When I2S acts as a slave, this bit must be configured to 0 or 1; when I2S acts as a master, this bit is configured according to functional requirements.

3. The duty cycle of the output clock is 50%.

9. Interrupts

9.1 Main Features

- Support 37 IRQ interrupts.
- Configurable 0~7 priority levels for each IRQ interrupt.

9.2 SysTick

SysTick calibration value is 0x147. Using a 32.768 kHz clock source for SysTick counting gives a reference time base of 10 ms.

9.3 Interrupt Vector Table

The interrupt vector Table is as follows:

Table 9-1 Interrupt Vectors

Position	Priority	Type of priority	Acronym	Description	Address
	-	-	-	Reserved	0x0000_0000
	-3	fixed	Reset	Reset	0x0000_0004
	-2	fixed	NMI_Handler	Secure area check error	0x0000_0008
	-1	fixed	HardFault_Handler	fault	0x0000_000C
	0	settable	MemManage Handler	fault	0x0000_0010
	1	settable	BusFault Handler	fault	0x0000_0014
	2	settable	UsageFault Handler	fault	0x0000_0018
	-	-	-	Reserved	0x0000_001C - 0x0000_002B
	3	settable	SVC_Handler	System service call via SWI instruction	0x0000_002C
	-	-	-	Reserved	0x0000_0030 - 0x0000_0037
	5	settable	PendSV_Handler	Pendable request for system service	0x0000_0038
	6	settable	SysTick_Handler	System tick timer	0x0000_003C
0	7	settable	sec	Include mpu	0x0000_0040
1	8	settable	rtc	Include tamper io, cyc, wakeup io	0x0000_0044

Position	Priority	Type of priority	Acronym	Description	Address
2	9	settable	wwdg		0x0000_0048
3	10	settable	efc		0x0000_004C
4	11	settable	uart3		0x0000_0050
5	12	settable	i2c2		0x0000_0054
6	13	settable	uart0		0x0000_0058
7	14	settable	uart1		0x0000_005C
8	15	settable	uart2		0x0000_0060
9	16	settable	lpuart		0x0000_0064
10	17	settable	ssp0		0x0000_0068
11	18	settable	ssp1		0x0000_006C
12	19	settable	qspi		0x0000_0070
13	20	settable	i2c0		0x0000_0074
14	21	settable	i2c1		0x0000_0078
15	22	settable	-		0x0000_007C
16	23	settable	adcctrl		0x0000_0080
17	24	settable	afec		0x0000_0084
18	25	settable	ssp2		0x0000_0088
19	26	settable	dmac1		0x0000_008C
20	27	settable	dacctrl		0x0000_0090
21	28	settable	lorac		0x0000_0094
22	29	settable	iom		0x0000_0098
23	30	settable	gptim0		0x0000_009C
24	31	settable	gptim1		0x0000_00A0
25	32	settable	gptim2		0x0000_00A4
26	33	settable	gptim3		0x0000_00A8
27	34	settable	basictim0		0x0000_00AC
28	35	settable	basictim1		0x0000_00B0
29	36	settable	lptim0		0x0000_00B4
30	37	settable	sac		0x0000_00B8
31	38	settable	dmac0		0x0000_00BC
32	39	settable	i2s		0x0000_00C0
33	40	settable	lcdctrl		0x0000_00C4
34	41	settable	pwr		0x0000_00C8
35	42	settable	lptim1		0x0000_00CC
36	43	settable	iwdg		0x0000_00D0

10. Embedded Flash

10.1 Introduction

- The whole Flash is divided into Flash info area and Flash main area
- Flash size:
  - Flash info area: 16 KB
  - Flash main area: 256 KB for ASR6601SE, 128 KB for ASR6601CB
- ● Page erase (4 KB) and Mass erase (all main flash area)

10.2 Main Features

- Flash operations include read, program, page erase and mass erase
- Read access latency
- Accessing acceleration
- Instruction prefetch, buffer deep 1
- Flash program operation supports single and continuous modes
- Option bytes in Flash info area
- Interrupt signals generation

10.3 Functional Description

10.3.1 Flash Info Area Division

The Flash info area is divided into four parts: Option Bytes, Factory Bytes, OTP and BootLoader. See the table below for details.

Table 10-1 Flash Info Area Division

Start Addres	Description	Size
0x10003000	Option Bytes	4KB
0x10002000	Factory Bytes	4KB
0x10001C00	OTP	1KB
0x10000000	BootLoader	7KB

10.3.2 EFC_CR Protection

By default, the EFC_CR register cannot be modified, to modify it, the user must configure the protection sequence correctly through the *EFC_PROTECT_SEQ* register in the following order. If there is an error in the configuration, then the configuration is invalid, and the protection sequence should be reconfigured.

(1) First write “0x8C9DAEBF” to EFC_PROTECT_SEQ register

(2) Then write “0x13141516” to EFC_PROTECT_SEQ register

10.3.3 Read Access Latency

In order to improve Flash read performance, the number of wait states (READ_NUM[19:16]) should be correctly programmed in *EFC_TIMING_CFG* register according to the frequency of SYS_CLK. The number of wait states (READ_NUM) equals to (READ_NUM+1) multiplied by SYS_CLK clock period. See following details:

- For 48MHz SYS_CLK frequency READ_NUM must ≥ 2.
- For 32MHz SYS_CLK frequency READ_NUM must ≥ 1.
- For 24MHz SYS_CLK frequency READ_NUM must ≥ 1.
- For 3.6MHz SYS_CLK frequency READ_NUM must ≥ 0.
- For 32kHz SYS_CLK frequency READ_NUM must ≥ 0.

Operations to switch to a high-frequency clock source for SYS_CLK:

(1) Modify the READ_NUM value in *EFC_TIMING_CFG* register to match the SYS_CLK after its clock source is switched.

(2) Wait for the READ_NUM_DONE status bit in *EFC_SR* register to be set.

(3) Modify the SYS_CLK_SEL field in *RCC_CR0* register to switch to the target clock source.

Operations to switch to a low-frequency clock source for SYS_CLK:

(1) Modify the SYS_CLK_SEL field in *RCC_CR0* register to switch to the target clock source.

(2) Modify the READ_NUM value in *EFC_TIMING_CFG* register to match the SYS_CLK after its clock source is switched.

(3) Wait for the READ_NUM_DONE status bit in *EFC_SR* register to be set.

Note: *When the user wants to switch to a high-frequency clock source, first increase the READ_NUM, and then configure the clock source selection bit; otherwise, first configure the clock source selection bit, and then decrease the READ_NUM.*

10.3.4 Accessing acceleration

Read acceleration is disabled by default. If READ_NUM < (2^HCLK_DIV), read acceleration can be enabled to achieve the maximum bus access efficiency. Note that read acceleration must be enabled after READ_NUM and HCLK_DIV are configured.

Note: *Read acceleration and instruction prefetch can’t be enabled at the same time.*

10.3.5 Instruction Prefetch

It is disabled by default. If READ_NUM ≥ (2^HCLK_DIV), read acceleration cannot be enabled. You can choose to enable instruction prefetch to improve access efficiency.

Note: *Read acceleration and instruction prefetch can't be enabled at the same time.*

10.3.6 Flash Program

There are two modes for Flash programming:

- **Single Programming Mode**

  In single mode, it programs 2 words (8 Bytes) at one time.

- **Continuous Programming Mode**

  In continuous mode, it programs a complete word line (512 Bytes) each time.

  During continuous programming, Flash cannot be read or executed, so the continuous programming code must be executed in RAM.

Steps for single programming:

(1) Set the PROG_EN bit in register *EFC_CR*.

(2) Write the low 4 Bytes data into register *EFC_PROG_DATA0*.

(3) Write the high 4 Bytes data into register *EFC_PROG_DATA1*.

(4) Write any value to the Flash address to be written into.

(5) Wait for the OPERATION_DONE bit in register *EFC_SR* to be set.

(6) Write 1 to the OPERATION_DONE bit in register *EFC_SR* to clear the flag.

Steps for continuous programming:

(1) Set the PROG_EN, WRITE_RELEASE_EN and PROG_MODE bits in register *EFC_CR*.

(2) Wait for the PROG_DATA_WAIT bit in register *EFC_SR* to be set.

(3) Write the low 4 Bytes data into register *EFC_PROG_DATA0*.

(4) Write the high 4 Bytes data into register *EFC_PROG_DATA1*.

(5) Write any value to the Flash address to be written into.

(6) Wait for the PROG_DATA_WAIT bit in register *EFC_SR* to be set.

(7) Continue to write data to the *EFC_PROG_DATA0* and *EFC_PROG_DATA1 registers*.

(8) Repeat **Step 6** and **Step 7** until 512 Bytes are written.

(9) Wait for the OPERATION_DONE bit in register *EFC_SR* to be set.

(10) Write 1 to the OPERATION_DONE bit in register *EFC_SR* to clear the flag.

10.3.7 Flash Erase

The Flash memory erase operation can be performed at page level (page erase) or on the whole memory (mass erase).

- **Page Erase**

  The page erase is measured in 4 Bytes.

- **Mass Erase**

  After a mass erase, the entire Flash main area will be 0xFF.

Steps for page erase:

(1) Set the PAGE_ERASE_EN bit in register *EFC_CR*.

(2) Write any value to the Flash address to be erased.

(3) Wait for the OPERATION_DONE bit in register *EFC_SR* to be set.

(4) Write 1 to the OPERATION_DONE bit in register *EFC_SR* to clear the flag.

Steps for mass erase:

(1) Set the MASS_ERASE_EN bit in register *EFC_CR*.

(2) Write any value to the Flash address 0x08000000.

(3) Wait for the OPERATION_DONE bit in register *EFC_SR* to be set.

(4) Write 1 to the OPERATION_DONE bit in register *EFC_SR* to clear the flag.

10.4 Flash Option Bytes

Flash option bytes is divided into option0 and option1.

10.4.1 Flash Option0

Option0 has 64 bits in total, and its format is as follows:

Table 10-2 Flash Option0

63-50	49-44	43-38	37-32	31-26	25	24-19
RESERVED	WR_PROTECT_END	WR_PROTECT_START	EXE_ONLY2_END	EXE_ONLY2_START	EXE_ONLY_KEEEP	EXE_ONLY1_END
18-13	**12-5**	**4**	**3**	**2**	**1**	**0**
EXE_ONLY1_START	DEBUG_LEVEL	RESERVED	SYS_SRAM_RESET	FLASH_BOOT1	USE_FLASH_BOOT0	FLASH_BOOT0

Bits 63-50 RESERVED: Must be kept, and cannot be modified.

Bits 49-44 WR_PROTECT_END: Write-protected area end.

When *WR_PROTECT_START* > *WR_PROTECT_END*, the write-protected area is disabled. It is disabled by default.

Bits 43-38 WR_PROTECT_START: Write-protected area start.

When *WR_PROTECT_START* > *WR_PROTECT_END*, the write-protected area is disabled. It is disabled by default.

Bits 37-32 EXE_ONLY2_END: Exe_Only2 area end.

When *EXE_ONLY2_START* > *EXE_ONLY2_END*, the Exe_Only2 area is disabled. It is disabled by default. Once enabled, this area can only be expanded but can't be disabled or narrowed.

Bits 31-26 EXE_ONLY2_START: Exe_Only2 area start.

When *EXE_ONLY2_START* > *EXE_ONLY2_END*, the Exe_Only2 area is disabled. It is disabled by default. Once enabled, this area can only be expanded but can't be disabled or narrowed.

Bit 25 EXE_ONLY_KEEP: Whether Exe_Only area is kept when the Debug_Level changes from 1 to 0.

- 0: not keep Exe_Only area
- 1: keep the Exe_Only area

This bit can only be set to 0 by software. When Debug_Level changes from 1 to 0, EXE_ONLY_KEEP is set to 1 automatically by hardware.

Bits 24-19 EXE_ONLY1_END: Exe_Only1 area end.

When *EXE_ONLY1_START* > *EXE_ONLY1_END*, the Exe_Only1 area is disabled. It is disabled by default. Once enabled, this area can only be expanded but can't be disabled or narrowed.

Bits 18-13 EXE_ONLY1_START: Exe_Only1 area start.

When *EXE_ONLY1_START* > *EXE_ONLY1_END*, the Exe_Only1 area is disabled. It is disabled by default. Once enabled, this area can only be expanded but can't be disabled or narrowed.

Bits 12-5 DEBUG_LEVEL: Debug_level configuration.

- AA: Level 0
- CC: Level 2
- Others: Level 1

Bit 4 RESERVED: Must be kept, and cannot be modified.

Bit 3 SYS_SRAM_RESET: Whether to clear system SRAM during system startup after its reset

- 1: clear system SRAM
- 0: not clear system SRAM

Bit 2 FLASH_BOOT1: This bit can be used to identify the boot mode.

Bit 1 USE_FLASH_BOOT0: This bit can be used to identify the boot mode.

Bit 0 FLASH_BOOT0: This bit can be used to identify the boot mode.

See table below for the boot mode configuration summary:

Table 10-3 ASR6601 Boot Mode Configuration

DEBUG_ LEVEL	USE_FLASH_ BOOT0	FLASH_ BOOT0	BOOT0 PIN	FLASH_ BOOT1	MAIN_FLASH_ EMPTY	Boot Config
2	X	X	X	X	X	Boot from Flash Main
<2	0	X	0	X	0	Boot from Flash Main
<2	0	X	0	X	1	Boot from Flash Bootloader
<2	0	X	1	1	X	Boot from Flash Bootloader
<2	0	X	1	0	X	Boot from System SRAM
<2	1	1	X	X	0	Boot from Flash Main
<2	1	1	X	X	1	Boot from Flash Bootloader
<2	1	0	X	1	X	Boot from Flash Bootloader
<2	1	0	X	0	X	Boot from System SRAM

10.4.2 Flash Option1

Option1 has 64 bits in total, and its format is as follows:

Table 10-4 Flash Option1

63-56	55	54-49	48	47-42	41-37
RESERVED	SYSRAM_HIDE_EN	SYSRAM_HIDE_START	FLASH_HIDE_EN	FLASH_HIDE_START	RETRAM_SECURE_END
36-32	**31-24**	**23-18**	**17-12**	**11-6**	**5-0**
RETRAM_SECURE_START	RESERVED	SYSRAM_SECURE_END	SYSRAM_SECURE_START	FLASH_SECURE_END	FLASH_SECURE_START

Bits 63-56 RESERVED: Must be kept, and cannot be modified.

Bit 55 SYSRAM_HIDE_EN: SysRamHide area enable control.

- 0: SysRamHide area enabled
- 1: SysRamHide area disabled

Only valid if FlashSecure area is enabled.

Bits 54-49 SYSRAM_HIDE_START: SysRamHide area start.

The configuration is only valid when the SysRamHide area is within the SysRamSecure area and the FlashSecure area is enabled by bits[11:0]. The SysRamHide area is from SysRamHideStart to SysRamSecureEnd.

Bit 48 FLASH_HIDE_EN: FlashHide area enable control.

- 0: FlashHide area enabled
- 1: FlashHide area disabled

Only valid if FlashSecure area is enabled.

Bits 47-42 FLASH_HIDE_START: FlashHide area start.

The configuration is only valid when the FlashHide area is within the FlashSecure area and the FlashSecure area is enabled by bits[11:0]. The FlashHide area is from FlashHideStart to FlashSecureEnd.

Bits 41-37 RETRAM_SECURE_END: RetRam Secure area end.

When *RETRAM_SECURE_START* > *RETRAM_SECURE_END*, the RetRam Secure area is disabled. The configuration is only valid when the FlashSecure area is enabled by bits[11:0].

Bits 36-32 RETRAM_SECURE_START: RetRam Secure area start.

When *RETRAM_SECURE_START* > *RETRAM_SECURE_END*, the RetRam Secure area is disabled. The configuration is only valid when the FlashSecure area is enabled by bits[11:0].

Bits 31-24 RESERVED: Must be kept, and cannot be modified.

Bits 23-18 SYSRAM_SECURE_END: SysRam Secure area end.

When *SYSRAM_SECURE_START* > *SYSRAM_SECURE_END*, the SysRam Secure area is disabled. The configuration is only valid when the FlashSecure area is enabled by bits[11:0].

Bits 17-12 SYSRAM_SECURE_START: SysRam Secure area start.

When *SYSRAM_SECURE_START* > *SYSRAM_SECURE_END*, the SysRam Secure area is disabled. The configuration is only valid when the FlashSecure area is enabled by bits[11:0].

Bits 11-6 FLASH_SECURE_END: Flash Secure area end.

When *FLASH _SECURE_START > FLASH _SECURE_END*, the Flash Secure area is disabled. The Flash Secure area enable is the master switch for enabling other secure areas. When the Flash Secure area is disabled, the erase operation is triggered.

Bits 5-0 FLASH_SECURE_START: Flash Secure area start.

When *FLASH_SECURE_START > FLASH _SECURE_END*, the Flash Secure area is disabled. The Flash Secure area enable is the master switch for enabling other secure areas. When the Flash Secure area is disabled, the erase operation is triggered.

10.5 Embedded Flash Registers

Base Address: 0x40020000

Table 10-5 Embedded Flash Registers Summary

Register	Offset	Description
EFC_CR	0x00	Control Register
EFC_INT_EN	0x04	Interrupt enable register
EFC_SR	0x08	Status Register
EFC_PROG_DATA0	0x0C	Program 编程数据 0
EFC_PROG_DATA1	0x10	Program 编程数据 1
EFC_TIMING_CFG	0x14	时钟配置寄存器
EFC_PROTECT_SEQ	0x18	保护序列
RESERVED	0x1C-0x28	保留
SERIAL_NUM_LOW	0x2C	芯片序列号低 32 位
SERIAL_NUM_HIGH	0x30	芯片序列号高 32 位
RESERVED	0x34-0x38	保留
OPTION_CSR_BYTES	0x3C	OPTION 控制及状态数据
OPTION_EXE_ONLY_BYTES	0x40	OPTION 只执行数据
OPTION_WR_PROTECT_BYTES	0x44	OPTION 写保护数据
OPTION_SECURE_BYTES0	0x48	OPTION 安全数据 0
OPTION_SECURE_BYTES1	0x4C	OPTION 安全数据 1

10.5.1 EFC_CR

Offset: 0x00

Reset Value: 0x00000000

31	30-10	9	8	7	6
INFO_BYTE_LOAD	RESERVED	ECC_DIS	OPTION_OPR_EN	RESERVED	WRITE_RELEASE_EN
w	r	r/w	r/w	r	r/w
5	**4**	**3**	**2**	**1**	**0**
PREFETCH_EN	READ_ACC_EN	PROG_MODE	PROG_EN	PAGE_ERASE_EN	MASS_ERASE_EN
r/w	r/w	r/w	r/w	r/w	r/w

Bit 31 INFO_BYTE_LOAD: Info byte load reset request.

- Write 0: no action
- Write 1: system will reset, and reload the information in the Flash info area, such as options. This bit is automatically cleared by hardware.

Bits 30-10 RESERVED: Must be kept, and cannot be modified.

Bit 9 ECC_DIS: ECC encoding disable.

Bit 8 OPTION_OPR_EN: Option operation enable.

- 0: Option operation disabled
- 1: Option operation enabled

Notes:

1. Any two of OPTION_OPR_EN, PROG_EN and PAGE_ERASE_EN cannot be enabled at the same time.

2. After each option operation is performed, the system should be reset for the configuration to take effect.

Bit 7 RESERVED: Must be kept, and cannot be modified.

Bit 6 WRITE_RELEASE_EN: When the system executes Flash program, erase (including Mass) and option operations, the AHB bus mode should be selected.

- 0: hold mode
- 1: release mode

Note: *Once configured in the release mode, the Flash cannot be read or executed during programming/erasing operation, otherwise, the FLASHBUSY_ERR error flag will be set. But you can access the EFC_SR register and wait the operation to be completed.*

Bit 5 PREFETCH_EN: Flash instruction prefetch enable.

- 0: prefetch disabled
- 1: prefetch enabled

Note: *Read acceleration and instruction prefetch can't be enabled at the same time.*

Bit 4 READ_ACC_EN: Flash read acceleration enable.

- 0: read acceleration disabled (hold mode)
- 1: read acceleration enabled (release mode)

PDF-XChange Editor Click to BUY NOW! www.pdf-xchange.com

Notes:

1. When READ_NUM < (2^HCLK_DIV), the read acceleration can be enabled. And it must be enabled after READ_NUM and HCLK_DIV configurations are completed.

2. Read acceleration and instruction prefetch can't be enabled at the same time.

Bit 3 PROG_MODE: flash program mode selection.

● 0: single programming mode. In this mode, the data in the *EFC_PROG_DATA1* and *EFC_PROG_DATA0* registers are written to the specified address in each program.

● 1: WL continuous programming mode. In this mode, a word line (512 Bytes) is programmed to the continuous address of the Flash memory automatically. During the procedure, the software checks the PROG_DATA_WAIT flag to determine whether to write new data into the *EFC_PROG_DATA1* and *EFC_PROG_DATA0* registers.

Notes:

1. The ECC encoding format in Flash is 64+8, so an even number of words are programmed each time.

2. In WL continuous programming mode, the WRITE_RELEASE_EN bit should be set to 1. During the programming process, only the EFC_SR, EFC_PROG_DATA1 and EFC_PROG_DATA0 registers can be read or written, the Flash cannot be read or executed.

Bit 2 PROG_EN: Flash programming enable.

- 0: write to the Flash memory does not trigger Flash programming operation
- 1: write to the Flash memory triggers Flash programming operation

Notes:

1. In single programming mode, the programming is started by writing data to the 8-Byte aligned Flash address. The data of register EFC_PROG_DATA0 will be written into the low 4-Byte address space, and the data of register EFC_PROG_DATA1 will be written into the high 4-Byte address space.

2. In WL continuous programming mode, programming is started by writing data to the Flash address, and the programming address is accumulated by 8 Bytes until the end of a WL programming.

Bit 1 PAGE_ERASE_EN: Flash page erasing enable.

- 0: write to the Flash memory does not trigger Flash page erasing operation
- 1: write to the Flash memory triggers Flash page erasing operation

Bit 0 MASS_ERASE_EN: Flash mass erasing enable

- 0: write to the Flash memory does not trigger Flash mass erasing operation
- 1: write to the Flash memory triggers Flash mass erasing operation

Notes:

1. When the bit is set, if there is a write to the address belonging to the Flash main area, mass erase is only performed on the main area; if there is a write to the address belonging to the Flash info area, mass erase is performed on both the main and info areas.

2. Do not perform mass erase on the Flash info area, otherwise the chip will be destroyed.

10.5.2 EFC_INT_EN

Offset: 0x04

Reset Value: 0x00000000

31-9	8	7	6	5
RESERVED	TWO_BIT_ERROR_INT_EN	ONE_BIT_CORRECT_INT_EN	PROG_ERR_INT_EN	PAGE_ERASE_ERR_INT_EN
r	r/w	r/w	r/w	r/w
4	**3**	**2**	**1**	**0**
OPTION_WR_ERR_INT_EN	FLASHBUSY_ERR_INT_EN	PROG_DATA_WAIT_INT_EN	RESERVED	OPERATION_DONE_INT_EN
r/w	r/w	r/w	r	r/w

Bits 31-9 RESERVED: Must be kept, and cannot be modified.

Bit 8 TWO_BIT_ERROR_INT_EN: ECC TWO_BIT_ERROR interrupt enable.

- 0: disabled
- 1: enabled

Bit 7 ONE_BIT_CORRECT_INT_EN: ECC ONE_BIT_CORRECT interrupt enable.

- 0: disabled
- 1: enabled

Bit 6 PROG_ERR_INT_EN: PROG_ERR interrupt enable.

- 0: disabled
- 1: enabled

Bit 5 PAGE_ERASE_ERR_INT_EN: PAGE_ERASE_ERR interrupt enable.

- 0: disabled
- 1: enabled

Bit 4 OPTION_WR_ERR_INT_EN: OPTION_WR_ERR interrupt enable.

- 0: disabled
- 1: enabled

Bit 3 FLASHBUSY_ERR_INT_EN: FLASHBUSY_ERR interrupt enable.

- 0: disabled
- 1: enabled

Bit 2 PROG_DATA_WAIT_INT_EN: PROG_DATA_WAIT interrupt enable.

- 0: disabled
- 1: enabled

Bit 1 RESERVED: Must be kept, and cannot be modified.

Bit 0 OPERATION_DONE_INT_EN: OPERATION_DONE interrupt enable.

- 0: disabled
- 1: enabled

10.5.3 EFC_SR

Offset: 0x08

Reset Value: 0x00000006

31-9	8	7	6	5
RESERVED	TWO_BIT_ERROR	ONE_BIT_CORRECT	PROG_ERR	PAGE_ERASE_ERR
r	r/w	r/w	r/w	r/w
4	**3**	**2**	**1**	**0**
OPTION_WR_ERR	FLASHBUSY_ERR	PROG_DATA_WAIT	READ_NUM_DONE	OPERATION_DONE
r/w	r/w	r/w	r	r/w

Bits 31-9 RESERVED: Must be kept, and cannot be modified.

Bit 8 TWO_BIT_ERROR: TWO_BIT_ERROR flag is set when the Flash memory is read.

- 0: no two-bit error occurred
- 1： 1: two-bit error occurred when reading the Flash memory and ECC did not correct

Bit 7 ONE_BIT_CORRECT: ONE_BIT_CORRECT flag is set when the Flash memory is read.

- 0: no one-bit error occurred
- 1: one-bit error occurred when reading the Flash memory and ECC corrected it

Bit 6 PROG_ERR: Some partitions within the Flash info area don't support programming operation(PROG_EN). Programming operation to these partitions will be blocked, and this bit will be set by hardware and cleared by software writing 1 to it.

- 0: no programming error occurred
- 1: programming error occurred

Note: *The option area cannot be written by direct program operations. The bootloader area cannot be programmed.*

Bit 5 PAGE_ERASE_ERR: The Flash info area don't support erasing operation. Erasing operation to the info area will be blocked, and this bit will be set by hardware and cleared by software writing 1 to it.

- 0: no page erase error occurred
- 1: a page erase error occurred

Bit 4 OPTION_WR_ERR: The Option area should be configured with the limitations respected, or the configuration is invalid and this bit is set by hardware. It is cleared by software writing 1 to it.

- 0: no write permission error on Option byte
- 1: a write permission error on Option byte occurred

The configuration for the Option area must respect the following limitations:

1. Flash EXE_Only1/EXE_Only2 area can't be disabled or narrowed once it is enabled.
2. Bit EXE_ONLY_KEEP can't be modified from 0 to 1.
3. When SECURE_AREA_EN=1, operations initiated by non-secure areas only act on the FLASH_SECURE_END/FLASH_SECURE_START bits in Option bytes to clear the secure_area_en status bit.

Bit 3 FLASHBUSY_ERR: When Flash is performing programming, erasing (including mass), and option operations, the read operation by the software will be blocked, the data returned by the bus is uncertain, it is an abnormal state, this bit will be set by hardware and cleared by software writing 1 to it.

- 0: no error occurred
- 1: read error occurred during a Flash operation

Bit 2 PROG_DATA_WAIT: Waiting for data to be written to the Flash memory in WL continuous programming mode. This bit is set by hardware and is cleared automatically by hardware when the software writes new data to the *FC_PROG_DATA0* and *EFC_PROG_DATA1* registers. It can also be cleared by software writing 1 to it.

- 0: the value of registers EFC_PROG_DATA0 and EFC_PROG_DATA1 has been written to the Flash memory
- 1: wait for the value of registers EFC_PROG_DATA0 and EFC_PROG_DATA1 to be written to the Flash memory

Bit 1 READ_NUM_DONE: READ_NUM configuration status flag, it indicates whether the READ_NUM configuration is complete. This bit is set and cleared by hardware.

- 0: in progress
- 1: complete

Bit 0 OPERATION_DONE: Flash operation status flag, it indicates whether Flash mass erase/page erase/program/option operation is complete. This bit is set by hardware and cleared by software writing 1 to it.

- 0: in progress
- 1: complete

10.5.4 EFC_PROG_DATA0

Offset: 0x0C

Reset Value: 0x00000000

31-0
PROG_DATA0
r/w

Bits 31-0 PROG_DATA0: programming data 0.

Note: *When programming, write data to register EFC_PROG_DATA0 first.*

10.5.5 EFC_PROG_DATA1

Offset: 0x10

Reset Value: 0x00000000

31-0
PROG_DATA1
r/w

Bits 31-0 PROG_DATA1: programming data 1.

Note: *When programming, write data to register EFC_PROG_DATA0 first.*

10.5.6 EFC_TIMING_CFG

Offset: 0x14

Reset Value: 0x00031D1D

31-20	19-16	15-0
RESERVED	READ_NUM	RESERVED
r	r/w	r

Bits 31-20 RESERVED: Must be kept, and cannot be modified.

Bit 19-16 READ_NUM: Flash read operation read wait count control, the read wait count is equal to (READ_NUM+1) SYS_CLK clock cycles.

- For 48 MHz SYS_CLK frequency, READ_NUM must ≥ 2.
- For 32 MHz SYS_CLK frequency, READ_NUM must ≥ 1.
- For 24 MHz SYS_CLK frequency, READ_NUM must ≥ 1.
- For 4 MHz SYS_CLK frequency, READ_NUM must ≥ 0.
- For 32 kHz SYS_CLK frequency, READ_NUM must ≥ 0.

***Note:** When changing the SYS_CLK clock source in register RCC_CR0, pay attention to the sequence of operations. If you intend to switch to a faster clock source, first increase the READ_NUM, and then configure the clock source selection bit; otherwise, first configure the clock source selection bit, and then decrease the READ_NUM.*

Bits 15-0 RESERVED: Must be kept, and cannot be modified.

10.5.7 EFC_PROTECT_SEQ

Offset: 0x18

Reset Value: 0x00000000

31-0
PROTECT_SEQ
w

Bits 31-0 PROTECT_SEQ: Protection sequence for the configuration of register *EFC_CR* By default, the EFC_CR register cannot be modified, to modify it, the user must configure the protection sequence correctly through the FC_PROTECT_SEQ register in the following order. If there is an error in the configuration, then the configuration is invalid, and the protection sequence should be reconfigured.

1. Write 0x8C9DAEBF.
2. Write 0x13141516.
3. You can operate with EFC_CR

10.5.8 SERIAL_NUM_LOW

Offset:0x2C

31-0
SERIAL_NUM_LOW
r

Bits 31-0 SERIAL_NUM_LOW: Less significant 32 bits of the chip serial number.

10.5.9 SERIAL_NUM_HIGH

Offset:0x30

31-0
SERIAL_NUM_HIGH
r

Bits 31-0 SERIAL_NUM_HIGH: Most significant 32 bits of the chip serial number.

10.5.10 OPTION_CSR_BYTES

Offset: 0x3C

Reset Value: 0x000000BD

31-7	6-5	4	3	2	1	0
RESERVED	DEBUG_LEVEL	SECURE_AREA_EN	SYS_SRAM_RST	FLASH_BOOT1	USE_FLASH_BOOT0	FLASH_BOOT0
r	r	r	r	r	r	r

Bits 31-7 RESERVED: Must be kept, and cannot be modified.

Bits 6-5 DEBUG_LEVEL: Debug level setting.

- 0: Level 0 ● 1: Level 1 ● 2: Level 2

Bit 4 SECURE_AREA_EN: Flash secure area status flag.

- 0: secure area disabled
- 1: secure area enabled

Bit 3 SYS_SRAM_RST: Clear system SRAM during system startup after its reset.

- 0: do not clear system SRAM
- 1: clear system SRAM

Bit 2 FLASH_BOOT1: This bit can be used for boot mode identification. See *Table* 7-7 for more details.

- 0: System SRAM boot
- 1: BootLoader

Bit 1 USE_FLASH_BOOT0: This bit can be used for boot mode identification. See *Table* 7-7 for more details.

- 0: Use BOOT0 pin
- 1: Use FLASH_BOOT0 option bit

Bit 0 FLASH_BOOT0: This bit can be used for boot mode identification, and the configuration is only valid when USE_FLASH_BOOT0=1. See *Table* 7-7 for more details.

- 0: FLASH_BOOT1 controls the startup mode
- 1: Boot from Main Flash

10.5.11 OPTION_EXE_ONLY_BYTES

Offset: 0x40

Reset Value: 0x00FC0FC0

31-25	24	23-18	17-12	11-6	5-0
RESERVED	EXE_ONLY_KEEP	EXE_ONLY2_END	EXE_ONLY2_START	EXE_ONLY1_END	EXE_ONLY1_START
r	r	r	r	r	r

Bits 31-25 RESERVED: Must be kept, and cannot be modified.

Bit 24 EXE_ONLY_KEEP: Keep Exe_Only area when the Debug_Level changes from 1 to 0.

- 0: erase ExeOnly area
- 1: keep the ExeOnly area

This bit can only be cleared by software.

Bits 23-18 EXE_ONLY2_END: Exe_Only2 area end offset.

If *EXEONLY2_START* > *EXEONLY2_END*, the ExeOnly2 area is disabled.

Bits 17-12 EXE_ONLY2_START: Exe_Only2 area start offset.

If *EXEONLY2_START* > *EXEONLY2_END*, the ExeOnly2 area is disabled. Once enabled, this area can only be expanded but can't be disabled or narrowed.

Bits 11-6 EXE_ONLY1_END: Exe_Only1 area end offset.

If *EXEONLY1_START* > *EXEONLY1_END*, the ExeOnly1 area is disabled.

Bits 5-0 EXE_ONLY1_START: Exe_Only1 area start offset.

If *EXEONLY1_START* > *EXEONLY1_END*, the ExeOnly1 area is disabled. Once enabled, this area can only be expanded but can't be disabled or narrowed.

10.5.12 OPTION_WR_PROTECT_BYTES

Offset: 0x44

Reset Value: 0x0003F03F

31-12	11-6	5-0
RESERVED	WRPROTECT_END	WRPROTECT_START
r	r	r

Bits 31-12 RESERVED: Must be kept, and cannot be modified.

Bits 11-6 WRPROTECT_END: Write-protected area end offset.
If *WRPROTECT_START* > *WRPROTECT_END*, the write-protected area is disabled.

Bits 5-0 WRPROTECT_START: Write-protected area start offset.
If *WRPROTECT_START* > *WRPROTECT_END*, the write-protected area is disabled

10.5.13 OPTION_SECURE_BYTES0

Offset: 0x48

Reset Value: 0x00FC0FC0

31-24	23-18	17-12	11-6	5-0
RESERVED	SYSRAM_SECURE_ END	SYSRAM_SECURE_ START	FLASH_SECURE_ END	FLASH_SECURE_ START
r	r	r	r	r

Bits 31-24 RESERVED: Must be kept, and can't be modified.

Bits 23-18 SYSRAM_SECURE_END: SysRam Secure area end.

If *SYSRAM_SECURE_START* > *SYSRAM_SECURE_END*, the SysRam Secure area is disabled.

The configuration is only valid when SECURE_AREA_EN=1.

Bits 17-12 SYSRAM_SECURE_START: SysRam Secure area start.

If *SYSRAM_SECURE_START* > *SYSRAM_SECURE_END*, the SysRam Secure area is disabled.

The configuration is only valid when SECURE_AREA_EN=1.

Bits 11-6 Flash Secure area end.

If *FLASH_SECURE_START* > *FLASH_SECURE_END*, the Flash Secure area is disabled.

Bits 5-0 FLASH_SECURE_START: Flash Secure area start.
If *FLASH_SECURE_START* > *FLASH_SECURE_END*, the Flash Secure area is disabled.
The Flash Secure area enable is the master switch for enabling other secure areas.
If the Flash Secure area is enabled, the SECURE_AREA_EN bit is set, which means all the other secure areas can be enabled.
If the Flash Secure area is disabled, the SECURE_AREA_EN bit is cleared, which triggers the erase operation.

10.5.14 OPTION_SECURE_BYTES1

Offset: 0x4C

Reset Value: 0x008103E0

31-24	23		22-17
RESERVED	SYSRAM_HIDE_ENABLE		SYSRAM_HIDE_START
r	r		r
16	**15-10**	**9-5**	**4-0**
FLASH_HIDE_ENABLE	FLASH_HIDE_START	RETRAM_SECURE_END	RETRAM_SECURE_START
r	r	r	r

Bits 31-24 RESERVED: Must be kept, and can't be modified.

Bit 23 SYSRAM_HIDE_ENABLE: SysRamHide area enable control.

- 0: SysRamHide area enabled
- 1: SysRamHide area disabled

The configuration is only valid when SECURE_AREA_EN=1.

Bits 22-17 SYSRAM_HIDE_START: SysRamHide area start.

The configuration is only valid when the SysRamHide area is within the SysRamSecure area and when SECURE_AREA_EN=1.

The SysRamHide area is from SYSRAM_HIDE_START to SYSRAM_SECURE_END.

Bit 16 FLASH_HIDE_ENABLE: FlashHide area enable control.

- 0: FlashHide area enabled
- 1: FlashHide area disabled

The configuration is only valid when SECURE_AREA_EN=1.

Bits 15-10 FLASH_HIDE_START: FlashHide area start.

The configuration is only valid when the FlashHide area is within the FlashSecure area and when SECURE_AREA_EN=1.
The FlashHide area is from FLASH_HIDE_START to FLASH_SECURE_END.

Bits 9-5 RETRAM_SECURE_END: RetRam Secure area end.
If *RETRAM_SECURE_START* > *RETRAM_SECURE_END*, the RetRam Secure area is disabled.
The configuration is only valid when SECURE_AREA_EN=1.

Bits 4-0 RETRAM_SECURE_START: RetRam Secure area start.
If *RETRAM_SECURE_START* > *RETRAM_SECURE_END*, the RetRam Secure area is disabled.
The configuration is only valid when SECURE_AREA_EN=1.

11. GPIO

11.1 Introduction

ASR6601 GPIOs are divided into four groups: Ports A, B, C, and D. The SFR registers of each group are allocated the same, and they are distinguished by different base addresses. PortD Pin8 ~ Pin15 are located in the AON domain, and the other IOs are located in the Main domain.

All GPIOs support input and output, pull-up and pull-down, push-pull output and open-drain output. The output drive current can be configured as 4mA or 8mA. All GPIOs can generate interrupts, which can be triggered by rising edge, falling edge or both edges. In Sleep/Stop0~2 mode, all GPIOs can be used for wake-up; while in Stop3 mode, only some GPIOs can be used to wake-up MCU. All GPIOs support alternate functions.

11.2 Output Configuration

GPIO data output is configured by the *GPIOx_OER* and *GPIOx_ODR* registers.

GPIO output can be set or cleared. Writing 1 to bits[15:0] in register *GPIOx_BRR* or writing 1 to bits[31-16] in register *GPIOx_BSRR* can **clear** the corresponding bit in register *GPIOx_ODR*. And writing 1 to bits[15:0] in register *GPIOx_BSRR* can **set** the corresponding bit in register *GPIOx_ODR*.

GPIO port is configured as **push-pull** output through register *GPIOx_OTYPER*. As to output in **open-drain** mode, for PortD Pin8 ~ PortD Pin15, it is enabled by configuring the *GPIOx_IER*, *GPIOx_OER, GPIOx_ODR and GPIOx_PSR* registers, and for other IO ports, it is enabled by configuring the *GPIOx_OER, GPIOx_IER, GPIOx_ODR and GPIOx_OTYPER* registers. Not implementing a real open drain structure, the open drain function is achieved by control of the *GPIOx_OER* and *GPIOx_ODR* registers.

GPIO can be configured as analog output.

11.3 Input Configuration

GPIO data input is enabled by configuring register *GPIOx_IER*, and you can read register *GPIOx_IDR* to get the input status.

Input floating mode is realized by configuring register *GPIOx_PER* to disable pull-up and pull-down.

Pull-up or pull-down is enabled by configuring register *GPIOx_PER*, and register *GPIOx_PSR* is used for pull-up or pull-down selection.

GPIO can be configured as analog input.

11.4 Output Drive Strength

High (8 mA) or low (4 mA) output drive strength is configured by *GPIOx_DSR* register.

11.5 GPIO Interrupts

All GPIOs support interrupts, which can be triggered by rising edge, falling edge or both edges.

Interrupts are enabled by configuring *GPIOx_INT_CR* register.

11.6 Wakeup from Sleep/Stop0~2 Mode

In Sleep or Stop 0/1/2 mode, MCU can be woken up at high level or low level, and the output wake-up signal is high level. GPIO00-GPIO63 can all be used for wakeup, four IOs make up a group. A group can generate a wakeup signal, and each IO in a group can wake up MCU at high level or low level. In Sleep/Stop0~2 mode, the wakeup function is enabled by configuring the *GPIOx_WU_EN* register, and the high-level or low-level wakeup is selected by configuring the *GPIOx_WU_LVL* register.

11.7 Wakeup from Stop3 Mode

GPIO00~GPIO55 of the Main domain, every 4 IO MUXs output a wake-up signal, a total of 14 wake-up signals. Support high-level wake-up or low-level wake-up and wake-up enable control, which is achieved by configuring the Stop3 wake-up enable register GPIOx_STOP3_WU_CR.

11.8 Alternate Function Configuration

GPIO can be used as general I/O or configured as alternate function. GPIO input/output is enabled or disabled by the *GPIOx_OER* and *GPIOx_IER* registers, while the alternate function input/output is enabled or disabled by alternate peripherals. The I/O pull-up or pull-down is configured by the *GPIOx_PER* and *GPIOx_PSR* registers.

As to alternate function control, 3-bit for each pin among PortD Pin8~Pin15, and 4-bit for each of the other pins. By default, PortA Pin6 and Pin7 are configured as SWD pins, and the other IOs are configured as GPIO.

The function of Portx Pin[7:0] is configured through the lower 8 Pin function MUX selection register GPIOx_AFRL, and the function of Portx Pin[15:8] is configured through the upper 8 Pin function MUX selection register GPIOx_AFRH.

11.9 Clock and Reset

There are four groups of APB bus clock and APB bus reset, each group has an independent bus clock and bus reset.

11.10 Power Domains

Main Domain:

Except for PortD Pin8~PortD Pin15, the corresponding pads are all in the Main domain.

AON (always-on) Domain:

The PADs corresponds to PortD Pin8~Pin15 are in the AlwaysOn domain. If they are configured as alternate function, they are directly controlled by the peripherals. Otherwise, they will be controlled by the GPIO registers in the AlwaysOn domain.

11.11 Low-power Mode Operation and Wakeup

1. In Sleep mode, all GPIOs can work and output wake-up signal.

2. In Stop0/Stop1/Stop2 mode, all GPIOs can work and output wake-up signal.

3. In Stop3 mode, GPIO00~GPIO55 can retain the state, and can be configured as wake-up signal.

4. In Stop3 mode, PortD Pin8~Pin15 in AlwaysOn domain can retain the state, CPU can also be woken up through RTC.

5. In Standby mode, PortD Pin8~PortD Pin15 can work, while the other IOs can't work.

11.12 SWD IO

Default Control: The GPIO alternate function low register selects SWD by default, and SWCLK pull-down (PortA Pin7) and SWDIO pull-up (PortA Pin6) are default.

Sealing control: After power-on, the IO status is controlled by the default state of the register until the DebugLevel judgment is completed. If sealing is found to be necessary, permanent sealing is performed; otherwise, it continues to be controlled by the register.

Software configuration: During software operation, the SWD can be disabled by controlling the multiplexing register. Note that it is a one-way seal, that is, it can only be disabled, and cannot be disabled and then enabled.

11.13 BOOT0 Control

Default Control: Since all IOs except the SWCLK and SWDIO IOs are analog IOs by default, the BOOT0, SWCLK and SWDIO pins require special control at power-on.

BOOT0 (GPIO02): BOOT0 is in input pull-down status before io_lock. After EFC is locked, it switches to GPIO mode.

11.14 GPIO Registers

GPIO Port A Base Address: 0x4001F000

GPIO Port B Base Address: 0x4001F400

GPIO Port C Base Address: 0x4001F800

GPIO Port D Base Address: 0x4001FC00

Table 11-1 GPIO Registers Summary

Register	Offset	Description
GPIOx_OER	0x00	通用输出使能寄存器
GPIOx_OTYPER	0x04	通用输出类型控制寄存器
GPIOx_IER	0x08	通用输入使能寄存器
GPIOx_PER	0x0C	上下拉使能寄存器
GPIOx_PSR	0x10	上下拉选择寄存器
GPIOx_IDR	0x14	输入数据寄存器
GPIOx_ODR	0x18	输出数据寄存器
GPIOx_BRR	0x1C	输出数据清零寄存器
GPIOx_BSRR	0x20	输出数据置位和清零寄存器
GPIOx_DSR	0x24	输出驱动能力寄存器
GPIOx_INT_CR	0x28	中断使能寄存器
GPIOx_FR	0x2C	中断沿标志寄存器
GPIOx_WU_EN	0x30	Sleep/Stop0~2 唤醒使能寄存器
GPIOx_WU_LVL	0x34	Sleep/Stop0~2 唤醒电平控制寄存器
GPIOx_AFRL	0x38	低 8 Pin 功能 MUX 选择寄存器
GPIOx_AFRH	0x3C	高 8 Pin 功能 MUX 选择寄存器
GPIOx_STOP3_WU_CR	0x40	Stop3 唤醒使能控制寄存器

11.14.1 GPIOx_OER (x=A, B, C, D)

Offset: 0x00

Reset Value: 0x0000FFFF

31-16	15-0
RESERVED	OEN
r-0h	rw-ffffh

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 OEN: Portx Pin[15:0] output enable.

- 0: output enabled
- 1: output disabled

11.14.2 GPIOx_OTYPER (x=A, B, C, D)

Offset: 0x04

Reset Value: 0x00000000

31-16	15-0
RESERVED	OTYPE
r-0h	rw-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 OTYPE: Portx Pin[15:0] output type control.

- 0: push-pull
- 1: open-drain

Note: *The output type of the pads in the AON domain (PortD_Pin[15:8]) is controlled by the GPIOx_IER, GPIOx_OER, GPIOx_ODR and GPIOx_PSR registers instead of this register. For the other pins, the open drain mode is enabled through the GPIOx_IER, GPIOx_OER, GPIOx_ODR and GPIOx_OTYPER registers.*

11.14.3 GPIOx_IER (x=A, B, C, D)

Offset: 0x08

Reset Value: 0x00000000

31-16	15-0
RESERVED	IE
r-0h	rw-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 IE: Portx Pin[15:0] input enable.

- 0: input disabled

PDF-XChange Editor Click to BUY NOW! www.pdf-xchange.com

- 1: input enabled

11.14.4 GPIOx_PER (x=A, B, C, D)

Offset: 0x0C

Reset Value: 0x00000000

31-16	15-0
RESERVED	PE
r-0h	rw-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 PE: Portx Pin[15:0] pull-up/pull-down enable.

- 0: pull-up/pull-down disabled
- 1: pull-up/pull-down enabled

GPIO pull-up and pull-down is selected by the *GPIOx_PSR* register. By default, pull-up/pull-down is disabled, and all the IOs except PortA_Pin[7:6] are in analog mode. PortA_Pin[7:6] are used as SWD function.

11.14.5 GPIOx_PSR (x=A, B, C, D)

Offset: 0x10

Reset Value: 0x00000000

31-16	15-0
RESERVED	PS
r-0h	rw-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 PS: Portx Pin[15:0] pull-up/pull-down selection.

- 0: pull-down
- 1: pull-up

11.14.6 GPIOx_IDR (x=A, B, C, D)

Offset:0x14

Reset Value: 0x00000000

31-16	15-0
RESERVED	ID
r-0h	r-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 ID: Portx Pin[15:0] input.

- 0: low level

- 1: high level

11.14.7 GPIOx_ODR (x=A, B, C, D)

Offset: 0x18

Reset Value: 0x00000000

31-16	15-0
RESERVED	OD
r-0h	rw-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 OD: Portx Pin[15:0] output.

- 0: low level
- 1: high level

11.14.8 GPIOx_BRR (x=A, B, C, D)

Offset: 0x1C

Reset Value: 0x00000000

31-16	15-0
RESERVED	BR
r-0h	w-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 BR: Portx Pin[15:0] output data clear.

- 0: no action
- 1: clear the corresponding bit of the GPIOx_ODR register

11.14.9 GPIOx_BSRR (x=A, B, C, D)

Offset: 0x20

Reset Value: 0x00000000

31-16	15-0
BR	BSR
w-0h	w-0h

Bits 31-16 BR: Portx Pin[15:0] output data clear.

- 0: no action
- 1: clear the corresponding bit of the GPIOx_ODR register

Note: If BSR and BR are both valid, BSR has higher priority.

Bits 15-0 BSR: Portx Pin[15:0] output data set.

- 0: no action
- 1: set the corresponding bit of the GPIOx_ODR register

Note: If BSR and BR are both valid, BSR has higher priority.

11.14.10 GPIOx_DSR (x=A, B, C, D)

Offset: 0x24

eset Value: 0x00000000

31-16	15-0
RESERVED	DS
r-0h	w-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 DS: Portx Pin[15:0] output drive strength configuration.

- 0: low drive strength (4 mA)
- 1: high drive strength (8 mA)

11.14.11 GPIOx_INT_CR (x=A, B, C, D)

Offset: 0x28

Reset Value: 0x00000000

2*n + 1	2*n
NEG_INT_EN	POS_INT_EN
rw-0h	rw-0h

Bits 2*n + 1 NEG_INT_EN: Portx Pin[15:0] enable interrupt triggered by falling edge.

- 0: interrupt triggered by falling edge disabled
- 1: interrupt triggered by falling edge enabled

Bits 2*n POS_INT_EN: Portx Pin[15:0] enable interrupt triggered by rising edge.

- 0: interrupt triggered by rising edge disabled
- 1: interrupt triggered by rising edge enabled

11.14.12 GPIOx_FR (x=A, B, C, D)

Offset: 0x2C

Reset Value: 0x00000000

2*n + 1	2*n
NEG_F	POS_F
rw1c-0h	rw1c-0h

Bits 2*n + 1 NEG_INT_F: Portx Pin[15:0] interrupt flag (falling edge)

- 0: no interrupt triggered by falling edge occurred
- 1: interrupt triggered by falling edge occurred

Bits 2*n POS_INT_F: Portx Pin[15:0] interrupt flag (rising edge)

- 0: no interrupt triggered by rising edge occurred
- 1: interrupt triggered by rising edge occurred

11.14.13 GPIOx_WU_EN (x=A, B, C, D)

Offset: 0x30

Reset Value: 0x00000000

31-16	15-0
RESERVED	WU_EN
r-0h	rw-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 WU_EN: Enable/disable Portx Pin[15:0] to wake-up CPU from Sleep/Stop0~2 modes.

- 0: disable Sleep/Stop0~2 wakeup
- 1: enable Sleep/Stop0~2 wakeup

11.14.14 GPIOx_WU_LVL (x=A, B, C, D)

Offset: 0x34

Reset Value: 0x00000000

31-16	15-0
RESERVED	WU_LVL
r-0h	rw-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 WU_LVL: Configure Portx Pin[15:0] CPU Sleep/Stop0~2 mode wakeup to high or low level.

- 0: wake-up at low level
- 1: wake-up at high level

11.14.15 GPIOx_AFRL (x=A, B, C, D)

Offset: 0x38

Reset Value: 0x00000000

31-28	27-24	23-20	19-16	15-12	11-8	7-4	3-0
AF7	AF6	AF5	AF4	AF3	AF2	AF1	AF0
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

Bits 31-28 AF7: Portx Pin7 function selection

- 0000：Function0
- 0001：Function1
- 0010：Function2
- 0011：Function3
- 0100：Function4
- 0101：Function5
- 0110：Function6
- 0111：Function7
- others：Reserved

位 27-24 AF6： Portx Pin6 功能 MUX 选择。

- 0000：Function0
- 0001：Function1
- 0010：Function2
- 0011：Function3
- 0100：Function4
- 0101：Function5
- 0110：Function6
- 0111：Function7
- others：Reserved

位 23-20 AF5： Portx Pin5 功能 MUX 选择。

- 0000：Function0
- 0001：Function1
- 0010：Function2
- 0011：Function3
- 0100：Function4
- 0101：Function5
- 0110：Function6
- 0111：Function7
- others：Reserved

位 19-16 AF4： Portx Pin4 功能 MUX 选择。

- 0000：Function0
- 0001：Function1
- 0010：Function2

- 0011：Function3
- 0100：Function4
- 0101：Function5
- 0110：Function6
- 0111：Function7
- others：Reserved

Bits 15-12 AF3: Portx Pin3 function selection.

- 0000：Function0
- 0001：Function1
- 0010：Function2
- 0011：Function3
- 0100：Function4
- 0101：Function5
- 0110：Function6
- 0111：Function7
- others：Reserved

Bits 11-8 AF2: Portx Pin2 function selection.

- 0000：Function0
- 0001：Function1
- 0010：Function2
- 0011：Function3
- 0100：Function4
- 0101：Function5
- 0110：Function6
- 0111：Function7
- others：Reserved

Bits 7-4 AF1: Portx Pin1 function selection.

- 0000：Function0
- 0001：Function1
- 0010：Function2
- 0011：Function3
- 0100：Function4
- 0101：Function5
- 0110：Function6
- 0111：Function7
- others：Reserved

Bits 3-0 AF0: Portx Pin0 function selection.

- 0000：Function0
- 0001：Function1
- 0010：Function2
- 0011：Function3
- 0100：Function4
- 0101：Function5

- 0110: Function6
- 0111: Function7
- others: Reserved

11.14.16 GPIOx_AFRH (x=A, B, C)

Offset: 0x3C

Reset Value: 0x00000000

31-28	27-24	23-20	19-16	15-12	11-8	7-4	3-0
AF15	AF14	AF13	AF12	AF11	AF10	AF9	AF8
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

Bits 31-28 AF15: Portx Pin15 function selection.

- 0000: Function0
- 0001: Function1
- 0010: Function2
- 0011: Function3
- 0100: Function4
- 0101: Function5
- 0110: Function6
- 0111: Function7
- others: Reserved

Bits 27-24 AF14: Portx Pin14 function selection.

- 0000: Function0
- 0001: Function1
- 0010: Function2
- 0011: Function3
- 0100: Function4
- 0101: Function5
- 0110: Function6
- 0111: Function7
- others: Reserved

Bits 23-20 AF13: Portx Pin13 function selection.

- 0000: Function0
- 0001: Function1
- 0010: Function2
- 0011: Function3
- 0100: Function4
- 0101: Function5
- 0110: Function6
- 0111: Function7
- others: Reserved

Bits 19-16 AF12: Portx Pin12 function selection.

- 0000：Function0
- 0001：Function1
- 0010：Function2
- 0011：Function3
- 0100：Function4
- 0101：Function5
- 0110：Function6
- 0111：Function7
- others：Reserved

Bits 15-12 AF11: Portx Pin11 function selection.

- 0000：Function0
- 0001：Function1
- 0010：Function2
- 0011：Function3
- 0100：Function4
- 0101：Function5
- 0110：Function6
- 0111：Function7
- others：Reserved

Bits 11-8 AF10: Portx Pin10 function selection.

- 0000：Function0
- 0001：Function1
- 0010：Function2
- 0011：Function3
- 0100：Function4
- 0101：Function5
- 0110：Function6
- 0111：Function7
- others：Reserved

Bits 7-4 AF9: Portx Pin9 function selection.

- 0000：Function0
- 0001：Function1
- 0010：Function2
- 0011：Function3
- 0100：Function4
- 0101：Function5
- 0110：Function6
- 0111：Function7
- others：Reserved

Bits 3-0 AF8: Portx Pin8 function selection.

- 0000：Function0
- 0001：Function1

- 0010: Function2
- 0011: Function3
- 0100: Function4
- 0101: Function5
- 0110: Function6
- 0111: Function7
- others: Reserved

11.14.17 GPIOD_AFRH

Offset: 0x3C

Reset Value: 0x00000000

31-24	23-21	20-18	17-15	14-12	11-9	8-6	5-3	2-0
RESERVED	AF15	AF14	AF13	AF12	AF11	AF10	AF9	AF8
r-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

Bits 31-24 RESERVED: Must be kept, and cannot be modified.

- 001: Function1
- 010: Function2
- 011: Function3
- 100: Function4
- 101: Function5
- 110: Function6
- 111: Function7

Bits 23-21 AF15: PortD Pin15 function selection.

- 001: Function1
- 010: Function2
- 011: Function3
- 100: Function4
- 101: Function5
- 110: Function6
- 111: Function7

Bits 20-18 AF14: PortD Pin14 function selection.

- 001: Function1
- 010: Function2
- 011: Function3
- 100: Function4
- 101: Function5
- 110: Function6
- 111: Function7

Bits 17-15 AF13: PortD Pin13 function selection.

- 001: Function1

- 010: Function2
- 011: Function3
- 100: Function4
- 101: Function5
- 110: Function6
- 111: Function7

Bits 14-12 AF12: PortD Pin12 function selection.

- 001: Function1
- 010: Function2
- 011: Function3
- 100: Function4
- 101: Function5
- 110: Function6
- 111: Function7

Bits 11-9 AF11: Pin11 function selection.

- 001: Function1
- 010: Function2
- 011: Function3
- 100: Function4
- 101: Function5
- 110: Function6
- 111: Function7

Bits 8-6 AF10: PortD Pin10 function selection.

- 001: Function1
- 010: Function2
- 011: Function3
- 100: Function4
- 101: Function5
- 110: Function6
- 111: Function7

Bits 5-3 AF9: PortD Pin9 function selection.

- 001: Function1
- 010: Function2
- 011: Function3
- 100: Function4
- 101: Function5
- 110: Function6
- 111: Function7

Bits 2-0 AF8: PortD Pin8 function selection.

- 001: Function1
- 010: Function2
- 011: Function3

- 100: Function4
- 101: Function5
- 110: Function6
- 111: Function7

11.14.18 GPIOA_STOP3_WU_CR

Offset: 0x40

Reset Value: 0x00000000

31-16	15	14	13-12	11
RESERVED	STOP3_WU_EN_G1	STOP3_WU_LVL_G3	STOP3_WU_SEL_G3	STOP3_WU_EN_G2
r-0h	rw-0h	rw-0h	rw-0h	rw-0h
10		**9-8**	**7**	**6**
STOP3_WU_LVL_G2		STOP3_WU_SEL_G2	STOP3_WU_EN_G1	STOP3_WU_LVL_G1
rw-0h		rw-0h	rw-0h	rw-0h
5-4		**3**	**2**	**1-0**
STOP3_WU_SEL_G1		STOP3_WU_EN_G0	STOP3_WU_LVL_G0	STOP3_WU_SEL_G0
rw-0h		rw-0h	rw-0h	rw-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.
Bit 15 STOP3_WU_EN_G3: PortA Group3 wake-up pin enable control in Stop3 mode.

- 0: disabled
- 1: enabled

Bit 14 STOP3_WU_LVL_G3: PortA Group3 wake-up pin level selection in Stop3 mode.

- 0: low level
- 1: high level

Bits 13-12 STOP3_WU_SEL_G3: PortA Pin Group3 wake-up source selection in Stop3 mode.

- 00: PortA Pin6
- 01: PortA Pin7
- 10: PortA Pin14
- 11: PortA Pin15

Bit 11 STOP3_WU_EN_G2: PortA Group2 wake-up pin enable control in Stop3 mode.

- 0: disabled
- 1: enabled

Bit 10 STOP3_WU_LVL_G2: PortA Group2 wake-up pin level selection in Stop3 mode.

- 0: low level
- 1: high level

Bits 9-8 STOP3_WU_SEL_G2: PortA Pin Group2 wake-up source selection in Stop3 mode.

- 00: PortA Pin8
- 01: PortA Pin9
- 10: PortA Pin10
- 11: PortA Pin11

Bit 7 STOP3_WU_EN_G1: PortA Group1 wake-up pin enable control in Stop3 mode.

- 0: disabled
- 1: enabled

Bit 6 STOP3_WU_LVL_G1: PortA Group1 wake-up pin level selection in Stop3 mode.

- 0: low level
- 1: high level

Bits 5-4 STOP3_WU_SEL_G1: PortA Pin Group1 wake-up source selection in Stop3 mode.

- 00: PortA Pin4
- 01: PortA Pin5
- 10: PortA Pin12
- 11: PortA Pin13

Bit 3 STOP3_WU_EN_G0: PortA Group0 wake-up pin enable control in Stop3 mode.

- 0: disabled
- 1: enabled

Bit 2 STOP3_WU_LVL_G0: PortA Group0 wake-up pin level selection in Stop3 mode.

- 0: low level
- 1: high level

Bits 1-0 STOP3_WU_SEL_G0: PortA Pin Group0 wake-up source selection in Stop3 mode.

- 00: PortA Pin0
- 01: PortA Pin1
- 10: PortA Pin2
- 11: PortA Pin3

11.14.19 GPIOx_STOP3_WU_CR (x=B, C)

Offset: 0x40

Reset Value: 0x00000000

31-16	15	14	13-12	11
RESERVED	STOP3_WU_EN_G3	STOP3_WU_LVL_G3	STOP3_WU_SEL_G3	STOP3_WU_EN_G2
r-0h	rw-0h	rw-0h	rw-0h	rw-0h
10		**9-8**	**7**	**6**
STOP3_WU_LVL_G2		STOP3_WU_SEL_G2	STOP3_WU_EN_G1	STOP3_WU_LVL_G1
rw-0h		rw-0h	rw-0h	rw-0h
5-4		**3**	**2**	**1-0**
STOP3_WU_SEL_G1		STOP3_WU_EN_G0	STOP3_WU_LVL_G0	STOP3_WU_SEL_G0
rw-0h		rw-0h	rw-0h	rw-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bit 15 STOP3_WU_EN_G3: Portx Pin Group3 wake-up enable control in Stop3 mode.

- 0: disabled
- 1: enabled

Bit 14 STOP3_WU_LVL_G3: Portx Pin Group3 wake-up level selection in Stop3 mode.

- 0: low level
- 1: high level

Bits 13-12 STOP3_WU_SEL_G3: Portx Pin Group3 wake-up source selection in Stop3 mode.

- 00: Portx Pin12
- 01: Portx Pin13
- 10: Portx Pin14
- 11: Portx Pin15

Bit 11 STOP3_WU_EN_G2: Portx Pin Group2 wake-up enable control in Stop3 mode.

- 0: disabled
- 1: enabled

Bit 10 STOP3_WU_LVL_G2: Portx Pin Group2 wake-up level selection in Stop3 mode.

- 0: low level
- 1: high level

Bits 9-8 STOP3_WU_SEL_G2: Portx Pin Group2 wake-up source selection in Stop3 mode.

- 00: Portx Pin8
- 01: Portx Pin9
- 10: Portx Pin10
- 11: Portx Pin11

Bit 7 STOP3_WU_EN_G1: Portx Pin Group1 wake-up enable control in Stop3 mode.

- 0: disabled
- 1: enabled

Bit 6 STOP3_WU_LVL_G1: Portx Pin Group1 wake-up level selection in Stop3 mode.

- 0: low level
- 1: high level

Bits 5-4 STOP3_WU_SEL_G1: Portx Pin Group1 wake-up source selection in Stop3 mode.

- 00: Portx Pin4
- 01: Portx Pin5
- 10: Portx Pin6
- 11: Portx Pin7

Bit 3 STOP3_WU_EN_G0: Portx Pin Group0 wake-up enable control in Stop3 mode.

- 0: disabled
- 1: enabled

Bit 2 STOP3_WU_LVL_G0: Portx Pin Group0 wake-up level selection in Stop3 mode.

- 0: low level
- 1: high level

Bits 1-0 STOP3_WU_SEL_G0: Portx Pin Group0 wake-up source selection in Stop3 mode.

- 00: Portx Pin0
- 01: Portx Pin1
- 10: Portx Pin2
- 11: Portx Pin3

11.14.20 GPIOD_STOP3_WU_CR

Offset: 0x40

Reset Value: 0x00000000

31-8		7	6
RESERVED		STOP3_WU_EN_G1	STOP3_WU_LVL_G1
r-0h		rw-0h	rw-0h
5-4	**3**	**2**	**1-0**
STOP3_WU_SEL_G1	STOP3_WU_EN_G0	STOP3_WU_LVL_G0	STOP3_WU_SEL_G0
rw-0h	rw-0h	rw-0h	rw-0h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bit 7 STOP3_WU_EN_G1: PortD Pin Group1 wake-up enable control in Stop3 mode.

- 0: disabled
- 1: enabled

Bit 6 STOP3_WU_LVL_G1: PortD Pin Group1 wake-up level selection in Stop3 mode.

- 0: low level
- 1: high level

Bits 5-4 STOP3_WU_SEL_G1: PortD Pin Group1 wake-up source selection in Stop3 mode.

- 00: PortD Pin4
- 01: PortD Pin5
- 10: PortD Pin6
- 11: PortD Pin7

Bit 3 STOP3_WU_EN_G0: PortD Pin Group0 Stop3 wake-up enable control in Stop3 mode.

- 0: disabled
- 1: enabled

Bit 2 STOP3_WU_LVL_G0: PortD Pin Group0 Stop3 wake-up level selection in Stop3 mode.

- 0: low level
- 1: high level

Bits 1-0 STOP3_WU_SEL_G0: PortD Pin Group0 Stop3 wake-up source selection in Stop3 mode.

- 00: PortD Pin0
- 01: PortD Pin1
- 10: PortD Pin2
- 11: PortD Pin3

12. LoRa Controller (LoRaC)

12.1 Introduction

LoRa Controller is used to control the internal RF TRX to transmit and reception LoRa signals.

12.2 Main Features

- SPI interface for RF TRX control
- Interrupt signal generation

12.3 Functional Description

12.3.1 Internal SPI Interface

There is an internal SPI interface in the LoRa Controller, which allows the LoRa Controller to directly control RF TRX through registers. The communication between the MCU and RF TRX is as follows:

(1) Initialize the internal SSP in LoRa Controller.

(2) Check the BUSY_DIG_SR bit in register *LORAC_SR* is 0, if it is 0, it means that

RF TRX is currently free for communication.

(3) Clear REG_NSS bit in register *LORAC_NSS_CR*.

(4) Write data into register *SSP_DR* which belonging to the internal SSP of LoRa Controller.

(5) Wait for the transmission to be completed.

(6) Read back the data through register *SSP_DR*.

(7) Repeat Steps 4 ~ Step 6 as required.

(8) Set REG_NSS bit in register *LORAC_NSS_CR*.

12.3.2 Timing Sequence of Power-on

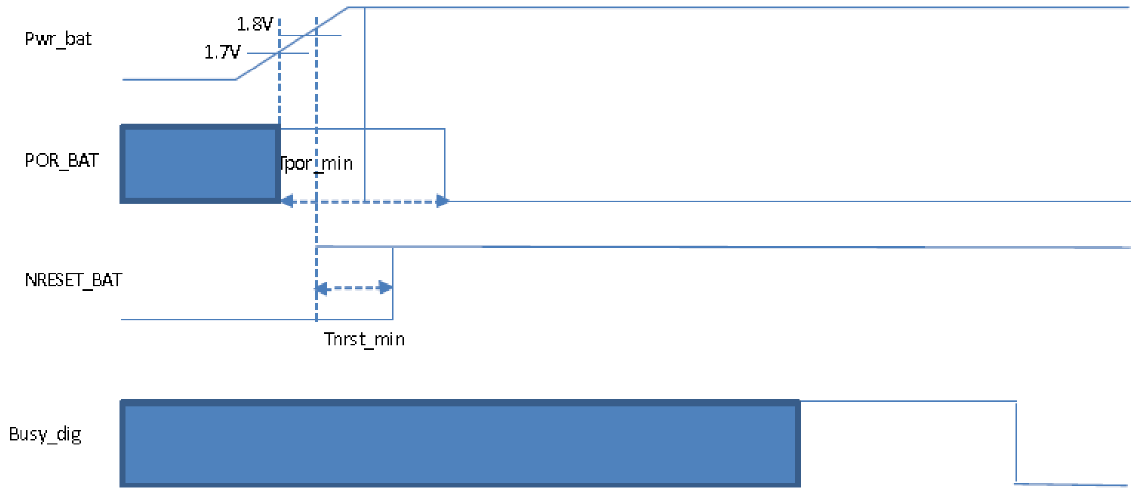

Figure 12-1 Power-on Timing Sequence

As shown in the figure above, the process of power-on is:

(1) Set NRESET_BAT bit in register *LORAC_CR1*.

(2) Clear POR_BAT bit in register *LORAC_CR1*.

(3) Wait for the BUSY_DIG_SR bit in register *LORAC_SR* to be cleared.

Tpor_min is 100 µs and Tnrst_min is 50 µs.

12.3.3 Interrupts

The LoRa Controller transparently transmits the RF TRX interrupt request, and generates the interrupt signal. Note that once the interrupt request of the LoRa Controller is triggered, software must send the *ClearIrqStatus* command to the RF TRX to clear the interrupt, otherwise the interrupt request will be triggered again.

12.4 LoRaC Registers

LORAC Base Address: 0x40009000

Table 12-1 LORAC Registers Summary

Register	Offset	Description
SSP_CR0	0x00	LORAC 内部 SSP 控制寄存器 0
SSP_CR1	0x04	LORAC 内部 SSP 控制寄存器 1
SSP_DR	0x08	LORAC 内部 SSP 数据寄存器
SSP_SR	0x0C	LORAC 内部 SSP 状态寄存器
SSP_CPSR	0x10	LORAC 内部 SSP 时钟分频寄存器
SSP_IMSC	0x14	LORAC 内部 SSP 中断设置寄存器
SSP_RIS	0x18	LORAC 内部 SSP 原始中断状态寄存器
SSP_MIS	0x1C	LORAC 内部 SSP 屏蔽中断状态寄存器
SSP_ICR	0x20	LORAC 内部 SSP 中断清除寄存器
SSP_DMACR	0x24	LORAC 内部 SSP DMA 控制寄存器
RESERVED	0x28-0xFC	保留
LORAC_CR0	0x100	LORAC 控制寄存器 0
LORAC_CR1	0x104	LORAC 控制寄存器 1
LORAC_SR	0x108	LORAC 状态寄存器
LORAC_NSS_CR	0x10C	LORAC NSS 控制寄存器
LORAC_SCK_CR	0x110	LORAC SCK 控制寄存器
LORAC_MOSI_CR	0x114	LORAC MOSI 控制寄存器
LORAC_MISO_SR	0x118	LORAC MISO 状态寄存器

12.4.1 SSP_CR0

Offset: 0x000

Reset Value: 0x00000000

31-16	15-8	7	6	5-4	3-0
RESERVED	SCR	SPH	SPO	FRF	DSS
r	r/w	r/w	r/w	r/w	r/w

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-8 SCR: Serial clock rate, used to set the SSP transfer rate.

$$F_{SSPCLKOUT} = \frac{F_{SSPCLK}}{CPSDVR \times (1+SCR)}$$

The formula to calculate the SSP transfer rate is as above, where CPSDVR is an even number ranging from 2 to 254.

Bit 7 SPH: SSP phase setting, only applied in Motorola SPI format.

Bit 6 SPO: SSP polarity setting, only applied in Motorola SPI format.

Bits 5-4 FRF: SSP frame formats setting.

- 0: Motorola SPI
- 1: Texas Instruments SPI
- 2: National Semiconductor Microwire
- 3: reserved

Bits 3-0 DSS: Data width setting.

- 0: reserved
- 1: reserved
- 2: reserved
- 3: 4 bits
- 4: 5 bits
- 5: 6 bits
- 6: 7 bits
- 7: 8 bits
- 8: 9 bits
- 9: 10 bits
- 10: 11 bits
- 11: 12 bits
- 12: 13 bits
- 13: 14 bits
- 14: 15 bits
- 15: 16 bits

12.4.2 SSP_CR1

Offset: 0x004

Reset Value: 0x00000000

31-4	3	2	1	0
RESERVED	SOD	MS	SSE	LBM
r	r/w	r/w	r/w	r/w

Bits 31-4 RESERVED: Must be kept, and cannot be modified.

Bit 3 SOD: SSP output disable in slave mode.

- 0: SSP output enabled in slave mode
- 1: SSP output disabled in slave mode

Bit 2 MS: Master/slave mode selection.

- 0: master mode
- 1: slave mode

Bit 1 SSE: SSP enable.

- 0: disabled
- 1: enabled

Bit 0 LBM: loopback mode.

- 0: normal mode
- 1: loopback mode

12.4.3 SSP_DR

Offset: 0x008

Reset Value: 0x00000000

31-16	15-0
RESERVED	DATA
r	r/w

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 DATA: SSP TX/RX data.

12.4.4 SSP_SR

Offset: 0x0C

Reset Value: 0x00000003

31-5	4	3	2	1	0
RESERVED	BSY	RFF	RNE	TNF	TFE
r	r	r	r	r	r

Bits 31-5 RESERVED: Must be kept, and cannot be modified.

Bit 4 BSY: SSP busy flag.

- 0: SSP is idle
- 1: SSP transfer in progress

Bit 3 RFF: RX FIFO full flag.

- 0: RX FIFO is not full
- 1: RX FIFO is full

Bit 2 RNE: RX FIFO not empty flag.

- 0: RX FIFO is empty
- 1: RX FIFO is not empty

Bit 1 TNF: TX FIFO not full flag.

- 0: TX FIFO is full
- 1: TX FIFO is not full

Bit 0 TFE: TX FIFO empty flag.

- 0: TX FIFO is not empty
- 1: TX FIFO is empty

12.4.5 SSP_CPSR

Offset: 0x00C

Reset Value: 0x00000000

31-8	7-0
RESERVED	CPSDVSR
r	r/w

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bits 7-0 CPSDVSR: Clock prescaler divider, must be an even number between 2~254.

12.4.6 SSP_IMSC

Offset: 0x010

Reset Value: 0x00000000

31-4	3	2	1	0
RESERVED	TXIM	RXIM	RTIM	RORIM
r	r/w	r/w	r/w	r/w

Bits 31-4 RESERVED: Must be kept, and cannot be modified.

Bit 3 TXIM: TX interrupt mask bit.

- 0: TX interrupt is masked
- 1: TX interrupt is not masked

Bit 2 RXIM: RX interrupt mask bit.

- 0: RX interrupt is masked
- 1: RX interrupt is not masked

Bit 1 RTIM: RX timeout interrupt mask bit.

- 0: RX timeout interrupt is masked
- 1: RX timeout interrupt is not masked

Bit 0 RORIM: RX overrun interrupt mask bit.

- 0: RX overrun interrupt is masked
- 1: RX overrun interrupt is not masked

12.4.7 SSP_RIS

Offset: 0x014

Reset Value: 0x00000008

31-4	3	2	1	0
RESERVED	TXRIS	RXRIS	RTRIS	RORRIS
r	r	r	r	r

Bits 31-4 RESERVED: Must be kept, and cannot be modified.

Bit 3 TXRIS: TX raw interrupt status.

Bit 2 RXRIS: RX raw interrupt status.

Bit 1 RTRIS: RX timeout raw interrupt status.

Bit 0 RORRIS: RX overrun raw interrupt status.

12.4.8 SSP_MIS

Offset: 0x018

Reset Value: 0x00000000

31-4	3	2	1	0
RESERVED	TXMIS	RXMIS	RTMIS	RORMIS
r	r	r	r	r

Bits 31-4 RESERVED: Must be kept, and cannot be modified.

Bit 3 TXMIS: TX masked interrupt status.

Bit 2 RXMIS: RX masked interrupt status.

Bit 1 RTMIS: RX timeout masked interrupt status.

Bit 0 RORMIS: RX overrun masked interrupt status.

12.4.9 SSP_ICR

Offset: 0x01C

Reset Value: 0x00000000

31-2	1	0
RESERVED	RTIC	RORIC
r	w	w

Bits 31-2 RESERVED: Must be kept, and cannot be modified.

Bit 1 RTIC: RX timeout interrupt clear. This bit is cleared by software writing 1 to it, writing 0 has no effect.

Bit 0 RORIC: RX overrun interrupt clear. This bit is cleared by software writing 1 to it, writing 0 has no effect.

12.4.10 SSP_DMACR

Offset: 0x020

Reset Value: 0x00000000

31-2	1	0
RESERVED	TXDMAE	RXDMAE
r	r/w	r/w

Bits 31-2 RESERVED: Must be kept, and cannot be modified.

Bit 1 TXDMAE: DMA TX enable.

- 0: DMA TX disabled
- 1: DMA TX enabled

Bit 0 RXDMAE: DMA RX enable.

- 0: DMA RX disabled
- 1: DMA RX enabled

12.4.11 LORAC_CR0

Offset: 0x100

Reset Value: 0x00000000

31-11	10	9	8	7-5	4-0
RESERVED	NSS_SEL	SCK_MOSI_SEL	RESERVED	IRQ_DIG_INT_EN	RESERVED
r	r/w	r/w	r	r/w	r

Bits 31-11 RESERVED: Must be kept, and cannot be modified.

Bit 10 NSS_SEL: NSS source selection for RF TRX.

- 0: from register LORAC_NSS_CR
- 1: from internal SSP of LORAC

Bit 9 SCK_MOSI_SEL: SCK/MOSI/MISO source selection for RF TRX.

- 0: from LORAC_SCK_CR, LORAC_MOSI_CR and LORA_MISO_SR
- 1: from internal SSP of LORAC

Bit 8 RESERVED: Must be kept, and cannot be modified.

Bits 7-5 IRQ_DIG_INT_EN: IRQ_DIG_INT high level interrupt enable.

Bit[5] corresponds to IRQ_DIG[0], bit[6] corresponds to IRQ_DIG[1] and bit[7] corresponds to IRQ_DIG[2].

- 0: disabled
- 1: enabled

Bits 4-0 RESERVED: Must be kept, and cannot be modified.

12.4.12 LORAC_CR1

Offset: 0x104

Reset Value: 0x00000080

31-8	7	6	5
RESERVED	POR_BAT	RESERVED	NRESET_BAT
r	r/w	r	r/w
4-3	**2**	**1**	**0**
RESERVED	CLK_32M_EN_BAT	TCXO_EN_BAT	PWRTCXO_EN_BAT
r	r/w	r/w	r/w

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bit 7 POR_BAT: POR_BAT control.

- 0: no reset
- 1: reset

Bit 6 RESERVED: Must be kept, and cannot be modified.

Bit 5 NRESET_BAT: NRESET_BAT control.

- 0: reset
- 1: no reset

Bits 4-3 RESERVED: Must be kept, and cannot be modified.

Bit 2 CLK_32M_EN_BAT: CLK_32M_EN_BAT control.

- 0: disabled
- 1: enabled

Bit 1 TCXO_EN_BAT: TCXO_EN_BAT control.

- 0: disabled
- 1: enabled

Bit 0 PWRTCXO_EN_BAT: PWRTCXO_EN_BAT control.

- 0: disabled
- 1: enabled

12.4.13 LORAC_SR

Offset: 0x108

Reset Value: 0x00000100

31-9	8	7-5	4-2	1	0
RESERVED	BUSY_DIG_SR	IRQ_DIG_SR	RESERVED	CLK_32M_RDY_BAT_SR	RESERVED
r	r	r	r	r	r

Bits 31-9 RESERVED: Must be kept, and cannot be modified.

Bit 8 BUSY_DIG_SR: BUSY_DIG status flag, it indicates whether the RF TRX is busy with processing commands. This bit is set and cleared by hardware.

- 0: RF TRX is not busy
- 1: RF TRX is busy with processing commands

Bits 7-5 IRQ_DIG_SR: IRQ_DIG flag, it indicates the RF TRX interrupt request. This bit is set and cleared by hardware. Noted that once the interrupt request is triggered, software must send the *ClearIrqStatus* command to the RF TRX to clear the interrupt, otherwise the interrupt request will be triggered again.

- 0: no interrupt
- 1: interrupt occurred

Bits 4-2 RESERVED: Must be kept, and cannot be modified.

Bit 1 CLK_32M_RDY_BAT_SR: CLK_32M_RDY_BAT status flag, it indicates whether the XO32M clock for RF TRX is ready. This bit is set and cleared by hardware.

- 0: not ready
- 1: ready

Bit 0 RESERVED: Must be kept, and cannot be modified.。

12.4.14 LORAC_NSS_CR

Offset: 0x10C

Reset Value: 0x00000001

31-1	0
RESERVED	REG_NSS
r	r/w

Bits 31-1 RESERVED: Must be kept, and cannot be modified.

Bit 0 REG_NSS: NSS control bit.

- 0: pull down NSS pin
- 1: pull up NSS pin

12.4.15 LORAC_SCK_CR

Offset: 0x110

Reset Value: 0x00000000

31-1	0
RESERVED	REG_SCK
r	r/w

Bits 31-1 RESERVED: Must be kept, and cannot be modified.

Bit 0 REG_SCK: SCK control bit.

- 0: pull down SCK pin
- 1: pull up SCK pin

12.4.16 LORAC_MOSI_CR

Offset: 0x114

Reset Value: 0x00000000

31-1	0
RESERVED	REG_MOSI
r	r/w

Bits 31-1 RESERVED: Must be kept, and cannot be modified.

Bit 0 REG_MOSI: MOSI control bit.

- 0: pull down MOSI pin
- 1: pull up MOSI pin

12.4.17 LORAC_MISO_SR

Offset: 0x118

Reset Value: 0x00000000

31-1	0
RESERVED	REG_MISO
r	r

Bits 31-1 RESERVED: Must be kept, and cannot be modified.

Bit 0 REG_MISO: MISO status flag, it indicates the status of MISO (RF TRX output pin). This bit is set and cleared by hardware.

- 0: low
- 1: high

13. UART

13.1 Introduction

ASR6601 UART unit supports UART and IrDA modes.

The transmit and receive FIFOs are independent, with a depth of 16. The FIFO threshold can be configured to 1/8, 1/4, 1/2, 3/4 and 7/8. Disabling the FIFO is equivalent to a depth of 1 character.

16 -bit baud rate divisor integer part and 6-bit baud rate divisor fractional part. Standard asynchronous communication bit, support 5, 6, 7 and 8-bit data, support parity check, support 1 or 2 stop bits. Sup-port DMA, support false start bit detection, support Line Break generation and detection, support hardware flow control.

The maximum bit rate in IrDA mode is 460800, and the maximum bit rate in Low-Power IrDA mode is 115200, half-duplex. Supports 3/16 and LowPower (1.41~2.23μs) bit lengths. In Low-Power IrDA mode, the bit length is approximated by dividing the UARTCLK frequency.

Each UART port can be uniquely identified by the ID register.

13.2 Clock and Reset

Each UART has independent APB bus clock and independent APB bus reset.

13.3 Reference Clock

UARTCLK frequency must meet the requirements of baud rate generation:

$$F_{UARTCLK(min)} >= 16 \times baudrate_{(max)}$$

$$F_{UARTCLK(max)} <= 16 \times 65535 \times baudrate_{(min)}$$

For example, to generate baud rates from 110 bps to 460800 bps, the UARTCLK frequency must be between 7.3728 MHz to 115.34 MHz.

In the meantime, the UARTCLK frequency cannot be greater than **5 /3** times the frequency of PCLK：

$$F_{UARTCLK} <= {}^{5}/_{3} * F_{PCLK}$$

For example, in UART mode, when UARTCLK is 14.7456 MHz, to generate 921600 baud, PCLK must be greater than or equal to 8.85276 MHz. This ensures that the UART has enough time to write the received data into the receive FIFO.

13.4 Baud Rate Generator

The baud rate generator contains free-running counters that generate the internal ×16 clocks, *Baud16* and *IrLPBaud16*. *Baud16* provides timing information for UART transmission and reception control. *Baud16* is a pulse stream with a width of one UARTCLK clock cycle and a frequency of 16 times the baud rate. *IrLPBaud16* provides timing information to generate the pulse width of the IrDA encoded transmit bit stream in low-power IrDA mode.

13.5 FIFO

The transmit FIFO and receive FIFO are independent, and they are enabled or disabled by the FEN bit in the UART Line Control Register (*UARTx_LCR_H*). The transmit FIFO is an 8-bit wide and 16 deep FIFO memory buffer. The receive FIFO is a 12-bit wide and 16 deep FIFO memory buffer, and it has four extra bits per character for status information. You can program the watermark level to $_1/_8$, $_1/_4$, $_1/_2$, $_3/_4$ or $_7/_8$ for each FIFO through the Interrupt FIFO Level Selection Register (*UARTx_IFLS*). When FIFO is disabled, the depth is 1 byte. The FIFO status can be read from the Flag Register (*UARTx_FR*).
Bits[10:8] of the receive FIFO are error bits indicating associated errors. Bit[11] of the receive FIFO serves as an overrun indicator.

Table 13-1 Receive FIFO Bit Functions

FIFO Bit	Function
11	Overrun 错误
10	Break 错误
9	奇偶校验错误
8	帧错误
7:0	被接收数据

13.6 UART Operation

13.6.1 Baud Rate Divider

The baud rate divisor consists of a 16-bit integer and a 6-bit fractional part. The 16-bit integer is written to register *UARTx_IBRD*. The 6-bit fractional part is written to register *UARTx_FBRD*. The fractional baud rate divider enables the use of any clock with a frequency >3.6864 MHz to act as UARTCLK, while it is still possible to generate all the standard baud rates. The Baud Rate Divisor has the following relationship to UARTCLK:

$$\textbf{\textit{Baud Rate Divisor = UARTCLK / (16 x BautRate) = }} BRD_I + BRD_F$$

BRD_I is the integer part and BRD_F is the fractional part separated by a decimal point:

16-bit Integer Part	.	6-bit Fractional Part

The 6-bit number can be calculated by taking the fractional part of the required baud rate divisor and multiplying it by 64 (that is, 2_n, where n is the effective width of the *UARTx_FBRD* register) and adding 0.5 to account for rounding errors:

$$\textit{Fractional Part} = BRD_F \times 2^n + 0.5$$

13.6.2 Data Transmission

Data received or transmitted is stored in two 16-Byte FIFOs, and the receive FIFO has four extra bits per character for status information.
For transmission, data is written into the TX FIFO through the Data Register (*UARTx_DR*).
Enable the UART through the UARTEN bit in the Data Register (*UARTx_CR*), then data starts transmitting with the data bit, stop bits, parity bit and other parameters indicated in the Line Control Register (*UARTx_LCR_H*) until the TX FIFO is empty.
Once data is written into the TX FIFO, the BUSY signal goes high and remains high while data is being transmitted.

Only when the TX FIFO is empty and the stop bits included in the last character have been transmitted from the shift register, the BUSY signal will go low. Even though the UART is no longer enabled, the BUSY signal is still high.

13.6.3 Data Reception

Enable the UART through the UARTEN bit in the Control Register (*UARTx_CR*) and configure the data width, stop bits, parity bit and other parameters by the Line Control Register (*UARTx_LCR_H*).
When the receiver is idle, UARTRXD is pulled low, Baud16 enables the receive counter to start running, and data is sampled on the 8th cycle of that counter in UART mode or the 4th cycle of the counter in IrDA mode to allow for the shorter logic 0 pulses.
If UARTRXD remains low on the 8th cycle of Baud16, then a valid start bit is detected, otherwise a false start bit is detected and is ignored.
If the start bit is valid, then data sampling is performed every 16th cycle of Baud16 according to the length configured by the WLEN bit in register *UARTx_LCR_H*. If parity mode is enabled, the parity bit will be checked.
Finally, if UARTRXD is high, a valid stop bit is confirmed, otherwise a framing error is occurred. The full character received is stored in the RX FIFO along with the associated error bits.

13.7 IrDA SIR Operation

The IrDA SIR ENDEC provides the function of converting between an UART data stream and half-duplex serial SIR interface. The role of the SIR ENDEC is to provide a digital encoded output, and decoded input to the UART. There are two modes of operation:

- **In IrDA mode**, a zero logic level is transmitted as high pulse, and the pulse width is specified as $_3/_{16}$ of the selected baud rate bit period on the nSIROUT signal, while logic one levels are transmitted as a LOW signal.
- **In low-power IrDA mode**, the width of the transmitted infrared pulse is set to three times the period of the internally generated IrLPBaud16 signal (1.63 µs, assuming a nominal frequency of 1.842 MHz).

The IrDA SIR physical layer specifies a half-duplex communication link, with a minimum 10ms delay between transmission and reception. This delay must be generated by software because it is not supported by the UART. The delay is required because the infrared receiver circuits might become biased.

13.7.1 Low-Power Divider

The IrLPBAUD16 signal is generated by dividing the UARTCLK signal according to the low power divider value configured by the ILPDVSR bit in register *UARTx_ILPR*.

$$\textit{Low-Power Divider} = (F_{UARTCLK} / F_{IrLPBAUD16})$$

$F_{IrLPBAUD16}$ is nominally 1.8432 MHz, which meets the requirement of **1.42MHz < $F_{IrLPBAUD16}$ < 2.12MHz**.

13.7.2 IrDA SIR Transmit Encoder

The SIR transmit encoder modulates the NRZ (Non Return-to-Zero) transmit bit stream output from the UART. The IrDA SIR physical layer specifies use of a RZI (Return to Zero, Inverted) modulation scheme, which represents logic 0 as an infrared light pulse. The modulated output pulse stream is transmitted to an external output driver and infrared LED.

In IrDA mode the transmitted pulse width is specified as three times the period of the internal ×16 clock (Baud16), that is, $3/16$ of a bit period.
In low-power IrDA mode the transmit pulse width is specified as $3/16$ of a 115200 bps bit period. This is implemented as three times the period of a nominal 1.8432 MHz clock (IrLPBaud16). In normal and low-power IrDA modes, when the fractional baud rate divider is used, the transmitted SIR pulse stream includes more jitter. The is because the Baud16 pulses cannot be generated at regular intervals when fractional division is used. That is, the Baud16 cycles have a different number of UARTCLK cycles. The worst case jitter in the SIR pulse stream can be up to three UARTCLK cycles. Provided that the UARTCLK is > 3.6864 MHz and the baud rate used for IrDA mode is ≤ 115200 bps, the jitter is less than 9%. This is within the limits of the SIR IrDA Specification where the maximum amount of jitter permitted is 13%.

13.7.3 IrDA SIR Receive Decoder

The SIR receive decoder demodulates the Return-to-Zero bit stream from the infrared detector and outputs the received NRZ serial bit stream to the UART received data input. The decoder input is normally HIGH in the idle state. The transmit encoder output has the opposite polarity to the decoder input.

START bit is detected when the decoder input is LOW.
To prevent the UART from responding to glitches on the received data input, SIRIN pulses less than $3/16$ of Baud16 will be ignored in IrDA mode; and SIRIN pulses less than $3/16$ of IrLPBaud16 will be ignored in low-power IrDA mode.

13.8 UART Character Frame

The UART character frame is shown below:

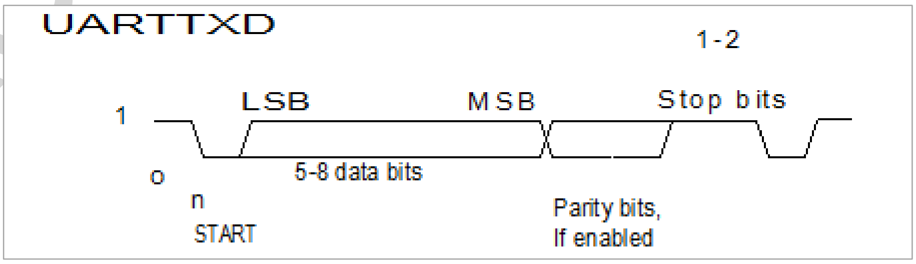

Figure 13-1 UART Character Frame

PDF-XChange Editor Click to BUY NOW! www.pdf-xchange.com

13.9 IrDA Data Modulation

The IrDA $_3/_{16}$ data modulation is shown below.

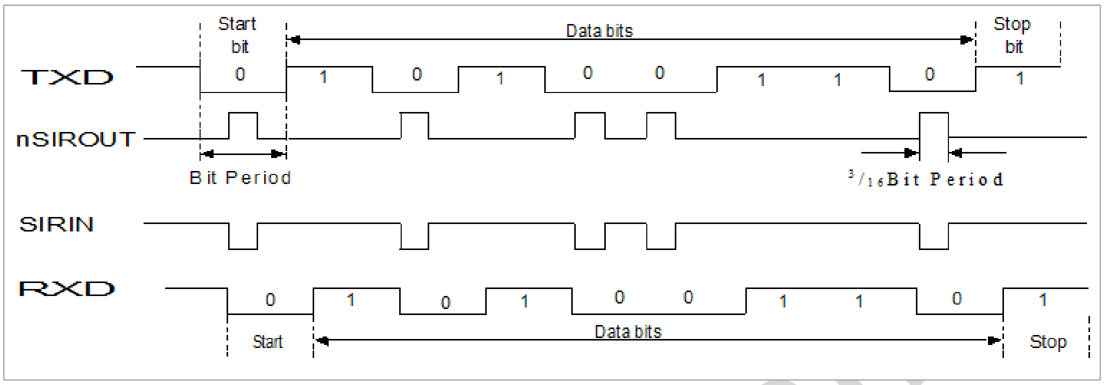

Figure 13-2 IrDA Data Modulation (3/16)

13.10 Hardware Flow Control

The hardware flow control is selectable using the CTSEn and RTSEn bits in the *UARTx_CR* Register.
When RTS flow control is enabled, the nUARTRTS signal is asserted until the receive FIFO is filled up to the watermark level.
When the CTS flow control is enabled, the transmitter can only transfer data when nUARTCTS signal is asserted and the transmit FIFO is not empty.

13.11 Interrupts

The UART supports the generation of Tx Done, Rx Done, Rx Timeout, Frame Error, Break Error, Parity Error and Overrun Error interrupts. The individual interrupts can be enabled or disabled by configuring the mask bits in the Interrupt Mask Set/Clear Register (*UARTx_IMSC*). The status of all interrupt signals, including the interrupt bits that are disabled, can be read from the Raw Interrupt Status Register (*UARTx_RIS*). The status of the enabled interrupt signals can be read from the Masked Interrupt Status Register (*UARTx_MIS*). The interrupt is cleared by writing “1” to the corresponding bit in the Interrupt Clear Register (*UARTx_ICR*).

13.12 DMA

The UART module supports DMA transmission and reception, which is configured by register *UARTx_DMACR*.

13.13 UART Registers

UART0 Base Address: 0x40003000

UART1 Base Address: 0x40004000

UART2 Base Address: 0x40010000

UART3 Base Address: 0x40011000

Table 13-2 UART Registers Summary

Register	Offset	Description
UARTx_DR	0x00	数据寄存器
UARTx_RSR_ECR	0x04	接收状态寄存器/错误清除寄存器
UARTx_RSV0[4]	0x08	4 x 4 字节保留
UARTx_FR	0x18	标志寄存器
UARTx_RSV1	0x1C	4 字节保留
UARTx_ILPR	0x20	红外低功耗计数寄存器
UARTx_IBRD	0x24	波特率整数寄存器
UARTx_FBRD	0x28	波特率小数寄存器
UARTx_LCR_H	0x2C	线控寄存器
UARTx_CR	0x30	控制寄存器
UARTx_IFLS	0x34	中断 FIFO 水位选择寄存器
UARTx_IMSC	0x38	中断掩码设置/清除寄存器
UARTx_RIS	0x3C	原始中断状态寄存器
UARTx_MIS	0x40	被掩中断状态寄存器
UARTx_ICR	0x44	中断清除寄存器
UARTx_DMACR	0x48	DMA 控制寄存器
UARTx_RSV2[997]	0x4C	4 x 997 字节保留

13.13.1 UARTx_DR (x=0, 1, 2, 3)

Offset: 0x000

Reset Value: 0x00000000

31-12	11	10	9	8	7-0
RESERVED	OE	BE	PE	FE	DATA
r-0h	r-0h	r-0h	r-0h	r-0h	rw-0h

Bits 31-12 RESERVED: Must be kept, and cannot be modified.

Bit 11 OE: Overrun error flag.

- 0: no overrun error
- 1: overrun occurred

Bit 10 BE: Break error flag.

- 0: no break error
- 1: break error occurred

When this bit is set, it indicates that the received data input was held LOW for longer than a full-word (defined as start, data, parity and stop bits) transmission time.

In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO.

Bit 9 PE: Parity error flag.

- 0: no parity error
- 1: parity error occurred

When this bit is set, it indicates that the parity of the received data character does not match the configuration of the EPS bit in the *UARTx_LCR_H* Register.

In FIFO mode, this error is associated with the character at the top of the FIFO.

Bit 8 FE: Framing error flag.

- 0: no framing error
- 1: framing error occurred

When this bit is set, it indicates that the received character did not have a valid stop bit.

In FIFO mode, this error is associated with the character at the top of the FIFO.

Bits 7-0 DATA: Transmit data character/Receive data character.

13.13.2 UARTx_RSR_ECR (x=0, 1, 2, 3)

Offset: 0x004

Reset Value: 0x00000000

31-4	3	2	1	0
RESERVED	OE	BE	PE	FE
r-0h	r-0h	r-0h	r-0h	r-0h

Bits 31-4 RESERVED: Must be kept, and cannot be modified.

Bit 3 OE: Overrun error flag.

- 0: no overrun error
- 1: overrun occurred

Bit 2 BE: Break error flag.

- 0: no break error
- 1: break error occurred

When this bit is set, it indicates that the received data input was held LOW for longer than a full-word (defined as start, data, parity and stop bits) transmission time.
In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO.

Bit 1 PE: Parity error flag.

- 0: no parity error
- 1: parity error occurred

When this bit is set, it indicates that the parity of the received data character does not match the configuration of the EPS bit in the *UARTx_LCR_H* Register.

In FIFO mode, this error is associated with the character at the top of the FIFO.

Bit 0 FE: Framing error flag.

- 0: no framing error
- 1: framing error occurred

When this bit is set, it indicates that the received character did not have a valid stop bit.

In FIFO mode, this error is associated with the character at the top of the FIFO.

13.13.3 UARTx_FR (x=0, 1, 2, 3)

Offset: 0x018

Reset Value: 0x00000000

31-8	7	6	5	4	3	2-0
RESERVED	TXFE	RXFF	TXFF	RXFE	BUSY	RESERVED
r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bit 7 TXFE: Transmit FIFO empty.

- 0: transmit FIFO/UART_DR register not empty
- 1: transmit FIFO/UART_DR empty

This bit is associated with the FEN bit in the *UARTx_LCR_H* register.

This bit does not indicate if there is data in the transmit shift register.

Bit 6 RXFF: Receive FIFO full.

- 0: receive FIFO/UART_DR not full
- 1: receive FIFO/UART_DR full

This bit is associated with the FEN bit in the *UARTx_LCR_H* register.

Bit 5 TXFF: Transmit FIFO full.

- 0: transmit FIFO/UART_DR not full
- 1: transmit FIFO/UART_DR full

This bit is associated with the FEN bit in the *UARTx_LCR_H* register.

Bit 4 RXFE: Receive FIFO empty.

- 0: receive FIFO/UART_DR not empty
- 1: receive FIFO/UART_DR empty

This bit is associated with the FEN bit in the *UARTx_LCR_H* register.

Bit 3 BUSY: UART busy.

- 0: no transfer
- 1: transfer in progress

This bit is set to 1 as soon as the transmit FIFO becomes non-empty, irrespective of whether the UART is enabled or not.

Bits 2-0 RESERVED: Must be kept, and cannot be modified.

13.13.4 UARTx_ILPR (x=0, 1, 2, 3)

Offset: 0x020

Reset Value: 0x00000000

31-8	7-0
RESERVED	ILPDVSR
r-0h	rw-0h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bits 7-0 ILPDVSR: low-power divisor value. Zero is an illegal value. Writing “0” results in no generation of IrLPBaud16 pulses.

13.13.5 UARTx_IBRD (x=0, 1, 2, 3)

Offset: 0x024

Reset Value: 0x00000000

31-16	15-0
RESERVED	BAUD_DIVINT
r-0h	rw-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 BAUD_DIVINT: The integer baud rate divisor.

13.13.6 UARTx_FBRD (x=0, 1, 2, 3)

Offset: 0x028

Reset Value: 0x00000000

31-6	5-0
RESERVED	BAUD_DIVFRAC
r-0h	rw-0h

Bits 31-6 RESERVED: Must be kept, and cannot be modified.

Bits 5-0 BAUD_DIVFRAC: The fractional baud rate divisor.

13.13.7 UARTx_LCR_H (x=0, 1, 2, 3)

Offset: 0x02C

Reset Value: 0x00000000

31-7	6-5	4	3	2	1	0
RESERVED	WLEN	FEN	STP2	EPS	PEN	BRK
r-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

Bits 31-7 RESERVED: Must be kept, and cannot be modified.

Bits 6-5 WLEN: Word length.

- 00: 5 bits
- 01: 6 bits
- 10: 7 bits
- 11: 8 bits

Bit 4 FEN: FIFO enable.

- 0: FIFO disabled
- 1: FIFO enabled

Bit 3 STP2: Stop bits selection.

- 0: 1 stop bit
- 1: 2 stop bits

Bit 2 EPS: Parity selection.

- 0: odd parity
- 1: even parity

This bit has no effect if the PEN bit is 0.

Bit 1 PEN: Parity enable.

- 0: parity checking disabled
- 1: parity checking enabled.

Bit 0 BRK: Send break.

- write 0: end the Break command
- write 1: a low-level is continually output on the UART_TXD pin, after completing transmission of the current character.

For the proper execution of the Break command, the software must set this bit for at least two complete frames.

13.13.8 UARTx_CR (x=0, 1, 2, 3)

Offset: 0x030

Reset Value: 0x00000000

<table>
<tr><th colspan="2">31-24</th><th>23-16</th><th>15</th><th>14</th><th>13-10</th></tr>
<tr><td colspan="2">RESERVED</td><td>RESERVED</td><td>CTSEn</td><td>RTSEn</td><td>RESERVED</td></tr>
<tr><td colspan="2">r-0h</td><td>r-0h</td><td>rw-0h</td><td>rw-0h</td><td>r-0h</td></tr>
<tr><th>9</th><th>8</th><th>7-3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>RXE</td><td>TXE</td><td>RESERVED</td><td>SIRLP</td><td>SIREN</td><td>UARTEN</td></tr>
<tr><td>rw-0h</td><td>rw-0h</td><td>r-0h</td><td>rw-0h</td><td>rw-0h</td><td>rw-0h</td></tr>
</table>

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bit 15 CTSEn: CTS hardware flow control enable.

- 0: CTS hardware flow control disabled
- 1: CTS hardware flow control enabled

Bit 14 RTSEn: RTS hardware flow control enable.

- 0: RTS hardware flow control disabled
- 1: RTS hardware flow control enabled

Bits 13-10 RESERVED: Must be kept, and cannot be modified.

Bit 9 RXE: Receiver enable.

- 0: receiver disabled. If the UART is disabled during of reception, then it completes the current character before stopping.
- 1: receiver enabled

Bit 8 TXE: Transmitter enable.

- 0: transmitter disabled. If the UART is disabled during transmission, then it completes the current character before stopping.
- 1: transmitter enabled

Bits 7-3 RESERVED: Must be kept, and cannot be modified.

Bit 2 SIRLP: Low-power IrDA SIR encoding mode selection.

- 0: low-level bits are transmitted with a pulse width of **3 /16** of the bit period.
- 1: low-level bits are transmitted with a pulse width of 3 times the period of the IrLPBaud16 input signal. Setting this bit helps reduce power consumption, but might reduce transmission distances.

Bit 1 SIRE: IrDA SIR enable.

- 0: IrDA SIR ENDEC is disabled. SIR_OUT remains LOW, and signal transitions on SIR_IN are ignored. Data is transmitted and received on UART_TXD and UART_RXD.
- 1: IrDA SIR ENDEC is enabled. UARTTXD remains HIGH, and signal transitions on UART_RXD are ignored. Data is transmitted and received on SIR_OUT and SIR_IN.

  This bit has no effect if the UARTEN bit is 0.

Bit 0 UARTEN: UART enable.

- 0: UART disabled. If the UART is disabled during transmission or reception, then it completes the current character before stopping.
- 1: UART enabled

13.13.9 UARTx_IFLS (x=0, 1, 2, 3)

Offset: 0x034

Reset Value: 0x00000000

31-6	5-3	2-0
RESERVED	RXIFLSEL	TXIFLSEL
r-0h	rw-0h	rw-0h

Bits 31-6 RESERVED: Must be kept, and cannot be modified.

Bits 5-3 RXIFLSEL: Receive interrupt FIFO level selection.

- 000: receive FIFO becomes ≥ **1/8** full
- 001: receive FIFO becomes ≥ **1/4** full
- 010: receive FIFO becomes ≥ **1/2** full
- 011: receive FIFO becomes ≥ **3/4** full
- 100: receive FIFO becomes ≥ **7/8** full
- 101~111: reserved

Bits 2-0 TXIFLSEL: Transmit interrupt FIFO level selection.

- 000: transmit FIFO becomes ≥ **1/8** full
- 001: transmit FIFO becomes ≥ **1/4** full
- 010: transmit FIFO becomes ≥ **1/2** full
- 011: transmit FIFO becomes ≥ **3/4** full
- 100: transmit FIFO becomes ≥ **7/8** full
- 101~111: reserved

13.13.10 UARTx_IMSC (x=0, 1, 2, 3)

Offset: 0x038

Reset Value: 0x00000000

31-16	15-11	10	9	8	7	6	5	4	3-0
RESERVED	RESERVED	OEIM	BEIM	PEIM	FEIM	RTIM	TXIM	RXIM	RESERVED
r-0h	r-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	r-0h

Bits 31-11 RESERVED: Must be kept, and cannot be modified.

Bit 10 OEIM: Overrun error interrupt mask bit.

- 0: overrun error interrupt disabled
- 1: overrun error interrupt enabled

Bit 9 BEIM: Break error interrupt mask bit.

- 0: break error interrupt disabled
- 1: break error interrupt enabled

Bit 8 PEIM: Parity error interrupt mask bit.

- 0: parity error interrupt disabled

- 1: parity error interrupt enabled

Bit 7 FEIM: Framing error interrupt mask bit.

- 0: framing error interrupt disabled
- 1: framing error interrupt enabled

Bit 6 RTIM: Receive timeout interrupt mask bit.

- 0: receive timeout interrupt disabled
- 1: receive timeout interrupt enabled

Bit 5 TXIM: Transmission completion interrupt mask bit.

- 0: transmission completion interrupt disabled
- 1: transmission completion interrupt enabled

Bit 4 RXIM: Reception completion interrupt mask bit.

- 0: reception completion interrupt disabled
- 1: transmission completion interrupt enabled

Bits 3-0 RESERVED: Must be kept, and cannot be modified.

13.13.11 UARTx_RIS (x=0, 1, 2, 3)

Offset: 0x03C

Reset Value: 0x00000000

31-16	15-11	10	9	8	7	6	5	4	3-0
RESERVED	RESERVED	OERIS	BERIS	PERIS	FERIS	RTRIS	TXRIS	RXRIS	RESERVED
r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h

Bits 31-11 RESERVED: Must be kept, and cannot be modified.

Bit 10 OERIS: Overrun error raw interrupt status.

Bit 9 BERIS: Break error raw interrupt status.

Bit 8 PERIS: Parity error raw interrupt status.

Bit 7 FERIS: Framing error raw interrupt status.

Bit 6 RTRIS: Receive timeout raw interrupt status.

Bit 5 TXRIS: Transmission completion raw interrupt status.

Bit 4 RXRIS: Reception completion raw interrupt status.

Bits 3-0 RESERVED: Must be kept, and cannot be modified.

13.13.12 UARTx_MIS (x=0, 1, 2, 3)

Offset: 0x040

Reset Value: 0x00000000

31-16	15-11	10	9	8	7	6	5	4	3-0
RESERVED	RESERVED	OEMIS	BEMIS	PEMIS	FEMIS	RTMIS	TXMIS	RXMIS	RESERVED
r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h

Bits 31-11 RESERVED: Must be kept, and cannot be modified.
Bit 10 OEMIS: Overrun error masked interrupt status.
Bit 9 BEMIS: Break error masked interrupt status.
Bit 8 PEMIS: Parity error masked interrupt status.
Bit 7 FEMIS: Framing error masked interrupt status.
Bit 6 RTMIS: Receive timeout masked interrupt status.
Bit 5 TXMIS: Transmission completion masked interrupt status.
Bit 4 RXMIS: Reception completion masked interrupt status.
Bits 3-0 RESERVED: Must be kept, and cannot be modified.

13.13.13 UARTx_ICR (x=0, 1, 2, 3)

偏移量：0x44

复位值：0x00000000

31-16	15-11	10	9	8	7	6	5	4	3-0
RESERVED	RESERVED	OEIC	BEIC	PEIC	FEIC	RTIC	TXIC	RXIC	RESERVED
r-0h	r-0h	w-0h	w-0h	w-0h	w-0h	w-0h	w-0h	w-0h	r-0h

Bits 31-11 RESERVED: Must be kept, and cannot be modified.

Bit 10 OEIC: Clear Overrun Error interrupt.

- 0: no action
- 1: clear overrun error interrupt

Bit 9 BEIC: Clear Break Error interrupt.

- 0: no action
- 1: clear break error interrupt

Bit 8 PEIC: Clear Parity Error interrupt.

- 0: no action
- 1: clear parity error interrupt

Bit 7 FEIC: Clear Framing Error interrupt.

- 0: no action
- 1: clear framing error interrupt

Bit 6 RTIC: Clear Receive Timeout interrupt.

- 0: no action
- 1: clear receive timeout interrupt

Bit 5 TXIC: Clear Transmission Completion interrupt.

- 0: no action
- 1: clear transmission completion interrupt

Bit 4 RXIC: Clear Reception Completion interrupt.

- 0: no action
- 1: clear reception completion interrupt

Bits 3-0 RESERVED: Must be kept, and cannot be modified.

13.13.14 UARTx_DMACR (x=0, 1, 2, 3)

Offset: 0x048

Reset Value: 0x00000000

31-3	2	1	0
RESERVED	DMAONERR	TXDMAE	RXDMAE
r-0h	rw-0h	rw-0h	rw-0h

Bits 31-3 RESERVED: Must be kept, and cannot be modified.

Bit 2 DMAONERR: DMA on error.

Bit 1 TXDMAE: Transmit DMA enable.

- 0: disabled
- 1: enabled

Bit 0 RXDMAE: Receive DMA enable.

- 0: disabled
- 1: enabled

13.13.15 UARTx_ID[8] (x=0, 1, 2, 3)

13.13.15.1 PeriphID0

Offset: 0x0FE0

Reset Value: 0x00000000

31-8	7-0
RESERVED	PARTNUMBER0
r-0h	r-11h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bits 7-0 PARTNUMBER0: 0x11.

13.13.15.2 PeriphID1

Offset: 0x0FE4

Reset Value: 0x00000000

31-8	7-4	3-0
RESERVED	DESIGNER0	PARTNUMBER1
r-0h	r-1h	r-0h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bits 7-4 DESIGNER0: 0x1.

Bits 3-0 PARTNUMBER1: 0x0.

13.13.15.3 PeriphID2

Offset: 0x0FE8

Reset Value: 0x00000000

31-8	7-4	3-0
RESERVED	REVISION0	DESIGNER1
r-0h	r-xh	r-0h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bits 7-4 REVISION0:

- 0x0: r1p0
- 0x1: r1p1
- 0x2: r1p3/r1p4
- 0x3: r1p5

Bits 3-0 DESIGNER1: 0x0.

13.13.15.4 PeriphID3

Offset: 0x0FEC

Reset Value: 0x00000000

31-8	7-0
RESERVED	CONFIGURATION
r-0h	r-0h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bits 7-0 CONFIGURATION: 0x00.

13.13.15.5 PCellID0

Offset: 0x0FD0

Reset Value: 0x00000000

31-8	7-0
RESERVED	CellID0
r-0h	r-dh

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bits 7-0 CellID0: 0x0d.

13.13.15.6 PCellID1

Offset: 0x0FD4

Reset Value: 0x00000000

31-8	7-0
RESERVED	CellID1
r-0h	r-f0h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bits 7-0 CellID1: 0xf0.

13.13.15.7 PCellID2

Offset: 0x0FD8

Reset Value: 0x00000000

31-8	7-0
RESERVED	CellID2
r-0h	r-5h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bits 7-0 CellID2**:** 0x05.

13.13.15.8 PCellID3

Offset: 0x0FDC

Reset Value: 0x00000000

31-8	7-0
RESERVED	CellID3
r-0h	r-b1h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bits 7-0 CellID3: 0xb1.

14. SSP

14.1 Introduction

All three SSP (synchronous serial port) can be configured as a master or slave device.

SSP support multiple frame formats, with configurable data width and transfer rate.

14.2 Main Features

- Master or slave operation
- Up to 16 MHz output
- 16-bit wide TX/RX FIFO with a depth of 8
- Multiple frame formats
- 4-bit to 16-bit data width
- DMA
- Interrupt signal generation

14.3 Functional Description

14.3.1 Basic Information

Four I/O pins (SSP_NSS, SSP_CLK, SSP_TX, SSP_RX) are dedicated to SSP communication with external devices.

1. **SSP_NSS**

   The chip select pin is active at low level.

2. **SSP_CLK**

   SSP clock pin acts as clock output in master mode and as the clock input in slave mode.

3. **SSP_TX**

   SSP TX pin is used to transmit data in both master and slave modes.

4. **SSP_RX**

   SSP RX pin is used to receive data in both master and slave modes.
   The connection between SSP and SPI device is shown in the figure below. Note the difference between SSP_TX/SSP_RX and SPI_MOSI/SPI_MISO.

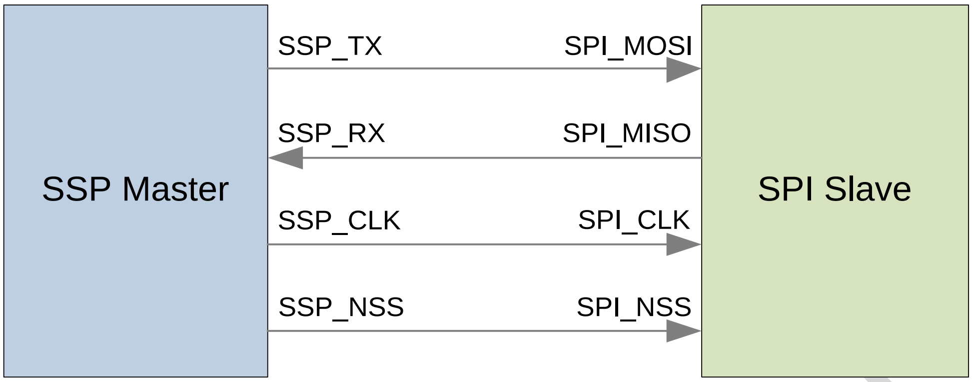

Figure 14-1 Connection between a SSP Master and a SPI Slave

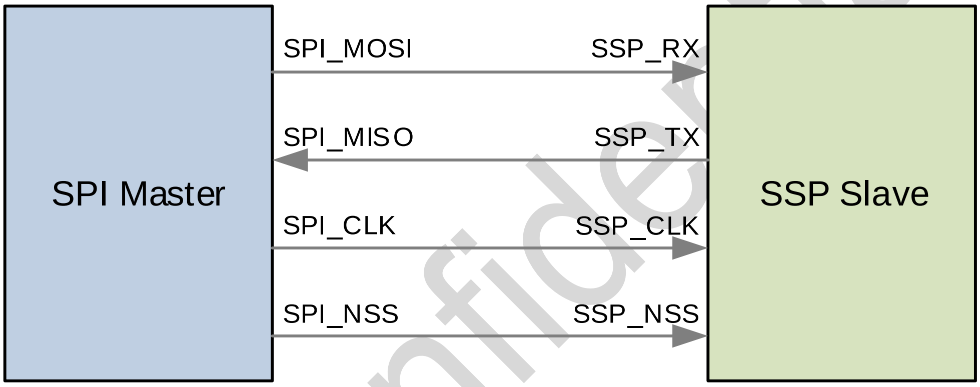

Figure 14-2 Connection between a SPI Master and a SSP Slave

14.3.2 Clock Division

SSP clock should meet below requirements:

(1) Up to 16 MHz output clock

(2) The clock frequency in master mode is at most 1/2 of PCLK clock frequency

(3) The clock frequency in slave mode is at most 1/12 of PCLK clock frequency

The formula to calculate clock output in master mode is below:

$$F_{SSPCLKOUT} = \frac{F_{SSPCLK}}{CPSDVR \times (1+SCR)}$$

Figure 14-3 MASTER mode clock output calculation

SSPCLK is the interface clock for SSP, and SSPCLKOUT is the output clock. For example, SSPCLK is 24MHz by default and 1MHz SSPCLKOUT is required, then the user should set bit CPSDVR in register *SSP_CPSR* to 2, and set bit SCR in register *SSP_CR0* to 11.

14.3.3 Data Format

SSP supports three frame formats:

- Motorola SPI
- Texas Instruments SPI
- National Semiconductor Microwire

14.3.4 DMA Transfer

SSP DMA Transfer Process:

(1) Enable the TXDMAE bit in register *SSP_DMACR*.
(2) Configure *SSP_DR* register as the destination address of DMA.
(3) Configure the memory address of the data to be sent as the source address of DMA.
(4) Configure the data width of DMA transfer to 8 bits by configuring the SRC_TR_WIDTH and DES_TR_WIDTH bits to 0 in the *DMA_CTLx* register.
(5) Configure the DMA burst length to 4 by configuring the SRC_MSIZE and DEST_MSIZE bits to 1 in the *DMA_CTLx* register.
(6) Configure the total length of DMA data transfer.
(7) Configure DMA handshake type to the corresponding SSP TX type (for example, for SSP0, configure it to DMA_HANDSHAKE_SSP_0_TX).
(8) Activate the DMA.

When the DMA transfer is completed, the CH_EN_x bit in the DMA_CHENREG register is cleared.

SSP DMA Reception Process:

(1) Enable the RXDMAE bit in register *SSP_DMACR*.
(2) Configure register *SSP_DR* as the source address of DMA.
(3) Configure the memory address of the data to be received as the destination address of DMA.
(4) Configure the data width of DMA transfer to 8 bits by configuring the SRC_TR_WIDTH and DES_TR_WIDTH bits to 0 in the *DMA_CTLx* register.
(5) Configure the DMA burst length to 4 by configuring the SRC_MSIZE and DEST_MSIZE bits to 1 in the *DMA_CTLx* register.
(6) Configure the total length of DMA data transfer.
(7) Configure DMA handshake type to the corresponding SSP RX type (for example, for SSP0, configure it to DMA_HANDSHAKE_SSP_0_RX).
(8) Activate the DMA.

When the DMA transfer is completed, the CH_EN_x bit in the DMA_CHENREG register is cleared.

14.3.5 Interrupts

SSP has four interrupts: SSP RX interrupt, SSP TX interrupt, SSP RX OVERRUN interrupt and SSP RX TIMEOUT.

1. SSP RX Interrupt

SSP RX interrupt is triggered when there are 4 or more locations in SSP RX FIFO.

2. SSP TX Interrupt

SSP TX interrupt is triggered when there are 4 or less locations in SSP TX FIFO.

3. SSP RX Overrun Interrupt

SSP RX overrun interrupt is triggered when the SSP RX FIFO is full and continues to receive data.

4. SSP RX Timeout Interrupt

SSP RX timeout interrupt is triggered when the SSP RX FIFO is not empty but SSP has not received any new data for the duration time of 32-bit data transfer.

14.4 SSP Registers

SSP0 Base Address: 0x40006000

SSP1 Base Address: 0x40012000

SSP2 Base Address: 0x40013000

Table 14-1 SSP Registers Summary

Register	Offset	Description
SSP_CR0	0x00	Control register 0
SSP_CR1	0x04	Control register 1
SSP_DR	0x08	Data register
SSP_SR	0x0C	Status register
SSP_CPSR	0x10	Clock Prescaler Register
SSP_IMSC	0x14	Interrupt Mask Set/Clear Register
SSP_RIS	0x18	Raw Interrupt Status register
SSP_MIS	0x1C	Masked Interrupt Status register
SSP_ICR	0x20	Interrupt Clear Register
SSP_DMACR	0x24	DMA Control Register

14.4.1 SSP_CR0

Offset: 0x000

Reset Value: 0x00000000

31-16	15-8	7	6	5-4	3-0
RESERVED	SCR	SPH	SPO	FRF	DSS
r	r/w	r/w	r/w	r/w	r/w

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-8 SCR: Serial clock rate, used to set the SSP transfer rate.

$$F_{SSPCLKOUT} = \frac{F_{SSPCLK}}{CPSDVR \times (1+SCR)}$$

The formula to calculate the SSP transfer rate is as above, where CPSDVR is an even number ranging from 2 to 254.

Bit 7 SPH: SSP phase setting, only applied in Motorola SPI format.

Bit 6 SPO: SSP polarity setting, only applied in Motorola SPI format.

Bits 5-4 FRF: SSP frame formats setting.

- 0: Motorola SPI
- 1: Texas Instruments SPI
- 2: National Semiconductor Microwire
- 3: reserved

Bits 3-0 DSS: Data width setting.

- 0: reserved
- 1: reserved
- 2: reserved
- 3: 4 bits
- 4: 5 bits
- 5: 6 bits
- 6: 7 bits
- 7: 8 bits
- 8: 9 bits
- 9: 10 bits
- 10: 11 bits
- 11: 12 bits
- 12: 13 bits
- 13: 14 bits
- 14: 15 bits
- 15: 16 bits

14.4.2 SSP_CR1

Offset: 0x004

Reset Value: 0x00000000

31-4	3	2	1	0
RESERVED	SOD	MS	SSE	LBM
r	r/w	r/w	r/w	r/w

Bits 31-4 RESERVED: Must be kept, and cannot be modified.

Bit 3 SOD: SSP output disable in slave mode.

- 0: SSP output enabled in slave mode
- 1: SSP output disabled in slave mode

Bit 2 MS: Master/slave mode selection.

- 0: master mode
- 1: slave mode

Bit 1 SSE: SSP enable.

- 0: disabled
- 1: enabled

Bit 0 LBM: loopback mode.

- 0: normal mode
- 1: loopback mode

14.4.3 SSP_DR

Offset: 0x008

Reset Value: 0x00000000

31-16	15-0
RESERVED	DATA
r	r/w

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 DATA: SSP TX/RX data.

14.4.4 SSP_SR

Offset: 0x00C

Reset Value: 0x00000003

31-5	4	3	2	1	0
RESERVED	BSY	RFF	RNE	TNF	TFE
r	r	r	r	r	r

Bits 31-5 RESERVED: Must be kept, and cannot be modified.

Bit 4 BSY: SSP busy flag.

- 0: SSP is idle
- 1: SSP transfer in progress

Bit 3 RFF: RX FIFO full flag.

- 0: RX FIFO is not full
- 1: RX FIFO is full

Bit 2 RNE: RX FIFO not empty flag.

- 0: RX FIFO is empty
- 1: RX FIFO is not empty

Bit 1 TNF: TX FIFO not full flag.

- 0: TX FIFO is full
- 1: TX FIFO is not full

Bit 0 TFE: TX FIFO empty flag.

- 0: TX FIFO is not empty
- 1: TX FIFO is empty

14.4.5 SSP_CPSR

Offset: 0x010

Reset Value: 0x00000000

31-8	7-0
RESERVED	CPSDVSR
r	r/w

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bits 7-0 CPSDVSR: Clock prescaler divider, must be an even number between 2~254.

14.4.6 SSP_IMSC

偏移量：0x00

复位值：0x00000000

31-4	3	2	1	0
RESERVED	TXIM	RXIM	RTIM	RORIM
r	r/w	r/w	r/w	r/w

Bits 31-4 RESERVED: Must be kept, and cannot be modified.

Bit 3 TXIM: TX interrupt mask bit.

- 0: TX interrupt is masked
- 1: TX interrupt is not masked

Bit 2 RXIM: RX interrupt mask bit.

- 0: RX interrupt is masked
- 1: RX interrupt is not masked

Bit 1 RTIM: RX timeout interrupt mask bit.

- 0: RX timeout interrupt is masked
- 1: RX timeout interrupt is not masked

Bit 0 RORIM: RX overrun interrupt mask bit.

- 0: RX overrun interrupt is masked
- 1: RX overrun interrupt is not masked

14.4.7 SSP_RIS

Offset: 0x018

Reset Value: 0x00000008

31-4	3	2	1	0
RESERVED	TXRIS	RXRIS	RTRIS	RORRIS
r	r	r	r	r

Bits 31-4 RESERVED: Must be kept, and cannot be modified.

Bit 3 TXRIS: TX raw interrupt status.

Bit 2 RXRIS: RX raw interrupt status.

Bit 1 RTRIS: RX timeout raw interrupt status.

Bit 0 RORRIS: RX overrun raw interrupt status.

14.4.8 SSP_MIS

Offset: 0x01C

Reset Value: 0x00000000

31-4	3	2	1	0
RESERVED	TXMIS	RXMIS	RTMIS	RORMIS
r	r	r	r	r

Bits 31-4 RESERVED: Must be kept, and cannot be modified.

Bit 3 TXMIS: TX masked interrupt status.

Bit 2 RXMIS: RX masked interrupt status.

Bit 1 RTMIS: RX timeout masked interrupt status.

Bit 0 RORMIS: RX overrun masked interrupt status.

14.4.9 SSP_ICR

Offset: 0x020

Reset Value: 0x00000000

31-2	1	0
RESERVED	RTIC	RORIC
r	w	w

Bits 31-2 RESERVED: Must be kept, and cannot be modified.

Bit 1 RTIC: RX timeout interrupt clear. This bit is cleared by software writing 1 to it, writing 0 has no effect.

Bit 0 RORIC: RX overrun interrupt clear. This bit is cleared by software writing 1 to it, writing 0 has no effect.

14.4.10 SSP_DMACR

Offset: 0x024

Reset Value: 0x00000000

31-2	1	0
RESERVED	TXDMAE	RXDMAE
r	r/w	r/w

Bits 31-2 RESERVED: Must be kept, and cannot be modified.

Bit 1 TXDMAE: DMA TX enable.

- 0: DMA TX disabled
- 1: DMA TX enabled

Bit 0 RXDMAE: DMA RX enable.

- 0: DMA RX disabled
- 1: DMA RX enabled

15. I2C

15.1 Introduction

The I2C bus interface unit supports master mode and slave mode. SDA is the data transmission line, and SCL is the interface clock line. It supports multi-host and bus arbitration functions. It supports 100Kbps standard rate mode and 400Kbps fast mode. It supports FIFO mode, the transmit FIFO depth is 8, the receive FIFO depth is 16, and the read and write pointers of FIFO are configurable.

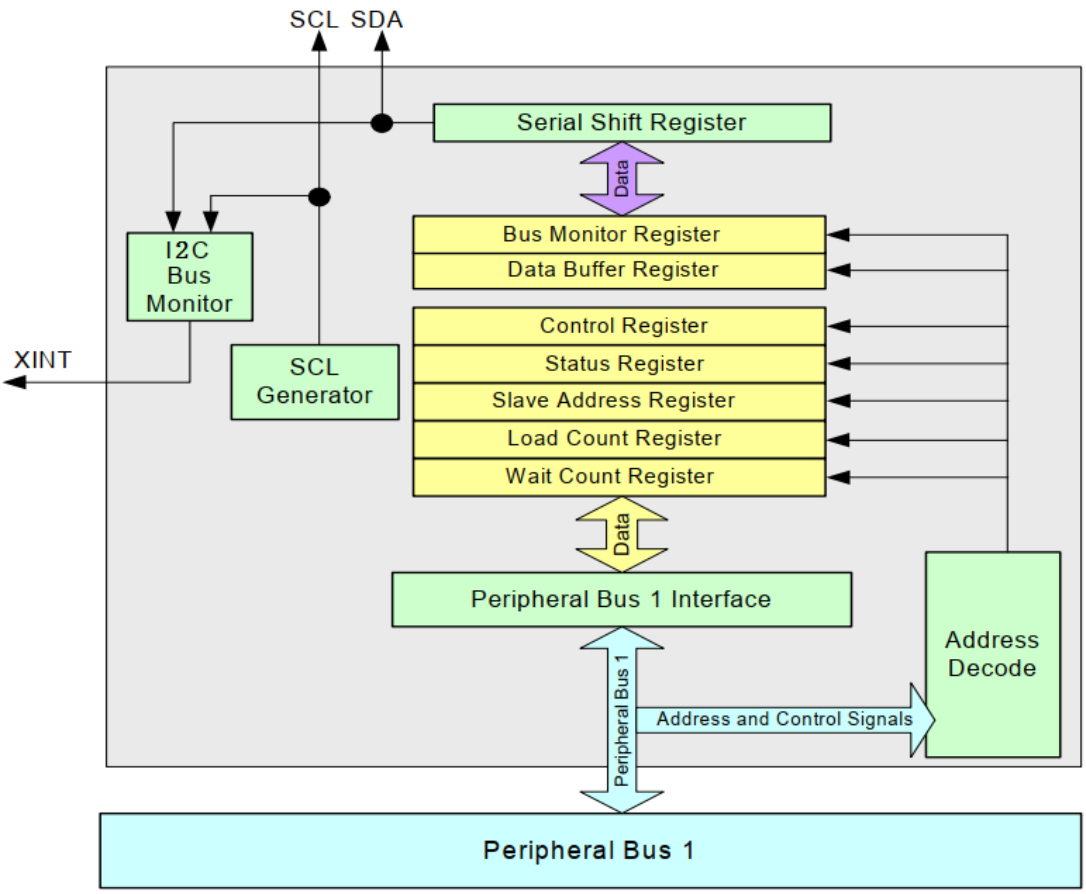

Figure 15-1 I2C Block Diagram

15.2 Start and Stop Conditions

Start condition: When SCL level is high, SDA level changes from high to low, thus generating a Start condition.

Stop condition: When SCL level is high, SDA level changes from low to high, thus generating a Stop condition.

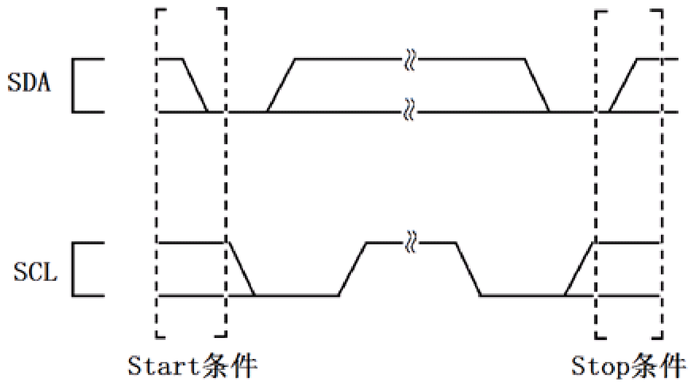

Figure 15-2 SDA and SCL Signals During Start and Stop Conditions

Start a byte transmission or generate Start, Repeated Start or Stop conditions by configuring the START and STOP bits in register *I2Cx_CR*.

Table 15-1 Start and Stop Conditions

Start 位	Stop 位	条件	描述
0	0	无 Start 和 Stop	当有多个数据字节将要被传输的时候，I2C 不会发送 Start 或者 Stop 条件
0	1	Start 或 Repeated Start	I2C 发送一个 Start 条件然后再发送 I2Cx_DBR 内的 8 位数据。Start 发送前，I2Cx_DBR 必须要包含 7 位的从地址和 1 位的 R/nW。 Repeated Start 条件，I2Cx_DBR 包含目标从设备地址和 R/nW 位，这允许主机在不释放总线的情况下进行多次传输。 接口停留在主机发送模式用于写，切换到主机接收模式用于读。
1	x	Stop 条件	在主机发送模式，I2Cx_DBR 内的 8 位数据发送完成之后在总线上发送一个 Stop 条件。 在主机接收模式，I2Cx_CR{ACKNAK}必需置 1 用来发送一个 NAK 脉冲，接收的数据被存入 I2Cx_DBR 寄存器，然后在总线上发送一个 Stop 条件。

1. Start Condition

The Start condition and the data in register *I2Cx_DBR* will be sent after bit TB in register *I2Cx_CR* is set. I2C bus stays in master transmit mode for write requests and stays in master receive mode for read requests.

For a Repeated Start, a change in Read, Write or the target slave address, the register *I2Cx_DBR* contains the updated target slave address and the R/nW bit.
The START condition will not be cleared by the I2C. If I2C loses bus arbitration when it starts to send a Start condition, it will try to resend the Start condition when the bus is freed.

2. No Start or Stop Condition

When I2C is transmitting multiple data bytes, the START and STOP bits in the *I2Cx_CR* register are set to 0, there is no Start or Stop condition. Software writes the data byte, and I2C sets bit ITE in register *I2Cx_SR* and clears bit TB in register *I2Cx_CR*. Then software writes a new byte to register *I2Cx_DBR* and sets bit TB in register *I2Cx_CR*, which starts the new byte transmission. This process continues until the START or STOP bit in register *I2Cx_CR* is set by software. After a Start, Stop or Repeated Start condition is transmitted, the START and STOP bits in register *I2Cx_CR* are not cleared automatically by I2C.

After each byte and ACK/NAK are transferred, I2C holds the SCL line low and waits until the TB bit in register *I2Cx_CR* is set.

3. Stop Condition

A Stop condition terminates a data transfer. In master transmit mode, the STOP and TB bits in register *I2Cx_CR* must be set to start the transmission of the last byte. In master receive mode, the ACKNAK, STOP and TB bits in register *I2Cx_CR* must be set to start the reception of the last byte. After a Stop condition is transmitted, software must clear the STOP bit in register *I2Cx_CR*.

15.3 Data Transfer Sequence

I2C transmits data in 1-byte increments and by following sequence:

1. Start
2. 7-bit slave address
3. R/nW bit
4. Acknowledge pulse
5. 8 data bits
6. Acknowledge pulse
7. Repeat of Step 5 and Step 6
8. Repeated Start (repeat Step 1) or Stop

15.4 Data and Addressing

The I2C Data Buffer Register (*I2Cx_DBR*) and the I2C Slave Address Register (*I2Cx_SAR*) manage data and slave addressing. *I2Cx_DBR* contains 1 byte of data or a 7-bit target slave address and the R/nW bit. *I2Cx_SAR* contains the ASR6601 I2C slave address when the I2C is in slave mode. After I2C receives a full byte of data and an ACK, it stores the data in register *I2Cx_DBR*. To transmit data, the CPU writes to the *I2Cx_DBR* register, and the I2C transmits the data to the serial bus when the TB bit in register *I2Cx_CR* is set.

1. I2C is in Master/Slave Transmit mode:

(1) Software writes data to the *I2Cx_DBR* register, which makes the I2C to start a master transaction or to send the next data byte after the ITE bit in register *I2Cx_SR* is set.

(2) When bit TB in register *I2Cx_CR* is set, the data in register *I2Cx_DBR* is transmitted.

(3) If the ITEIE bit in register *I2Cx_CR* is set, an I2Cx_DBR transmit-empty interrupt is triggered after a byte and an ACK is transferred.

(4) When the I2C is ready to send the next byte before the CPU writes to the *I2Cx_DBR* register and there is no Stop condition, the I2C is in a wait state until the CPU writes to the *I2Cx_DBR* register and sets the TB bit in the *I2Cx_CR* register.

Note: *In FIFO mode, software writes to the TX FIFO instead of the I2Cx_DBR register.*

2. I2C is in Master/Slave Receive mode:

(1) When a full byte of data is received (if the DRFIE bit in register *I2Cx_CR* is set), the I2Cx_DBR receive-full interrupt is generated and the IRF bit in register *I2Cx_SR* is set, the CPU then reads the I2Cx_DBR register to retrieve the data.

(2) After the ACK cycle is completed, I2C transfers data from the shift register to the *I2Cx_DBR* register.

(3) I2C is in wait state until the *I2Cx_DBR* register is read by the CPU.

(4) After the CPU reads the *I2Cx_DBR* register, the I2C updates the ACKNAK and TB bits in register *I2Cx_CR* to allow the transmission of the next byte.

Note: *In FIFO mode, software reads from the RX FIFO instead of the I2Cx_DBR register.*

3. Addressing a Slave Device:

As a master device, the I2C must form and send the first byte of a transaction. This byte consists of a 7-bit slave address and a R/nW bit. After the first byte is transmitted, the I2C must receive an ACK from the slave device. When it is a Write transaction, the I2C remains in master transmit mode, and the slave device remains in slave receive mode. When it is a Read transaction, the I2C switches to master receive mode immediately after receiving an ACK, and the slave device switches to slave transmit mode. When a NAK is returned, the I2C automatically sends a Stop condition and sets the BED bit in register *I2Cx_SR* to abort the current transaction.

PDF-XChange Editor Click to BUY NOW! www.pdf-xchange.com

15.5 Acknowledge

Each byte transmission must be accompanied by an ACK generated by the master or slave receiver. The transmitter must release the SDA line for the receiver to transmit the ACK.

In master transmit mode, if the target slave receiver does not generate an ACK, the SDA line remains high, which indicates a NAK. The lack of an ACK causes I2C to set the BED bit in register *I2Cx_SR* and generate an interrupt. I2C automatically generates a Stop condition and aborts the transmission.

In master receive mode, I2C sends a NAK to notify the slave transmitter to stop sending data. The ACKNAK bit in the *I2Cx_CR* register controls the generation of ACK/NAK on the bus. According to the I2C protocol, the BED bit in the *I2Cx_SR* register is not set for a master receive mode NAK. I2C automatically sends the ACK every time it receives a byte from the bus. Before the master receiver receives the last byte, software must set the ACKNAK bit in the *I2Cx_CR* register to generate a NAK. The NAK is sent after the last byte to indicate that the last byte has been sent.

In slave receive mode, I2C automatically acknowledges its own slave address, irrespective of whether the ACKNAK bit in the *I2Cx_CR* register is set. In slave mode, I2C automatically sends the ACK after receiving a byte, irrespective of whether the ACKNAK bit in the *I2Cx_CR* register is set.

In slave transmit mode, receiving a NAK indicates that the last byte has been transferred. The master then sends a Stop condition or Repeated Start condition. The UB bit in register *I2Cx_SR* remains set until a Stop condition or Repeated Start condition is received.

15.6 Arbitration

I2C bus arbitration is required by a multi-master capability. Bus arbitration is used when two or more masters simultaneously generate a Start condition within the minimum time of a Start condition.

Arbitration can last for a long time. If the slave address and the R/nW bit are the same, the arbitration moves to the data. Due to the Wired-And nature of the I2C bus, no data is lost if two or all masters are outputting the same bus state. If the address, the R/nW bit, or the data are different, the master that transitioned to the high state (the master data is different from the SDA line) loses arbitration and aborts the data transfer. The I2C bus sets bit ALD in register *I2Cx_SR*, then returns to the idle state.

In FIFO mode, software must empty the FIFOs once arbitration is lost. This can be done by clearing the read and write pointer registers for TxFIFO and RxFIFO.

15.7 I2C Master Mode

When software starts a read or write operation on the I2C bus, the I2C switches from the default slave receive mode to master transmit mode. The Start condition is followed by the 7-bit slave address and the R/nW bit.

After receiving an ACK, the I2C enters one of the two master modes:

- Master transmit: I2C writes data
- Master receive: I2C reads data

The CPU writes to register *I2Cx_CR* to start a master transaction.

Table 15-2 Master Transactions

I2C Master Action	Op. mode	Definition
Generate clock output	Master transmit Master receive	● The master drives the SCL line. ● The SCLE and UE bits in the *I2Cx_CR* register must be set.
Write slave address to I2Cx_DBR	Master transmit Master receive	● CPU writes to bits[7:1] in the *I2Cx_DBR* register before a Start condition is enabled. ● The first 7 bits are sent after Start.
Write R/nW bit to I2Cx_DBR	Master transmit Master receive	● CPU writes the R/nW control bit to the least significant bit in register *I2Cx_DBR*. ● If the R/nW bit is low, master remains a master transmitter, if the R/nW bit is high, master switches to a master receiver
Send Start condition	Master transmit Master receive	After the 7-bit target slave address and the R/nW bit are written into the *I2Cx_DBR* register, ● Software sets the START bit in register *I2Cx_CR*. ● Software sets the TB bit in register *I2Cx_CR* to initiate the Start condition.
Initiate first data byte transmission	Master transmit Master receive	● CPU writes one data byte to the *I2Cx_DBR* register. ● Software sets the TB bit in register *I2Cx_CR* and I2C starts the transmission of the Byte. ● The TB bit in register *I2Cx_CR* is cleared and the ITE bit in register *I2Cx_SR* is set when the transfer is complete.
Arbitrate for I2C bus	Master transmit Master receive	If 2 or more masters send a Start condition within the same clock period, then bus arbitration must occur. ● I2C arbitrates as long as there is a need. Bus arbitration takes place during the transmission of target slave address, R/nW bit or data, and it continues until all masters except one master lose the bus. No data is lost. ● If I2C loses arbitration, the ALD bit in register *I2Cx_SR* is set, and I2C switches to slave receive mode. ● If I2C loses arbitration when it starts to send the target slave address, it will try to resend the address when the bus is freed.
Write one data byte to I2Cx_DBR	Master transmit only	● 如果 I2Cx_SR{ITE} 被置 1 并且 I2Cx_CR{TB} 被清 0，当 I2Cx_DBR 空中断被使能，那么中断产生。 ● CPU 写一个字节到 I2Cx_DBR 寄存器，并且根据需要设置合适的 Start/Stop 条件组合，然后把 I2Cx_CR{TB} 置

		1 发送数据。数据的 8 个位被从位移寄存器搬到串行总线上。若发送前 I2Cx_CR{STOP} 置 1，那么在数据的 8 个位传输完成之后会跟随一个 Stop 条件。
Wait for ACK from slave receiver	Master transmit only	作为发送方，主机产生 ACK 的时钟，并且将 SDA 线释放给接收的从机发送 ACK。
Read one byte from I2Cx_DBR	Master receive only	在 I2Cx_CR{ACKNAK} 被读取之后，位移寄存器内的 8 位数据被搬到 I2Cx_DBR 寄存器， ● 当 I2Cx_SR{IRF} 被置 1 且 I2Cx_CR{TB} 被清 0 时，CPU 读取 I2Cx_DBR 寄存器。可以使能 I2Cx_DBR 寄存器接收满中断通知 CPU。 ● 当 I2Cx_DBR 被读取完，如果 I2Cx_SR{ACKNAK} 被清 0（代表 ACK），软件必须把 I2Cx_CR{ACKNAK} 清 0 并且把 I2Cx_CR{TB} 置 1 来开始下一个字节的读取。 ● 如果 I2Cx_SR{ACKNAK} 被置 1（代表 NAK），I2Cx_CR{TB} 被清除，I2Cx_CR{STOP} 被置 1，且 I2Cx_SR{UB} 被置 1，最后一个字节已经被读取到 I2Cx_DBR 寄存器，I2C 正在发送 Stop 条件。 ● 如果 I2Cx_SR{ACKNAK} 被置 1（代表 NAK），并且 I2Cx_CR{TB} 被清 0，但是 I2Cx_CR{STOP} 被清 0，软件有两个选择： 1. 把 I2Cx_CR{START} 置 1，将新的目标从机地址写入 I2Cx_DBR，把 I2Cx_CR{TB} 置 1，发送一个 Repeated Start 条件。 2. 把 I2Cx_CR{MA} 置 1，并且保持 I2Cx_CR{TB}为 0，仅发送一个 Stop 条件。
Transmit ACK to slave transmitter	Master receive only	● 作为接收主机一方，在 ACK 期间，产生 ACK 时钟，并且驱动 SDA 线 ● 如果下一个字节为最后一个事务，软件需要把 I2Cx_CR{ACKNAK} 置 1 来产生 NAK。
Generate a Repeated Start	Master transmit Master receive	使用 Repeated Start 代替 Stop 条件可以在不释放总线的情况下继续新的传输 ● Repeated Start 条件在最后一个字节数据被传输后产生 ● 软件必须把 7 位的目标从机地址和 1 位的 R/nW 位写入 I2Cx_DBR 寄存器，然后把 I2Cx_CR{START} 置 1，再把 I2Cx_CR{TB} 置 1
Generate Stop	Master transmit Master receive	● Stop 条件在最后一个字节数据被传输后产生 ● I2Cx_CR{STOP} 要在最后一个字节被传输前置 1

15.8 FIFO Mode

The FIFO mode can only be used when the I2C is in *Master Mode*.

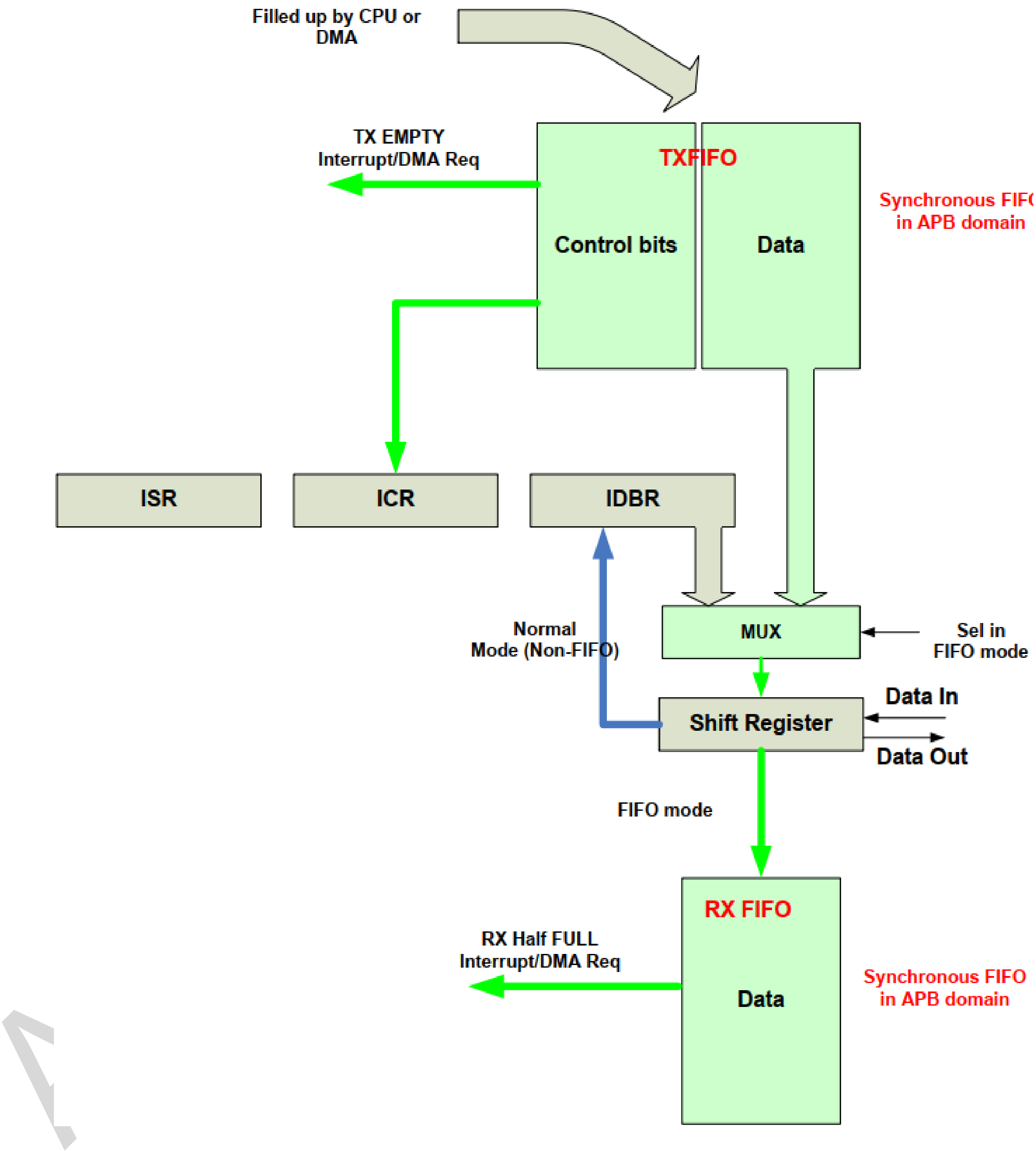

Figure 15-3 FIFO Mode Block Diagram

FIFOs can be used for both transmission and reception to help reducing the empty and full interrupts of register I2Cx_DBR. The FIFOs allow reading and writing multiple bytes without interrupting the CPU after each byte.

DMA is used to improve I2C transactions (typically more than 8 Bytes). The entire transaction can be completed by DMA without generating multiple FIFO interrupts.

The FIFO mode is backward compatible, and it is disabled by clearing the FIFO_EN bit in the *I2Cx_CR* register.

Transmit FIFO has a width of 12 (4 control bits and 8 data bits) and a depth of 8. The 4 control bits are bits[3:0] in register *I2Cx_CR*, which are required for each data byte transmission. After a byte is transmitted, the next byte is copied from the TX FIFO into the shift register, and the control bits are copied into the *I2Cx_CR* register. This byte is now transferred, and it continues like that until the Stop bit is set.

Receive FIFO has a width of 8 (8 data bits) and a depth of 16, which is used to store the received data. The control bits for each byte and dummy data are put in the corresponding position in the TX FIFO. When the RX FIFO is half full, an interrupt or DMA request is sent out for the data in the RX FIFO to be read out.

In order to support the FIFO mode and fully utilize its capabilities, the following status and control bits need to be configured:

(1) Set the FIFO_EN bit in register *I2Cx_CR* to enable the FIFO mode.

(2) Set the TXBEGIN bit in register *I2Cx_CR* to start a transaction.

(3) Bits[31:28] in register *I2Cx_CR* enables or disables all the FIFO related interrupts, and bits [31:28] in register *I2Cx_SR* is used to inquire the interrupt status.

(4) TXDONE interrupt is generated when each transaction is completed (that is, a Stop condition is sent).

(5) The DMA_EN bit in register *I2Cx_CR* is used to enable/disable DMA mode.

In DMA mode, all the FIFO related interrupts must be disabled in register *I2Cx_CR* (bits[31:28]), and the DMA_EN bit in this register must be set. In this way, all DMA requests are sent to the DMA without interrupting the CPU. The TXDONE_IE bit in the *I2Cx_CR* register needs to be set in both FIFO and DMA modes to generate an interrupt to the CPU after each transaction is completed.

15.9 I2C Slave Mode

Table 15-3 Slave Transactions

Operation	Op.mode	Description
Slave receive mode (default)	Slave receive only	● I2C 监视所有从机地址事务 ● I2Cx_CR{UE} 必须置 1 ● I2C 监视总线上的 Start 条件。若检测到 Start 条件，接口读取前 8 位数据，并把前 7 位与自身从机地址做比较，若匹配则响应 ACK ● 若首字节的第 8 位（R/nW）为低，那么 I2C 保持在从机接收模式，并把 I2Cx_SR{SAD} 清 0。若 R/nW 为高，I2C 切换到从机发送模式，并把 I2Cx_SR{SAD} 置 1
Set the slave address detection bit	Slave receive Slave transmit	● 用来指示接口检测到匹配的 I2C 寻址 ● 若使能 I2Cx_CR{SADIE}，在匹配的从机地址被接收和 ACK 响应之后，中断产生，I2Cx_SR{SAD} 置 1
Read one byte from *I2Cx_DBR*	Slave receive only	● 8 位数据从总线上读取到位移寄存器，在整个字节被接收完成和 ACK/NAK 完成之后，位移寄存器内的数据被搬到 I2Cx_DBR 寄存器 ● 当 I2Cx_SR{IRF} 置 1，且 I2Cx_CR{TB} 清 0，若使能 I2Cx_CR{DRFIE}，I2Cx_DBR 接收满中断产生 ● 软件从 I2Cx_DBR 读取数据，并根据需要配置 I2Cx_CR{ACKNAK}，把 I2Cx_CR{TB} 置 1，这个操作使从机退出等待模式，接续接收主机的数据
Transmit ACK to master transmitter	Slave receive only	● 作为接收从机，I2C 在 SCL 为高的时候将 SDA 线拉低产生 ACK ● ACK/NAK 由 I2Cx_CR{ACKNAK} 控制
Write one byte to *I2Cx_DBR*	Slave transmit only	● I2Cx_SR{ITE} 置 1，I2Cx_CR{TB} 清 0，若使能 I2Cx_CR{ITEIE} 中断，I2Cx_DBR 发送空中断产生 ● 软件把数据写入 I2Cx_DBR 寄存器，然后把 I2Cx_CR{TB} 置 1 开始数据的发送
等待接收主机的 ACK	仅从机发送	作为发送从机，I2C 释放 SDA 线等待接收主机拉低响应 ACK

15.10 Clock and Reset

Each I2C interface has independent APB bus clock and independent APB bus reset. Software must ensure that the I2C unit is disabled (*I2Cx_CR*[UE]=0) before reset, and ensure that the I2C bus is idle (*I2Cx_SR*[IBB]=0) when the unit is enabled after reset. When reset, all registers except the *I2Cx_SAR*, return to the default reset condition. *I2Cx_SAR* is not affected by a reset.

Steps for I2C clock reset:

1. Set the UR bit in the *I2Cx_CR* register, and clear the remaining bits of this register.
2. Clear the *I2Cx_SR* register.
3. Clear the UR bit in the *I2Cx_CR* register.

15.11 Interrupts

I2C interrupts are configured by register *I2Cx_CR*, and the interrupt status can be obtained by querying the corresponding bit in register *I2Cx_SR*.

15.12 DMA

DMA (Direct Memory Access) is enabled by setting the DMA_EN bit in register *I2Cx_CR* to support transmission and reception.

15.13 I2C Registers

I2C0 Base Address: 0x40007000

I2C1 Base Address: 0x40014000

I2C2 Base Address: 0x40015000

Table 15-4 I2C Registers Summary

Register	Offset	Description
I2Cx_CR	0x00	控制寄存器
I2Cx_SR	0x04	状态寄存器
I2Cx_SAR	0x08	从地址寄存器
I2Cx_DBR	0x0C	数据 Buffer 寄存器
I2Cx_LCR	0x10	加载计数寄存器
I2Cx_WCR	0x14	等待计数寄存器
I2Cx_RST_CYCL	0x18	复位周期寄存器
I2Cx_BMR	0x1C	总线监视寄存器
I2Cx_WFIF0	0x20	发送 FIFO 寄存器
I2Cx_WFIFO_WPTR	0x24	发送 FIFO 写指针寄存器
I2Cx_WFIFO_RPTR	0x28	发送 FIFO 读指针寄存器
I2Cx_RFIFO	0x2C	接收 FIFO 寄存器
I2Cx_RFIFO_WPTR	0x30	接收 FIFO 写指针寄存器
I2Cx_RFIFO_RPTR	0x34	接收 FIFO 读指针寄存器
I2Cx_RESV[2]	0x38	4 x 2 字节保留
I2Cx_WFIFO_STATUS	0x40	写 FIFO 状态寄存器
I2Cx_RFIFO_STATUS	0x44	读 FIFO 状态寄存器

15.13.1 I2Cx_CR (x=0, 1, 2)

Offset: 0x000

Reset Value: 0x00000200

31	30	29	28	27	26	25	24
RXOV_IE	RXF_IE	RXHF_IE	TXE_IE	TXDONE_IE	MSDE	MSDIE	SSDIE
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h
23	**22**	**21**	**20**	**19**	**18**	**17-16**	
SADIE	BEIE	RESERVED	DRFIE	ITEIE	ALDIE	RESERVED	
rw-0h	rw-0h	r-0h	rw-0h	rw-0h	rw-0h	r-0h	
15	**14**	**13**	**12**	**11**	**10**	**9-8**	
RESERVED	UE	SCLE	MA	IBRR	UR	MODE	
r-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-2h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMA_EN	RESERVED	FIFOEN	TXBEGIN	TB	ACKNAK	STOP	START
rw-0h	r-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

Bit 31 RXOV_IE: Receive FIFO overrun interrupt enable.

- 0: receive FIFO overrun interrupt disabled
- 1: receive FIFO overrun interrupt enabled

Bit 30 RXF_IE: Receive FIFO full interrupt enable.

- 0: receive FIFO full interrupt disabled
- 1: receive FIFO full interrupt enabled

Bit 29 RXHF_IE: Receive FIFO half full interrupt enable.

- 0: Receive FIFO half full interrupt disabled
- 1: Receive FIFO half full interrupt enabled

Bit 28 TXE_IE: Transmit FIFO empty interrupt enable.

- 0: Transmit FIFO empty interrupt disabled
- 1: Transmit FIFO empty interrupt enabled

Bit 27 TXDONE_IE: Transaction done interrupt enable.

- 0: transaction done interrupt disabled
- 1: transaction done interrupt enabled

Bit 26 MSDE: Master Stop detection enable.

- 0: Master Stop detection disabled
- 1: Master Stop detection enabled

Bit 25 MSDIE: Master Stop detection interrupt enable.

- 0: Master Stop detection interrupt disabled
- 1: Master Stop detection interrupt enabled

Bit 24 SSDIE: Slave Stop detection interrupt enable.

- 0: Slave Stop detection interrupt disabled
- 1: Slave Stop detection interrupt enabled

Bit 23 SADIE: Slave address detection interrupt enable.

- 0: Slave address detection interrupt disabled
- 1: Slave address detection interrupt enabled

Bit 22 BEIE: Bus error interrupt enable.

- 0: Bus error interrupt disabled
- 1: Bus error interrupt enabled

Bit 21 RESERVED: Must be kept, and cannot be modified.

Bit 20 DRFIE: I2Cx_DBR receive-full interrupt enable.

- 0: I2Cx_DBR receive-full interrupt disabled
- 1: I2Cx_DBR receive-full interrupt enabled

Bit 19 ITEIE: I2Cx_DBR transmit-empty interrupt enable.

- 0: I2Cx_DBR transmit-empty interrupt disabled
- 1: I2Cx_DBR transmit-empty interrupt enabled

Bit 18 ALDIE: Arbitration loss detection interrupt enable.

- 0: Arbitration loss detection interrupt disabled
- 1: Arbitration loss detection interrupt enabled

Bits 17-15 RESERVED: Must be kept, and cannot be modified.

Bit 14 UE: I2C unit enable.

- 0: I2C unit disabled
- 1: I2C unit enabled (the default is slave receive mode)

Software must ensure that the I2C bus is idle before enabling the I2C unit, and ensure that the internal I2C clock is enabled before setting or clearing this bit.

Bit 13 SCLE: SCL enable.

- 0: disable the I2C from driving the SCL line
- 1: enable the I2C clock output for master-mode operation

Bit 12 MA: Master abort.

This bit is used for the I2C to generate a Stop condition in master mode.

- 0: Stop condition is generated when the STOP bit in this register is set
- 1: Stop condition is generated without data transmission

In master transmit mode, after a data byte is transmitted, the TB bit in this register is cleared and the ITE bit in register *I2Cx_SR* is set. When no more data bytes need to be sent, setting the MA bit to generate a Stop condition to free the bus. In master receive mode, when a NAK is sent with the STOP bit=0 and without a Repeated Start condition followed, setting the MA bit to generate a Stop condition to free the bus. The TB bit in this register must remain clear.

Bit 11 IBRR: I2C bus reset request.

- 0: no action
- 1: I2C bus reset, and this bit is cleared automatically

Bit 10 UR: Unit reset.

- 0: no action
- 1: reset the I2C unit

Bits 9-8 MODE: Bus clock mode for the master.

- 00: standard mode – 100 Kbps
- 01: fast mode – 400 Kbps

Bit 7 DMA_EN: DMA enable.

- 0: DMA requests disabled
- 1: DMA requests enabled

Bit 6 RESERVED: Must be kept, and cannot be modified.

Bit 5 FIFOEN: FIFO mode enable.

- 0: FIFO mode disabled
- 1: FIFO mode enabled

Bit 4 TXBEGIN: Transaction begin.

- 0: no action
- 1: new transaction begins

This bit is cleared by hardware after a Stop condition is generated, and the software needs to set it again to start a new transaction.

Bit 3 TB: Transfer byte, used to send or receive a byte on the I2C bus.

- 0: cleared by I2C when one byte is sent or received
- 1: send or receive a byte

The I2C unit monitors this bit to determine whether the byte transfer has completed. In master or slave mode, after each byte including the ACK is transferred, I2C holds the SCL line low until the TB bit is set.

Bit 2 ACKNAK: The positive/negative acknowledge (ACK/NAK) control bit in master receive mode.

- 0: send ACK after receiving a data byte
- 1: send NAK after receiving a data byte

In slave mode, when the I2C responds to its slave address or the reception is complete, it automatically sends an ACK, regardless of whether the ACKNAK bit is set.

Bit 1 STOP: Generate a Stop condition.

- 0: no action
- 1: generate Stop condition

This bit is used to generate a Stop condition on the I2C bus after the transmission of the next data byte in master mode. In master receive mode, the ACKNAK bit and the STOP bit must be set to 1 at the same time.

Bit 0 START: Generate a Start condition.

- 0: no action
- 1: generate a Start condition

This bit is used to generate a Start condition on the I2C bus in master mode.

PDF-XChange Editor Click to BUY NOW! www.pdf-xchange.com

15.13.2 I2Cx_SR (x=0, 1, 2)

Offset: 0x004

Reset Value: 0x00000000

31	**30**	**29**	**28**	**27**	**26**	**25**
RXOV	RXF	RXHF	TXE	TXDONE	MSD	RESERVED
rw1c-0h	rw1c-0h	rw1c-0h	rw1c-0h	rw1c-h	r1ch	r-0h
24	**23**	**22**	**21**	**20**	**19**	**18**
SSD	SAD	BED	RESERVED	IRF	ITE	ALD
rw1c-0h	rw1c-0h	rw1c-0h	r-0h	rw1c-0h	rw1c-0h	rw1c-0h
17	**16**	**15**	**14**	**13-8**	**7-0**	
RESERVED	IBB	UB	ACKNAK	RESERVED	RESERVED	
r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	

Bit 31 RXOV: Receive FIFO overrun flag.

- 0: no receive FIFO overrun occurred
- 1: receive FIFO overrun occurred, and it is cleared by software writing 1 to it

Bit 30 RXF: Receive FIFO full flag.

- 0: receive FIFO is not full
- 1: receive FIFO is full, and it is cleared by software writing 1 to it

Bit 29 RXHF: Receive FIFO half-full flag.

- 0: receive FIFO is not half full
- 1: receive FIFO is half full, and it is cleared by software writing 1 to it

Bit 28 TXE: Transmit FIFO empty flag.

- 0: transmit FIFO is not empty
- 1: transmit FIFO is empty, and it is cleared by software writing 1 to it

Bit 27 TXDONE: Transaction done flag (used in FIFO mode).

- 0: transaction in progress
- 1: transaction is done, and it is cleared by software writing 1 to it

Bit 26 MSD: Master Stop detection flag (only valid in master mode).

- 0: no master Stop detected
- 1: master Stop detected, and it is cleared by software writing 1 to it

Bit 25 RESERVED: Must be kept, and cannot be modified.

Bit 24 SSDIE: Slave Stop detection flag.

- 0: no slave Stop detected
- 1: slave Stop was detected, and it is cleared by software writing 1 to it

Bit 23 SAD: Slave address detection flag.

- 0: no matching slave address detected
- 1: matching slave address detected, and it is cleared by software writing 1 to it

Bit 22 BED: Bus error detection flag.

- 0: no bus error detected
- 1: bus error detected, and it is cleared by software writing 1 to it
  This bit is set in two cases:
  - As a master transmitter, the I2C did not receive an ACK after sending a byte.
  - As a slave receiver, the I2C generates a NAK.

Bit 21 RESERVED: Must be kept, and cannot be modified.

Bit 20 IRF: I2Cx_DBR receive full flag.

- 0: the I2Cx_DBR register has not received a new data byte or the I2C bus is idle
- 1: the I2Cx_DBR register received a new data byte, and it is cleared by software writing 1 to it

Bit 19 ITE: I2Cx_DBR transmit empty flag.

- 0: the data byte transmit in progress
- the I2C has finished transmitting a byte on the I2C bus, and it is cleared by software writing 1 to it

Bit 18 ALD: Arbitration loss detection flag, used in multi-master scenarios.

- 0: I2C wins the arbitration or no arbitration took place
- 1: I2C loses the arbitration, and it is cleared by software writing 1 to it

Bit 17 RESERVED: Must be kept, and cannot be modified.

Bit 16 IBB: I2C bus busy flag.

- 0: Bus is idle or the bus is being used by the I2C interface
- 1: Bus is busy but not used by the I2C interface

Bit 15 UB: I2C unit busy flag.

- 0: I2C unit is idle
- 1: I2C unit is busy

Bit 14 ACKNAK: ACK/NAK status flag.

- 0: I2C received or sent an ACK
- 1: I2C received or sent a NAK

In slave transmit mode, this bit is used to determine whether the byte transmitted is the last one. This bit is updated after each byte and ACK/NAK information is received.

Bits 13-0 RESERVED: Must be kept, and cannot be modified.

15.13.3 I2Cx_SAR (x=0, 1, 2)

Offset: 0x008

Reset Value: 0x00000000

31-7	6-0
RESERVED	SLAVE_ADDRESS
r-0h	rw-0h

Bits 31-7 RESERVED: Must be kept, and cannot be modified.

Bits 6-0 SLAVE_ADDRESS: The ASR6601 I2C slave address used in slave mode.

15.13.4 I2Cx_DBR (x=0, 1, 2)

Offset: 0x00C

Reset Value: 0x00000000

31-8	7-0
RESERVED	DATA_BUFFER
r-0h	rw-0h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bits 7-0 DATA_BUFFER: Buffer for I2C bus transmit/receive data.

15.13.5 I2Cx_LCR (x=0, 1, 2)

Offset: 0x010

Reset Value: 0x18183a7e

31-18	17-9	8-0
RESERVED	FLV	SLV
r-1818h	rw-1dh	rw-7eh

Bits 31-18 RESERVED: Must be kept, and cannot be modified.

Bits 17-9 FLV: Phase decrementer load value for fast mode SCL in master mode.

Bits 8-0 SLV: Phase decrementer load value for standard mode SCL in master mode.

15.13.6 I2Cx_WCR (x=0, 1, 2)

Offset: 0x14

Reset Value: 0x0000143a

31-5	4-0
RESERVED	COUNT
r-a1h	rw-1ah

Bits 31-5 RESERVED: Must be kept, and cannot be modified.

Bits 4-0 COUNT: Counter values for defining the setup and hold times in standard and fast modes.

15.13.7 I2Cx_RST_CYCL (x=0, 1, 2)

Offset: 0x018

Reset Value: 0x00000000

31-4	3-0
RESERVED	RST_CYC
r-0h	rw-0h

Bits 31-4 RESERVED: Must be kept, and cannot be modified.

Bits 3-0 RST_CYC: Serial bus reset SCL cycle count.

15.13.8 I2Cx_BMR (x=0, 1, 2)

Offset: 0x01C

Reset Value: 0x00000003

31-2	1	0
RESERVED	SCL	SDA
r-0h	r-1h	r-1h

Bits 31-2 RESERVED: Must be kept, and cannot be modified.

Bit 1 SCL: SCL pin state.

Bit 0 SDA: SDA pin state.

15.13.9 I2Cx_WFIF0 (x=0, 1, 2)

Offset: 0x020

Reset Value: 0x00000000

31-12	11-8	7-0
RESERVED	CONTROL	DATA
r-0h	w-0h	w-0h

Bits 31-12 RESERVED: Must be kept, and cannot be modified.

Bits 11-8 CONTROL: I2C bus transmit/receive data control bits.

Bits 7-0 DATA: I2C bus send data for write transactions and dummy data for read transactions.

15.13.10 I2Cx_WFIFO_WPTR (x=0, 1, 2)

Offset: 0x024

Reset Value: 0x00000000

31-4	3-0
RESERVED	DATA
r-0h	rw-0h

Bits 31-4 RESERVED: Must be kept, and cannot be modified.

Bits 3-0 DATA: The position in the Transmit FIFO where the software will write the next entry.

15.13.11 I2Cx_WFIFO_RPTR (x=0, 1, 2)

Offset: 0x028

Reset Value: 0x00000000

31-4	3-0
RESERVED	DATA
r-0h	rw-0h

Bits 31-4 RESERVED: Must be kept, and cannot be modified.

Bits 3-0 DATA: The position in the Transmit FIFO where the hardware will read the next entry.

15.13.12 I2Cx_RFIFO (x=0, 1, 2)

Offset: 0x02C

Reset Value: 0x00000000

31-8	7-0
RESERVED	DATA
r-0h	r-0h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bits 7-0 DATA: I2C bus receive data for read transactions.

15.13.13 I2Cx_RFIFO_WPTR (x=0, 1, 2)

Offset: 0x030

Reset Value: 0x00000000

31-4	3-0
RESERVED	DATA
r-0h	r-0h

Bits 31-4 RESERVED: Must be kept, and cannot be modified.

Bits 3-0 DATA: The position in the Receive FIFO where the hardware will write the next entry.

15.13.14 I2Cx_RFIFO_RPTR (x=0, 1, 2)

Offset: 0x034

Reset Value: 0x00000000

31-4	3-0
RESERVED	DATA
r-0h	r-0h

Bits 31-4 RESERVED: Must be kept, and cannot be modified.

Bits 3-0 DATA: The position in the Receive FIFO where the software will read the next entry.

15.13.15 I2Cx_WFIFO_STATUS (x=0, 1, 2)

Offset: 0x040

Reset Value: 0x00000000

31-16	15-9	8-1	0
RESERVED	WFIFO_SIZE	WFIFO_EMPTY	WFIFO_FULL
r-0h	r-0h	r-0h	r-0h

Bits 31-6 RESERVED: Must be kept, and cannot be modified.

Bits 5-2 WFIFO_SIZE: The Transmit FIFO size.

Bit 1 WFIFO_EMPTY: Transmit FIFO empty.

Bit 0 WFIFO_FULL: Transmit FIFO full.

15.13.16 I2Cx_RFIFO_STATUS (x=0, 1, 2)

Offset: 0x044

Reset Value: 0x00000000

31-24	23-16	15-8	7-4
RESERVED	RESERVED	RESERVED	RFIFO_SIZE
r-0h	r-0h	r-0h	r-0h
3	**2**	**1**	**0**
RFIFO_EMPTY	RFIFO_FULL	RFIFO_HALFFULL	RFIFO_OVERRUN
r-0h	r-0h	r-0h	r-0h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bits 7-4 RFIFO_SIZE: The Receive FIFO size.

Bit 3 RFIFO_EMPTY: Receive FIFO empty.

Bit 2 RFIFO_FULL: Receive FIFO full.

Bit 1 RFIFO_HALFFULL: Receive FIFO half full.

Bit 0 RFIFO_OVERRUN: Receive FIFO overrun.

16. ADC

16.1 Introduction

The 12-bit ADC (Analog to Digital Converter) has 8 external channels and 7 internal channels for measuring signals with up to 1M sampling rate. The internal VBAT/3 channel allows the ADC to measure the VBAT/3 signal. ADC analog input channels can be configured in single ended (range: 0.1V~1.1V) or differential mode (range: -1.0~1.0V). The ADC conversion supports a programmable channel sequence with a length between 1 to 16 in continuous, single, or discontinuous sampling modes. The conversion can be initiated by software or hardware configurable trigger sources. In addition, the ADC supports DMA request and interrupt generation.

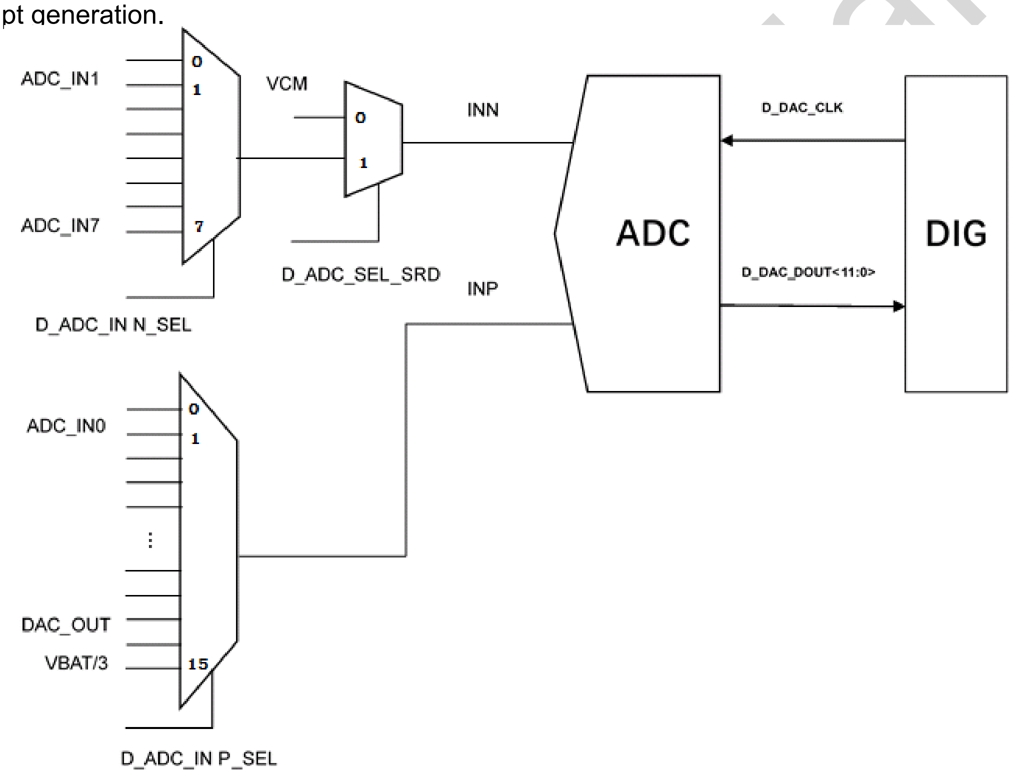

Figure 16-1 ADC Diagram

16.2 ADC Input Mode

Channels can be configured to be either single-ended input or differential input through the differential mode selection register (*ADC_DIFFSEL*). External channels support both single ended and differential modes, and internal channels only support single-ended mode. A fixed combination is required in differential mode, channel 0 and 1 is a differential group, channel 2 and 3 is a differential group, channel 4 and 5 is a differential group, and channel 6 and 7 is a differential group. The width of the last data result of a conversion stored in the data buffer is 12-bit, where the MSB is a sign bit and the other 11 bits are data bits in differential input mode, but no sign bit is presented in single-ended mode, all 12 bits are data bits.

16.3 Sampling Channels

- **8 External Channels:** In single-ended mode, each channel is independent. In differential mode, every two channels form a group and cannot be split.

- **7 Internal Channels:** include DAC output, internal VRef, VBAT/3 (battery voltage), Vts internal temperature sensor) and a channel dedicated for internal tests. The internal channels do not support differential mode.

Table 16-1 ADC Sampling Channels

Channel No.	Signal	Source
1	ADC_PAD_IN<0>	gpio11
2	ADC_PAD_IN<1>	gpio08
3	ADC_PAD_IN<2>	gpio05
4	ADC_PAD_IN<3>	gpio04
5	ADC_PAD_IN<4>	gpio50
6	ADC_PAD_IN<5>	gpio49
7	ADC_PAD_IN<6>	gpio48
8	ADC_PAD_IN<7>	gpio47
9	OPA0_ADC_OUT	
10	OPA1_ADC_OUT	
11	OPA2_ADC_OUT	
12	DCTEST_OUT	
13	TD_OUT_TEST	
14	DAC_CORE_AOUT	
15	VBAT31	

To generate VBAT31 signal, it is necessary to enable VBAT/3 voltage division circuit by setting the D_VBAT_DIV3_EN bit in the *RESV1* register of the analog part. This channel is nominally 1/3 of VBAT and it is 1/3.06 of VBAT actually.

16.4 Trigger Source

- **Software trigger:** The conversion starts immediately when a rising edge on the START bit of ADC_CR is detected.
- **Hardware trigger:** The conversion is triggered by Timer or IO, containing 10 selectable trigger sources with a configurable level.

16.5 Low-power Operation

A new trigger request can only be received after the *ADC_DR* register has been read or the EOC flag is cleared, which can prevent overrun but might bypass some trigger requests.

16.6 Overrun

Configure the ADC_DR register to hold old data or update with new data when an overrun occurs.

16.7 Conversion Modes

The sampling mode is configured by the CONV_MODE bit in register *ADC_CFGR*:
The ADC conversion supports a programmable channel sequence with a length between 1 to 16, and the channels can be configured in single-ended or differential mode. In differential mode, only the P input of the channels in the sequence need to be configured. A channel can be selected more than once in the sequence, and thus the conversion of the same channel will be performed multiple times in each sequence. The conversion sequence is configured through the *ADC_SEQR0* and *ADC_SEQR1* registers, and every 4 bits configures one channel number. The two 32-bit registers have 64 bits in total, and thus up to 16 channels can be configured to be converted.

- **Continuous Mode:** When a software or hardware trigger event occurs, the ADC performs a sequence of conversions. After the conversions are completed, the ADC automatically restarts and continuously performs the same sequence of conversions until a STOP command is issued by software.
- **Single Mode:** When a software or hardware trigger event occurs, the ADC performs a single sequence of conversions and then stops automatically after the conversions are completed.
- **Discontinuous Mode:** Each conversion defined in the sequence requires a hardware or software trigger event. When a sequence of conversions is completed, a new trigger event restarts the conversion of the first channel defined in the sequence. While in continuous and single modes, the complete sequence is converted upon a single trigger event.

16.8 Voltage Reference

The reference voltage is configured through the D_ADC_SEL_VREF bit in the RST register of the analog part. The external or internal reference voltage is configured by clearing or setting this bit, and the default value is 1.

- **Internal Voltage Reference:** VRef, 1.2V.
- **External Voltage Reference:** VREFP/3, VREFP≤3.6V. VREFP is connected to VDDA in the ASR6601CB (48-pin) chip.

16.9 Data Buffer

For the 12-bit data buffer, the most significant bit is the sign bit in differential input mode.

ADC Value	Definition (differential)	Definition (single-ended)
1111_1111_1111	+Vref(1)	+Vref(1)
…	…	…
…	…	…
…	…	…
…	…	…
…	…	…
1000_0000_0001	+Vref/2048(1)	+Vref/2+Vref/4096(1)
1000_0000_0000	0	+Vref/2(1)
0111_1111_1111	-Vref/2048(1)	+Vref/2-Vref/4096(1) …
…	…	
…	…	…
…	…	…
…	…	…
…	…	…
0000_0000_0000	-Vref(1)	0

(1) *This value should be calibrated by software to correct error on the ADC hardware.*

The measure range in differential mode is -1.0~1.0V, and the measure range in single-ended mode is 0.1~1.1V. In order to correct the error on the ADC hardware, ASR6601 is calibrated before leaving the factory. The calibration data (Offset and Gain) are stored in Flash. The user needs to convert the data read from register *ADC_DR* to get the final AD value. The formula is as follows:

$$V = (Vout – Offset) / Gain$$

where **Vout** is the value readed from the data buffer.

16.10 DMA

When the 12-bit data buffer is full, the DMA request is genetated if the DMA_EN bit in register ADC_CFGR is set. DMA request is disabled by writing ‘0’ to the DMA_EN bit.

PDF-XChange Editor Click to BUY NOW! www.pdf-xchange.com

16.11 Interrupts

The interrupt sources include the end of conversion (EOC), end of a sequence of conversions (EOS), and a data overrun (OVERRUN). The interrupts are enabled through register *ADC_IER*, and the interrupt status is inquired through the *ADC_ISR* register.

16.12 Wakeup

The MCU wakes up from the Sleep mode if an interrupt or event is generated.

16.13 Clock and Reset

The ADC bus reset and the ADC clock reset are independent. The ADC module supports the APB bus clock. The ADC interface clock source can be one of the following sources (divided or not): sys_clk, apb_x_pclk, pll_clk or rco48m_clk.

16.14 ADC Registers

Base Address: 0x40017000

Table 16-2 ADC Registers Summary

Register	Offset	Description
ADC_CR	0x00	控制寄存器
ADC_CFGR	0x04	配置寄存器
ADC_SEQR0	0x08	通道采样序列控制寄存器 0
ADC_SEQR1	0x0C	通道采样序列控制寄存器 1
ADC_DIFFSEL	0x10	采样通道差分/单端选择寄存器
ADC_ISR	0x14	中断和状态寄存器
ADC_IER	0x18	中断使能寄存器
ADC_DR	0x1C	数据寄存器

16.14.1 ADC_CR

Offset: 0x00

Reset Value: 0x00000000

31-4	3	2	1	0
RESERVED	STOP	START	DIS	EN
r-0h	rw-0h	rw-0h	w-0h	rw-0h

Bits 31-4 RESERVED: Must be kept, and cannot be modified.

Bit 3 STOP: ADC stop conversion command.

- 0: no action
- 1: Stop the ADC. Read 1 means that the STOP command is in progress

Notes:

1. *Software writes 1 to this bit to stop and discard an ongoing conversion, thus the conversion sequence is reset; this bit is cleared automatically by hardware.*
2. *After the START bit is cleared, software must wait for 3 ADCCLK ticks before reconfigure the START bit; or wait for 1 CLK_DIV (set in register ADC_CFGR) tick to set the DIS bit to disable the ADC.*
3. *Software is allowed to set this bit only when START=1 and STOP=0.*
4. *Before setting the STOP bit, it is recommended to disable the trigger source first, or keep the trigger level in an invalid state.*

Bit 2 START: ADC start conversion command.

- 0: no action
- 1: start the ADC. Read 1 means that the ADC conversion is being performed

This bit is set by software to start the ADC conversion. Software is allowed to set this bit only when EN=1 and DIS=0. Whether an ADC conversion starts immediately (software trigger mode) or won't start until a hardware trigger event occurs depends on the TRIG_SEL[18:17] configuration bits in register *ADC_CFGR*.

This bit is automatically cleared by hardware in the following the following circumstances:

1. In single conversion mode, if software trigger is selected (TRIG_SEL=00 in register *ADC_CFGR*), the START bit is cleared when the EOS flag in register *ADC_ISR* is set.
2. In discontinuous conversion mode, if software trigger is selected, the START bit is cleared when the EOC flag in register *ADC_ISR* is set.
3. In any case, after the execution of the STOP command, the START and STOP bits are cleared by hardware at the same time.

Bit 1 DIS: ADC disable.

- 0: no action
- 1: disable the ADC

Software is allowed to set this bit only when EN=1 and START=0 (no conversion in progress).

Bit 0 EN: ADC enable.

- 0: no action
- 1: enable the ADC. Read 1 means that the ADC is enabled

This bit is set by software to enable the ADC module. The software is allowed to set this bit only when all bits of register *ADC_CR* equal 0. Reading this bit reflects whether the ADC is enabled or not. The software must wait at least 100us for the ADC analog circuit to stabilize after initialization before it enables the ADC conversion.

16.14.2 ADC_CFGR

Offset: 0x004

Reset Value: 0x00000002

31-24	23	22	21-20	19	18-17
RESERVED	RESERVED	WAIT_MODE	CONV_MODE	OVERRUN_MODE	TRIG_SEL
r-0h	r-0h	r-0h	r-0h	r-0h	rw-0h

16	15-13	12	11-8	7-0
EXT_TRIG_SEL[3]	EXT_TRIG_SEL[2:0]	DMA_EN	CLK_DIV[11:8]	CLK_DIV[7:0]
rw-0h	rw-0h	rw-0h	rw-0h	rw-2h

Bits 31-23 RESERVED: Must be kept, and cannot be modified.

Bit 22 WAIT_MODE: Wait conversion mode.

- 0: wait conversion mode disabled
- 1: wait conversion mode enabled

Wait for the conversion mode, that is, a new trigger request can only be received after register ADC_DR has been read or the EOC flag (in ADC_ISR) is cleared, which can prevent overrun but may bypass some trigger requests.
Software is allowed to write this bit only when START=0 (in register *ADC_CR*).

Bits 21-20 CONV_MODE: ADC conversion mode selection.

- 00: single conversion mode
- 01: continuous conversion mode
- 1x: discontinuous conversion mode

Software is allowed to write this bit only when START=0 (in register *ADC_CR*).

Notes:

1. *In single conversion mode, when a software or hardware trigger event occurs, the ADC performs a single sequence of conversions (set by ADC_SEQR0/1). After the conversions are completed, the ADC stops until a new trigger occurs.*
2. *In continuous conversion mode, when a software or hardware trigger event occurs, the ADC performs a sequence of conversions (set by ADC_SEQR0/1). After the conversions are completed, the ADC automatically re-starts and continuously performs the same sequence of conversions until a STOP command is issued by software.*
3. *In discontinuous conversion mode, each conversion defined in the sequence (set by ADC_SEQR0/1) requires a hardware or software trigger event. When a sequence of conversions is completed, a new trigger event restarts the conversion of the first channel defined in the sequence.*

Bit 19 OVERRUN_MODE: Overrun management mode.

- 0: the old data in the ADC_DR register is hold when an overrun is occurred
- 1: the ADC_DR register is overwritten with the newly converted data when an overrun is occurred

Software is allowed to write this bit only when START=0 (in register *ADC_CR*).

Bits 18-17 TRIG_SEL: Trigger mode and polarity selection.

- 00: software trigger. The conversion starts immediately when a rising edge on the START bit of ADC_CR is detected
- 01: hardware trigger detection on the rising edge
- 10：硬件触发，下降沿触发
- 10: hardware trigger detection on the falling edge

Software is allowed to write this bit only when START=0 (in register *ADC_CR*).

When a hardware trigger is selected, after the START bit is configured, software must wait for 3 ADCCLK ticks before receiving the trigger signal.

Bits 16-13 EXT_TRIG_SEL: External trigger selection for the start of ADC conversion.

- 0000~0100: reserved
- 0101: GPIO47
- 0110: GPIO31
- 0111: GPIO19
- 1000: GPIO10
- 1001: GPTIM1_TRGO
- 1010: GPTIM0_CH2_OUT
- 1011: GPTIM3_TRGO
- 1100: GPTIM0_CH3_OUT
- 1101: GPTIM0_TRGO
- 1110: GPTIM2_CH1_OUT
- 1111: reserved

Notes:

1. *Software is allowed to write this bit only when START=0 (in register ADC_CR).*
2. *If the TRGO signal of GPTIMx is used as the trigger source for the ADC conversion, the MMS bit in the GPTIM0_CR2 and GPTIM1_CR2 registers can only be configured as 0b100 (OC0REF), 0b101 OC1REF), 0b110 (OC2REF) or 0b111 (OC3REF). For GPTIM2_CR2 and GPTIM3_CR2, only 0b100 (OC0REF) or 0b101 (OC1REF) can be selected.*
3. *To achieve timed trigger or periodic trigger, you need to configure the selected channel as output mode, select the corresponding output mode, and configure the corresponding GPTIMx_ARR and GPTIMx_CCRx according to the required time.*

Bit 12 DMA_EN: DMA enable.

- 0: DMA disabled
- 1: DMA enabled

Bits 11-0 CLK_DIV: ADCCLK prescale.

- 000: not divided
- 001: not divided
- n: ADC_IP_CLK=ADCCLK/n, 50% duty cycle

Notes:

1. *This bit can only be configured when all bits of the ADC_CR register are 0; the clock source selection for ADCCLK is configured in the RCC_CR2 register.*
2. *The clock division and clock source selection need to consider the data readout speed. The ADC samples every 16 ADC clock cycles, if a high-speed ADC clock source is chosen, the converted data cannot be read in time by the software or the DMA, which may cause overflow.*

16.14.3 ADC_SEQR0

偏移量：0x08

复位值：0x00000000

注意：仅在 ADC_CR{START} 和 ADC_CR{EN} 为 0 时，才能够配置该寄存器。

31-28	27-24	23-20	19-16	15-12	11-8	7-4	3-0
SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

Bits 31-28 SEL7: Select the channel number from 1 to 15 as the 7th in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

Bits 27-24 SEL6: Select the channel number from 1 to 15 as the 6th in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

Bits 23-20 SEL5: Select the channel number from 1 to 15 as the 5th in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

Bits 19-16 SEL4: Select the channel number from 1 to 15 as the 4th in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

Bits 15-12 SEL3: Select the channel number from 1 to 15 as the 3th in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

Bits 11-8 SEL2: Select the channel number from 1 to 15 as the 2nd in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

Bits 7-4 SEL1: Select the channel number from 1 to 15 as the 1st in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

Bits 3-0 SEL0: Select the channel number from 1 to 15 as the 0th in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

16.14.4 ADC_SEQR1

Offset: 0x00C

Reset Value: 0x00000000

Note: *Software is allowed to configure this register only when START=0 and EN=0 (in ADC_CR).*

31-28	27-24	23-20	19-16	15-12	11-8	7-4	3-0
SEL15	SEL14	SEL13	SEL12	SEL11	SEL10	SEL9	SEL8
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

Bits 31-28 SEL15: Select the channel number from 1 to 15 as the 15th in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

Bits 27-24 SEL14: Select the channel number from 1 to 15 as the 14th in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

Bits 23-20 SEL13: Select the channel number from 1 to 15 as the 13th in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

Bits 19-16 SEL12: Select the channel number from 1 to 15 as the 12th in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

Bits 15-12 SEL11: Select the channel number from 1 to 15 as the 11th in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

Bits 11-8 SEL10: Select the channel number from 1 to 15 as the 10th in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

Bits 7-4 SEL9: Select the channel number from 1 to 15 as the 9th in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

Bits 3-0 SEL8: Select the channel number from 1 to 15 as the 8th in the conversion sequence.
If 0 rather than 1 to 15 is configured, it marks the end of the sequence and itself cannot be converted. If the channel numbers selected by bits SEL**x** are the same, the conversion of the same channel in a sequence will be performed multiple times.
In differential input mode, only the channel number of the positive input needs to be configured by software, and the channel number of the negative input is selected automatically by hardware according to register *ADC_DIFFSEL*.

16.14.5 ADC_DIFFSEL

Offset: 0x010

Reset Value: 0x00000000

Note: *Software is allowed to configure this register only when START=0 and EN=0 (in ADC_CR).*

31-16	15-9	8-1	0
RESERVED	SEL1	SEL0	RESERVED
r-0h	r-0h	rw-0h	r-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-9 SEL1: Channels 9 to 15 are internal channels.

These channels can only be configured in single-ended mode. These bits are read-only.

Bits 8-1 SEL0: Differential or single-ended mode selection for Channels 1 to 8.

Each bit controls a channel with the same number as it:

- 0: channel x is configured in single-ended mode
- 1: channel x is configured in differential mode

In Differential mode, a group consists of two adjacent external channels, such as Group 1 consisting of Channel 2 and Channel 3, so the corresponding two control bits of this register should be set to 1 at the same time.

位 0 RESERVED：保留，不可更改。

16.14.6 ADC_ISR

Offset: 0x014

Reset Value: 0x00000000

Note: *It is recommended to clear all bits of this register before software sets the START bit (in register ADC_CR).*

31-3	2	1	0
RESERVED	OVERRUN	EOS	EOC
r-0h	rw1c-0h	rw1c-0h	rw1c-0h

Bits 31-3 RESERVED: Must be kept, and cannot be modified.

Bit 2 OVERRUN: ADC conversion overrun flag.

- 0: no overrun
- 1: overrun occurred

This bit is set by hardware when an overrun occurs and a new conversion is completed when the EOC flag was already set, but the *ADC_DR* register has not been read or software writing 1 to clear this bit was not configured.

It is cleared by software writing 1 to it.

Bit 1 EOS: End of sequence of conversions flag.

- 0: conversion sequence in progress
- 1: conversion sequence completed

This bit is set by hardware when a sequence of conversions (set by *ADC_SEQR0/1*) is completed.

It is cleared by software writing 1 to it.

Bit 0 EOC: End of conversion flag.

- 0: channel conversion in progress
- 1: channel conversion completed

This flag is set by hardware at the end of each conversion of a channel (when the newly converted data is stored in the *ADC_DR* register).

It is cleared by software writing 1 to it or by reading the *ADC_DR* register.

16.14.7 ADC_IER

偏移量：0x18

复位值：0x00000000

31-3	2	1	0
RESERVED	OVERRUN_INT_EN	EOS_INT_EN	EOC_INT_EN
r-0h	rw-0h	rw-0h	rw-0h

Bits 31-3 RESERVED: Must be kept, and cannot be modified.

Bit 2 OVERRUN_INT_EN: ADC conversion overrun interrupt enable.

- 0: overrun interrupt disabled
- 1: overrun interrupt enabled

Bit 1 EOS_INT_EN: End of conversion sequence interrupt enable.

- 0: end of conversion sequence interrupt disabled
- 1: end of conversion sequence interrupt enabled

Bit 0 EOC_INT_EN: End of conversion interrupt enable.

- 0: end of conversion interrupt disabled
- 1: end of conversion interrupt enabled

16.14.8 ADC_DR

Offset: 0x01C

Reset Value: 0x00000000

31-12	11-0
RESERVED	DATA
r-0h	r-0h

Bits 31-12 RESERVED: Must be kept, and cannot be modified.

Bits 11-0 DATA: ADC converted data. In differential mode, bit[11] is the sign bit.

17. Basic timer (BSTIM)

17.1 Introduction

BSTIMER (Basic Timer) contains a 16bits counter, supports auto-reloading function, and supports up to 16-bits programmable frequency division counter. There are two BSTIMERs, named BSTIMER0 and BSTIMER1.

17.2 Main features

BSTIMER includes the following functions:

- 16bits counter, up-counting, auto-reloading
- Prescaler
- DMA control
- Single pulse mode
- Master mode
- Update event management
- Debug mode control
- Interrupt signal generation

BSTIMER structure diagram:

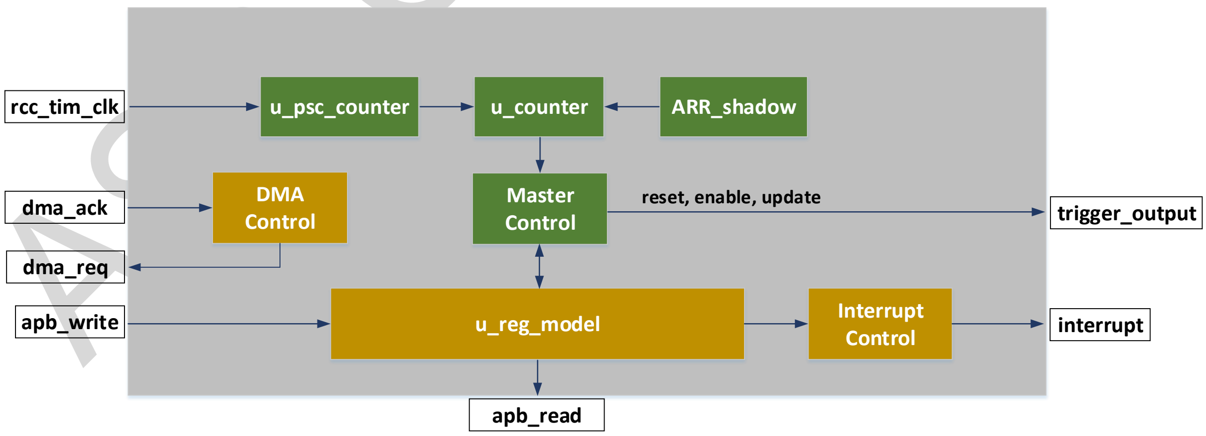

Figure 17-1 BSTIMER Diagram

- **rcc_tim_clk:** BSTIMER clock
- **dma_ack:** DMA ACK
- **dma_req:** BSTIMER DMA Request
- **apb_write:** APB bus write
- **apb_read:** APB bus read
- **trigger_output: trigger_output:** BSTIMER TRGO output
- **interrupt:** BSTIMER interrupt

17.3 Clock source

The clock source of the BSTIMER interface is PCLK and cannot be set to other clock sources. Please refer to the RCC chapter for clock enable and complex bits configuration.

17.4 Counter

The counter only supports upward counting. When counting to ARR, the counter value will change from ARR to 0, and then continue counting. At the same time, the status flag UIF is set. If the update event interrupt request is enabled, that is, UIE is set, an interrupt will also be generated, indicating that a counting cycle is completed. In the next counting cycle, the counter continues to count from 0, and so on.

17.5 Auto-reload

The ARPE Bits of Register BSTIM_CR1 can be configured by software to set whether to enable the ARR Shadow Register. If ARPE=0, the Shadow Register is disabled. The value written by the software is directly updated to the ARR for counter use. If ARPE=1, the value written by the software is the value will not take effect immediately, until the update event arrives, the value will be updated to ARR for counter use.

17.6 Prescaler

BSTIMER supports 16-bit (1~65535) programmable prescaler, which is implemented by the frequency division counter BSTIM_PSC. The interface clock is used as the clock of the prescaler, and the CEN of the register BSTIM_CR1 is used as the count enable of the prescaler. When the prescalet counts to the pre-loaded frequency division value, it outputs a pulse as the count enable of the next level counter, and then the prescaler returns to zero and counts again, and so on.

The division value of the division counter uses the shadow register by default, that is, the software write operation will not take effect immediately, but the new division value will be written into the shadow register until the update event (UG event is set, count overflow) arrives, at which time the division value will officially take effect. The software read operation reads the written register value, not the shadow register. If there are multiple write operations before the update event arrives, the previously written value will be overwritten. The counting and division waveforms are as follows:

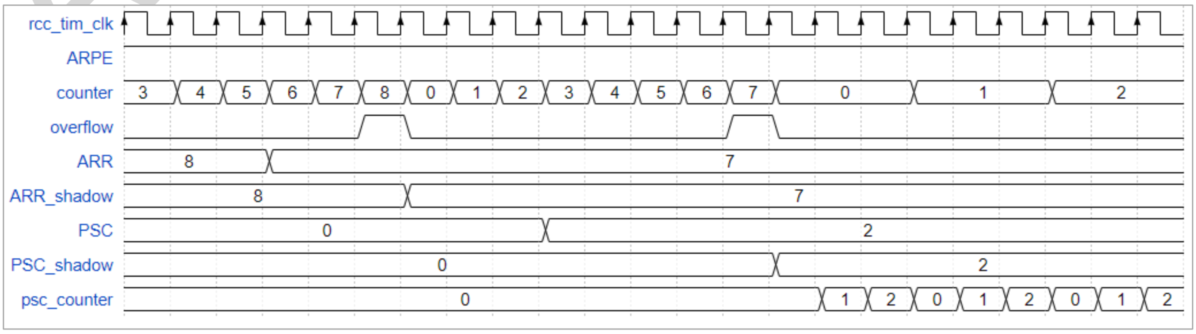

Figure 17-2 Counting and Dividing Waveforms

17.7 DMA

BSTIMER supports the DMA function. After enabling the DMA function, all its Registers except BSTIM_SR and BSTIM_EGR can transfer data to each other and memory. BSTIM_SR can only read data, and BSTIM_EGR can only write data. Enable DMA through UDE Bits of Register BSTIM_DIER , when there is an update event, a DMA Request will be generated. DMA 返回的 The ACK signal returned by DMA clears the module's DMA Request signal.

17.8 Single pulse mode

BSTIMER supports single pulse counting mode, which can be enabled by setting the OPM bit of register BSTIM_CR1. In this mode, when the counter counts to the ARR value, it will return to zero and stop counting (CEN hardware automatically clears to zero), and will not count again unless it is initialized again, as shown in the following figure:

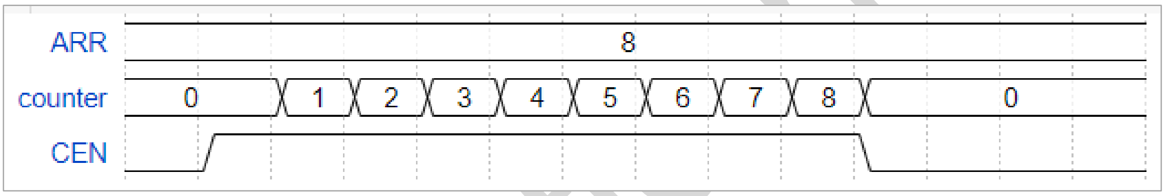

Figure 17-3 Single pulse waveform

17.9 Mode selection

BSTIMER can be cascaded with other internal modules and used as a host to control the DAC by generating a trigger output signal (TRGO). The source of the TRGO signal can be selected by software configuration of the MMS bit of the BSTIM_CR2 register, as follows:

- MMS=3'b000：Reset mode, in which case the UG flag will be output as a TRGO signal to the external slave.
- MMS=3'b001：Enable mode, in which case the counter's count enable CEN will be output as the TRGO signal to the external slave.
- MMS=3'b010：Update mode, in which case the update event is output to the external slave as the TRGO signal.
- Other values of MMS are reserved.

17.10 Update event management

Update events include the following event sources:

1. Counter overflow event (overflow), that is, the value of counter changes from ARR to 0.
2. UG Set Bits (software Set Bits), that is, configure the UG Bits of Register BSTIM_EGR.

The control signals related to update event management are URS and UDIS in BSTIM_CR1 register . The specific controls are as follows:

- If UDIS=0, URS=0, the overflow and UG bits setting will initialize the counter and prescaler. If the shadow register is enabled, the update event will update the written value to the shadow register (ARR depends on ARPE).
  UIF will set. If interrupt or DMA is enabled, interrupt or DMA Request will be generated.
- If UDIS=0, URS=1, the overflow and UG bits setting will initialize the counter and prescaler. If the shadow register is enabled, the update event will update the written value to the shadow register (ARR depends on ARPE).
  UIF will only set Bits in the case of overflow. If interrupt or DMA is enabled, interrupt or DMA Request will be generated.
- If UDIS=1 (URS ignored), only UG bit setting will still initialize counter and prescaler, but the shadow register will not be updated, and UIF will not set, so no corresponding interrupt or DMA request will be generated.

17.11 Debug mode control

BSTIMER can be configured by software whether to stop counting under debug. The DEBUG mode counting control of BSTIMER0 and BSTIMER1 is implemented through the CR2 Register of SYSCFG. If this function is enabled, BSTIMER will stop counting when entering the system debug mode (the counter will not be initialized).

17.12 Interrupts

The interrupt signal of BSTIMER is as follows:

Table 17-1 BSTIMER interrupts

Interrupt	Description
Update event interrupt	Counter overflow and UG bit setting can generate update event interrupt.

The interrupts above are enabled by configuring the UIE bit in the BSTIM_DIER register, and the interrupt status can be obtained through the BSTIM_SR register.

17.13 BSTIMER registers

BSTIMER0 Base Address: 0x4000C000

BSTIMER1 Base Address: 0x4001C000

Table 17-2 BSTIMER Registers Summary

Register	Offset	Description
BSTIM_CR1	0x00	控制寄存器 1
BSTIM_CR2	0x04	控制寄存器 2
BSTIM_DIER	0x0c	DMA/中断使能寄存器
BSTIM_SR	0x10	状态寄存器
BSTIM_EGR	0x14	事件寄存器
BSTIM_CNT	0x24	计数器寄存器
BSTIM_PSC	0x28	计数器分频值
BSTIM_ARR	0x2c	计数器重装载值

17.13.1 BSTIM_CR1

Offset: 0x000

Reset Value: 0x00000000

31-8	7	6-4	3	2	1	0
RESERVED	ARPE	RESERVED	OPM	URS	UDIS	CEN
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bit 7 ARPE: Reload shadow Register enable.

- 0: BSTIM_ARR Shadow Register disabled
- 1: BSTIM_ARR Shadow Register enabled

Bits 6-4 RESERVED: Must be kept, and cannot be modified.

Bit 3 OPM: Single pulse mode enable.

- 0: single pulse mode disabled
- 1: single pulse mode Enabled, counter stops counting at the next update event

Bit 2 URS: Update event source selection, this bit only affects the interrupt (UIF) and DMA flag bits, and does not affect the internal logic.

- 0: Counter overflow and UG bit can set UIF bit
- 1: Only counter overflow events can set UIF bit

Bit 1 UDIS: Update events are disable.

- 0: Update event enabled, update events can be generated.
- 1: The update event is disabled, the shadow Register and UIF will not be updated, but at this time the counter and prescaler can still be initialized by the UG bit Set event.

Bit 0 CEN: Counter enable, CEN is cleared by hardware in single pulse mode.

- 0: Counter disabled
- 1: Counter enabled

17.13.2 BSTIM_CR2

Offset: 0x004

Reset Value: 0x00000000

31-7	6-4	3-0
RESERVED	MMS	RESERVED
rw-0h	rw-0h	rw-0h

Bits 31-7 RESERVED: Must be kept, and cannot be modified.

Bits 6-4 MMS: Master mode selection, TRGO output can be configured.

- 000: Reset mode, UG will be output as TRGO signal
- 001: Enable mode, CEN will be output as TRGO signal
- 010: Update mode, update events (internal signals) will be output as TRGO signals
- Other values: Reserved

Bits 3-0 RESERVED: Must be kept, and cannot be modified.

17.13.3 BSTIM_DIER

Offset: 0x00C

Reset Value: 0x00000000

31-9	8	7-1	0
RESERVED	UDE	RESERVED	UIE
rw-0h	rw-0h	rw-0h	rw-0h

Bits 31-9 RESERVED: Must be kept, and cannot be modified.

Bit 8 UDE: Update event DMA Request Enable.

- 0: disable update event DMA Request
- 1: enable update event DMA Request

Bit 0 UIE: Update event Interrupt Request enable.

Bit 0 UIE: Update event Interrupt Request enable.

- 0: Update event Interrupt Request disabled
- 1: Update event Interrupt Request enabled

PDF-XChange Editor
Click to BUY NOW!
www.pdf-xchange.com

17.13.4 BSTIM_SR

Offset: 0x010

Reset Value: 0x00000000

31-1	0
RESERVED	UIF
r-0h	r-0h

Bits 31-1 RESERVED: Must be kept, and cannot be modified.

Bit 0 UIF: Update event flag.

- 0: no update event
- 1: update event occurs

17.13.5 BSTIM_EGR

Offset: 0x014

Reset Value: 0x00000000

31-1	0
RESERVED	UG
w-0h	w-0h

Bits 31-1 RESERVED: Must be kept, and cannot be modified.

Bit 0 UG: Update event generation enable.

- 0: update event generation disabled
- 1: update event generation enabled

17.13.6 BSTIM_CNT

Offset: 0x024

Reset Value: 0x00000000

31-16	15-0
RESERVED	CNT
rw-0h	rw-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 CNT: Counter count value.

17.13.7 BSTIM_PSC

Offset: 0x028

Reset Value: 0x00000000

31-16	15-0
RESERVED	PSC
rw-0h	rw-0h

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 PSC: Clock division value is PSC+1.

17.13.8 BSTIM_ARR

Offset: 0x02C

Reset Value: 0x0000ffff

31-16	15-0
RESERVED	ARR
rw-0h	rw-ffffh

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 ARR: Counter auto-reload value.

18. RTC

18.1 Introduction

The Real-time Clock is an independent BCD timer/counter. It has two 32-bit registers, which contain the seconds, minutes, hours (12-hour or 24-hour format), day of week, date of month, month, and year, expressed in binary coded decimal format (BCD). In addition, there is a 32-bit register used to indicate sub-seconds value. The RTC supports operation under low-power mode.

18.2 Main Features

The features of RTC are as follows:

- Calendar with the seconds, minutes, hours (12-hour or 24-hour format), day of week, date of month, month, and year, expressed in binary coded decimal format (BCD).
- RTC frequency calibration with a resolution of about 0.5 ppm with a range from -1024 ppm to +1024 ppm.
- Wake-up from low-power modes.
- Tamper/wakeup IO detection at high or low level with configurable filter.
- 32-bit counter for periodic count.
- 2 Alarms, support calendar matching.
- Clear retention SRAM on tamper/wakeup alarm.
- Internal signal output by GPIO, including Alarm0 pulse, Alarm1 pulse, periodic counter pulse and second signal.
- Calendar values reading.
- Sub-seconds value reading.
- Period counting value reading.
- Interrupt signal generation.

18.3 Interface Clock

Both XO32K and RCO32K can be RTC clock source and XO32K accuracy is higher than RCO32K.

See *RCC* Chapter for clock configuration details.

18.4 Calendar

The RTC calendar time and date are accessed through two types of registers, which are the asynchronous registers and the synchronous registers.

- *RTC_SYNDATA* indicates the seconds, minutes and hours; *RTC_SYNDATA_H* indicates the day of week, date of month, month and year.
- *RTC_CALENDAR_R* and *RTC_CALENDAR_R_H* are **synchronous registers**. *RTC_CALENDAR_R* indicates the seconds, minutes and hours; *RTC_CALENDAR_R_H* indicates the day of week, date of month, month and year.

18.4.1 Calendar reading

This document only introduces reading the RTC calendar values by **synchronous registers**. The synchronous registers should be read several times with the same result obtained to ensure that the data is correct. Follow below steps to read the calendar:

(1) Read the *RTC_SUB_SECOND* register to get the subsecond_count value.

(2) Read the value of the *RTC_CALENDAR_R* register for two consecutive times, if the values read are different, then continue reading until the values read for two consecutive times are the same.

(3) Read the value of the *RTC_CALENDAR_R_H* register for two consecutive times, if the values read are different, then continue reading until the values read for two consecutive times are the same.

(4) Read the value of the *RTC_SUB_SECOND* register again, if the value is not equal to the value in Step 1, then software will restart reading the calendar from Step 1.

(5) When the subseconds downcounter reaches 0, the value of the *RTC_CALENDAR_R* or *RTC_CALENDAR_R_H* register may have no change, so if the subsecond_count value is 0, then the software will restart reading the calendar from Step 1; if subsecond_count value is not 0, then it indicates that the complete calendar time has been read correctly.

For converting subsecond_count to sub-second (unit: microsecond), first obtain the frequency of the RTC interface clock (fRTCCLK) through the RTC_CLK_SEL bit in the *RCC_CR1* register, then use the formula below to calculate the sub-second:

sub-second=(1000000*SUBSECOND_COUNT)/fRTCCLK

18.4.2 Calendar setting

The *RTC_CALENDAR_H* and *RTC_CALENDAR* registers are used to set the calendar. RTC_CALENDAR_H sets the year, month, date of month and day of week. RTC_CALENDAR sets the hours, minutes and seconds. Since the *RTC_SUB_SECOND* register is read-only, the subsecond is read-only. Follow below steps to set the calendar:

(1) Read the *RTC_SR1* register, and wait for all WRITE_XXX_DONE bits and the SECOND_SR bit (bits[11:1]) to be set. After that, writing to the *RTC_CALENDAR_H* and *RTC_CALENDAR* registers is allowed.

(2) Configure the year, month, date of month and day of week in the *RTC_CALENDAR_H* register.

(3) Read the *RTC_SR1* register, and wait for all WRITE_XXX_DONE bits and the SECOND_SR bit (bits[11:1]) to be set. After that, writing to the *RTC_CALENDAR_H* and *RTC_CALENDAR* registers is allowed.

(4) Configure the hours, minutes and seconds in the *RTC_CALENDAR* register.

18.5 RTC PPM Calibration

The RTC frequency can be calibrated with a resolution of about 0.5 ppm with a range from -1024 ppm to +1024 ppm. Configure register *RTC_PPMADJUST* to set the adjustment value. When the value is set to 0x7FFF, it means to adjust 0 ppm, that is, no adjustment is required. Follow below steps to conduct the PPM calibration:

(1) Read the *RTC_SR1* register, and wait for all WRITE_XXX_DONE bits and the SECOND_SR bit bits[11:1]) to be set. After that, writing to the *RTC_PPMADJUST* register is allowed.

(2) Configure the adjustment value in the *RTC_PPMADJUST* register.

18.6 Wake-up from Low-power Mode

RTC can wake up the MCU from Sleep, Stop or Standby mode through an interrupt or wakeup signal.

Table 18-1 RTC Wakeup Source

Mode	Description
Sleep	RTC 所有中断可以使设备从 sleep 模式中唤醒
Stop0、Stop1、Stop2、Stop3	Wakeup/tamper IO、alarm、cyc 的唤醒信号可以使设备从 stop 模式中唤醒
Standby	Wakeup/tamper IO、alarm、cyc 的唤醒信号可以使设备从 standby 模式中唤醒

Enable the wakeup/tamper IO, RTC alarm and periodic count signal for wake-up through the corresponding bit in register *RTC_CR*：

Table 18-2 Bits to Enable Wake-up Signals

功能	RTC_CR 寄存器 bit 信息
WAKEUP_IO0	WAKEUP0_WKEN1
WAKEUP_IO1	WAKEUP1_WKEN1
WAKEUP_IO2	WAKEUP2_WKEN1
TAMPER	TAMPER_WKEN1
ALARM0	RTC_ALARM0_WKEN
ALARM1	RTC_ALARM1_WKEN
CYC	CYC_WKEN

18.7 Tamper/Wakeup IO Detection

The tamper/wakeup IO input events can be configured for edge detection or level detection with filtering. Edge detection means to detect the rising or falling edge of GPIO, while level detection means to detect the high or low level of GPIO. If GPIO is active at high level, a tamper detection event is generated when a high level is detected on GPIO input; if GPIO is active at low level, a tamper detection event is generated when a high level is detected on GPIO input. When an input event is detected, the following actions can be conducted:

- Erase the retention SRAM.
- Generate an interrupt, capable to wakeup from Sleep mode.
- Generate a wakeup signal (WAKEUP_IO0/WAKEUP_IO1/WAKEUP_IO2/TAMPER), capable to wakeup from Stop and Standby modes.

18.7.1 Tamper/Wakeup Initialization and Configuration

Before Tamper/Wakeup initialization, the corresponding GPIO should be configured as tamper/wakeup function. In addition, if it is level detection, GPIO should be configured with pull-up or pull-down. If GPIO is active at high level, pull down GPIO; if GPIO is active at low level, pull up GPIO.

Taking Tamper as an example, the initialization and configuration process is as follows:

(1) If it is level detection, set the filter length by configuring bit TAMPER_FILTER_CFG in the *RTC_CR* register, configure the active level by bit TAMPER_LEVEL_SEL, and then enable the tamper pin level wakeup by the TAMPER_WKEN0 bit. ***If it is edge detection, ignore this step.***

(2) Configure bit TAMPER_WKEN1 in register *RTC_CR* to enable TAMPER_SR to wake up the MCU from Stop or Standby mode. ***If no such need, ignore this step***.

(3) Set the TAMPER_EN bit in the *RTC_CR* register to enable tamper detection.

18.7.2 Retention SRAM Erase Operation

When the tamper/wakeup IO input event is detected, the hardware can erase the retention SRAM. This is configured by setting the corresponding bit of RTC_RET_SRAM_ERASE_EN in register *RTC_CR2*. Bit0 corresponds to wakeup IO0, bit1 corresponds to wakeup IO1, bit2 corresponds to wakeup IO2, and bit3 corresponds to tamper function.

18.8 Periodic Counter

The periodic counter generates interrupts or wakeup events at regular intervals. The regular interval is set according to the configured CYC_MAX_VALUE in the *RTC_CYC_MAX_VALUE* register. Obtain the RTC interface clock frequency (fRTCCLK) through the RTC_CLK_SEL bit in the *RCC_CR1* register, and then use the formula below to calculate the regular interval (in microseconds):

Regular interval= (1000000 * CYC_MAX_VALUE) / fRTCCLK

During the periodic count, the number of elapsed cycles is read from the CYC_CNT_VALUE bits in the *RTC_CYC_CNT_VALUE* register. On this basis, the interval (in microseconds) from the start of the ongoing counting to the current moment can be calculated by the formula below:

Interval= (1000000 * CYC_CNT_VALUE) / fRTCCLK

Follow below steps to configure the periodic count:

(1) When the regular interval is known, calculate the CYC_MAX_VALUE according to the above formula, and configure this value to register *RTC_CYC_MAX_VALUE*.

(2) Configure bit CYC_WKEN in register *RTC_CR* to enable CYC_SR to wake up CPU from Stop or Standby mode. ***If no such need, ignore this step***.

(3) Set bit CYC_START_COUNTER in register *RTC_CR* to enable periodic counter.

18.9 RTC Alarms

RTC provides two alarms: Alarm 0 and Alarm 1. Both support mask selection and matching with calendar. With Mask configuration, each calendar field (sub-seconds, seconds, minutes, hours, date or day of week) can be independently selected to match the values programmed in the alarm registers. Note that for the date and the day of week, we can only choose one of them for the match.

If bit ALARM**x**_WEEK_SEL (Alarm**x** means Alarm 0 or Alarm 1, similarly hereinafter) is 0 in the register *RTC_ALARM**x***, the date is selected for the match; if bit ALARM**x**_WEEK_SEL is 1, the day of week is selected for the match.

If the sub-seconds and seconds are not involved but the minutes are involved in Alarm**x** comparison, when Alarm**x** values match with those of the RTC Calendar, 60 interrupts or/and 60 wake-up events are generated at a one-second interval in one minute. If the sub-seconds, seconds, and minutes are not involved but the hours are involved in Alarm**x** comparison, when Alarm**x** values match with those of the RTC Calendar, 3600 interrupts or/and 3600 wake-up events are generated at a one-second interval in one hour. Whether the interrupts or/and wake-up events are generated depends on whether the alarm interrupt or/and the alarm wake-up is enabled.

The seconds, minutes, hours, date or day mask are configured through the ALARM**x**_MASK bit field in the *RTC_ALARM**x*** register, and the sub-seconds mask is configured through the RTC_ALARM**x**_SUB_MASK bit field in the *RTC_ALARM**x**_SUB* register. The sub-seconds value is set by the RTC_ALARM**x**_SUB_VALUE bit field in the *RTC_ALARM**x**_SUB* register.

The RTC_ALARM**x**_SUB_VALUE indicates RTC clock cycles, and the formula for converting clock cycles to time is the same as that in periodic count.

Take Alarm 0 as an example to describe the alarm configuration process as follows:

(1) Set the calendar.
(2) Configure the Alarm 0 values (including the hours, minutes, seconds, date or day) by the ALARM0_VALUE bit field in the *RTC_ALARM0* register.
(3) Configure the sub-seconds value for Alarm 0 through the RTC_ALARM0_SUB_VALUE bit field in the *RTC_ALARM0_SUB* register.
(4) Configure the seconds, minutes, hours, date or day mask for Alarm 0.
(5) Configure the sub-seconds mask for Alarm 0.
(6) Whether the ALARM0_SR interrupt or ALARM0_SR wake-up is enabled depends on the specific needs. They are configured through the ALARM0_SR_INT_EN bit in register *RTC_CR1* and the RTC_ALARM0_WKEN bit in register *RTC_CR*.
(7) Enable the Alarm 0 through the ALARM0_EN bit in register *RTC_ALARM0*.
(8) Enable the calendar by setting the RTC_START_RTC bit in register *RTC_CR*.

18.10 Internal Signal Output through IO

The internal signals that can be output through IO include Alarm 0 pulse, Alarm 1 pulse, periodic counter pulse, and the second signal. The alarm pulse and periodic counter pulse are pulses with a width of one RTC clock cycle. The Alarm pulse is output when the programmed values match with the Calendar. The periodic counter pulse is output every time the programmed count value is reached. The second signal is a square wave with a duty cycle of 50% and the frequency is 1 Hz. The RTC IO can output inverted levels. When the RTC_OUT_POL bit of the *RTC_CR2* register is 0, it means that the level is non-inverted, and when this bit is 1, it means that the level is inverted. Configure the RTC_OUT_SEL bit of the *RTC_CR2* register to select the RTC IO output signal.

18.11 Interrupts

The interrupt signals of RTC are as follows:

Table 18-3 RTC Interrupts

Interrupt	Description
闹钟 0 中断	闹钟 0 定时时间到产生的中断
闹钟 1 中断	闹钟 1 定时时间到产生的中断
周期计数中断	计数满一个周期时产生的中断
Tamper 中断	Tamper 检测到输入事件时产生的中断
Wakeupio0 中断	Wakeupio0 检测到输入事件时产生的中断
Wakeupio1 中断	Wakeupio1 检测到输入事件时产生的中断
Wakeupio2 中断	Wakeupio2 检测到输入事件时产生的中断
秒中断	秒信号在每一秒产生的中断

The above interrupts are enabled by configuring the *RTC_CR1* register. The second signal interrupt status is indicated by the SECOND_SR bit in the *RTC_SR1* register, and the other interrupts' status is indicated by the *RTC_SR* register.

18.12 RTC Registers

Base Address: 0x4000E000

Table 18-4 RTC Registers Summary

Register	Offset	Description
RTC_CR	0x00	控制寄存器
RTC_ALARM0	0x04	闹钟 0 寄存器
RTC_ALARM1	0x08	闹钟 1 寄存器
RTC_PPMADJUST	0x0c	PPM 调整寄存器
RTC_CALENDAR	0x10	日历配置时分秒寄存器
RTC_CALENDAR_H	0x14	日历配置年月日星期寄存器
RTC_CYC_MAX_VALUE	0x18	周期计数值配置寄存器
RTC_SR	0x1c	中断状态寄存器
RTC_ASYNDATA	0x20	日历异步读取时分秒寄存器
RTC_ASYNDATA_H	0x24	日历异步读取年月日星期寄存器
RTC_CR1	0x28	中断使能寄存器
RTC_SR1	0x2c	操作状态寄存器
RTC_CR2	0x30	控制寄存器 2
RTC_SUB_SECOND	0x34	读取 sub-second 寄存器
RTC_CYC_CNT_VALUE	0x38	只读周期计数值寄存器
RTC_ALARM0_SUB	0x3c	闹钟 0 的 sub-second 寄存器
RTC_ALARM1_SUB	0x40	闹钟 1 的 sub-second 寄存器
RTC_CALENDAR_R	0x44	日历同步读取时分秒寄存器
RTC_CALENDAR_R_H	0x48	日历同步读取年月日星期寄存器

18.12.1 RTC_CR

Offset: 0x000

Reset Value: 0x00000000

31-29	28	27	26	25
RESERVED	RTC_START_RTC	RTC_ALARM0_WKEN	RTC_ALARM1_WKEN	CYC_WKEN
r-0h	rw-0h	rw-0h	rw-0h	rw-0h
24	**23**	**22**	**21**	**20**
CYC_START_COUNTER	TAMPER_EN	TAMPER_LEVEL_SEL	TAMPER_WKEN0	TAMPER_WKEN1
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h
19-18	**17**	**16**	**15**	**14**
TAMPER_FILTER_CFG	WAKEUP0_EN	WAKEUP0_LEVEL_SEL	WAKEUP0_WKEN0	WAKEUP0_WKEN1
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h
13-12	**11**	**10**	**9**	**8**
WAKEUP0_FILTER_CFG	WAKEUP1_EN	WAKEUP1_LEVEL_SEL	WAKEUP1_WKEN0	WAKEUP1_WKEN1
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

7-6	5	4	3	2	1-0
WAKEUP1_FILTER_CFG	WAKEUP2_EN	WAKEUP2_LEVEL_SEL	WAKEUP2_WKEN0	WAKEUP2_WKEN1	WAKEUP2_FILTER_CFG
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

Bits 31-29 RESERVED: Must be kept, and cannot be modified.

Bit 28 RTC_START_RTC: RTC calendar enable.

- 0: disabled
- 1: enabled

Bit 27 RTC_ALARM0_WKEN: ALARM0_SR wake-up enable.

- 0: disabled
- 1: enabled

Bit 26 RTC_ALARM1_WKEN: ALARM1_SR wake-up enable.

- 0: disabled
- 1: enabled

Bit 25 CYC_WKEN: CYC_SR wake-up enable.

- 0: disabled
- 1: enabled

Bit 24 CYC_START_COUNTER: Periodic counter enable.

- 0: disabled
- 1: enabled

Bit 23 TAMPER_EN: Tamper enable.

- 0: disabled
- 1: enabled

Bit 22 TAMPER_LEVEL_SEL: Tamper active level selection.

- 0: low level
- 1: high level

Bit 21 TAMPER_WKEN0: Tamper level wake-up enable.

- 0: disabled
- 1: enabled

When TAMPER_EN is set to 0, this configuration is still valid.

Bit 20 TAMPER_WKEN1: TAMPER_SR wake-up enable.

- 0: disabled
- 1: enabled

Bits 19-18 TAMPER_FILTER_CFG: Tamper filter control.

- 0: none
- 1: filter length is 1 RTC interface clock cycle
- 2: filter length is 3 RTC interface clock cycles
- 3: filter length is 7 RTC interface clock cycles

Bit 17 WAKEUP0_EN: WAKEUP0 enable.

- 0: disabled
- 1: enabled

Bit 16 WAKEUP0_LEVEL_SEL: WAKEUP0 active level selection.

- 0: low level
- 1: high level

Bit 15 WAKEUP0_WKEN0: WAKEUP0 level wake-up enable.

- 0: disabled
- 1: enabled

When WAKEUP0_EN is set to 0, this configuration is still valid.

Bit 14 WAKEUP0_WKEN1: WAKEUP0_SR wake-up enable.

- 0: disabled
- 1: enabled

Bits 13-12 WAKEUP0_FILTER_CFG: WAKEUP0 filter control.

- 0: none
- 1: filter length is 1 RTC interface clock cycle
- 2: filter length is 3 RTC interface clock cycles
- 3: filter length is 7 RTC interface clock cycles

Bit 11 WAKEUP1_EN: WAKEUP1 enable.

- 0: disabled
- 1: enabled

Bit 10 WAKEUP1_LEVEL_SEL: WAKEUP1 active level selection.

- 0: low level
- 1: high level

Bit 9 WAKEUP1_WKEN0: WAKEUP1 level wake-up enable.

- 0: disabled
- 1: enabled

When WAKEUP1_EN is set to 0, this configuration is still valid.

Bit 8 WAKEUP1_WKEN1: WAKEUP1_SR wake-up enable.

- 0: disabled
- 1: enabled

Bits 7-6 WAKEUP1_FILTER_CFG: WAKEUP1 filter control.

- 0: none
- 1: filter length is 1 RTC interface clock cycle
- 2: filter length is 3 RTC interface clock cycles
- 3: filter length is 7 RTC interface clock cycles

Bit 5 WAKEUP2_EN: WAKEUP2 enable.

- 0: disabled
- 1: enabled

Bit 4 WAKEUP2_LEVEL_SEL: WAKEUP2 active level selection.

- 0: low level
- 1: high level

Bit 3 WAKEUP2_WKEN0: WAKEUP2 level wake-up enable.

- 0: disabled
- 1: enabled

When WAKEUP2_EN is set to 0, this configuration is still valid.

Bit 2 WAKEUP2_WKEN1: WAKEUP2_SR wake-up enable.

- 0: disabled
- 1: enabled

Bits 1-0 WAKEUP2_FILTER_CFG: WAKEUP2 filter control.

- 0: none
- 1: filter length is 1 RTC interface clock cycle
- 2: filter length is 3 RTC interface clock cycles
- 3: filter length is 7 RTC interface clock cycles

18.12.2 RTC_ALARM0

Offset: 0x004

Reset Value: 0x00000000

31	30	29-26	25-0
ALARM0_EN	ALARM0_WEEK_SEL	ALARM0_MASK	ALARM0_VALUE
rw-0h	rw-0h	rw-0h	rw-0h

Bit 31 ALARM0_EN: Alarm 0 enable.

- 0: disabled
- 1: enabled

Bit 30 ALARM0_WEEK_SEL: Date or day of week selection for Alarm 0.

- 0: match the date
- 1: match the day of week

Bits 29-26 ALARM0_MASK: Alarm 0 mask configuration.

[26] Alarm 0 seconds mask

- 0: match the seconds
- 1: seconds are not involved in Alarm 0 comparison

[27] Alarm 0 minutes mask

- 0: match the minutes
- 1: minutes are not involved in Alarm 0 comparison

[28] Alarm 0 hours mask

- 0: match the hours
- 1: hours are not involved in Alarm 0 comparison

[29] Alarm 0 date or day of week mask

- 0: match the date or day of week
- date or day of week is not involved in Alarm 0 comparison

Bits 25-0 ALARM0_VALUE: Alarm 0 value configuration. When the calendar sub-seconds, seconds, minutes, hours, date or day of week match the values programmed in this register and the *RTC_ALARM0_SUB* register, the ALARM0_SR bit is set.

[3:0]: second units

[6:4]: second tens

[10:7]: minute units

[13:11]: minute tens

[17:14]: hour units

[19:18]: hour tens

[23:20]: bits[23:20] configure date units or bits[22:20] configure day of week

[25:24]: date tens

18.12.3 RTC_ALARM1

Offset: 0x008

Reset Value: 0x00000000

31	30	29-26	25-0
ALARM1_EN	ALARM1_WEEK_SEL	ALARM1_MASK	ALARM1_VALUE
rw-0h	rw-0h	rw-0h	rw-0h

Bit 31 ALARM1_EN: Alarm 1 enable.

- 0: disabled
- 1: enabled

Bit 30 ALARM1_WEEK_SEL: Date or day of week selection for Alarm 1.

- 0: match the date
- 1: match the day of week

Bits 29-26 ALARM1_MASK: Alarm 1 mask configuration.

[26] Alarm 1 seconds mask

- 0: match the seconds
- 1: seconds are not involved in Alarm 1 comparison.

[27] Alarm 1 minutes mask

- 0: match the minutes
- 1: minutes are not involved in Alarm 1 comparison.

[28] Alarm 1 hours mask

- 0: match the hours
- 1: hours are not involved in Alarm 1 comparison.

[29] Alarm 1 date or day of week mask

- 0: match the date or day of week
- 1: date or day of week is not involved in Alarm 1 comparison

Bits 25:0 ALARM1_VALUE: Alarm 1 value configuration. When the calendar sub-seconds, seconds, minutes, hours, date or day of week match the values programmed in this register and the *RTC_ALARM1_SUB* register, the ALARM1_SR bit is set.

[3:0]: second units

[6:4]: second tens

[10:7]: minute units

[13:11]: minute tens

[17:14]: hour units

[19:18]: hour tens

[23:20]: bits[23:20] configure date units or bits[22:20] configure day of week

[25:24]: date tens

18.12.4 RTC_PPMADJUST

Offset: 0x00c

Reset Value: 0x00007fff

31-16	15-0
RESERVED	PPMADJUST_VALUE
r-0h	rw-7fffh

Bits 31-16 RESERVED: Must be kept, and cannot be modified.

Bits 15-0 PPMADJUST_VALUE: The RTC clock frequency can be calibrated with a resolution of about 0.5 ppm with a range from -1024 ppm to +1024 ppm.

- 77ff: increase frequency of RTC by 1024 ppm
- 7800: increase frequency of RTC by 1023.5 ppm
- ...
- 7ffd: increase frequency of RTC by 1 ppm
- 7ffe: increase frequency of RTC by 0.5 ppm
- 7fff: no adjustment
- 8000: decrease frequency of RTC by 0.5 ppm
- 8001: decrease frequency of RTC by 1 ppm
- ...
- 87fe: decrease frequency of RTC by 1023.5 ppm
- 87ff: decrease frequency of RTC by 1024 ppm

18.12.5 RTC_CALENDAR

Offset: 0x010

Reset Value: 0x00000000

31-20	19-0
RESERVED	CALENDAR_VALUE
r-0h	w-0h

Bits 31-20 RESERVED: Must be kept, and cannot be modified.

Bits 19-0 CALENDAR_VALUE: RTC calendar time values.

[3:0]: second units

[6:4]: second tens

[10:7]: minute units

[13:11]: minute tens

[17:14]: hour units

[19:18]: hour tens

18.12.6 RTC_CALENDAR_H

Offset: 0x014

Reset Value: 0x00000841

31-22	21-0
RESERVED	CALENDAR_H_VALUE
r-0h	w-841h

Bits 31-22 RESERVED: Must be kept, and cannot be modified.

Bits 21-0 CALENDAR_H_VALUE: RTC calendar date values.

[3:0]: date units

[5:4]: date tens

[9:6]: month units

[10]: month tens

[13:11]: week day units

[17:14]: year units

[17:14]: year units

18.12.7 RTC_CYC_MAX_VALUE

Offset: 0x018

Reset Value: 0x00008000

31-0
CYC_MAX_VALUE
rw-8000h

Bits 31-0 CYC_MAX_VALUE: The programmed count value for the periodic counter to reach. When the periodic counter reaches the CYC_MAX_VALUE, the periodic counter status flag (bit CYC_SR) is set. The periodic counter is clocked by the RTC interface clock.

18.12.8 RTC_SR

Offset: 0x01c

Reset Value: 0x00000000

31-7	6	5	4
RESERVED	ALARM0_SR	ALARM1_SR	CYC_SR
r-0h	rw-0h	rw-0h	rw-0h
3	**2**	**1**	**0**
TAMPER_SR	WAKEUP0_SR	WAKEUP1_SR	WAKEUP2_SR
rw-0h	rw-0h	rw-0h	rw-0h

Bits 31-7 RESERVED: Must be kept, and cannot be modified.

Bit 6 ALARM0_SR: Alarm 0 flag.

This flag is set by hardware and cleared by software writing 1 to it.

- 0: Alarm 0 values doesn't match the Calendar
- 1: Alarm 0 values match the Calendar

Bit 5 ALARM1_SR: Alarm 1 flag.

This flag is set by hardware and cleared by software writing 1 to it.

- 0: Alarm 1 values doesn't match the Calendar
- 1: Alarm 1 values match the Calendar

Bit 4 CYC_SR: Periodic counter flag.

This flag is set by hardware and cleared by software writing 1 to it.

- 0: CYC_MAX_VALUE is not reached
- 1: CYC_MAX_VALUE is reached

Bit 3 TAMPER_SR: Tamper flag.

This flag is set by hardware and cleared by software writing 1 to it.

- 0: tamper pin active level is not detected
- 1: tamper pin active level is detected

Bit 2 WAKEUP0_SR: Wakeup0 flag.

This flag is set by hardware and cleared by software writing 1 to it.

- 0: the Wakeup0 active level is not detected
- 1: the Wakeup0 active level is detected

Bit 1 WAKEUP1_SR: Wakeup1 flag.

This flag is set by hardware and cleared by software writing 1 to it.

- 0: Wakeup1 active level is not detected
- 1: Wakeup1 active level is detected

Bit 0 WAKEUP2_SR: Wakeup2 flag.

This flag is set by hardware and cleared by software writing 1 to it.

- 0: Wakeup2 active level is not detected
- 1: Wakeup2 active level is detected

18.12.9 RTC_ASYNDATA

Offset: 0x020

Reset Value: 0x00000000

31-20	19-0
RESERVED	SYN_DATA
r-0h	r-0h

Bits 31-20 RESERVED: Must be kept, and cannot be modified.

Bits 19-0 SYN_DATA: RTC calendar time values. This register is read-only by software.

[3:0]: second units

[6:4]: second tens

[10:7]: minute units

[13:11]: minute tens

[17:14]: hour units

[19:18]: hour tens

18.12.10 RTC_ASYNDATA_H

Offset: 0x024

Reset Value: 0x00000000

31-22	21-0
RESERVED	SYN_DATA_H
r-0h	r-0h

Bits 31-22 RESERVED: Must be kept, and cannot be modified.

Bits 21-0 SYN_DATA_H: RTC calendar date values. This register is read-only by software.

[3:0]: date units

[5:4]: date tens

[9:6]: month units

[10]: month tens

[13:11]: week day units

[17:14]: year units

[21:18]: year tens

18.12.11 RTC_CR1

Offset: 0x028

Reset Value: 0x00000000

31-8	7	6
RESERVED	SECOND_SR_INT_EN	ALARM0_SR_INT_EN
r-0h	rw-0h	rw-0h
5	**4**	**3**
ALARM1_SR_INT_EN	CYC_SR_INT_EN	TAMPER_SR_INT_EN
rw-0h	rw-0h	rw-0h
2	**1**	**0**
WAKEUP0_SR_INT_EN	WAKEUP1_SR_INT_EN	WAKEUP2_SR_INT_EN
rw-0h	rw-0h	rw-0h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bit 7 SECOND_SR_INT_EN: SECOND_SR interrupt enable.

- 0: disabled
- 1: enabled

Bit 6 ALARM0_SR_INT_EN: ALARM0_SR interrupt enable.

- 0: disabled
- 1: enabled

Bit 5 ALARM1_SR_INT_EN: ALARM1_SR interrupt enable.

- 0: disabled
- 1: enabled

Bit 4 CYC_SR_INT_EN: CYC_SR (periodic counter) interrupt enable.

- 0: disabled
- 1: enabled

Bit 3 TAMPER_SR_INT_EN: TAMPER_SR interrupt enable.

- 0: disabled
- 1: enabled

Bit 2 WAKEUP0_SR_INT_EN: WAKEUP0_SR interrupt enable.

- 0: disabled
- 1: enabled

Bit 1 WAKEUP1_SR_INT_EN: WAKEUP1_SR interrupt enable.

- 0: disabled
- 1: enabled

Bit 0 WAKEUP2_SR_INT_EN: WAKEUP2_SR interrupt enable.

- 0: disabled
- 1: enabled

18.12.12 RTC_SR1

Offset: 0x02c

Reset Value: 0x00000dff

31-12	11	10	9
RESERVED	WRITE_ALARM0_SUB_DONE	WRITE_ALARM1_SUB_DONE	SECOND_SR
r-0h	r-1h	r-1h	rw-0h

8	7	6
WRITE_RTCCR2_DONE	WRITE_RTCCR_DONE	WRITE_ALARM0_DONE
r-1h	r-1h	r-1h
5	**4**	**3**
WRITE_ALARM1_DONE	WRITE_PPMADJUST_DONE	WRITE_CALENDAR_DONE
r-1h	r-1h	r-1h
2	**1**	**0**
WRITE_CYC_MAX_VALUE_DONE	WRITE_RTCSR_DONE	READ_CALENDAR_DONE
r-1h	r-1h	r-1h

Bits 31-12 RESERVED: Must be kept, and cannot be modified.

Bit 11 WRITE_ALARM0_SUB_DONE: The complete flag of the write operation to register *RTC_ALARM0_SUB*. This bit is set and cleared by hardware.

- 0: write operation in progress
- 1: write operation completed

Bit 10 WRITE_ALARM1_SUB_DONE: The complete flag of the write operation to register *RTC_ALARM1_SUB*. This bit is set and cleared by hardware.

- 0: write operation in progress
- 1: write operation completed

Bit 9 SECOND_SR: Second signal interrupt status. This bit is set by hardware and cleared by software writing 1 to it.

- 0: no second signal interrupt generated
- 1: second signal interrupt generated

Bit 8 WRITE_RTCCR2_DONE: The complete flag of the write operation to register *RTC_CR2*. Thisbit is set and cleared by hardware.

- 0: write operation in progress
- 1: write operation completed

Bit 7 WRITE_RTCCR_DONE: The complete flag of the write operation to register *RTC_CR*. This bit is set and cleared by hardware.

- 0: write operation in progress
- 1: write operation completed

Bit 6 WRITE_ALARM0_DONE: The complete flag of the write operation to register *RTC_ALARM0*. This bit is set and cleared by hardware.

- 0: write operation in progress
- 1: write operation completed

Bit 5 WRITE_ALARM1_DONE: The complete flag of the write operation to register *RTC_ALARM1*. This bit is set and cleared by hardware.

- 0: write operation in progress
- 1: write operation completed

Bit 4 WRITE_PPMADJUST_DONE: The complete flag of the write operation to register *RTC_PPMADJUST*. This bit is set and cleared by hardware.

- 0: write operation in progress
- 1: write operation completed

Bit 3 WRITE_CALENDAR_DONE: The complete flag of the write operations to registers *RTC_CALENDAR* and *RTC_CALENDAR_H*. This bit is set and cleared by hardware.

- 0: write operation in progress
- 1: write operation completed

Bit 2 WRITE_CYC_MAX_VALUE_DONE: The complete flag of the write operation to register *RTC_CYC_MAX_VALUE*. This bit is set and cleared by hardware.

- 0: write operation in progress
- 1: write operation completed

Bit 1 WRITE_RTCSR_DONE: The complete flag of the write operation to register *RTC_SR*. This bit is set and cleared by hardware.

- 0: write operation in progress
- 1: write operation completed

Bit 0 READ_CALENDAR_DONE: The complete flag of the read operations to registers *RTC_CALENDAR_R* and *RTC_CALENDAR_R_H*. This bit is set and cleared by hardware.

- 0: write operation in progress
- 1: write operation completed

18.12.13 RTC_CR2

Offset: 0x030

Reset Value: 0x00000000

31-8	7	6-4	3-0
RESERVED	RTC_OUT_POL	RTC_OUT_SEL	RTC_RET_SRAM_ERASE_EN
r-0h	rw-0h	rw-0h	rw-0h

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bit 7 RTC_OUT_POL: RTC IO output polarity.

- 0: RTC IO output level is non-inverted
- 1: RTC IO output level is inverted

Bits 6-4 RTC_OUT_SEL: RTC IO output selection.

- 0-3: no output
- 4: alarm 0 pulse
- 5: alarm 1 pulse
- 6: cyc pulse
- 7: second signal (50% duty cycle)

Bits 3-0 RTC_RET_SRAM_ERASE_EN: If enabled, the Retention SRAM is erased upon a tamper or a wakeup event. [0]: wakeup0, [1]: wakeup1, [2]: wakeup2, [3]: tamper.

- 0: disabled
- 1: enabled

18.12.14 RTC_SUB_SECOND

Offset: 0x034

Reset Value: 0x00000000

31-15	14-0
RESERVED	RTC_SUB_SECOND_VALUE
r-0h	r-0h

Bits 31-15 RESERVED: Must be kept, and cannot be modified.

Bits 14-0 RTC_SUB_SECOND_VALUE: The subsecond count value of the RTC calendar counter. This register should be read several times with the same result obtained to ensure that the data is correct.

18.12.15 RTC_CYC_CNT_VALUE

Offset: 0x038

Reset Value: 0x00000000

31-0
CYC_CNT_VALUE
r-0h

Bits 31-0 CYC_CNT_VALUE: Periodic counter value. This register should be read several times with the same result obtained to ensure that the data is correct.

18.12.16 RTC_ALARM0_SUB

Offset: 0x03c

Reset Value: 0x00000000

31-20	19-16	15	14-0
RESERVED	RTC_ALARM0_SUB_MASK	RESERVED	RTC_ALARM0_SUB_VALUE
r-0h	rw-0h	r-0h	rw-0h

Bits 31-20 RESERVED: Must be kept, and cannot be modified.

Bits 19-16 RTC_ALARM0_SUB_MASK: Alarm 0 sub-second mask configuration. If sub-seconds are used in Alarm 0, it is recommended not to perform RTC PPM calibration.

- 0: No comparison on sub-seconds for Alarm 0.
- 1: RTC_ALARM0_SUB_VALUE [14:1] are not involved in Alarm 0 comparison. Only bit0 is compared.
- 2: RTC_ALARM0_SUB_VALUE [14:2] are not involved in Alarm 0 comparison. Only bits[1:0] are compared.
- N: [N-1:0] matches of the sub-second count, and other sub-second count bits are ineffective.

Bit 15 RESERVED: Must be kept, and cannot be modified.

Bits 14-0 RTC_ALARM0_SUB_VALUE: Alarm 0 sub-seconds value. When the calendar subseconds, seconds, minutes, hours, date or day of week match the values programmed in this register and the *RTC_ALARM0* register, the ALARM0_SR bit is set.

18.12.17 RTC_ALARM1_SUB

Offset: 0x040

Reset Value: 0x00000000

31-20	19-16	15	14-0
RESERVED	RTC_ALARM1_SUB_MASK	RESERVED	RTC_ALARM1_SUB_VALUE
r-0h	rw-0h	r-0h	rw-0h

Bits 31-20 RESERVED: Must be kept, and cannot be modified.

Bits 19-16 RTC_ALARM1_SUB_MASK: Alarm 1 sub-second mask configuration. If sub-seconds are used in Alarm 1, it is recommended not to perform RTC PPM calibration.

- 0: No comparison on sub-seconds for Alarm 1.
- 1: RTC_ALARM1_SUB_VALUE [14:1] are not involved in Alarm 1 comparison. Only bit0 is compared.
- 2: RTC_ALARM1_SUB_VALUE [14:2] are not involved in Alarm 1 comparison. Only bits[1:0] are compared.
- N: [N-1:0] matches of the sub-second count, and other sub-second count bits are ineffective.

Bit 15 RESERVED: Must be kept, and cannot be modified.

Bits 14-0 RTC_ALARM1_SUB_VALUE: Alarm 1 sub-seconds value. When the calendar subseconds, seconds, minutes, hours, date or day of week match the values programmed in this register and the *RTC_ALARM1* register, the ALARM1_SR bit is set.

18.12.18 RTC_CALENDAR_R

Offset: 0x044

Reset Value: 0x00000000

31-20	19-0
RESERVED	CALENDAR_SYNC
r-0h	r-0h

Bits 31-20 RESERVED: Must be kept, and cannot be modified.。

Bits 19-0 CALENDAR_SYNC: RTC_CALENDAR_R register values (seconds, minutes and hours). This register should be read several times with the same result obtained to ensure that the data is correct.

18.12.19 RTC_CALENDAR_R_H

Offset: 0x048

Reset Value: 0x00000841

31-22	21-0
RESERVED	CALENDAR_H_SYNC
r-0h	r-841h

Bits 31-22 RESERVED: Must be kept, and cannot be modified.

Bits 21-0 CALENDAR_H_SYNC: RTC_CALENDAR_R_H register values (date or day of week, month and year). This register should be read several times with the same result obtained to ensure that the data is correct.

19. Low-power UART (LPUART)

19.1 Introduction

LPUART (Low-power Universal Asynchronous Receiver/Transmitter) is a low-power serial port peripheral. When the 32K clock is used, the LPUART communications can be up to 9600 baud/s. Even in Deepsleep mode, the LPUART can be woken up by received data.

LPUART supports CTS (Clear To Send)/RTS (Require To Send) flow control.

DMA (direct memory access) can be used for data transmission and reception.

19.2 Main Features

- Programmable baud rate
- Programmable data format (support 5, 6, 7 or 8 data bits, 1 or 2 stop bits and 1 or no parity bit)
- DMA capability
- 1 byte deep TX FIFO/RX FIFO
- CTS/RTS flow control
- Interrupt generation
- Wakeup CPU from low-power modes

19.3 Functional Description

19.3.1 Data Format

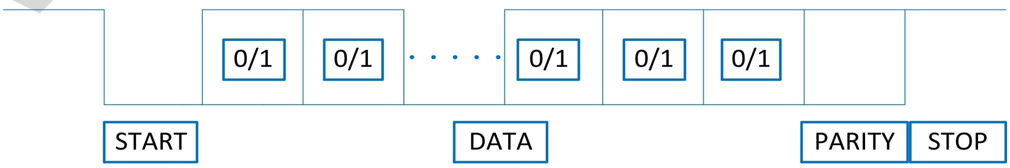

Figure 19-1 LPUART Data Format

When LPUART is idle, its data line should be kept at high level.

For data transmission, the start bit (START), data bits (DATA), parity bit (PARITY) and stop bits (STOP) are sequentially transmitted. The meaning of each bit is as follows:

(1) **Start Bit:** 0 signal is sent first to indicate the start of data transmission.

(2) **Data Bits:** 5, 6, 7 or 8 data bits are transmitted in sequence.

(3) **Parity Bit:** After the data bits, the parity bit is transmitted, or it can be configured as no parity bit.

(4) **Stop Bit:** 1 or 2 stop bits mark the end of data transmission.

19.3.2 Baud Rate Generation

The LPUART baud rate divisor consists of an integer part and a fractional part. This is mainly configured through the LPUART_BAUD_RATE_INT and LPUART_BAUD_RATE_FRA bits in the *LPUART_CR0* register.

Taking an LPUART interface clock frequency of 32.768kHz and 9600 baud/s as an example, the baud rate divisor is 32768/9600=3.413. Thus, set the integer part of the baud rate divisor to **3** through the LPUART_BAUD_RATE_INT bit, and set the fractional part of the baud rate divisor to **7** (0.413*16=6.608, rounded to 7) through the LPUART_BAUD_RATE_FRA bit.

19.3.3 CTS/RTS Flow Control

The connection between two LPUART devices is shown in the following figure:

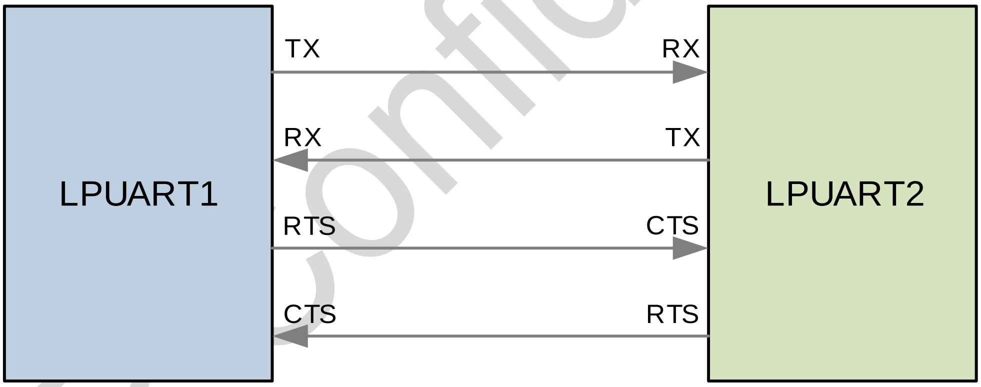

Figure 19-2 Connection between Two LPUART Devices

RTS (Require to Send) is an output signal used to determine whether the device is ready to receive data. It is active low, so the low level indicates that the device is ready for data reception.

CTS (Clear to Send) is an input signal used to determine whether the device can send data to the other. It is active low, so the low level indicates that the device can send data to the other.

19.3.4 DMA

LPUART DMA Transmit Process:

(1) Enable the DMA_TX_EN bit in register *LPUART_CR1*.

(2) Configure register *LPUART_DATA* as the destination address of DMA.

(3) Configure the memory address of the data to be sent as the source address of DMA.

(4) Configure the data width of DMA transfer to 8 bits by configuring the SRC_TR_WIDTH and DES_TR_WIDTH bits to 0 in the *DMA_CTLx* register.

(5) Configure the DMA burst length to 1 by configuring the SRC_MSIZE and DEST_MSIZE bits to 0 in the *DMA_CTLx* register.

(6) Configure the total length of DMA data transfer.

(7) Configure DMA handshake type to DMA_HANDSHAKE_LPUART_TX.

(8) Activate the DMA.

When the DMA transfer is completed, the CH_EN_x bit in the DMA_CHENREG register is cleared.

LPUART DMA Receive Process:

(1) Enable the DMA_RX_EN bit in register *LPUART_CR1*.

(2) Configure register *LPUART_DATA* as the source address of DMA.

(3) Configure the memory address of the data to be received as the destination address of DMA.

(4) Configure the data width of DMA transfer to 8 bits by configuring the SRC_TR_WIDTH and DES_TR_WIDTH bits to 0 in the *DMA_CTLx* register.

(5) Configure the DMA burst length to 1 by configuring the SRC_MSIZE and DEST_MSIZE bits to 0 in the *DMA_CTLx* register.

(6) Configure the total length of DMA data transfer.

(7) Configure DMA handshake type to DMA_HANDSHAKE_LPUART_RX.

(8) Activate the DMA.

When the DMA transfer is completed, the CH_EN_x bit in the DMA_CHENREG register is cleared.

19.3.5 Interrupts

LPUART interrupt signals:

- TX_DONE interrupt
- TXFIFO_EMPTY interrupt
- RXFIFO_NOT_EMPTY interrupt
- RX_OVERFLOW interrupt
- STOP_ERR interrupt
- PARITY_ERR interrupt
- START_INVALID interrupt
- RX_DONE interrupt
- START_VALID interrupt

19.3.6 CPU Wakeup from Low-power Modes

RX low-level, START_VALID and RX_DONE signals can be used to wakeup the CPU from low power modes. LPUART wakeup is enabled by configuring the LPUART_WAKEUP_EN[[24:22] bits in register *LPUART_CR0*.

19.4 LPUART Registers

Base Address:0x40005000

Table 19-1 LPUART Registers Summary

Register	Offset	Description
LPUART_CR0	0x00	Control Register 0
LPUART_CR1	0x04	Control Register 1
LPUART_SR0	0x08	Status Register 0
LPUART_SR1	0x0C	Status Register 1
LPUART_DATA	0x10	Data Register

19.4.1 LPUART_CR0

Offset: 0x00

Reset Value: 0x00000E13

31-27	26	25	24-22	21-10
RESERVED	LPUART_RTS_EN	LPUART_RX_EN	LPUART_WAKEUP_EN	LPUART_BAUD_RATE_INT
r	r/w	r/w	r/w	r/w
9-6	**5**	**4-2**	**1-0**	
LPUART_BAUD_RATE_FRA	LPUART_STOP_LEN	LPUART_PARITY_CFG	LPUART_DATA_LEN	
r/w	r/w	r/w	r/w	

Bits 31-27 RESERVED: Must be kept, and cannot be modified.

Bit 26 LPUART_RTS_EN: LPUART RTS flow control enable.

- 0: disabled
- 1: enabled

Bit 25 LPUART_RX_EN: LPUART reception enable.

- 0: disabled
- 1: enabled

Bits 24-22 LPUART_WAKEUP_EN: LPUART wakeup enable.

[22] Enable RX low-level signal as a wakeup source

- 0: disabled
- 1: enabled

[23] Enable START_VALID signal as a wakeup source

- 0: disabled
- 1: enabled

[24] Enable RX_DONE signal as a wakeup source

- 0: disabled
- 1: enabled

Bits 21-10 LPUART_BAUD_RATE_INT: The integer part of the baud rate divisor.

The frequency division factor is equal to the UART interface clock frequency/baud rate.

Take the UART interface clock frequency as 32.768KHz and the baud rate as 9600 as an example, the frequency division factor is 32768/9600=3.413, lpuart_baud_rate_int is configured as 3, and lpuart_baud_rate_fra is configured as 0.413*16=6 or 7.

Bits 9-6 LPUART_BAUD_RATE_FRA: The fractional part of the baud rate divisor.

Bit 5 LPUART_STOP_LEN: LPUART STOP bits configuration.

- 0: 1 stop bit
- 1: 2 stop bits

Bits 4-2 LPUART_PARITY_CFG: LPUART parity bit configuration.

- 0: even parity
- 1: odd parity
- 2: parity bit is 0
- 3: parity bit is 1
- >3: no parity

Bits 1-0 LPUART_DATA_LEN: LPUART data length.

Data width=LPUART_DATA_LEN+5

19.4.2 LPUART_CR1

Offset: 0x004

Reset Value: 0x00000000

31-13	12	11	10	9
RESERVED	LPUART_CTS_EN	DMA_TX_EN	DMA_RX_EN	LPUART_TX_EN
r	r/w	r/w	r/w	r/w
8	**7**	**6**	**5**	**4**
TX_DONE_INT_EN	TXFIFO_EMPTY_INT_EN	RXFIFO_NOT_EMPTY_INT_EN	RX_OVERFLOW_INT_EN	STOP_ERR_INT_EN
r/w	r/w	r/w	r/w	r/w
3	**2**	**1**	**0**	
PARITY_ERR_INT_EN	START_INVALID_INT_EN	RX_DONE_INT_EN	START_VALID_INT_EN	
r/w	r/w	r/w	r/w	

Bits 31:13 RESERVED: Must be kept, and cannot be modified.

Bit 12 LPUART_CTS_EN: LPUART CTS flow control enable.

- 0: disabled
- 1: enabled

Bit 11 DMA_TX_EN: DMA transmission requests enable.

- 0: disabled
- 1: enabled

Bit 10 DMA_RX_EN: DMA reception requests enable.

- 0: disabled
- 1: enabled

Bit 9 LPUART_TX_EN: LPUART transmission enable.

- 0: disabled
- 1: enabled

Bit 8 TX_DONE_INT_EN: TX_DONE interrupt enable.

- 0: disabled
- 1: enabled

Bit 7 TXFIFO_EMPTY_INT_EN: TXFIFO_EMPTY interrupt enable.

- 0: disabled
- 1: enabled

Bit 6 RXFIFO_NOT_EMPTY_INT_EN: RXFIFO_NOT_EMPTY interrupt enable.

- 0: disabled
- 1: enabled

Bit 5 RX_OVERFLOW_INT_EN: RX_OVERFLOW interrupt enable.

- 0: disabled
- 1: enabled

Bit 4 STOP_ERR_INT_EN: STOP_ERR interrupt enable.

- 0: disabled
- 1: enabled

Bit 3 PARITY_ERR_INT_EN: PARITY_ERR interrupt enable.

- 0: disabled
- 1: enabled

Bit 2 START_INVALID_INT_EN: START_INVALID interrupt enable.

- 0: disabled
- 1: enabled

Bit 1 RX_DONE_INT_EN: RX_DONE interrupt enable.

- 0: disabled
- 1: enabled

Bit 0 START_VALID_INT_EN: START_VALID interrupt enable.

- 0: disabled
- 1: enabled

19.4.3 LPUART_SR0

Offset: 0x008

Reset Value: 0x00000000

31-6		5	4
RESERVED		RX_OVERFLOW_SR	STOP_ERR_SR
r		r/w	r/w
3	**2**	**1**	**0**
PARITY_ERR_SR	START_INVALID_SR	RX_DONE_SR	START_VALID_SR
r/w	r/w	r/w	r/w

Bits 31-6 RESERVED: Must be kept, and cannot be modified.

Bit 5 RX_OVERFLOW_SR: RX_OVERFLOW flag is used to indicate whether a RX buffer overflow has occurred. This bit is set by hardware and cleared by software writing 1 to it.

- 0: no RX buffer overflow

- 1: RX buffer overflow occurred

Bit 4 STOP_ERR_SR: STOP_ERR flag is used to indicate whether a Stop error has occurred. This bit is set by hardware and cleared by software writing 1 to it.

- 0: no Stop error
- 1: Stop error occurred

Bit 3 PARITY_ERR_SR: PARITY_ERR flag is used to indicate whether a parity error has occurred. This bit is set by hardware and cleared by software writing 1 to it.

- 0: no parity error
- 1: parity error occurred

Bit 2 START_INVALID_SR: START_INVALID flag is used to indicate whether an invalid Start bit has been received. This bit is set by hardware and cleared by software writing 1 to it.

- 0: no invalid Start
- 1: invalid Start bit has been received

Bit 1 RX_DONE_SR: RX_DONE flag is used to indicate whether the data reception is completed. This bit is set by hardware and cleared by software writing 1 to it.

- 0: data reception not completed
- 1: data reception completed

Bit 0 START_VALID_SR: START_VALID flag is used to indicate whether a valid Start bit has been received. This bit is set by hardware and cleared by software writing 1 to it.

- 0: no valid Start
- 1: valid Start bit has been received

19.4.4 LPUART_SR1

Offset: 0x00C

Reset Value: 0x00000016

31-6		5	4
RESERVED		TX_DONE	TXFIFO_EMPTY
r		r/w	r
3	**2**	**1**	**0**
RXFIFO_NOT_EMPTY	WRITE_CR0_DONE	WRITE_SR0_DONE	RESERVED
r	r	r	r

Bits 31-6 RESERVED: Must be kept, and cannot be modified.

Bit 5 TX_DONE: TX_DONE flag. This bit is set by hardware and cleared by software writing 1 to it.

- 0: data transmission in progress
- 1: data transmission completed

Bit 4 TXFIFO_EMPTY: TXFIFO_EMPTY flag. This bit is set by hardware and cleared by software writing to the *LPUART_DATA* register.

- 0: non-empty
- 1: empty

Bit 3 RXFIFO_NOT_EMPTY: RXFIFO_NOT_EMPTY flag. This bit is set by hardware and cleared by software reading the *LPUART_DATA* register.

- 0: empty
- 1: non-empty

Bit 2 WRITE_CR0_DONE: The status of a write operation to the *LPUART_CR0* register. This bit is set and cleared by hardware.

- 0: write operation to the *LPUART_CR0* register in progress
- 1: write operation to the *LPUART_CR0* register completed

Bit 1 WRITE_SR0_DONE: The status of a write operation to the *LPUART_SR0* register. This bit is set and cleared by hardware.

- 0: write operation to the *LPUART_SR0* register in progress
- 1: write operation to the *LPUART_SR0* register completed

Bit 0 RESERVED: Must be kept, and cannot be modified.

19.4.5 LPUART_DATA

Offset: 0x010

Reset Value: 0x00000000

31-8	7-0
RESERVED	LPUART_DATA
r	r/w

Bits 31-8 RESERVED: Must be kept, and cannot be modified.

Bits 7-0 LPUART_DATA: LPUART TX/RX data.

Notes:

1. *If the data width is less than 8 bits, the less significant bits of the LPUART_DATA register is valid.*
2. *Before reading the LPUART_DATA register, check the RXFIFO_NOT_EMPTY flag to ensure that there is data in RXFIFO; before writing to the LPUART_DATA register, check the TXFIFO_EMPTY flag to ensure that the TXFIFO can be written.*

20. Low-power timer (LPTIM)

20.1 Introduction

LPTIMER（Low Power Timer）是一个 16 位的计时器，由于其有多种时钟源，LPTIMER 能够在除 standby 模式外的所有工作模式下运行，并且支持从所有低功耗工作模式中唤醒。有两个 LPTIMER，分别为 LPTIMER0 和 LPTIMER1。

20.2 Main features

LPTIMER 包括如下功能：

- 支持选择内部时钟与外部时钟作为计数时钟
- 16bits 计数器，加法计数，支持自动加载
- 支持两种计数模式，单次计数和连续计数
- 支持软件触发和外部触发源触发计数
- 分频计数器
- 支持产生 PWM
- 支持单脉冲、Set-once、Timeout 模式输出
- 支持 DEBUG 模式控制
- 支持产生通道输出事件、匹配事件、溢出事件、触发事件、DOWN 事件、UP 事件作为唤醒信号输出
- 正交解码
- 中断信号产生

LPTIMER 的框图如下:

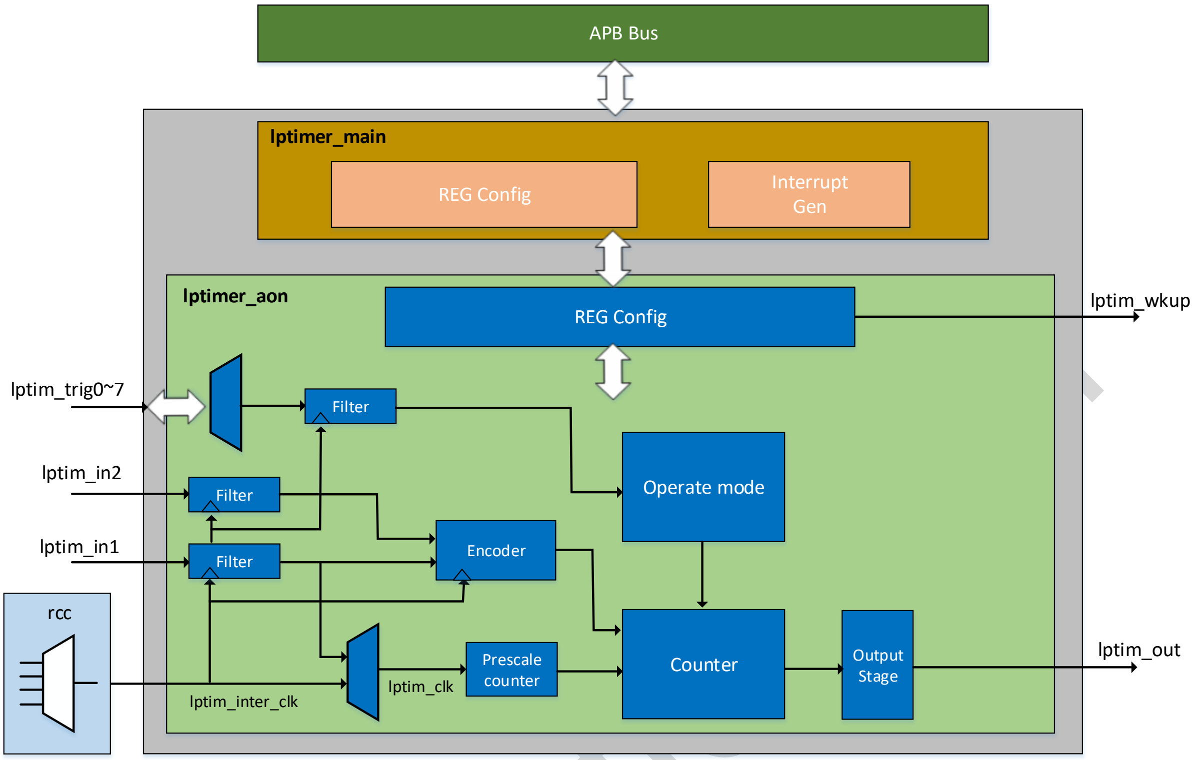

Figure 20-1 LPTIMER Diagram

- **lptim_trig0~7**：LPTIMER 的外部触发源
- **lptim_in2**：LPTIMER 的 IN2 pin 脚
- **lptim_in1**：LPTIMER 的 IN1 pin 脚
- **lptim_wkup**：LPTIMER 的唤醒信号
- **lptim_out**：LPTIMER 的 OUT pin 脚

20.3 Interface clock

LPTIMER 接口时钟源分内部时钟和外部时钟，内部时钟源有 PCLK0、RCO3.6M、XO32K、RCO32K，外部时钟源通过 IN1 的 GPIO 输入。时钟配置和选择可以参考 RCC 章节。

20.4 Counter clock selection

LPTIMER 除了接口时钟有内部和外部之分外，计数时钟也有内部和外部之分，其内部与外部的时钟源与接口时钟的一致。控制计数时钟选择的寄存器 bit 位为寄存器 LPTIM_CFGR 的 COUNTMODE，值为 0 表示计数器由内部时钟控制，值为 1 表示计数器由外部时钟控制，如果 RCC 模块的寄存器 RCC_CR1 的 LPTIM1_EXT_CLK_SEL 位或 LPTIM_EXT_CLK_SEL 位

为 0，即表示 LPTIMER0 或 LPTIMER1 的接口时钟为内部时钟，则 COUNTMODE 的值可以为 0 或 1，即计数时钟既可以为内部时钟也可以为外部时钟；如果 LPTIM1_EXT_CLK_SEL 位或 LPTIM_EXT_CLK_SEL 位为 1，则 LPTIMER0 或 LPTIMER1 的寄存器 LPTIM_CFGR 的 COUNTMODE 位只能设置为 0，这时 0 不是表示计数时钟为内部时钟，是表示 COUNTMODE 值需要清零，计数时钟只能为外部时钟。

20.5 Counter

除编码模式外，计数器仅支持向上计数，计数到 ARR 时产生 ARRM 中断，计数器回到 0 重新计数。若使能 timeout 模式，则除了计数器值增加到 ARR 时清零外，触发信号也可以清零计数器重新计数。若使能编码模式，则计数器的计数方向由硬件控制，向上计数到 ARR 时产生 ARRM 事件并清零计数器，向下计数到 0 时则重新加载 ARR 到计数器。

20.6 Counting modes

LPTIMER 支持两种计数模式，单次计数和连续计数。单次计数模式下，计数器停止阶段第一个到来的触发信号（硬件或软件）会触发计数器开始计数，计数过程中的触发信号将会被忽略，计数到 ARR 时计数器会停止计数，直到下一次触发信号到来才会再次开始计数，依次类推。连续计数模式下，一旦触发（硬件或软件），计数器会一直计数下去，从 0 到 ARR，然后回到 0 再次计数，如此循环往复。

两种计数模式可以在任意时刻切换（前提是 enable 置位），例如，配置 LPTIMER 为单次计数模式，若置位寄存器 LPTIM_CR 的 CNTSTRT，则计数器计数到 ARR 值将不会停止计数；配置 LPTIMER 为连续计数模式，若置位寄存器 LPTIM_CR 的 SNGSTRT，则计数器计数到 ARR 时将会停止计数，直到下一次触发信号到来。因此状态图如下：

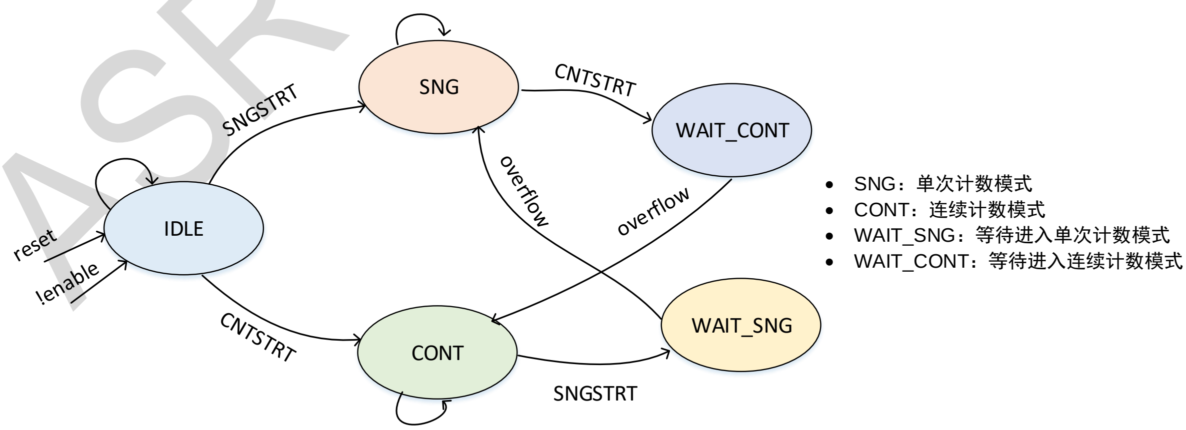

Figure 20-2 Counting mode conversion Diagram

20.7 Trigger sources

触发 LPTIMER 计数有两种方式，一种是软件触发，另一种是外部触发源触发。通过寄存器 LPTIM_CFGR 的 TRIGEN 位段进行控制，当值为 0 时为软件触发，非零时为外部触发，当为外部触发时，可以设置外部触发信号上升沿有效、下降沿有效或双沿有效。LPTIMER 有 8 种触发输入源可以选择使用，**LPTIMER0** 的外部触发源如下表：

Table 20-1 LPTIMER0 external trigger source

TRIGSEL	External Trigger	Comment
lptim_ext_trig0	lptim_etr	Lptimer etr pin input
lptim_ext_trig1	comp0	Comp0 output
lptim_ext_trig2	comp1	Comp1 output
lptim_ext_trig3	rtc_cyc_counter	RTC cyc counter output pulse
lptim_ext_trig4	rtc_alarm0	RTC alarm0 output pulse
lptim_ext_trig5	rtc_alarm1	RTC alarm1 output pulse
lptim_ext_trig6	gpio	GPIO58
lptim_ext_trig7	gpio	GPIO59

LPTIMER1 的外部触发源如下表：

Table 20-2 LPTIMER1 external trigger source

TRIGSEL	External Trigger	Comment
lptim_ext_trig0	lptim_etr	Lptimer etr pin input
lptim_ext_trig1	comp0	Comp0 output
lptim_ext_trig2	comp1	Comp1 output
lptim_ext_trig3	rtc_cyc_counter	RTC cyc counter output pulse
lptim_ext_trig4	rtc_alarm0	RTC alarm0 output pulse
lptim_ext_trig5	rtc_alarm1	RTC alarm1 output pulse
lptim_ext_trig6	gpio	GPIO60
lptim_ext_trig7	gpio	GPIO61

20.8 Prescaler

计数使能信号可以被软件配置分频，支持 1、2、4、8、16、32、64、128 分频，通过配置寄存器 LPTIM_CFGR 的 PRESC 位段进行分频配置。该分频通过计数器实现，即上一级电路产生的计数使能信号将作为该分频计数器的计数使能，当分频计数器计数到预先加载的分频值后，输出一个脉冲，作为下一级计数器的计数使能，然后分频计数器归零重新计数，依次类推。

20.9 PWM

LPTIMER 可以产生 PWM 波形，波形的极性可以通过寄存器 LPTIM_CFGR 的 WAVPOL 比特控制，占空比可以通过寄存器 LPTIM_CMP 和 LPTIM_ARR 的值进行控制。以软件触发和内部时钟计数为例，配置 PWM 的流程如下：

(1) 配置寄存器 LPTIM_CFGR 的 COUNTMODE 为 0，即设置内部时钟计数。

(2) 寄存器 LPTIM_CFGR 的 PRESC 为默认值，即不设置计数器分频。

(3) 配置寄存器 LPTIM_CFGR 的 PRELOAD 的值为 0，即不使能寄存器 LPTIM_CMP 和 LPTIM_ARR 的缓存功能。如果需要也可以使能。

(4) 配置寄存器 LPTIM_CFGR 的 WAVPOL 为 0，即波形输出不反相。

(5) 配置寄存器 LPTIM_CFGR 的 WAVE 为 0。

(6) 寄存器 LPTIM_CFGR 的 TRIGEN 位段的值为 0，即软件触发。

(7) 使能 LPTIMER，就是置位寄存器 LPTIM_CR 的 ENABLE。

(8) 设置寄存器 LPTIM_ARR 和 LPTIM_CMP 的值。

(9) 使能连续计数功能，通过置位寄存器 LPTIM_CR 的 CNTSTRT 实现。

20.10 Single-pulse, Set-once, Timeout output mode

单脉冲模式下，计数器未计数时，检测到第一个触发信号，则计数使能置位，若计数到 ARR 或 enable 清零或模块复位，则计数使能清零，计数过程中的触发信号将会被忽略，如下图：

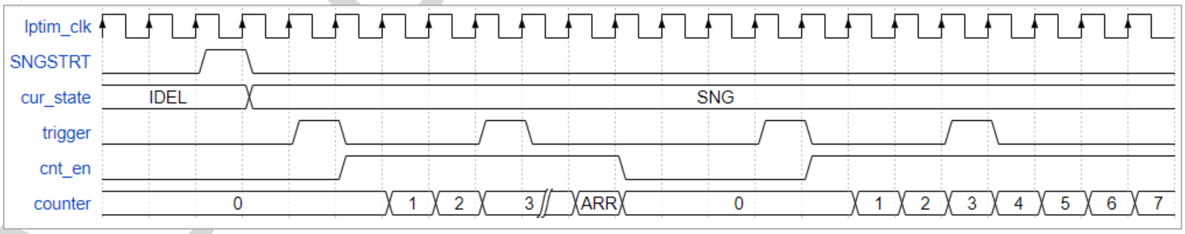

Figure 20-3 Single pulse counting

单脉冲模式通过配置寄存器 LPTIM_CFGR 的 WAVE 为 0 以及寄存器 LPTIM_CR 的 SNGSTRT 为 1 实现。

Set-once 模式下，检测到第一个触发信号后，计数使能置位，若计数到 ARR，则计数使能清零，计数过程中的触发信号将会被屏蔽，屏蔽信号通过 mask 实现，即检测到第一个触发信号后，mask 有效，屏蔽之后的所有触发信号，如下图：

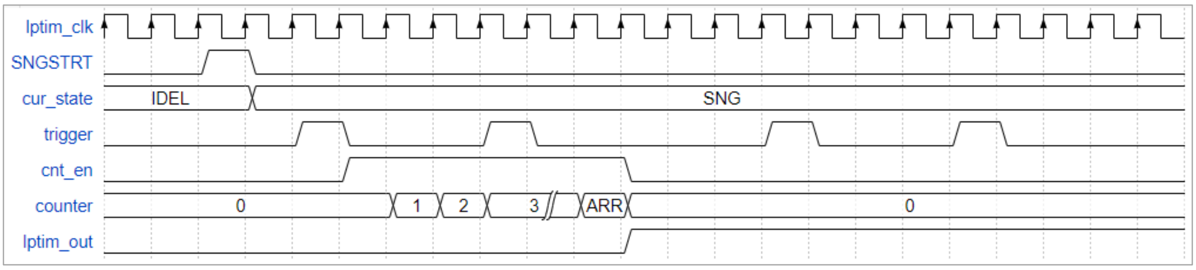

Figure 20-4 Set-once counting

Set-once 模式通过配置寄存器 LPTIM_CFGR 的 WAVE 为 1 以及寄存器 LPTIM_CR 的 SNGSTRT 为 1 实现。

Timeout 模式与连续计数模式类似，一旦被触发，计数使能一直有效，区别是，计数过程中的触发信号会让计数器从 0 开始重新计数，且输出波形也会被清除，如下图：

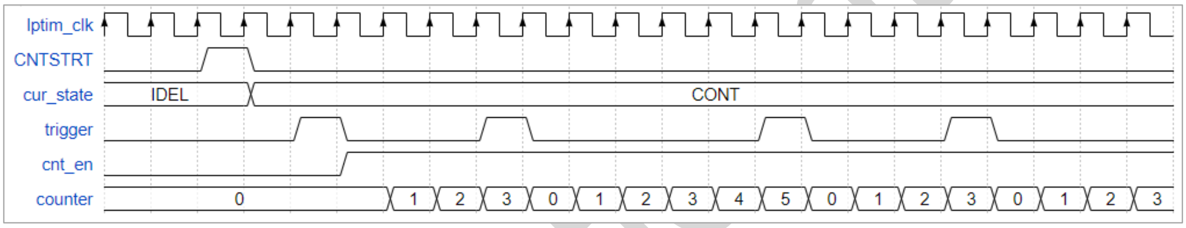

Figure 20-5 Timeout counting

Timeout 模式通过配置寄存器 LPTIM_CFGR 的 WAVE 为 0 以及寄存器 LPTIM_CR 的 CNTSTRT 为 1 实现。

20.11 Quadrature encoder mode

LPTIMER 支持正交编码计数功能，可以通过 IN1 和 IN2 输入正交信号，进行计数和方向检测。编码模式共有三种，仅在上升沿计数、下降沿计数以及双边沿计数，编码模式的使能通过寄存器 LPTIM_CFGR 的 ENC 控制，编码模式的边沿控制通过寄存器 LPTIM_CFGR 的 CKPOL 来实现。在此功能下，两个通道输入可以配置数字滤波功能，滤波使能通过寄存器 LPTIM_CFGR 的 CKFLT_ENABLE 控制，滤波值通过寄存器 LPTIM_CFGR 的 CKFLT 进行配置。通过两个通道信号的组合，可以产生计数使能和方向控制信号，控制计数器加减。具体的组合方式见下表：

Table 20-3 Quadtature encoder channel signals

编码模式	IN1/IN2 电平	IN1		IN2	
		上升沿	下降沿	上升沿	下降沿
上升沿计数	高电平	向下计数	-	向上计数	-
	低电平	向上计数	-	向下计数	-
下降沿计数	高电平	-	向上计数	-	向下计数
	低电平	-	向下计数	-	向上计数
双沿计数	高电平	向下计数	向上计数	向上计数	向下计数
	低电平	向上计数	向下计数	向下计数	向上计数

IN1 和 IN2 输入信号频率必须小于 LPTIMER 时钟频率的 1/4。

20.12 DEBUG mode control

LPTIMER 可由软件配置 debug 下是否停止计数，通过 SYSCFG 的 CR2 寄存器来实现 LPTIMER0 和 LPTIMER1 的 DEBUG 模式计数控制，如果使能该功能，则进入系统 debug 模式时，LPTIMER 停止计数（计数器不会被初始化）。

20.13 Wake-up signals

LPTIMER 有 6 种唤醒信号输出，分别是，

- **通道输出信号**，此时通道输出将作为唤醒信号输出。
- **匹配事件（CMPM）**，此时计数器与寄存器 LPTIM_CMP 的匹配事件将作为唤醒信号输出。
- **溢出事件（ARRM）**，此时 overflow 事件将作为唤醒信号输出。
- **触发事件（EXTTRIG）**，此时有效的触发事件将作为唤醒信号输出。
- **DOWN 事件**，若计数方向由向上计数变为向下计数，DOWN 事件会置位，此时 DOWN 事件会作为唤醒信号输出。
- **UP 事件**，若计数方向由向下计数变为向上计数，UP 事件会置位，此时 UP 事件会作为唤醒信号输出。

以上唤醒信号除了通道输出信号，均为 LPTIM_ISR 寄存器的标志位，且有独立的使能位，使能位分别为寄存器 LPTIM_CFGR 的 *OUT_WKUP_EN*、*CMPM_WKUP_EN*、*ARRM_WKUP_EN*、*EXTTRIG_WKUP_EN*、*DOWN_WKUP_EN*、*UP_WKUP_EN* 比特位，唤醒信号与相应使能位是 AND 的关系，各唤醒源之间是 OR 的关系。

20.14 Interrupts

LPTIMER 的中断信号如下:

Table 20-4 LPTIMER interrupts

中断名称	描述
DOWN 中断	编码模式下，表示计数方向由向上变为向下
UP 中断	编码模式下，表示计数方向由向下变为向上
ARROK 中断	表示 ARR 值加载完成
CMPOK 中断	表示 CMP 值加载完成
EXTTRIG 中断	表示检测到有效触发边沿
ARRM 中断	表示计数器值到达 ARR
CMPM 中断	表示计数器值与 CMP 匹配

上述中断的使能通过配置寄存器 LPTIM_IER 实现，所有中断的中断状态都可以通过寄存器 LPTIM_SR1 获得。

20.15 LPTIMER registers

LPTIMER0 基地址：0x4000D000

LPTIMER1 基地址：0x4000D800

Table 20-5 LPTIMER Registers Summary

寄存器	偏移量	描述
LPTIM_ISR	0x00	状态寄存器
LPTIM_ICR	0x04	状态清除寄存器
LPTIM_IER	0x08	中断使能寄存器
LPTIM_CFGR	0x0c	配置寄存器，该寄存器需在 LPTIM_CR 寄存器的 ENABLE 清零时修改
LPTIM_CR	0x10	控制寄存器
LPTIM_CMP	0x14	比较寄存器
LPTIM_ARR	0x18	重装载寄存器
LPTIM_CNT	0x1c	计数器寄存器
LPTIM_CSR	0x20	清除状态标志寄存器，表示使用寄存器 LPTIM_ICR 清除 LPTIM_ISR 某些状态位时的是否清除完成标志
LPTIM_SR1	0x24	中断标志寄存器，中断标志位会被寄存器 LPTIM_ICR 立即清零

20.15.1 LPTIM_ISR

偏移量：0x00

复位值：0x00000180

31-9	8	7	6	5	4	3	2	1	0
RESERVED	CROK	CFGROK	DOWN	UP	ARROK	CMPOK	EXTTRIG	ARRM	CMPM
r-0h	r-1h	r-1h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h

位 31-9 RESERVED：保留。

位 8 CROK：上一次对寄存器 LPTIM_CR 的写操作状态。该位由硬件控制，写操作前需要检查上一次写操作是否完成。

- 0：正在进行写操作
- 1：上一次对 LPTIM_CR 的写操作已经完成

位 7 CFGROK：上一次对 LPTIM_CFGR 的写操作状态。该位由硬件控制，写操作前需要检查上一次写操作是否完成。

- 0：正在进行写操作
- 1：上一次对 LPTIM_CFGR 的写操作已经完成

位 6 DOWN：编码模式下计数方向由向上变为向下。

- 0：计数方向未发生由上向下的变化
- 1：计数方向由向上变为向下

可以通过写 LPTIM_ICR 寄存器清零，但是需要时间同步清零脉冲，因此无法立即清除。

位 5 UP：编码模式下计数方向由向下变为向上。

- 0：计数方向未发生由下向上的变化
- 1：计数方向由向下变为向上

可以通过写 LPTIM_ICR 寄存器清零，但是需要时间同步清零脉冲，因此无法立即清除。

位 4 ARROK：ARR 值加载状态。

- 0：未加载完成
- 1：加载完成

可以通过写 LPTIM_ICR 寄存器清零，但是需要时间同步清零脉冲，因此无法立即清除。

位 3 CMPOK：CMP 值加载状态。

- 0：未加载完成
- 1：加载完成

可以通过写 LPTIM_ICR 寄存器清零，但是需要时间同步清零脉冲，因此无法立即清除。

位 2 EXTTRIG：是否检测到有效触发边沿。

- 0：未检测到有效触发边沿
- 1：检测到有效触发边沿

可以通过写 LPTIM_ICR 寄存器清零，但是需要时间同步清零脉冲，因此无法立即清除。

位 1 ARRM：计数器值是否到达 ARR 值。

- 0：计数器值未到达 ARR
- 1：计数器值到达 ARR

可以通过写 LPTIM_ICR 寄存器清零，但是需要时间同步清零脉冲，因此无法立即清除。

位 0 CMPM：计数器值与 CMP 值匹配状态。

- 0：计数器值与 CMP 值未匹配
- 1：计数器值与 CMP 值匹配

可以通过写 LPTIM_ICR 寄存器清零，但是需要时间同步清零脉冲，因此无法立即清除。

20.15.2 LPTIM_ICR

偏移量：0x04

复位值：0x00000000

31-7	6	5	4	3	2	1	0
RESERVED	DOWNCF	UPCF	ARROKCF	CMPOKCF	EXTTRIGCF	ARRMCF	CMPMCF
w-0h	w-0h	w-0h	w-0h	w-0h	w-0h	w-0h	w-0h

位 31-7 RESERVED：保留。

位 6 DOWNCF：清除 DOWN 标志位。软件写 1 清除标记位，该位由硬件清零。

- 0：无操作
- 1：清除操作

位 5 UPCF：清除 UP 标志位。软件写 1 清除标记位，该位由硬件清零。

- 0：无操作
- 1：清除操作

位 4 ARROKCF：清除 ARROK 标志位。软件写 1 清除标记位，该位由硬件清零。

- 0：无操作
- 1：清除操作

位 3 CMPOKCF：清除 CMPOK 标志位。软件写 1 清除标记位，该位由硬件清零。

- 0：无操作
- 1：清除操作

位 2 EXTTRIGCF：清除 EXTTRIG 标志位。软件写 1 清除标记位，该位由硬件清零。

- 0：无操作
- 1：清除操作

位 1 ARRMCF：清除 ARRM 标志位。软件写 1 清除标记位，该位由硬件清零。

- 0：无操作
- 1：清除操作

位 0 CMPMCF：清除 CMPM 标志位。软件写 1 清除标记位，该位由硬件清零。

- 0：无操作
- 1：清除操作

20.15.3 LPTIM_IER

偏移量：0x08

复位值：0x00000000

31-7	6	5	4	3	2	1	0
RESERVED	DOWNIE	UPIE	ARROKIE	CMPOKIE	EXTTRIGIE	ARRMIE	CMPMIE
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 31-7 RESERVED：保留。

位 6 DOWNIE：DOWN 中断使能。

- 0：禁用中断
- 1：使能中断

位 5 UPIE：UP 中断使能。

- 0：禁用中断
- 1：使能中断

位 4 ARROKIE：ARROK 中断使能。

- 0：禁用中断
- 1：使能中断

位 3 CMPOKIE：CMPOK 中断使能。

- 0：禁用中断
- 1：使能中断

位 2 EXTTRIGIE：EXTTRIG 中断使能。

- 0：禁用中断
- 1：使能中断

位 1 ARRMIE：ARRM 中断使能。

- 0：禁用中断
- 1：使能中断

位 0 CMPMIE：CMPM 中断使能。

- 0：禁用中断
- 1：使能中断

20.15.4 LPTIM_CFGR

偏移量：0x0c

复位值：0x00000000

31	30	29	28	27	26
RESERVED	OUT_WKUP_EN	DOWN_WKUP_EN	UP_WKUP_EN	EXTTRIG_WKUP_EN	ARRM_WKUP_EN
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h
25	**24**	**23**	**22**	**21**	**20**
CMPM_WKUP_EN	ENC	COUNTMODE	PRELOAD	WAVPOL	WAVE
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h
19	**18-17**	**16**	**15-13**	**12**	**11-9**
TIMEOUT	TRIGEN	RESERVED	TRIGSEL	RESERVED	PRESC
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h
8	**7-6**	**5**	**4-3**	**2-1**	**0**
TRGLT_ENABLE	TRGFLT	CKFLT_ENABLE	CKFLT	CKPOL	RESERVED
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 31 RESERVED：保留。

位 30 OUT_WKUP_EN：LPTIM_OUT 唤醒使能。

- 0：LPTIM_OUT 不能触发唤醒信号
- 1：LPTIM_OUT 可以触发唤醒信号

位 29 DOWN_WKUP_EN：DOWN 事件唤醒使能。

- 0：DOWN 事件不能触发唤醒信号
- 1：DOWN 事件可以触发唤醒信号

位 28 UP_WKUP_EN：UP 事件唤醒使能。

- 0：UP 事件不能触发唤醒信号
- 1：UP 事件可以触发唤醒信号

位 27 EXTTRIG_WKUP_EN：外部触发事件唤醒使能。

- 0：外部触发事件不能触发唤醒信号
- 1：外部触发事件可以触发唤醒信号

位 26 ARRM_WKUP_EN：计数溢出事件唤醒使能（ENC 模式除外）。

- 0：计数溢出不能触发唤醒信号
- 1：计数溢出事件触发唤醒信号

位 25 CMPM_WKUP_EN：计数匹配事件唤醒使能。

- 0：计数匹配不能触发唤醒信号
- 1：计数匹配事件触发唤醒信号

位 24 ENC：编码模式使能。

- 0：禁用编码模式

- 1：使能编码模式

位 23 COUNTMODE：计数模式选择。

- 0：计数器由内部时钟控制
- 1：计数器由外部时钟控制

位 22 PRELOAD：寄存器缓存使能。

- 0：ARR 和 CMP 直接由软件操作
- 1：ARR 和 CMP 由更新事件更新

位 21 WAVPOL：输出波形极性。

- 0：输出不反相
- 1：输出反相

位 20 WAVE：波形形状。

- 0：禁用 Set-once，选择 PWM 或单脉冲模式
- 1：使能 Set-once 模式

位 19 TIMEOUT：Timeout 模式使能。

- 0：禁用 Timeout 模式
- 1：使能 Timeout 模式

位 18-17 TRIGEN：外部触发使能及极性选择。

- 00：软件触发
- 01：外部触发上升沿有效
- 10：外部触发下降沿有效
- 11：外部触发双沿有效

位 16 RESERVED：保留。

位 15-13 TRIGSEL：外部触发源选择。

- 000：lptim_ext_trig0
- 001：lptim_ext_trig1
- 010：lptim_ext_trig2
- 011：lptim_ext_trig3
- 100：lptim_ext_trig4
- 101：lptim_ext_trig5
- 110：lptim_ext_trig6
- 111：lptim_ext_trig7

位 12 RESERVED：保留。

位 11-9 PRESC：时钟分频。

- 000：/1
- 001：/2
- 010：/4
- 011：/8
- 100：/16
- 101：/32
- 110：/64
- 111：/128

位 8 TRGLT_ENABLE：触发输入滤波器使能，必须先配置滤波器长度，再使能。

- 0：禁用触发输入滤波器
- 1：使能触发输入滤波器

位 7-6 TRGFLT：触发输入滤波器配置。

- 00：无操作
- 01：使能滤波器，滤波器长度 N=2
- 10：使能滤波器，滤波器长度 N=4
- 11：使能滤波器，滤波器长度 N=8

位 5 CKFLT_ENABLE：外部时钟滤波器使能，必须先配置滤波器长度，再使能。

- 0：禁用外部时钟滤波器
- 1：使能外部时钟滤波器

位 4-3 CKFLT：外部时钟滤波器配置。

- 00：无操作
- 01：使能滤波器，滤波器长度 N=2
- 10：使能滤波器，滤波器长度 N=4
- 11：使能滤波器，滤波器长度 N=8

位 2-1 CKPOL：Encoder 模式控制。

- 00：选择 Encoder 模式 1，上升沿计数
- 01：选择 Encoder 模式 2，下降沿计数
- 10：选择 Encoder 模式 3，双沿计数
- 11：保留

位 0 RESERVED：保留。

20.15.5 LPTIM_CR

偏移量：0x10

复位值：0x00000000

31-3	2	1	0
RESERVED	CNTSTRT	SNGSTRT	ENABLE
rw-0h	rw-0h	rw-0h	rw-0h

位 31-7 RESERVED：保留。

位 2 CNTSTRT：连续计数模式使能。

- 0：不使能
- 1：使能连续计数模式，写 1 开始连续计数模式，若在连续计数模式过程中置位 SNGSTRT，则在下一次计数到 ARR 时停止计数（切换到单次计数模式）。该比特位需在 ENABLE 置位后修改。

位 1 SNGSTRT：单次计数模式使能。

- 0：不使能
- 1：使能单次计数模式，写 1 开始单次计数模式，若在单次计数模式过程中置位 CNTSTRT，则在下一次计数到 ARR 时继续计数（切换到连续计数模式）。该比特位需在 ENABLE 置位后修改。

位 0 ENABLE：LPTIMER 使能。

- 0：禁用 LPTIMER
- 1：使能 LPTIMER

20.15.6 LPTIM_CMP

偏移量：0x14

复位值：0x00000000

31-16	15-0
RESERVED	CMP
rw-0h	rw-0h

位 31-16 RESERVED：保留。

位 15-0 CMP：比较值，需在寄存器 LPTIM_CR 的 ENABLE 置位后才能修改。

20.15.7 LPTIM_ARR

偏移量：0x18

复位值：0x00000001

31-16	15-0
RESERVED	ARR
rw-0h	rw-0h

位 31-16 RESERVED：保留。

位 15-0 ARR：重装载值，需在寄存器 LPTIM_CR 的 ENABLE 置位后才能修改。

20.15.8 LPTIM_CNT

偏移量：0x1c

复位值：0x00000000

31-16	15-0
RESERVED	CNT
r-0h	r-0h

位 31-16 RESERVED：保留。

位 15-0 CNT：计数结果，读该值时，连续两次读到的结果一致才算有效。

20.15.9 LPTIM_CSR

偏移量：0x20

复位值：0x0000001f

31-5	4	3	2	1	0
RESERVED	DOWN_CLR_DONE	UP_CLR_DONE	EXTTRIG_CLR_DONE	ARRM_CLR_DONE	CMPM_CLR_DONE
r-0h	r-0h	r-0h	r-0h	r-0h	r-0h

位 31-5 RESERVED：保留。

位 4 DOWN_CLR_DONE：DOWN 清除完成。

- 0：正在清除 DOWN 标志位
- 1：清除成功

位 3 UP_CLR_DONE：UP 清除完成。

- 0：正在清除 UP 标志位
- 1：清除成功

位 2 EXTTRIG_CLR_DONE：EXTTRIG 清除完成。

- 0：正在清除 EXTTRIG 标志位
- 1：清除成功

位 1 ARRM_CLR_DONE：ARRM 清除完成。

- 0：正在清除 ARRM 标志位
- 1：清除成功

位 0 CMPM _CLR_DONE：CMPM 清除完成。

- 0：正在清除 CMPM 标志位
- 1：清除成功

20.15.10 LPTIM_SR1

偏移量：0x24

复位值：0x00000000

31-7	6	5	4	3	2	1	0
RESERVED	DOWN	UP	ARROK	CMPOK	EXTTRIG	ARRM	CMPM
r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h

位 31-7 RESERVED：保留。

位 6 DOWN：编码模式下计数方向由向上变为向下。

- 0：计数方向未发生由上向下的变化
- 1：计数方向由向上变为向下

位 5 UP：编码模式下计数方向由向下变为向上。

- 0：计数方向未发生由下向上的变化
- 1：计数方向由向下变为向上

位 4 ARROK：ARR 值加载状态。

- 0：未加载完成
- 1：加载完成

位 3 CMPOK：CMP 值加载状态。

- 0：未加载完成
- 1：加载完成

位 2 EXTTRIG：是否检测到有效触发边沿。

- 0：未检测到有效触发边沿
- 1：检测到有效触发边沿

位 1 ARRM：计数器值是否到达 ARR 值。

- 0：计数器值未到达 ARR
- 1：计数器值到达 ARR

位 0 CMPM：计数器值与 CMP 值匹配状态。

- 0：计数器值与 CMP 值未匹配
- 1：计数器值与 CMP 值匹配

21. DMA

21.1 Introduction

DMA 支持外设到外设，外设到 memory，memory 到外设，memory 到 memory 这四种数据搬移方式，支持数据位宽为 8 位、16 位或 32 位，并支持数据的 Auto-reloading 以及数据的链表（LLI）。共有两个 DMA，分别为 DMA0 和 DMA1，每个 DMA 有 4 个 channel。两个 DMA 相互独立，可以同时工作，每个 DMA 中的 4 个 channel 也是相互独立的，可以同时运行。

21.2 Main features

- 传输数据长度的配置
- 支持数据搬移方式的配置
- 支持 Auto-reloading
- 支持 LLI

21.3 Transfer data length configuration

DMA 可以传输多个 block 的数据，传输每个 block 的数据时先以 burst 方式传送，后面有不够 burst 的数据长度的数据时再以 single 方式发送。外设的数据传输如下图：

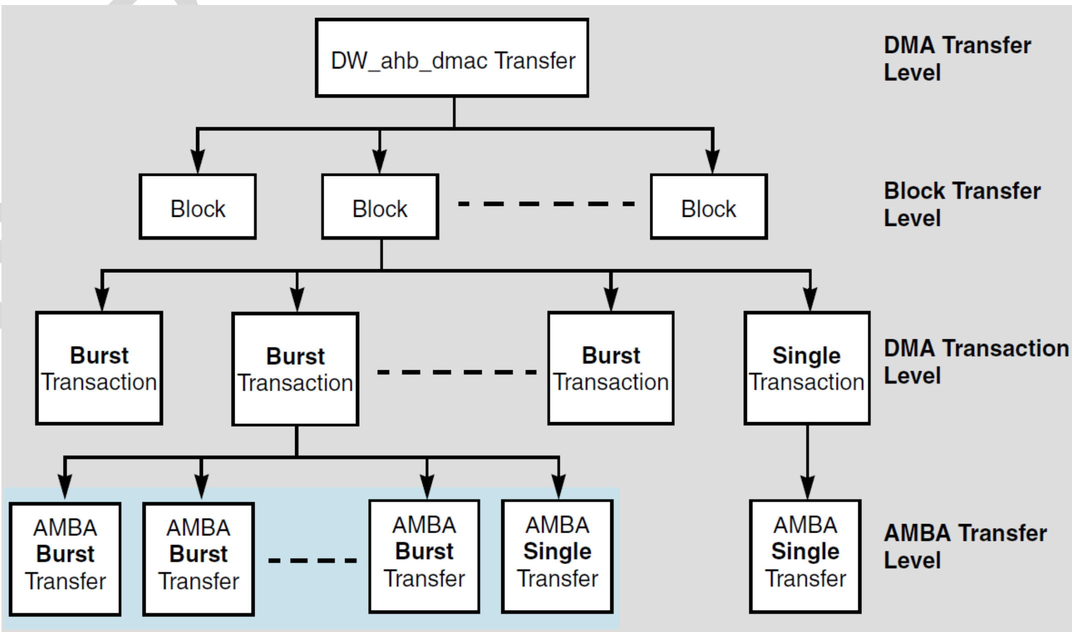

Figure 21-1 Data transfer

DMA 的源和目的数据位宽通过 *DMA_CTLx* 寄存器的 *SRC_TR_WIDTH* 和 *DST_TR_WIDTH* 位段进行配置（x 为 0、1、2 或 3），此位段值为 000 时表示 8bit，为 001 表示 16bit，为 002 表示 32bit。

DMA 的源和目的 burst 数据长度通过 *DMA_CTLx* 寄存器的 *SRC_MSIZE* 和 *DEST_MSIZE* 位段进行配置，此位段值为 000 时表示 1，为 001 表示 4，为 002 表示 8，那么转化为 Bytes 就是 *SRC_MSIZE (DEST_MSIZE) * (数据位宽的 bit 数 / 8)*。DMA 的 burst 数据长度 Bytes 需要与外设的输入或输出 FIFO 长度一致，否则可能导致数据丢失。

DMA 的 block size 通过 *DMA_CTLx* 寄存器的 *BLOCK_TS* 位进行配置，最多为 12 个 bit，那么 block size 最大为 4095，转换为 Bytes 时为 *BLOCK_TS * (数据位宽的 bit 数 / 8)*。

21.4 Data trasfer methods

DMA 支持外设到外设，外设到 memory，memory 到外设，memory 到 memory 四种数据搬移方式。外设到外设指数据的源和目的都为外设；外设到 memory 指源为外设，目的为 memory；memory 到外设指源为 memory，目的为外设；memory 到 memory 指源和目的都为 memory。数据搬移方式通过 *DMA_CTLx* 寄存器的 *TT_FC* 位段进行配置。除了 memory 到 memory 的搬移方式，其他几种方式都要配置外设与 DMA 之间的握手信号即 handshake。外设的 handshake 的值如下表所示：

Table 21-1 Handshake value

Handshake 值	外设信号	外设信号描述
4	lorac_tx	LORA 的 tx
5	lorac_rx	LORA 的 rx
6	dacctrl	DAC
7	adcctrl	ADC
10	i2c2_tx	I2C2 的 tx
11	i2c2_rx	I2C2 的 rx
12	i2c1_tx	I2C1 的 tx
13	i2c1_rx	I2C1 的 rx
14	i2c0_tx	I2C0 的 tx
15	i2c0_rx	I2C0 的 rx
16	ssp2_tx	SSP2 的 tx
17	ssp2_rx	SSP2 的 rx
18	ssp1_tx	SSP1 的 tx
19	ssp1_rx	SSP1 的 rx
20	ssp0_tx	SSP0 的 tx
21	ssp0_rx	SSP0 的 rx
22	lpuart_tx	LPUAR 的 tx
23	lpuart_rx	LPUAR 的 rx

Handshake 值	外设信号	外设信号描述
24	uart3_tx	UART3 的 tx
25	uart3_rx	UART3 的 rx
26	uart2_tx	UART2 的 tx
27	uart2_rx	UART2 的 rx
28	uart1_tx	UART1 的 tx
29	uart1_rx	UART1 的 rx
30	uart0_tx	UART0 的 tx
31	uart0_rx	UART0 的 rx
32	gptim0_ch3	GPTIMER0 的 channel3
33	gptim0_ch2	GPTIMER0 的 channel2
34	gptim0_ch1	GPTIMER0 的 channel1
35	gptim0_ch0	GPTIMER0 的 channel0
36	gptim0_trg	GPTIMER0 的 trigger
37	gptim0_up	GPTIMER0 的 update
38	Gptim1_ch3	GPTIMER1 的 channel3
39	Gptim1_ch2	GPTIMER1 的 channel2
40	Gptim1_ch1	GPTIMER1 的 channel1
41	Gptim1_ch0	GPTIMER1 的 channel0
42	Gptim1_trg	GPTIMER1 的 trigger
43	Gptim1_up	GPTIMER1 的 update
44	gptim2_ch1	GPTIMER2 的 channel1
45	gptim2_ch0	GPTIMER2 的 channel0
46	gptim2_trg	GPTIMER2 的 trigger
47	gptim2_up	GPTIMER2 的 update
48	Gptim3_ch1	GPTIMER3 的 channel1
49	Gptim3_ch0	GPTIMER3 的 channel0
50	Gptim3_trg	GPTIMER3 的 trigger
51	Gptim3_up	GPTIMER3 的 update
52	basictim1_up	BSTIMER1 的 update
53	basictim0_up	BSTIMER0 的 update

21.5 LLI

当有多块不连续的 memory 的数据需要搬移到外设或 memory 时，可以使用 LLI（即链表方式），如下图所示：

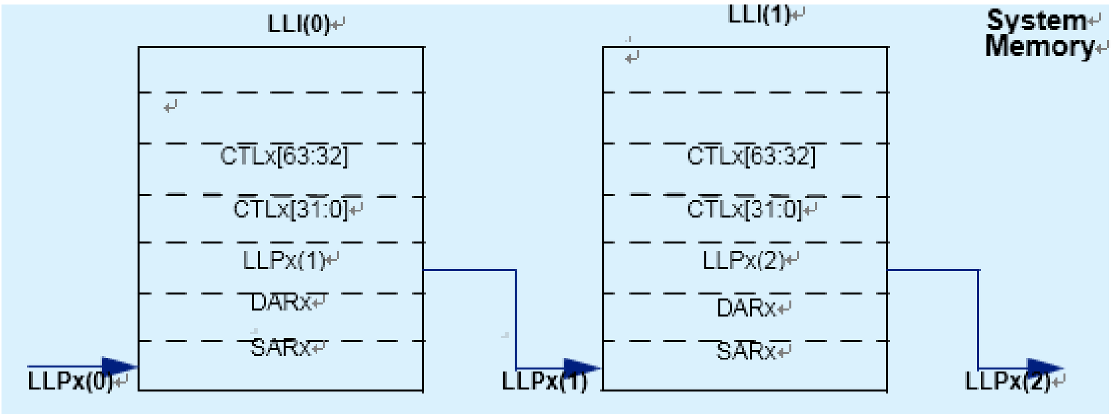

Figure 21-2 LLI chain table

LLI(0)、LLI(1)表示配置 block0、block1 的信息，包括源目的地址、数据位宽、burst 长度和 block 长度。LLPx 表示当前 block 指向下个 block 的地址，第一个 block 的 LLP 指向第二个 block 的地址即 LLI(1)的首地址，依次类推，最后一个 block 的 LLP 为 0。每个 block 长度是可以不一样的，并且 memory 的首地址也是不一样的。

21.6 Auto-reloading

Auto-reloading 指 block 中的 memory 的数据被搬完或 memory 都被写入完成，然后重新从此 memory 的起始地址开始搬送或写入数据，如此循环往复，直至把所用的 DMA 的 channel 去使能后才会停止。DMA 的源和目的都可以使用 Auto-reloading 功能，只要其为 memory 就可以。

21.7 Interrupts

DMA 的中断信号如下:

Table 21-2 DMA interrupts

中断名称	描述
DMA 块传输完成中断	DMA 块传输完成后产生的中断
DMA 目的端处理完成中断	DMA 目的端处理完成后产生的中断
DMA 源端处理完成中断	DMA 源端处理完成后产生的中断
DMA 传输出错中断	DMA 传输过程中产生错误时发生的中断
DMA 完全传输完成中断	DMA 完全传输完成后产生的中断

通过配置 *DMA_MaskBlock*, *DMA_MaskDstTran*, *DMA_MaskSrcTran*, *DMA_MaskErr* 和 *DMA_MaskTfr* 寄存器来使能上述中断。

通过 *DMA_StatusBlock*, *DMA_StatusDstTran*, *DMA_StatusSrcTran*, *DMA_StatusErr* 和 *DMA_StatusTfr* 寄存器可以获得所有中断的状态。

通过配置 *DMA_ClearBlock*、*DMA_ClearDstTran*、*DMA_ClearSrcTran*、*DMA_ClearErr* 和 *DMA_ClearTfr* 寄存器来清除中断状态。

21.8 DMA registers

DMA0 基地址：0x40023000

DMA1 基地址：0x40024000

Table 21-3 DMA Registers Summary

寄存器	偏移量	描述
DMA_SARx	0x00	源地址寄存器，x 表示 channel 0、1、2、3，对应的偏移量分别为 0x00、0x58、0xb0、0x108
DMA_DARx	0x08	目的地址寄存器，x 表示 channel 0、1、2、3，对应的偏移量分别为 0x08、0x60、0xb8、0x110
DMA_LLPx	0x10	链表指针寄存器，x 表示 channel 0、1、2、3，对应的偏移量分别为 0x10、0x68、0xc0、0x118
DMA_CTLx	0x18	通道控制寄存器，x 表示 channel 0、1、2、3，对应的偏移量分别为 0x18、0x70、0xc8、0x120
DMA_CFGx	0x40	通道配置寄存器，x 表示 channel 0、1、2、3，对应的偏移量分别为 0x40、0x98、0xf0、0x148
DMA_StatusTfr	0x2e8	DMA 完全传输完成中断状态寄存器
DMA_StatusBlock	0x2f0	DMA 块传输完成中断状态寄存器
DMA_StatusSrcTran	0x2f8	DMA 源端处理完成中断状态寄存器
DMA_StatusDstTran	0x300	DMA 目的端处理完成中断状态寄存器
DMA_StatusErr	0x308	DMA 传输出错中断状态寄存器
DMA_MaskTfr	0x310	DMA 完全传输完成中断使能寄存器
DMA_MaskBlock	0x318	DMA 块传输完成中断使能寄存器
DMA_MaskSrcTran	0x320	DMA 源端处理完成中断使能寄存器
DMA_MaskDstTran	0x328	DMA 目的端处理完成中断使能寄存器
DMA_MaskErr	0x330	DMA 传输出错中断使能寄存器
DMA_ClearTfr	0x338	DMA 完全传输完成中断状态清除寄存器
DMA_ClearBlock	0x340	DMA 块传输完成中断状态清除寄存器
DMA_ClearSrcTran	0x348	DMA 源端处理完成中断状态清除寄存器
DMA_ClearDstTran	0x350	DMA 目的端处理完成中断状态清除寄存器
DMA_ClearErr	0x358	DMA 传输出错中断状态清除寄存器
DMA_DmaCfgReg	0x398	DMA 使能寄存器
DMA_ChEnReg	0x3a0	DMA channel 使能寄存器

21.8.1 DMA_SARx

偏移量：0x00、0x58、0xb0、0x108

复位值：0x0000000000000000

63-32	31-0
RESERVED	SAR
r-0h	rw-0h

位 63-32 RESERVED：保留。

位 31-0 SAR：DMA 源地址寄存器。

21.8.2 DMA_DARx

偏移量：0x08、0x60、0xb8、0x110

复位值：0x0000000000000000

63-32	31-0
RESERVED	DAR
r-0h	rw-0h

位 63-32 RESERVED：保留。

位 31-0 DAR：DMA 目的地址寄存器。

21.8.3 DMA_LLPx

偏移量：0x10、0x68、0xc0、0x118

复位值：0x0000000000000000

63-32	31-0
RESERVED	LOC
r-0h	rw-0h

位 63-32 RESERVED：保留。

位 31-0 LOC：下一个 LLI 链表的首地址。

21.8.4 DMA_CTLx

偏移量：0x18、0x70、0xc8、0x120

复位值：0x0000000200308801

63-45	44	43-32	31-29	28
RESERVED	DONE	BLOCK_TS	RESERVED	LLP_SRC_EN
r-0h	rw-0h	rw-2h	rw-0h	rw-0h
27	**26-25**	**24-23**	**22-20**	**19**
LLP_DST_EN	SMS	DMS	TT_FC	RESERVED
rw-0h	rw-0h	rw-0h	rw-3h	rw-0h
18	**17**	**16-14**	**13-11**	**10-9**
DST_SCATTER_EN	SRC_GATHER_EN	SRC_MSIZE	DEST_MSIZE	SINC
rw-0h	rw-0h	rw-1h	rw-1h	rw-0h

8-7	6-4	3-1	0
DINC	SRC_TR_WIDTH	DST_TR_WIDTH	INT_EN
rw-0h	rw-0h	rw-0h	rw-1h

位 63-45 RESERVED：保留。

位 44 DONE：LLI 链表中一个 block 是否传输完成。

- 0：完成
- 1：未完成

位 43-32 BLOCK_TS：block 的长度。

位 31-29 RESERVED：保留。

位 28 LLP_SRC_EN：DMA 源使能 LLI 链表。

- 0：去使能
- 1：使能

位 27 LLP_DST_EN：DMA 目的使能 LLI 链表。

- 0：去使能
- 1：使能

位 26-25 SMS：DMA 源的 AHB master 选择。

- 00：AHB master 1
- 01：AHB master 2
- 10：AHB master 3
- 11：AHB master 4

位 24-23 SMS：DMA 目的的 AHB master 选择。

- 00：AHB master 1
- 01：AHB master 2
- 10：AHB master 3
- 11：AHB master 4

位 22-20 TT_FC：DMA 数据搬移方式选择。

- 000：DMA 流控的 Memory 到 Memory 方式
- 001：DMA 流控的 Memory 到外设方式
- 010：DMA 流控的外设到 Memory 方式
- 011：DMA 流控的外设到外设方式
- 其它值：无效

位 19 RESERVED：保留。

位 18 DST_SCATTER_EN：DMA 目的使能 Scatter。

- 0：去使能
- 1：使能

位 17 SRC_GATHER_EN：DMA 源使能 Gather。

- 0：去使能
- 1：使能

位 16-14 SRC_MSIZE：DMA 源的 Burst 长度配置。

- 000：1
- 001：4
- 010：8
- 其它值：无效

位 13-11 DEST_MSIZE：DMA 目的的 Burst 长度配置。

- 000：1
- 001：4
- 010：8
- 其它值：无效

位 10-9 SINC：DMA 源地址控制。

- 00：递增
- 01：递减
- 10：不变化
- 11：不变化

位 8-7 DINC：DMA 目的地址控制。

- 00：递增
- 01：递减
- 10：不变化
- 11：不变化

位 6-4 SRC_TR_WIDTH：DMA 源数据位宽配置。

- 000：8bit
- 001：16bit
- 010：32bit
- 其它值：无效

位 3-1 DST_TR_WIDTH：DMA 目的数据位宽配置。

- 000：8bit
- 001：16bit

- 010：32bit
- 其它值：无效

位 0 INT_EN： DMA 中断使能。

- 0：去使能
- 1：使能

21.8.5 DMA_CFGx

偏移量：0x40、0x98、0xf0、0x148

复位值：0x0000000400020e00

63-47	46-43	42-39	38	37	36-34
RESERVED	DEST_PER	SRC_PER	SS_UPD_EN	DS_UPD_EN	PROTCTL
r-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-1h
33	**32**	**31**	**30**	**29-20**	**19**
FIFO_MODE	FCMODE	RELOAD_DST	RELOAD_SRC	RESERVED	SRC_HS_POL
rw-0h	rw-0h	rw-0h	rw-0h	r-0h	rw-0h
18	**17**	**16**	**15-14**	**13-12**	**11**
DST_HS_POL	LOCK_B	LOCK_CH	LOCK_B_L	LOCK_CH_L	HS_SEL_SRC
rw-0h	rw-1h	rw-0h	rw-0h	rw-0h	rw-1h
10	**9**	**8**	**7-5**	**4-0**	
HS_SEL_DST	FIFO_EMPTY	CH_SUSP	CH_PRIOR	RESERVED	
rw-1h	r-1h	rw-0h	rw-0h	r-0h	

位 63-47 RESERVED： 保留。

位 46-43 DEST_PER： DMA 目的握手接口，有效值为 0 至 3。

位 42-39 SRC_PER： DMA 源握手接口，有效值为 0 至 3。

位 38 SS_UPD_EN： DMA 源状态更新使能。

- 0：去使能
- 1：使能

位 37 DS_UPD_EN： DMA 目的状态更新使能。

- 0：去使能
- 1：使能

位 36-34 PROTCTL： 保护控制。

位 33 FIFO_MODE： FIFO 模式选择。

- 0：可以获取全部 FIFO
- 1：只能获得一半 FIFO

位 32 FCMODE： 源端流控模式选择。

- 0：源端的请求发出就处理
- 1：直到目的端有请求发生才会处理源端的请求

位 31 RELOAD_DST：DMA 目的使能 Auto-reloading。

- 0：去使能
- 1：使能

位 30 RELOAD_SRC：DMA 源使能 Auto-reloading。

- 0：去使能
- 1：使能

位 29-20 RESERVED：保留。

位 19 SRC_HS_POL：DMA 源握手接口信息极性。

- 0：高有效
- 1：低有效

位 18 SRC_HS_POL：DMA 目的握手接口信息极性。

- 0：高有效
- 1：低有效

位 17 LOCK_B：总线锁定控制。

- 0：不锁定
- 1：锁定

位 16 LOCK_CH：DMA channel 锁定控制。

- 0：不锁定
- 1：锁定

位 15-14 LOCK_B_L：总线锁定延时。

- 00：等到 DMA 传输完成
- 01：等到 block 传输完成
- 10：等到 DMA 处理完成

位 13-12 LOCK_CH_L：DMA channel 锁定延时。

- 00：等到 DMA 传输完成
- 01：等到 block 传输完成
- 10：等到 DMA 处理完成

位 11 HS_SEL_SRC：DMA 源握手信号选择。

- 0：硬件握手
- 1：软件握手

位 10 HS_SEL_DST：DMA 目的握手信号选择。

- 0：硬件握手
- 1：软件握手

位 9 FIFO_EMPTY：DMA channel FIFO 是否为空。

- 0：非空
- 1：空

位 8 CH_SUSP：DMA channel FIFO 是否暂停。

- 0：非暂停
- 1：暂停

位 7-5 CH_PRIOR：DMA channel 优先级配置，有效值为 0 至 3，0 为最低优先级，3 为最高优先级。

位 4-0 RESERVED：保留。

21.8.6 DMA_StatusTfr

偏移量：0x2e8

复位值：0x0000000000000000

63-4	3	2	1	0
RESERVED	CHAN3_STATUS	CHAN2_STATUS	CHAN1_STATUS	CHAN0_STATUS
r-0h	r-0h	r-0h	r-0h	r-0h

位 63-4 RESERVED：保留。

位 3 CHAN3_STATUS：DMA channel3 的传输完成状态。

- 0：未完成
- 1：完成

位 2 CHAN2_STATUS：DMA channel2 的传输完成状态。

- 0：未完成
- 1：完成

位 1 CHAN1_STATUS：DMA channel1 的传输完成状态。

- 0：未完成
- 1：完成

位 0 CHAN0_STATUS：DMA channel0 的传输完成状态。

- 0：未完成
- 1：完成

21.8.7 DMA_StatusBlock

偏移量：0x2f0

复位值：0x0000000000000000

63-4	3	2	1	0
RESERVED	CHAN3_STATUS	CHAN2_STATUS	CHAN1_STATUS	CHAN0_STATUS
r-0h	r-0h	r-0h	r-0h	r-0h

位 63-4 RESERVED：保留。

位 3 CHAN3_STATUS：DMA channel3 的块传输完成状态。

- 0：未完成
- 1：完成

位 2 CHAN2_STATUS： DMA channel2 的块传输完成状态。

- 0：未完成
- 1：完成

位 1 CHAN1_STATUS： DMA channel1 的块传输完成状态。

- 0：未完成
- 1：完成

位 0 CHAN0_STATUS： DMA channel0 的块传输完成状态。

- 0：未完成
- 1：完成

21.8.8 DMA_StatusSrcTran

偏移量：0x2f8

复位值：0x0000000000000000

63-4	3	2	1	0
RESERVED	CHAN3_STATUS	CHAN2_STATUS	CHAN1_STATUS	CHAN0_STATUS
r-0h	r-0h	r-0h	r-0h	r-0h

位 63-4 RESERVED： 保留。

位 3 CHAN3_STATUS： DMA channel3 的源端传输完成状态。

- 0：未完成
- 1：完成

位 2 CHAN2_STATUS： DMA channel2 的源端传输完成状态。

- 0：未完成
- 1：完成

位 1 CHAN1_STATUS： DMA channel1 的源端传输完成状态。

- 0：未完成
- 1：完成

位 0 CHAN0_STATUS： DMA channel0 的源端传输完成状态。

- 0：未完成
- 1：完成

21.8.9 DMA_StatusDstTran

偏移量：0x300

复位值：0x0000000000000000

63-4	3	2	1	0
RESERVED	CHAN3_STATUS	CHAN2_STATUS	CHAN1_STATUS	CHAN0_STATUS
r-0h	r-0h	r-0h	r-0h	r-0h

位 63-4 RESERVED：保留。

位 3 CHAN3_STATUS：DMA channel3 的目的端传输完成状态。

- 0：未完成
- 1：完成

位 2 CHAN2_STATUS：DMA channel2 的目的端传输完成状态。

- 0：未完成
- 1：完成

位 1 CHAN1_STATUS：DMA channel1 的目的端传输完成状态。

- 0：未完成
- 1：完成

位 0 CHAN0_STATUS：DMA channel0 的目的端传输完成状态。

- 0：未完成
- 1：完成

21.8.10 DMA_StatusErr

偏移量：0x308

复位值：0x0000000000000000

63-4	3	2	1	0
RESERVED	CHAN3_STATUS	CHAN2_STATUS	CHAN1_STATUS	CHAN0_STATUS
r-0h	r-0h	r-0h	r-0h	r-0h

位 63-4 RESERVED：保留。

位 3 CHAN3_STATUS：DMA channel3 的传输错误状态。

- 0：未出错
- 1：出错

位 2 CHAN2_STATUS：DMA channel2 的传输错误状态。

- 0：未出错
- 1：出错

位 1 CHAN1_STATUS：DMA channel1 的传输错误状态。

- 0：未出错
- 1：出错

位 0 CHAN0_STATUS：DMA channel0 的传输错误状态。

- 0：未出错
- 1：出错

21.8.11 DMA_MaskTfr

偏移量：0x310

复位值：0x0000000000000000

63-12	11	10	9	8
RESERVED	INT_MASK_WE_3	INT_MASK_WE_2	INT_MASK_WE_1	INT_MASK_WE_0
r-0h	w-0h	w-0h	w-0h	w-0h
7-4	**3**	**2**	**1**	**0**
RESERVED	INT_MASK_3	INT_MASK_2	INT_MASK_1	INT_MASK_0
r-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 63-12 RESERVED：保留。

位 11 INT_MASK_WE_3：DMA channel3 的传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 10 INT_MASK_WE_2：DMA channel2 的传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 9 INT_MASK_WE_1：DMA channel1 的传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 8 INT_MASK_WE_0：DMA channel0 的传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 7-4 RESERVED：保留。

位 3 INT_MASK_3：DMA channel3 的传输完成中断使能。

- 0：去使能
- 1：使能

位 2 INT_MASK_2：DMA channel2 的传输完成中断使能。

- 0：去使能
- 1：使能

位 1 INT_MASK_1：DMA channel1 的传输完成中断使能。

- 0：去使能
- 1：使能

位 0 INT_MASK_0：DMA channel0 的传输完成中断使能。

- 0：去使能
- 1：使能

21.8.12 DMA_MaskBlock

偏移量：0x318

复位值：0x0000000000000000

63-12	11	10	9	8
RESERVED	INT_MASK_WE_3	INT_MASK_WE_2	INT_MASK_WE_1	INT_MASK_WE_0
r-0h	w-0h	w-0h	w-0h	w-0h
7-4	**3**	**2**	**1**	**0**
RESERVED	INT_MASK_3	INT_MASK_2	INT_MASK_1	INT_MASK_0
r-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 63-12 RESERVED：保留。

位 11 INT_MASK_WE_3：DMA channel3 的块传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 10 INT_MASK_WE_2：DMA channel2 的块传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 9 INT_MASK_WE_1：DMA channel1 的块传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 8 INT_MASK_WE_0：DMA channel0 的块传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 7-4 RESERVED：保留。

位 3 INT_MASK_3：DMA channel3 的块传输完成中断使能。

- 0：去使能
- 1：使能

位 2 INT_MASK_2：DMA channel2 的块传输完成中断使能。

- 0：去使能
- 1：使能

位 1 INT_MASK_1：DMA channel1 的块传输完成中断使能。

- 0：去使能
- 1：使能

位 0 INT_MASK_0：DMA channel0 的块传输完成中断使能。

- 0：去使能
- 1：使能

21.8.13 DMA_MaskSrcTran

偏移量：0x320

复位值：0x0000000000000000

63-12	11	10	9	8
RESERVED	INT_MASK_WE_3	INT_MASK_WE_2	INT_MASK_WE_1	INT_MASK_WE_0
r-0h	w-0h	w-0h	w-0h	w-0h
7-4	**3**	**2**	**1**	**0**
RESERVED	INT_MASK_3	INT_MASK_2	INT_MASK_1	INT_MASK_0
r-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 63-12 RESERVED：保留。

位 11 INT_MASK_WE_3：DMA channel3 的源端传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 10 INT_MASK_WE_2：DMA channel2 的源端传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 9 INT_MASK_WE_1：DMA channel1 的源端传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 8 INT_MASK_WE_0：DMA channel0 的源端传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 7-4 RESERVED：保留。

位 3 INT_MASK_3：DMA channel3 的源端传输完成中断使能。

- 0：去使能
- 1：使能

位 2 INT_MASK_2：DMA channel2 的源端传输完成中断使能。

- 0：去使能
- 1：使能

位 1 INT_MASK_1：DMA channel1 的源端传输完成中断使能。

- 0：去使能
- 1：使能

位 0 INT_MASK_0：DMA channel0 的源端传输完成中断使能。

- 0：去使能
- 1：使能

21.8.14 DMA_MaskDstTran

偏移量：0x328

复位值：0x0000000000000000

63-12	11	10	9	8
RESERVED	INT_MASK_WE_3	INT_MASK_WE_2	INT_MASK_WE_1	INT_MASK_WE_0
r-0h	w-0h	w-0h	w-0h	w-0h
7-4	**3**	**2**	**1**	**0**
RESERVED	INT_MASK_3	INT_MASK_2	INT_MASK_1	INT_MASK_0
r-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 63-12 RESERVED：保留。

位 11 INT_MASK_WE_3：DMA channel3 的目的端传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 10 INT_MASK_WE_2：DMA channel2 的目的端传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 9 INT_MASK_WE_1：DMA channel1 的目的端传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 8 INT_MASK_WE_0：DMA channel0 的目的端传输完成中断掩码写使能。

- 0：去使能
- 1：使能

位 7-4 RESERVED：保留。

位 3 INT_MASK_3：DMA channel3 的目的端传输完成中断使能。

- 0：去使能
- 1：使能

位 2 INT_MASK_2：DMA channel2 的目的端传输完成中断使能。

- 0：去使能
- 1：使能

位 1 INT_MASK_1：DMA channel1 的目的端传输完成中断使能。

- 0：去使能
- 1：使能

位 0 INT_MASK_0：DMA channel0 的目的端传输完成中断使能。

- 0：去使能
- 1：使能

21.8.15 DMA_MaskErr

偏移量：0x330

复位值：0x0000000000000000

63-12	11	10	9	8
RESERVED	INT_MASK_WE_3	INT_MASK_WE_2	INT_MASK_WE_1	INT_MASK_WE_0
r-0h	w-0h	w-0h	w-0h	w-0h
7-4	**3**	**2**	**1**	**0**
RESERVED	INT_MASK_3	INT_MASK_2	INT_MASK_1	INT_MASK_0
r-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 63-12 RESERVED：保留。

位 11 INT_MASK_WE_3：DMA channel3 的传输出错中断掩码写使能。

- 0：去使能
- 1：使能

位 10 INT_MASK_WE_2：DMA channel2 的传输出错中断掩码写使能。

- 0：去使能
- 1：使能

位 9 INT_MASK_WE_1：DMA channel1 的传输出错中断掩码写使能。

- 0：去使能
- 1：使能

位 8 INT_MASK_WE_0：DMA channel0 的传输出错中断掩码写使能。

- 0：去使能
- 1：使能

位 7-4 RESERVED：保留。

位 3 INT_MASK_3：DMA channel3 的传输出错中断使能。

- 0：去使能
- 1：使能

位 2 INT_MASK_2：DMA channel2 的传输出错中断使能。

- 0：去使能
- 1：使能

位 1 INT_MASK_1：DMA channel1 的传输出错中断使能。

- 0：去使能
- 1：使能

位 0 INT_MASK_0：DMA channel0 的传输出错中断使能。

- 0：去使能
- 1：使能

21.8.16 DMA_ClearTfr

偏移量：0x338

复位值：0x0000000000000000

63-4	3	2	1	0
RESERVED	CHAN3_CLEAR	CHAN2_CLEAR	CHAN1_CLEAR	CHAN0_CLEAR
r-0h	w-0h	w-0h	w-0h	w-0h

位 63-4 RESERVED：保留。

位 3 CHAN3_CLEAR：DMA channel3 的传输完成状态清除。

- 0：不操作
- 1：清除

位 2 CHAN2_CLEAR：DMA channel2 的传输完成状态清除。

- 0：不操作
- 1：清除

位 1 CHAN1_CLEAR：DMA channel1 的传输完成状态清除。

- 0：不操作
- 1：清除

位 0 CHAN0_CLEAR：DMA channel0 的传输完成状态清除。

- 0：不操作
- 1：清除

21.8.17 DMA_ClearBlock

偏移量：0x340

复位值：0x0000000000000000

63-4	3	2	1	0
RESERVED	CHAN3_CLEAR	CHAN2_CLEAR	CHAN1_CLEAR	CHAN0_CLEAR
r-0h	w-0h	w-0h	w-0h	w-0h

位 63-4 RESERVED：保留。

位 3 CHAN3_CLEAR：DMA channel3 的块传输完成状态清除。

- 0：不操作
- 1：清除

位 2 CHAN2_CLEAR：DMA channel2 的块传输完成状态清除。

- 0：不操作
- 1：清除

位 1 CHAN1_CLEAR：DMA channel1 的块传输完成状态清除。

- 0：不操作
- 1：清除

位 0 CHAN0_CLEAR： DMA channel0 的块传输完成状态清除。

- 0：不操作
- 1：清除

21.8.18 DMA_ClearSrcTran

偏移量：0x348

复位值：0x0000000000000000

63-4	3	2	1	0
RESERVED	CHAN3_CLEAR	CHAN2_CLEAR	CHAN1_CLEAR	CHAN0_CLEAR
r-0h	w-0h	w-0h	w-0h	w-0h

位 63-4 RESERVED： 保留。

位 3 CHAN3_CLEAR： DMA channel3 的源端传输完成状态清除。

- 0：不操作
- 1：清除

位 2 CHAN2_CLEAR： DMA channel2 的源端传输完成状态清除。

- 0：不操作
- 1：清除

位 1 CHAN1_CLEAR： DMA channel1 的源端传输完成状态清除。

- 0：不操作
- 1：清除

位 0 CHAN0_CLEAR： DMA channel0 的源端传输完成状态清除。

- 0：不操作
- 1：清除

21.8.19 DMA_ClearDstTran

偏移量：0x350

复位值：0x0000000000000000

63-4	3	2	1	0
RESERVED	CHAN3_CLEAR	CHAN2_CLEAR	CHAN1_CLEAR	CHAN0_CLEAR
r-0h	w-0h	w-0h	w-0h	w-0h

位 63-4 RESERVED： 保留。

位 3 CHAN3_CLEAR： DMA channel3 的目的端传输完成状态清除。

- 0：不操作
- 1：清除

位 2 CHAN2_CLEAR： DMA channel2 的目的端传输完成状态清除。

- 0：不操作

- 1：清除

位 1 CHAN1_CLEAR：DMA channel1 的目的端传输完成状态清除。

- 0：不操作
- 1：清除

位 0 CHAN0_CLEAR：DMA channel0 的目的端传输完成状态清除。

- 0：不操作
- 1：清除

21.8.20 DMA_ClearErr

偏移量：0x358

复位值：0x0000000000000000

63-4	3	2	1	0
RESERVED	CHAN3_CLEAR	CHAN2_CLEAR	CHAN1_CLEAR	CHAN0_CLEAR
r-0h	w-0h	w-0h	w-0h	w-0h

位 63-4 RESERVED：保留。

位 3 CHAN3_CLEAR：DMA channel3 的传输出错状态清除。

- 0：不操作
- 1：清除

位 2 CHAN2_CLEAR：DMA channel2 的传输出错状态清除。

- 0：不操作
- 1：清除

位 1 CHAN1_CLEAR：DMA channel1 的传输出错状态清除。

- 0：不操作
- 1：清除

位 0 CHAN0_CLEAR：DMA channel0 的传输出错状态清除。

- 0：不操作
- 1：清除

21.8.21 DMA_DmaCfgReg

偏移量：0x398

复位值：0x0000000000000000

63-1	0
RESERVED	DMA_EN
r-0h	rw-0h

位 63-1 RESERVED：保留。

位 0 DMA_EN：DMA 使能控制。

- 0：去使能
- 1：使能

21.8.22 DMA_ChEnReg

偏移量：0x3a0

复位值：0x0000000000000000

63-12	11	10	9	8
RESERVED	CH_EN_WE_3	CH_EN_WE_2	CH_EN_WE_1	CH_EN_WE_0
r-0h	w-0h	w-0h	w-0h	w-0h
7-4	**3**	**2**	**1**	**0**
RESERVED	CH_EN_3	CH_EN_2	CH_EN_1	CH_EN_0
r-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 63-12 RESERVED：保留。

位 11 CH_EN_WE_3：DMA channel3 的使能控制信息的写使能。

- 0：去使能
- 1：使能

位 10 CH_EN_WE_2：DMA channel2 的使能控制信息的写使能。

- 0：去使能
- 1：使能

位 9 CH_EN_WE_1：DMA channel1 的使能控制信息的写使能。

- 0：去使能
- 1：使能

位 8 CH_EN_WE_0：DMA channel0 的使能控制信息的写使能。

- 0：去使能
- 1：使能

位 7-4 RESERVED：保留。

位 3 CH_EN_3：DMA channel3 的使能控制，当 DMA 传输完成后，硬件自动将此 channel 去使能。

- 0：去使能

- 1：使能

位 2 CH_EN_2：DMA channel2 的使能控制，当 DMA 传输完成后，硬件自动将此 channel 去使能。

- 0：去使能
- 1：使能

位 1 CH_EN_1：DMA channel1 的使能控制，当 DMA 传输完成后，硬件自动将此 channel 去使能。

- 0：去使能
- 1：使能

位 0 CH_EN_0：DMA channel0 的使能控制，当 DMA 传输完成后，硬件自动将此 channel 去使能。

- 0：去使能
- 1：使能

22. GPTIMER

22.1 Introduction

ASR6601 共有 4 个通用定时器（GPTIMER），其中 GPTIMER0 和 GPTIMER1 有 4 路通道，GPTIMER2 和 GPTIMER3 有 2 路通道，即 GPTIMER2 和 GPTIMER3 没有通道 2 和 3。

GPTIMER 包含 16-bit 计数器，支持自动重装载功能，且支持最多 16-bit 可编程的分频计数器，4 路通道可独立配置为输入或输出，支持输入捕获、输出比较等功能，计数时钟和计数模式可软件配置，支持连接霍尔器件即支持编码模式（仅适用于 GPTIMER0 和 GPTIMER1），支持 DMA 配置，有独立中断输出，支持编码功能等。基于丰富的通道配置和功能，该 GPTIMER 可用于定时计数、测量输入脉冲宽度（us-ms 级）、产生 PWM 波形等应用。

22.2 Main features

- 16-bit 计数器，支持自动重装载，可配置边沿对齐（向上、向下）计数和中间对齐（向上/向下）计数
- 16-bit 可编程分频计数器（分频系数 1-65535），可在计数过程中配置
- 最多 4 路独立通道，可完成输入捕获、输出比较、PWM 波形输出、单脉冲波形输出
- 支持通道输出极性选择，和输入边沿配置
- 支持与外部输入或其他模块（GPTIMER、ADC、DAC）同步
- 独立的 DMA 通道，最多 6 组 DMA 请求，包括更新事件、触发事件以及 4 组通道事件（捕获、比较）
- 支持正交编码功能
- 支持外部触发通道输入时钟用于计数并且支持外部触发通道输入触发信号，支持通道输入时钟用于计数
- 支持通道重映射，即把其它模块的 GPIO 信号或内部信号映射到通道或外部通道

GPTIMER 的架构框图如下：

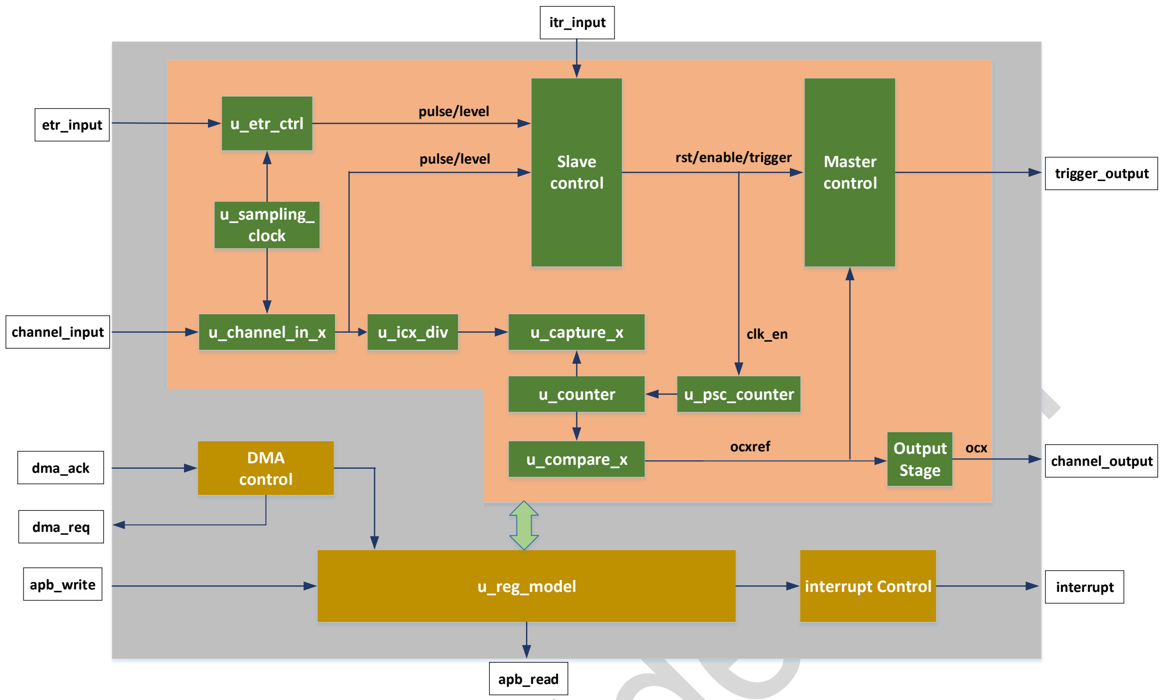

Figure 22-1 GPTIMER diagram

Table 22-1 GPTIMER module introduction

模块名称	描述
slave control	从模式控制器
master control	主模式控制器
u_etr_ctrl	ETR 通道控制，包括极性、分频、滤波等配置
u_channel_in_x	输入通道 x 控制，包括极性、滤波及边沿配置
u_icx_div	输入通道 x 事件分频器
u_sampling_clock	产生滤波器的采样时钟
u_capture_x	输入通道 x 捕获功能
u_compare_x	输出通道 x 比较功能
u_psc_counter	16-bit 分频计数器
u_counter	16-bit 计数器
u_reg_model	寄存器相关配置
output stage	输出控制
interrupt control	中断控制
dma control	DMA 功能控制
itr_input	其它 GPTIMER 的内部输入
etr_input	外部触发通道的输入
channel_input	通道输入
dma_ack	DMA 回复的 ACK
dma_req	向 DMA 发送的请求
apb_write	apb 总线写

模块名称	描述
apb_read	apb 总线读
trigger_output	主模式下的信号输出，为内部信号，不会输出到外部
channel_output	通道输出

22.3 Counter

GPTIMER 的计数器共 16-bit，支持向上、向下、中间对齐计数，计数时钟可选，可软件配置计数使能与关闭，软件可随时读写（建议不要在计数过程中写入，以免发生未知错误）。

22.3.1 Counter clock

GPTimer 共有四种计数时钟源，分别是内部时钟、外部时钟模式 1、外部时钟模式 2 以及内部触发信号控制计数。其中，内部时钟为默认方式（SMS==3'b000），时钟来自 RCC，只要 CEN 置位，则分频计数器和该计数器便开始计数，其他三种情况，均使用相应信号作为计数使能，并不是作为真正的时钟。

外部时钟模式 1（SMS==3'b111，TS==3'b100/101/110），该模式下，计数器由所选择的通道输入的上升沿或下降沿或双沿作为计数器的计数使能控制计数，例如选择通道 0 的上升沿控制计数，则每个上升沿都会让计数器加 1（向上计数、不分频），波形如下图：

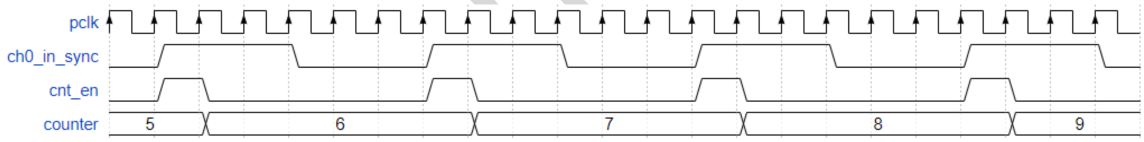

Figure 22-2 External clock mode 1 counting

外部时钟模式 2（ECE==1），该模式下，计数器由 ETR 的上升沿或下降沿作为计数器的计数使能控制计数，例如配置 ETR 的上升沿有效，则波形如下图所示。

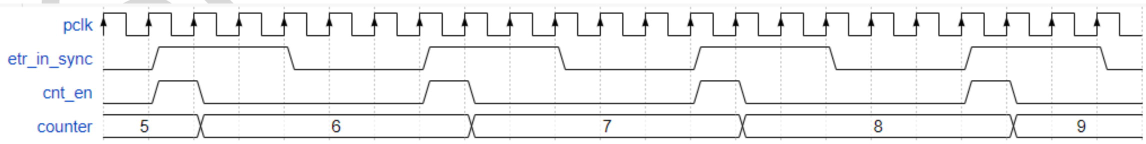

Figure 22-3 External clock mode 2 counting

GPTIMER 还可以选择内部触发信号控制计数（SMS==3'b111，TS==3'b001/010/011），即可以由上一级 GPTIMER 的触发输出信号作为该 GPTIMER 的计数时钟，从而实现 GPTIMER 的级联，该情况下，上一级 GPTIMER 相当于一个分频计数器，波形如下图所示。

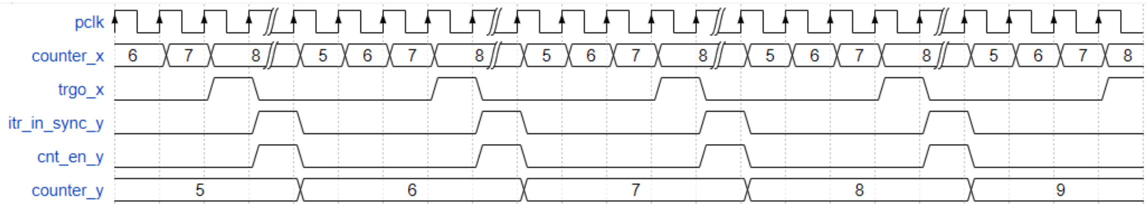

Figure 22-4 Internal trigger signal for clock counting

当 ETR 做为计数时钟输入时可以有两种方式实现，一种是外部时钟模式 1，配置 SMS==3'b111，TS==3'b111，另一种是外部时钟模式 2，配置 ECE==1

22.3.2 Auto-reload

GPTIMER 支持自动重装载功能，向上计数时，计数到重装载值（ARR）后，将会归零重新计数，向下计数时，会从 ARR 开始计数，计数到 0 后回到 ARR 重新计数，中间对齐计数时，计数器从 0 开始计数到 ARR-1，接着从 ARR 计数到 0。

ARR 可软件配置（ARPE）是否使用启用影子寄存器，如果 ARPE=0，则禁用影子寄存器，软件写入的值同步更新到 ARR 供计数器使用，如果 ARPE=1，则软件写入的值不会立即生效，直到更新事件到来，才会将该值更新到影子寄存器中供计数器使用。

22.3.3 Up-count

若配置为向上计数模式，则计数器使能且有计数时钟后，会从 0 开始递加到 ARR，产生向上溢出事件（overflow），然后归零重新开始计数。计数过程中如果 UG 置位（软件或硬件），则计数器包括分频计数器会被初始化（归零）。时序上，overflow 标志将在最后一个计数值期间产生，如果启用影子寄存器，则 ARR、PSC、CCRx 等寄存器将会在下一轮计数开始时更新到相应的影子寄存器，波形如下图所示。

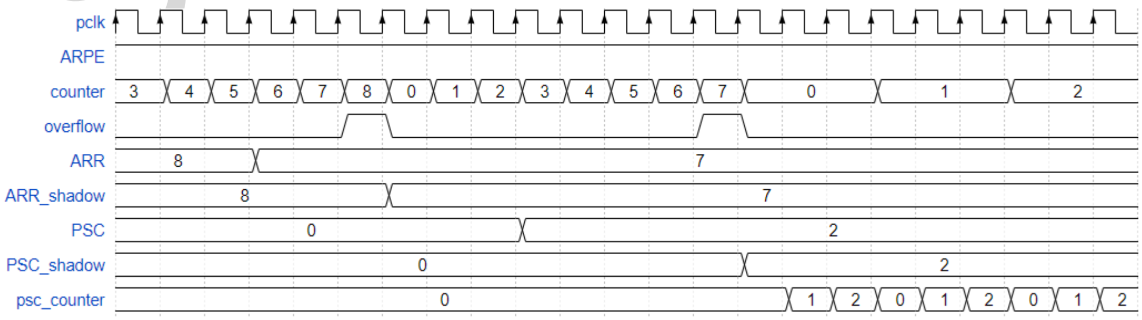

Figure 22-5 Up-counting

22.3.4 Down-count

若配置为向下计数模式，则计数器使能且有计数时钟后，会从 ARR 开始递减到 0，产生向下溢出事件（underflow），然后回到 ARR 重新开始计数。计数过程中如果 UG 置位（软件或硬件），则计数器包括分频计数器会被初始化（计数器回到 ARR，分频计数器归零）。时序上，underflow 标志将在最后一个计数值（CNT=0）期间产生，但是请注意，如果启用影子寄存器，则 ARR 寄存器将会在下一轮计数开始之前（CNT=0）更新到相应的影子寄存器，以保证下一轮计数过程可以使用最新的装载值和分频值，PSC 和 CCRx 则与之前相同，将在 underflow 下一时钟更新到影子寄存器，波形如下图所示。

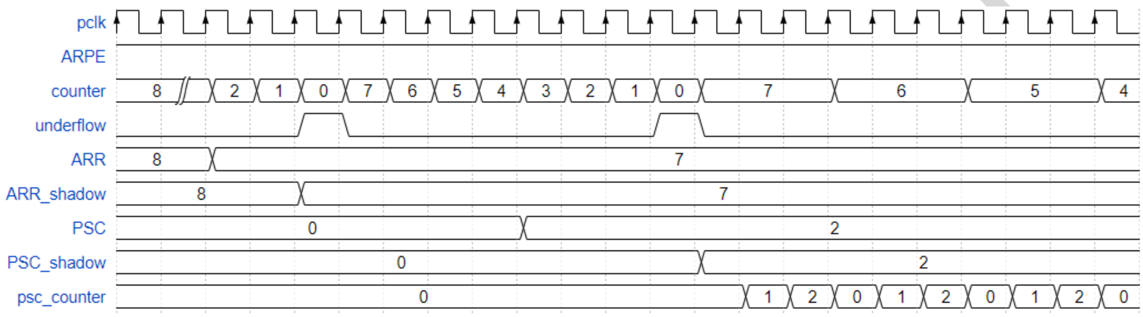

Figure 22-6 Down-counting

22.3.5 Center-aligned count

若配置为中间对齐计数模式，则计数器使能且有计数时钟后，会从 0 开始递增到 ARR-1，产生 overflow 事件，然后从 ARR 递减到 1，产生 underflow 事件，再从 0 开始重新计数。计数过程中如果 UG 置位（软件或硬件），则计数器包括分频计数器会被初始化（归零）。请注意，如果启用影子寄存器，则 ARR 和 PSC 寄存器将会在向上计数到老的 ARR-1 时更新到相应的影子寄存器，以保证在向下计数时可以使用新的 ARR 和新的 PSC，CCRx 的更新与之前情况相同。向下计数时，将会在产生 underflow 后更新 ARR、PSC 和 CCRx 的影子寄存器。在该模式下，计数方向由硬件控制，软件配置无效。波形如图，

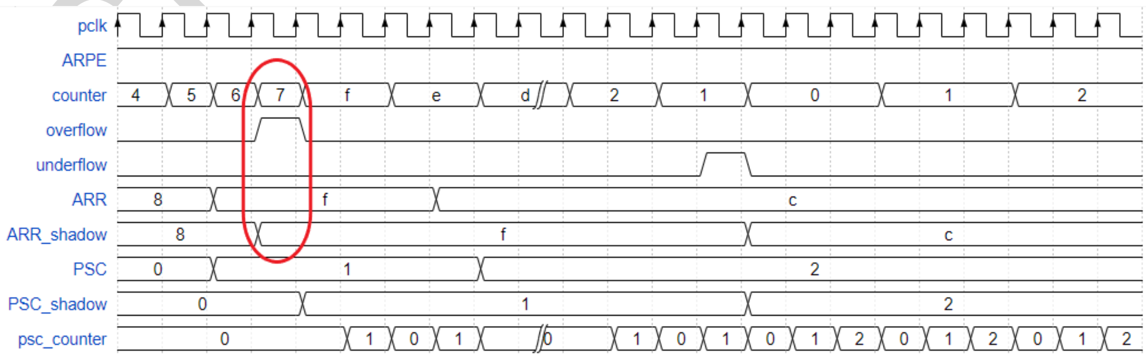

Figure 22-7 Center-aligned counting

22.4 Prescaler

GPTIMER 支持 16-bit（1~65535）可编程分频，这一功能通过该分频计数器实现。上一级电路产生的计数使能信号将作为该分频计数器的使能控制计数，当分频计数器计数到预先加载的分频值后，输出一个脉冲，作为下一级计数器的计数使能，然后分频计数器归零重新计数，如此循环。

分频计数器的分频值默认启用影子寄存器，即软件的写操作不会立即生效，而是直到更新事件（UG 置位、计数溢出）到来，才会将新的分频值写入影子寄存器，此时该分频值才正式生效。软件读操作读取的是写入的寄存器值，而不是影子寄存器，如果在更新事件到来前有多次写操作，则会覆盖之前写入的值。

举一个例子说明分频计数器，如配置为 4 分频，则输入 4 个高电平，才会输出一个有效的脉冲，波形如下图（通道 0，无滤波，选择通道 0 上升沿作为有效脉冲，配置 ic0 为 4 分频）。

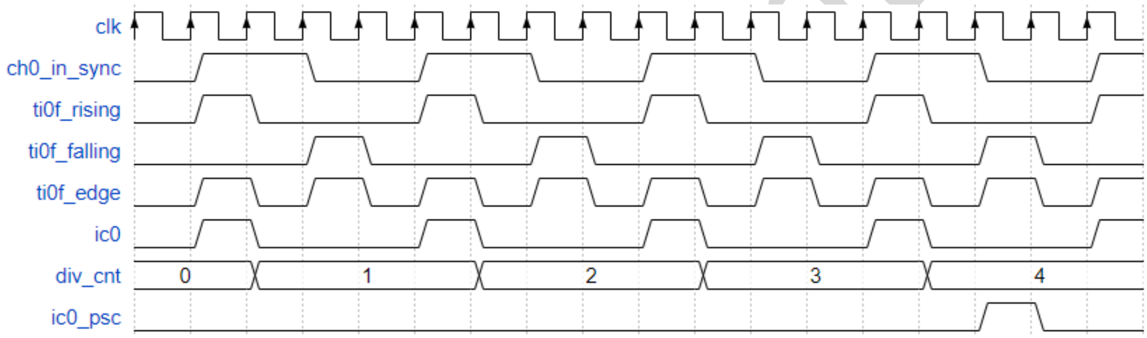

Figure 22-8 Prescaler

22.5 Capture mode

各输入通道和外部触发通道均可以选择数字滤波功能，该数字滤波功能通过使用高频的采样时钟（频率至少是输入信号的 4 倍）对输入信号采样。GPTIMER 内部所有 Flip-Flop 的时钟均由 pclk 提供。软件可以配置采样时钟的频率（CKD 分别为 pclk、pclk/2，pclk/4），通过采用计数器实现分频，如配置采样频率为 pclk 的 4 分频，则计数器由 pclk 控制计数，每 4 个 pclk 周期产生一个脉冲（宽度为 pclk 的一个周期），用于后级计数器的使能信号。在各通道内，用户还可以再次配置数字滤波器的采样时钟分频，即配置 ETF 的值，滤波原理上述相同。

22.6 Troughput

每个 GPTIMER 的各个通道有多路来源，这些信号来源与 GPTIMER 均为异步关系，因此在模块内部需要做同步处理。同步后的通道输入信号，可以根据软件配置进行滤波处理，滤波的采样频率和窗口长度均可以软件配置（ICxF），滤波后的信号由一个边沿检测器产生边沿信号，可以由软件配置有效电平（或有效边沿）。处理后的通道信号可以作为从模式控制器的控制信号，编码模式输入信号，也可以作为输入捕获使能信号（可配置分频）。每个输入通道可映射到当前通道、相邻通道或内部触发信号 TRC（CCxS[1:0]配置），具体方案见表格（以通道 0 为例），其中 ti0fp0 为映射到通道 0 的输入信号，ti1fp0 为映射到通道 1 的输入信号。

Table 22-2 Input channels polarity configuration

{CCONP，CCOP}	有效脉冲（应用于输入捕获、复位模式、触发模式、外部时钟模式）		有效电平（应用于 Gate 模式、编码模式）
	ti0fp0	ti1fp0	ti0fp
2'b00	通道 0 上升沿	通道 1 上升沿	通道 0 高电平
2'b01	通道 0 下降沿	通道 1 下降沿	通道 0 低电平
2'b10	保留	保留	保留
2'b11	通道 0 双沿	通道 1 双沿	通道 0 高电平

Table 22-3 Input channel mapping

CCxS	icx 映射
2'b01	tixfpx（x 代表当前通道）
2'b10	tiyfpx（y 代表相邻通道）
2'b11	trc（仅适用于 TS=3'b000、3'b001、3'b010、3'b100）

此外，通道 0 与其他通道不同，可以软件配置（TI0S 置位）通道 0 连接到通道 0、通道 1 和通道 2 的异或输出，此时该通道的其他功能依然有效，该功能仅适用于 GPTIMER0 和 GPTIMER1。

22.6.1 Input capture

输入捕获仅在通道被配置为输入模式且 CCxE 置位时被激活，可以由软件（CCxG）或硬件（当前通道、相邻通道或内部互联信号）触发捕获行为。当有效的捕获触发信号产生时，GPTIMER 会把当前 counter 的值锁存到相应的 CCRx 寄存器中，并且置位 CCxIF 标志位，如果使能了相应中断或 DMA 屏蔽位，则会产生中断信号或 DMA 请求。如果 CCxIF 置位时（未被软件清除）又发生了不止一次捕获行为，则 CCxOF 置位，指示发生了捕获溢出事件，读取 CCxR 寄存器（或 SR 寄存器相应位写 0）可以清除 CCxIF 和 CCxOF。波形如下图所示。

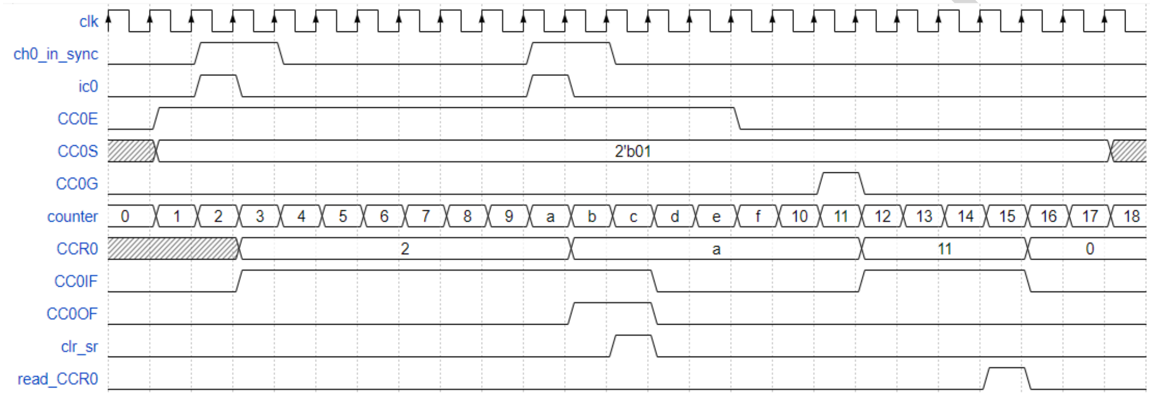

Figure 22-9 Input capture

22.6.2 Output compare

输出比较功能仅在通道被配置为输出模式且 CCxE 置位时被激活，该功能通过比较 counter 值与 CCRx 的值，控制通道输出高低翻转，进而输出特定的波形。

22.6.2.1 CCRx preload

CCRx 寄存器的写入有两种方式，若 CCxPE 置位，则软件写入的 CCRx 值不会直接被使用，真正起作用的是影子寄存器，作为缓冲，直到更新事件发生后，才会将 CCRx 的值更新到影子寄存器中；若 OCxPE 复位，则软件写入的 CCRx 值会直接被使用，影子寄存器禁用。

22.6.2.2 Output compare modes

当匹配（CNT==CCR）发生时，通道输出会根据配置的模式进行翻转，且 CCxIF 标志位会置位，若使能了相应的中断或 DMA 屏蔽位，则会产生中断或 DMA 请求，具体的模式控制如下表格所示。

Table 22-4 Output waveform description

比较模式	计数模式	输出波形
冻结模式	Any	无论 CNT 如何变化，输出维持不变
SET 模式	Any	在 CNT==CCR 后，输出高电平
RESET 模式	Any	在 CNT==CCR 后，输出低电平
TOGGLE 模式	Any	在 CNT==CCR 时，翻转当前电平
强制 RESET 模式	Any	选择该模式后，直接输出低电平，忽略比较结果
强制 SET 模式	Any	选择该模式后，直接输出高电平，忽略比较结果
PWM1 模式	向上计数（边沿对齐 pwm）	CNT<CCR 时，输出高电平，CNT>=CCR 时，输出低电平。如果 CCR>ARR，则输出一直为高电平（100%PWM），如果 CCR==0，则输出一直为低电平（0%PWM）。
	向下计数（边沿对齐 pwm）	CNT<=CCR 时，输出高电平，CNT>CCR 时，输出低电平。如果 CCR>ARR，则输出一直为高电平（100%PWM）。注意：0%PWM 模式在该情况下不支持。
	中间计数（中间对齐 pwm）	相当于向上计数与向下计数相结合。如果 CCR>=ARR，则输出一直为高电平（100%PWM），如果 CCR==0，则输出一直为低电平（0%PWM）。
PWM2 模式	向上计数（边沿对齐 pwm）	CNT<CCR 时，输出低电平，CNT>=CCR 时，输出高电平。如果 CCR>ARR，则输出一直为低电平（0%PWM），如果 CCR==0，则输出一直为高电平（100%PWM）。
	向下计数（边沿对齐 pwm）	CNT<=CCR 时，输出低电平，CNT>CCR 时，输出高电平。如果 CCR>ARR，则输出一直为低电平（0%PWM）。注意：100%PWM 模式在该情况下不支持。
	中间计数（中间对齐 pwm）	相当于向上计数与向下计数相结合。如果 CCR>=ARR，则输出一直为低电平（0%PWM），如果 CCR==0，则输出一直为高电平（100%PWM）。

各模式下输出波形如下所示（以向上计数为例）：

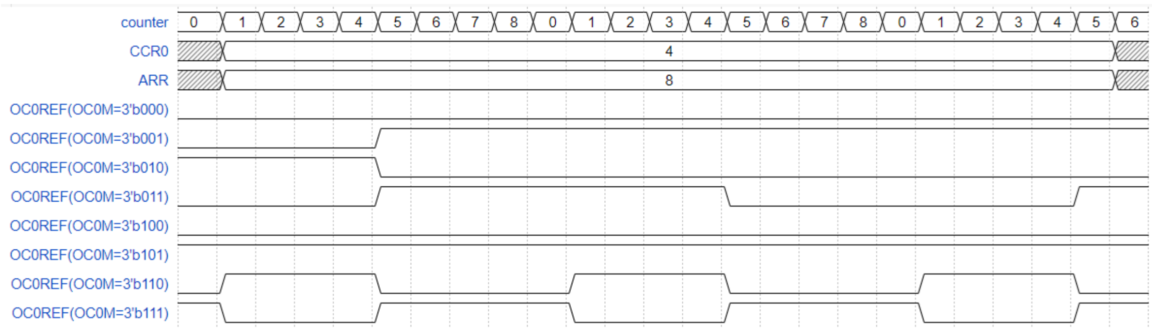

Figure 22-10 Output compare mode waveforms

其中，PWM 模式下还支持通过配置 ARR 和 CCR 控制输出 0%和 100%波形，边沿对齐计数的 PWM2 波形如下图所示。

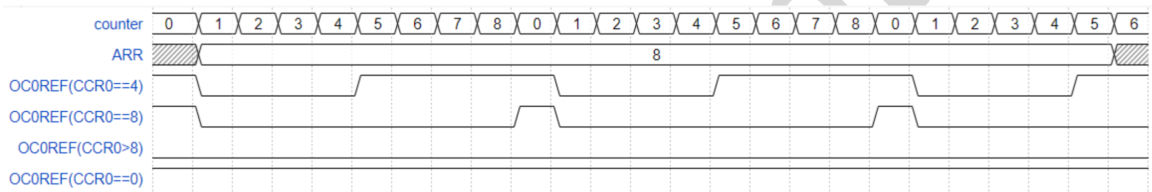

Figure 22-11 PWM2 edge-aligned counting

中间对齐计数的 PWM2 波形如下图所示。

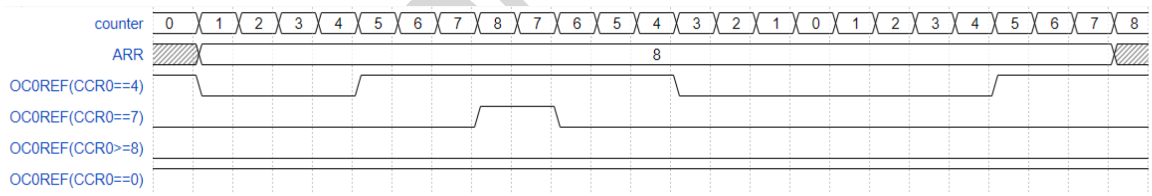

Figure 22-12 PWM2 center-aligned counting

22.6.2.3 Single pulse fast output

单脉冲模式下（OPM 置位），两个 PWM 模式可以配置为快速输出模式（置位 OCxFE），使能快速模式后，输出波形将忽略 CNT 和 CCR 的比较结果，改为由触发信号（根据 TS 选择）上升沿控制电平翻转，输出信号电平等同于匹配事件发生后的电平，例如，配置 GPTIMER 通道 0 为输出模式，选择 PWM1 模式，触发信号选择 ETR 输入，则当 ETR 输入高电平后，通道 0 立刻输出高电平（OCxP=0 情况下），该功能可以有效减少从触发信号边沿到波形输出之间的延迟。使能快速模式时的单脉冲输出波形如下图：

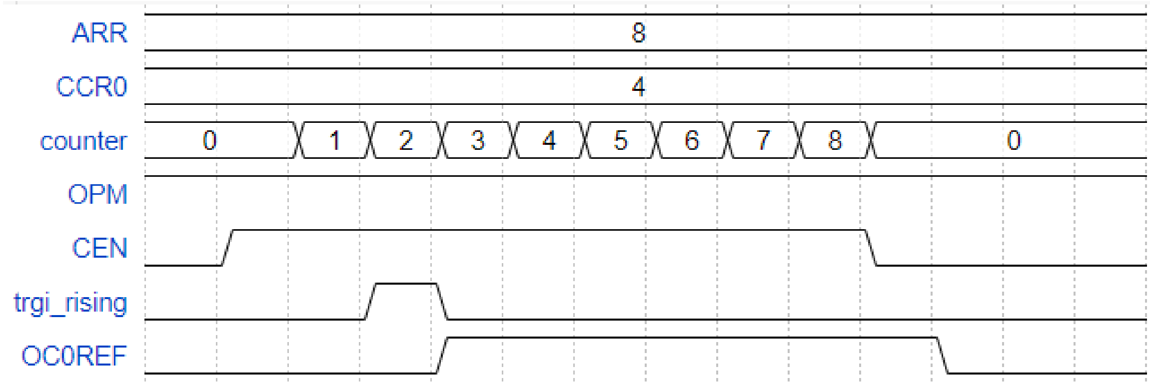

Figure 22-13 Single pulse output waveform in fast mode

22.6.2.4 Brake signal

输出波形除了受计数值的影响，还可以通过外部触发信号（ETR）硬件清零，若要使用该功能，需提前使能 OCxCE 位，同时保证 ETR 禁用分频（ETP=2'b00），且 ETR 不得作为计数时钟。使能该功能后（OCxCE=1），ETR 的电平有效（默认高电平）时，通道输出将被清除，更改 ETR 的有效电平时通过配置 ETP 实现。关闭该功能后（OCxCE=0），通道输出不会立刻恢复，而是等到下一次计数周期开始才会恢复正常输出。开启和关闭外部触发信号清除通道输出功能的对比波形如下图：

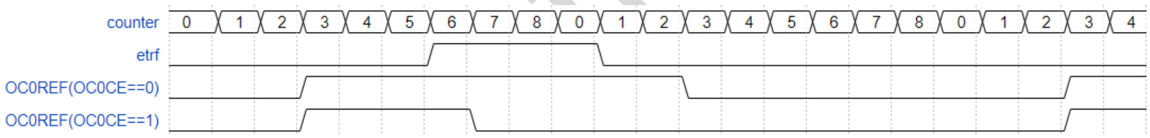

Figure 22-14 External brake signal trigger

22.7 Trigger input channels

每个 GPTIMER 的 ETR 有多路来源，通过 MUX 选择一路输入到模块内，这些信号来源与 GPTIMER 均为异步关系，因此在模块内部需要做同步处理。同步后的 ETR 信号，可以根据软件配置选择有效电平（或有效边沿）、配置分频（1、2、4、8）以及滤波处理，滤波的采样频率和窗长度均可以软件配置（ETF）。

22.8 Update event management

更新事件主要有以下事件源：

1. 计数器的溢出事件（overflow 和 underflow）
2. UG 置位

与更新事件管理相关的控制信号主要是 URS 和 UDIS，具体控制如下：

- 若 UDIS=0，URS=0，则 underflow、overflow、UG 置位会初始化 counter 和 pre-scale counter（center-aligned 模式下 counter 不会被 overflow 清零，也不会被 underflow 加载 ARR），如果启用影子寄存器，更新事件将会把写入的值更新到影子寄存器中（ARR 取决于 ARPE，CCRx 取决于 OCxPE），UIF 会置位，如果使能了中断或 DMA 屏蔽位，则会产生中断或 DMA 请求。
- 若 UDIS=0，URS=1，则 underflow、overflow、UG 置位会初始化 counter 和 pre-scale counter（center-aligned 模式下 counter 不会被 overflow 清零，也不会被 underflow 加载 ARR），如果启用影子寄存器，更新事件将会把写入的值更新到影子寄存器中（ARR 取决于 ARPE，CCRx 取决于 OCxPE），UIF 只会在 overflow 或 underflow 情况下置位，如果使能了中断或 DMA 屏蔽位，则会产生中断或 DMA 请求，该配置可以有效避免输入捕获模式下 UG 置位初始化计数器时，同时产生捕获中断和更新中断的情况。
- 若 UDIS=1（忽略 URS），则 underflow、overflow、UG 置位会初始化 counter 和 pre-scale counter（center-aligned 模式下 counter 不会被 overflow 清零，也不会被 underflow 加载 ARR），但是影子寄存器不会被更新，且 UIF 不会置位，因此不会产生相应中断或 DMA 请求。

22.9 Quadratute encoder mode

该 GPTIMER 支持正交编码计数功能，可以通过通道 0 和通道 1 输入正交信号，进行计数和方向检测。编码模式共有三种，仅在通道 0 边沿计数、仅在通道 1 边沿计数以及在通道 1 和通道 2 边沿计数。在此功能下，两个通道输入可以配置数字滤波功能，极性配置和分频配置无效。通过两个通道信号的组合，可以产生计数使能和方向控制信号，控制计数器加减（如果 CEN 使能），因此在该模式下，软件配置计数方向无效。具体的组合方式见下表，

Table 22-5 Encoder mode

编码模式	通道 0/1 电平	通道 0 边沿		通道 1 电平	
		上升沿	下降沿	上升沿	下降沿
编码模式 1（在通道 1 边沿计数）	高电平	-	-	向上计数	向下计数
	低电平	-	-	向下计数	向上计数
编码模式 2（在通道 0 边沿计数）	高电平	向下计数	向上计数	-	-
	低电平	向上计数	向下计数	-	-
编码模式 3（在所有通道边沿计数）	高电平	向下计数	向上计数	向上计数	向下计数
	低电平	向上计数	向下计数	向下计数	向上计数

在编码模式下，计数器同样是在 0-ARR 之间计数，向上计数到 ARR 时会产生 overflow，然后回到 0 重新计数，向下计数到 0 时会产生 underflow，然后回到 ARR 重新计数。此外，在该模式下，输入捕获（通道 2 和通道 3）、输出比较、分频、触发输出功能依然适用。编码模式 1 的计数波形图如下：

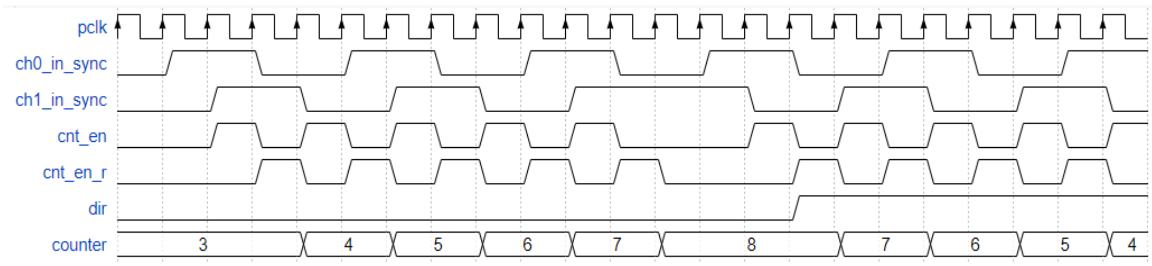

Figure 22-15 Counting waveform of encoder mode 1

22.10 Slave Mode Control

GPTIMER 支持级联操作，作为外部或内部模块的从机。从模式的触发输入信号 TRGI 有多路来源，通过 TS[2:0] 进行选择，结构如上图，其中 ITRx 来自于内部其他 GPTIMER 的触发输出信号（TRGO），具体映射关系见下表。

Table 22-6 GPTIMER internal trigger input mapping

从机 GPTIMER	ITR0	ITR1	ITR2
GPTIMER0	GPTIMER2	GPTIMER3	GPTIMER1
GPTIMER1	GPTIMER0	GPTIMER3	GPTIMER2
GPTIMER2	GPTIMER3	GPTIMER0	-
GPTIMER3	GPTIMER1	GPTIMER2	-

从模式控制主要有以下四种方式：

1. **复位模式**：TRGI 的上升沿将会初始化计数器和分频计数器，并且可以更新影子寄存器（UDIS=0 时），从模式下复位模式波形图如下：

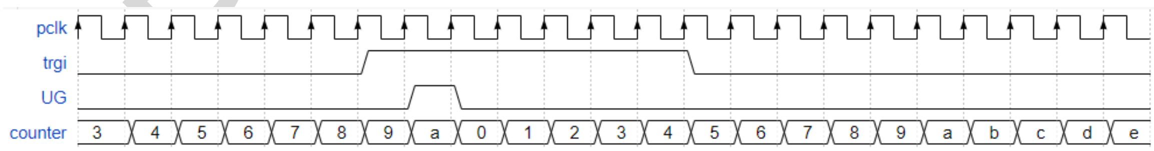

Figure 22-16 Reset mode waveform in slave mode

2. **门控模式**：TRGI 电平可以控制计数器的运行和停止，默认有效电平下，高电平时计数器计数，低电平时计数器停止计数（不是复位），在该模式下，CEN 需要软件置位。从模式下门控模式波形图如下：

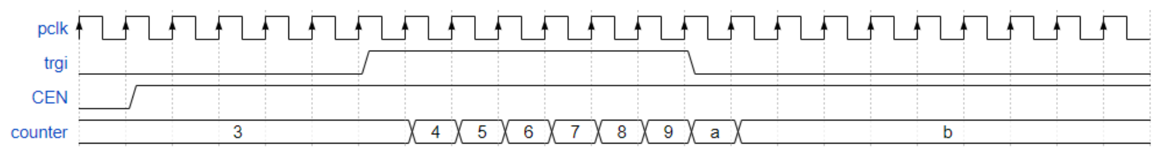

Figure 22-17 Gated mode waveform in slave mode

3. **触发模式**：TRGI 的上升沿可以控制计数器开始计数，但是无法控制计数器是否停止，在该模式下，CEN 不需要软件置位。从模式下触发模式波形图如下：

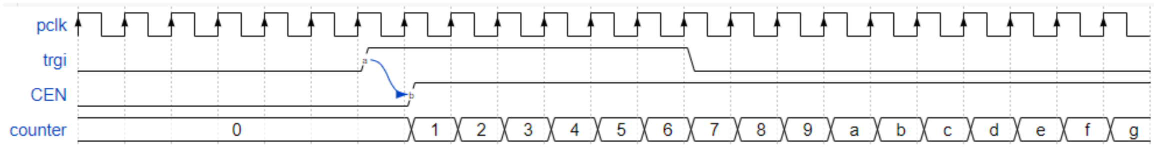

Figure 22-18 Trigger mode waveform in slave mode

4. **时钟模式（即外部时钟模式 1）**：TRGI 的上升沿作为计数器的计数使能控制计数，此时分频电路依然有效。

在从模式下，TRGI 的上升沿会置位 TIF 标志位，如果使能了相应中断或 DMA 屏蔽位，则会产生中断或 DMA 请求。但是门控模式有一些特殊，在该模式下，除上升沿外，下降沿也可以置位 TIF。

另外使用 GPTIMER 级联时，需要保证主从的时钟同频同相，否则会发生未知错误。

22.11 Master mode control

GPTIMER 也可以作为主模式使用，通过产生触发输出信号（TRGO）来控制其他 GPTIMER 或 ADC 和 DAC。TRGO 信号的来源可以由软件配置，具体如下：

- MMS=3'b000：复位模式，此时 UG 标志位将作为 TRGO 信号输出给外部从机。
- MMS=3'b001：使能模式，此时计数器的计数使能将作为 TRGO 信号输出给外部从机。如果当前 GPTIMER 同时处于从机门控模式，则该信号为门控信号，否则直接将 CEN 作为 TRGO 信号输出。
- MMS=3'b010：更新模式，此时将更新事件作为 TRGO 信号输出。
- MMS=3'b011：通道 0 比较脉冲模式，此时如果 CC0IF 将置位，则输出一个脉冲作为 TRGO 信号，无论此时 CC0IF 是否已经置位。
- MMS=3'b100：比较模式 1，此时将 OC0REF 作为 TRGO 信号输出。
- MMS=3'b101：比较模式 2，此时将 OC1REF 作为 TRGO 信号输出。
- MMS=3'b110：比较模式 3，此时将 OC2REF 作为 TRGO 信号输出。
- MMS=3'b111：比较模式 4，此时将 OC3REF 作为 TRGO 信号输出。

注意：后 *4* 种模式输出的信号 *OCxREF*，并不是最终的通道输出，而是内部信号。

GPTIMER 配置为主机使能模式时，有一种特殊应用，即同步启动主机和从机的计数器。但是因为主机的 CEN 作为 TRGO 输出到从机并使能从机计数器需要两个时钟的延迟（假定主从时钟同频同相），因此在使用这一功能时，内部会把主机的 CEN 信号用两级寄存器延迟 2 个时钟周期，以保证同步，该功能可以软件配置是否使能（MSM）。

22.12 Output control

GPTIMER0 和 GPTIMER1 共 4 路通道输出，GPTIMER2 和 GPTIMER3 共 2 路通道输出，同时有相应的输出使能信号，通道输出仅在 CCxE 置位时有效，此时可以通过 CCxP 控制输出极性，输出极性指输出有效电平为高电平还是低电平。输出使能信号为高有效，即在 CCxE 置位时有效，同时需保证通道被正确配置为输出模式，输出模式通过 CCxS 配置。

22.13 Channels remapping

通道重映射就是把 GPTIMER 的通道或外部触发通道 ETR 的输入信号从其他外部或内部信号映射过来。GPTIMER0 的 ETR 通道、通道 0 和通道 3 支持重映射，GPTIMER0 的通道 2 支持重映射，GPTIMER2 的 ETR 通道、通道 0 和通道 1 支持重映射，GPTIMER3 的 ETR 通道、通道 0 支持重映射。

22.14 Debug mode control

GPTIMER 可由软件配置 debug 下是否停止计数，如果使能该功能，则进入系统 debug 模式时，GPTIMER 停止计数（计数器不会初始化）。

22.15 DMA

GPTIMER 共有 6 个 DMA 请求源，分别是 update 事件（UIF）、4 路通道事件（捕获事件、比较匹配）（CCxIF）以及触发事件（TIF），可以由独立的屏蔽位配置是否使能相应的 DMA 请求。对于通道事件 DMA，可以软件（CCDS 位）配置通道的 DMA 请求源，若 CCDS=0，各通道 DMA 请求来自于各通道的事件，如捕获、比较匹配事件；若 CCDS=1，则各通道的 DMA 请求均来自于更新事件，通道事件将被屏蔽。

各 DMA 请求仅在无相应应答信号、DMA 使能开启且 DMA 事件发生时置位，在 DMA 请求置位时，应答信号可以清除 DMA 请求，否则 DMA 请求将一直保持置位状态。

除常规的 DMA 操作外，GPTIMER 还支持 burst 功能，即一个 DMA 请求可以连续读写多个内部寄存器。DBL 位可以选择 burst 长度，最多 18 个，DBA 可以选择 burst 的起始地址，DMAR 寄存器的地址可以作为 DMA 的目标地址或源地址（DMA 内部不需要设置每次递增）。当某一个 DMA 请求置位时，GPTIMER 根据 DBL 和 DBA 的值，计算出每一次读写操作的实际地址，实际地址计算方法为：*CR1 + (DBA + index) x 4*，其中 index 的值为 0 至 DBL。

注意： *寄存器组中间有保留寄存器地址，该地址也将包含在 DMA 的 burst 操作中，实际使用时需注意配置的长度。例如，起始地址选择 ARR 寄存器（0x2C），DBL 配置为 5'b00010（3 个 burst），则 DMA 实际操作的三个寄存器分别是 0x2C、 0x30、 0x34，其中 0x30 为保留寄存器，因此无法写入且读出永远为 0。*

22.16 Interrupts

GPTIMER 共有 6 个中断源，分别是 update 事件（UIF）、4 路通道事件（捕获事件、比较匹配）（CCxIF）以及触发事件（TIF），各中断可以由独立的中断屏蔽位选择是否使能，中断标志位与相应屏蔽位是 AND 的关系，中断之间是 OR 的关系。GPTIMER 的中断信号如下表：

Table 22-7 GPTIMER interrupts

中断名称	描述
触发事件中断	触发源产生事件时的中断
通道 3 事件中断	通道 3 产生捕获或比较事件时的中断
通道 2 事件中断	通道 2 产生捕获或比较事件时的中断
通道 1 事件中断	通道 1 产生捕获或比较事件时的中断
通道 0 事件中断	通道 0 产生捕获或比较事件时的中断
更新事件中断	产生更新事件时的中断

上述中断的使能分别通过配置寄存器 DIER 的 TIE、CC3IE、CC2IE、CC1IE、CC0IE、UIE 位实现。

22.17 GPTIMER registers

GPTIMER0 基地址：0x4000A000

GPTIMER1 基地址：0x4001A000

GPTIMER2 基地址：0x4000B000

GPTIMER3 基地址：0x4001B000

Table 22-8 GPTIMER Registers Summary

寄存器	偏移量	描述
GPTIM_CR1	0x00	控制寄存器 1
GPTIM_CR2	0x04	控制寄存器 2
GPTIM_SMCR	0x08	从模式控制寄存器
GPTIM_DIER	0x0C	DMA/中断使能寄存器
GPTIM_SR	0x10	状态寄存器
GPTIM_EGR	0x14	事件寄存器
GPTIM_CCMR1	0x18	捕获比较模式寄存器 1
GPTIM_CCMR2	0x1C	捕获比较模式寄存器 2
GPTIM_CCER	0x20	捕获比较使能寄存器
GPTIM_CNT	0x24	计数寄存器

GPTIM_PSC	0x28	计数器分频值寄存器
GPTIM_ARR	0x2C	计数器重装载值寄存器
GPTIM_CCR0	0x34	通道 0 捕获比较寄存器
GPTIM_CCR1	0x38	通道 1 捕获比较寄存器
GPTIM_CCR2	0x3C	通道 2 捕获比较寄存器
GPTIM_CCR3	0x40	通道 3 捕获比较寄存器
GPTIM_DCR	0x48	DMA 控制寄存器
GPTIM_DMAR	0x4C	DMA 地址寄存器
GPTIM_OR	0x50	通道重映射寄存器

22.17.1 GPTIM_CR1

偏移量：0x00

复位值：0x0000

15-10	9-8	7	6-5	4	3	2	1	0
RESERVED	CKD	ARPE	CMS	DIR	OPM	URS	UDIS	CEN
r-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 15-10 RESERVED：保留。

位 9-8 CKD：采样时钟分频。

- 00：fDTS = fpclk
- 01：fDTS = fpclk
- 10：fDTS = fpclk
- 11：fDTS = reserved

位 7 ARPE：重装载影子寄存器使能。

- 0：ARR 影子寄存器除能
- 1：ARR 影子寄存器使能

位 6-5 CMS：中间计数模式选择。

- 00：边沿对齐计数模式，DIR 控制向上或向下计数
- 01：中间对齐模式 1。输出比较中断标志位仅在向下计数过程中置位
- 10：中间对齐模式 2。输出比较中断标志位仅在向上计数过程中置位
- 11：中间对齐模式 3。输出比较中断标志位在向上和向下计数过程中均置位

位 4 DIR：计数方向选择。中间对齐模式和编码模式，该位由硬件控制。

- 0：向上计数
- 1：向下计数

位 3 OPM：单脉冲模式使能。

- 0：单脉冲模式除能
- 1：单脉冲模式使能，计数器在下一次更新事件停止计数

位 2 URS：更新事件源选择，该位仅影响中断和 DMA 标志位（UIF），不影响内部逻辑。

- 0：计数器溢出、UG 位置位、从模式 reset 模式下的触发，均可以置位 UIF
- 1：只有计数器溢出事件可以置位 UIF

位 1 UDIS：更新事件除能。

- 0：更新事件使能，中间对齐模式 1。输出比较中断标志位仅在向下计数过程中置位均可以产生更新事件
- 1：更新事件除能，影子寄存器和 UIF 均不会被更新，但是此时计数器和分频计数器仍可以被 UG 置位事件初始化

位 0 CEN：计数器使能，触发模式下 CEN 由硬件置位，单脉冲模式下 CEN 由硬件清零。

- 0：计数器除能
- 1：计数器使能

22.17.2 GPTIM_CR2

偏移量：0x04

复位值：0x0000

15-8	7	6-4	3	2-0
RESERVED	TI0S	MMS	CCDS	RESERVED
r-0h	rw-0h	rw-0h	rw-0h	r-0h

位 15-8 RESERVED：保留。

位 7 TI0S：通道 1 源异或选择（该功能仅 timer0 和 timer1 支持）。

- 0：通道 0 映射到通道 0 输入
- 1：通道 0 为通道 0、1、2 的异或输出

位 6-4 MMS：主模式选择，可以配置 TRGO 输出。

- 000：复位模式，UG 将作为 TRGO 信号输出
- 001：使能模式，CNT_EN（不是 CEN）将作为 TRGO 信号输出
- 010：更新模式，更新事件（内部信号）将作为 TRGO 信号输出
- 011：比较脉冲模式，每次 CC0IF 将要置位时 TRGO 会输出一个脉冲，即使 CC0IF 已经置位
- 100：比较模式，OC0REF（内部信号）作为 TRGO 信号输出
- 101：比较模式，OC1REF（内部信号）作为 TRGO 信号输出
- 110：比较模式，OC2REF（内部信号）作为 TRGO 信号输出
- 111：比较模式，OC3REF（内部信号）作为 TRGO 信号输出

位 3 CCDS：通道 DMA 请求源选择（该功能仅 gptimer0 和 gptimer1 支持）。

- 0：各通道的 DMA 请求（不包含更新事件请求和触发事件请求）由通道事件（捕获、比较）产生
- 1：各通道的 DMA 请求（不包含更新事件请求和触发事件请求）由更新事件产生

位 2-0 RESERVED：保留。

22.17.3 GPTIM_SMCR

偏移量：0x08

复位值：0x0000

15	14	13-12	11-8	7	6-4	3	2-0
ETP	ECE	ETPS	ETF	MSM	TS	RESERVED	SMS
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	r-0h	rw-0h

位 15 ETP：外部触发极性选择（配置极性时最好先不要选择模式（SMS），以防内部信号翻转触发未知错误）。

- 0：外部触发输入不反相
- 1：外部触发输入反相

位 14 ECE：外部时钟模式 2 使能。

- 0：禁用外部时钟模式 2
- 1：使能外部时钟模式 2

位 13-12 ETPS：外部触发输入分频（该分频主要用于 50%占空比降频，如 24M 信号 2 分频为 12M，电平延展一倍）择。

- 00：不分频
- 01：2 分频
- 10：4 分频
- 11：8 分频

位 11-8 ETF：外部触发输入滤波器配置。

- 0000：禁用滤波器
- 0001：滤波器采样频率 fsampling=fpclk，滤波器长度 N=2
- 0010：滤波器采样频率 fsampling=fpclk，滤波器长度 N=4
- 0011：滤波器采样频率 fsampling=fpclk，滤波器长度 N=8
- 0100：滤波器采样频率 fsampling=fDTS/2，滤波器长度 N=6
- 0101：滤波器采样频率 fsampling=fDTS/2，滤波器长度 N=8
- 0110：滤波器采样频率 fsampling=fDTS/4，滤波器长度 N=6
- 0111：滤波器采样频率 fsampling=fDTS/4，滤波器长度 N=8
- 1000：滤波器采样频率 fsampling=fDTS/8，滤波器长度 N=6
- 1001：滤波器采样频率 fsampling=fDTS/8，滤波器长度 N=8
- 1010：滤波器采样频率 fsampling=fDTS/16，滤波器长度 N=5
- 1011：滤波器采样频率 fsampling=fDTS/16，滤波器长度 N=6
- 1100：滤波器采样频率 fsampling=fDTS/16，滤波器长度 N=8
- 1101：滤波器采样频率 fsampling=fDTS/32，滤波器长度 N=5
- 1110：滤波器采样频率 fsampling=fDTS/32，滤波器长度 N=6
- 1111：滤波器采样频率 fsampling=fDTS/32，滤波器长度 N=8

位 7 MSM：主从模式同步（使用该功能时，需保证两个 timer 的时钟同频同相）。

- 0：无动作
- 1：TRGI 触发输入将延迟产生作用，以便与从计数器同时开始计数

位 6-4 TS：触发源选择，选择 TRGI 的来源（配置该位时 SMS 必须处于清零状态）。

- 000：ITR0
- 001：ITR1
- 010：ITR2（timer2 和 timer3 无此通道）
- 011：保留
- 100：通道 0 边沿检测输出
- 101：通道 0 滤波器输出
- 110：通道 1 滤波器输出
- 111：外部触发输入

位 3 RESERVED：保留。

位 2-0 SMS：从模式选择（选择模式前最好先配置好通道参数，以防内部信号翻转触发未知错误）。

- 000：禁用从模式
- 001：编码模式 1，计数器仅在通道 1 边沿计数
- 010：编码模式 2，计数器仅在通道 0 边沿计数
- 011：编码模式 3，计数器在通道 0 和 1 的边沿计数
- 100：复位模式，TRGI 的上升沿将复位计数器
- 101：门控模式，计数器仅在 TRGI 高电平期间计数
- 110：触发模式，计数器在 TRGI 上升沿将开始计数，该模式仅控制计数的开始
- 111：外部时钟模式 1，TRGI 的上升沿作为计数器计数时钟

22.17.4 GPTIM_DIER

偏移量：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8
RESERVED	TDE	RESERVED	CC3DE	CC2DE	CC1DE	CC0DE	UDE
r-0h	rw-0h	r-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED	TIE	RESERVED	CC3IE	CC2IE	CC1IE	CC0IE	UIE
r-0h	rw-0h	r-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 15 RESERVED：保留。

位 14 TDE：触发事件 DMA 请求使能。

- 0：禁用触发事件 DMA 请求
- 1：使能触发事件 DMA 请求

位 13 RESERVED：保留。

位 12 CC3DE：通道 3 事件 DMA 请求使能。

- 0：禁用通道 3 事件 DMA 请求
- 1：使能通道 3 事件 DMA 请求

位 11 CC3DE：通道 2 事件 DMA 请求使能。

- 0：禁用通道 2 事件 DMA 请求
- 1：使能通道 2 事件 DMA 请求

位 10 CC3DE： 通道 1 事件 DMA 请求使能。

- 0：禁用通道 1 事件 DMA 请求
- 1：使能通道 1 事件 DMA 请求

位 9 CC3DE： 通道 0 事件 DMA 请求使能。

- 0：禁用通道 0 事件 DMA 请求
- 1：使能通道 0 事件 DMA 请求

位 8 UDE： 更新事件 DMA 请求使能。

- 0：禁用更新事件 DMA 请求
- 1：使能更新事件 DMA 请求

位 7 RESERVED： 保留。

位 6 TIE： 触发事件中断请求使能。

- 0：禁用触发事件中断请求
- 1：使能触发事件中断请求

位 5 RESERVED： 保留。

位 4 CC3IE： 通道 3 事件中断请求使能。

- 0：禁用通道 3 事件中断请求
- 1：使能通道 3 事件中断请求

位 3 CC2IE： 通道 2 事件中断请求使能。

- 0：禁用通道 2 事件中断请求
- 1：使能通道 2 事件中断请求

位 2 CC1IE： 通道 1 事件中断请求使能。

- 0：禁用通道 1 事件中断请求
- 1：使能通道 1 事件中断请求

位 1 CC0IE： 通道 0 事件中断请求使能。

- 0：禁用通道 0 事件中断请求
- 1：使能通道 0 事件中断请求

位 0 UIE： 通道 0 事件中断请求使能。

- 0：禁用通道 0 事件中断请求
- 1：使能通道 0 事件中断请求

22.17.5 GPTIM_SR

偏移量：0x10

复位值：0x0000

15-13		12	11	10	9	8-7
RESERVED		CC3OF	CC2OF	CC1OF	CC0OF	RESERVED
r-0h		rw-0h	rw-0h	rw-0h	rw-0h	r-0h
6	**5**	**4**	**3**	**2**	**1**	**0**
TIF	RESERVED	CC3IF	CC2IF	CC1IF	CC0IF	UIF
rw-0h	r-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 15-13 RESERVED：保留。

位 12 CC3OF：通道 3 overcapture 标志（写 0 清零）。

- 0：无 overcapture
- 1：发生了至少 1 次 overcapture

位 11 CC2OF：通道 2 overcapture 标志（写 0 清零）。

- 0：无 overcapture
- 1：发生了至少 1 次 overcapture

位 10 CC1OF：通道 1 overcapture 标志（写 0 清零）。

- 0：无 overcapture
- 1：发生了至少 1 次 overcapture

位 9 CC0OF：通道 0 overcapture 标志（写 0 清零）。

- 0：无 overcapture
- 1：发生了至少 1 次 overcapture

位 8-7 RESERVED：保留。

位 6 TIF：触发事件中断标志（写 0 清零）。

- 0：无触发事件
- 1：触发事件发生

位 5 RESERVED：保留。

位 4 CC3IF：通道 3 捕获/比较事件标志（比较模式：写 0 清零；捕获模式：读 ccrx 寄存器或写 0 均可清零）。

- 0：无事件
- 1：捕获或比较事件发生

位 3 CC3IF：通道 2 捕获/比较事件标志（比较模式：写 0 清零；捕获模式：读 ccrx 寄存器或写 0 均可清零）。

- 0：无事件
- 1：捕获或比较事件发生

位 2 CC3IF：通道 1 捕获/比较事件标志（比较模式：写 0 清零；捕获模式：读 ccrx 寄存器或写 0 均可清零）。

- 0：无事件

- 1：捕获或比较事件发生

位 1 CC3IF：通道 0 捕获/比较事件标志（比较模式：写 0 清零；捕获模式：读 ccrx 寄存器或写 0 均可清零）。

- 0：无事件
- 1：捕获或比较事件发生

位 0 UIF：更新事件标志（读 SR 或写 0 可清零该位）。

- 0：无事件
- 1：更新事件发生

22.17.6 GPTIM_EGR

偏移量：0x14

复位值：0x0000

15-7	6	5	4	3	2	1	0
RESERVED	TG	RESERVED	CC3G	CC2G	CC1G	CC0G	UG
r-0h	w-0h	r-0h	w-0h	w-0h	w-0h	w-0h	w-0h

位 15-7 RESERVED：保留。

位 6 TG：触发产生。

- 0：无动作
- 1：产生一次触发事件，TIF 置位

位 5 RESERVED：保留。

位 4 CC3G：通道 3 事件产生。

- 0：无动作
- 1：输入模式时产生捕获动作，输出模式时产生比较动作，两种模式下 CC3IF 置位

位 3 CC2G：通道 2 事件产生。

- 0：无动作
- 1：输入模式时产生捕获动作，输出模式时产生比较动作，两种模式下 CC2IF 置位

位 2 CC1G：通道 1 事件产生。

- 0：无动作
- 1：输入模式时产生捕获动作，输出模式时产生比较动作，两种模式下 CC1IF 置位

位 1 CC0G：通道 0 事件产生。

- 0：无动作
- 1：输入模式时产生捕获动作，输出模式时产生比较动作，两种模式下 CC0IF 置位

位 0 UG：更新事件产生。

- 0：无动作
- 1：产生一次更新事件

22.17.7 GPTIM_CCMR1

偏移量：0x18

复位值：0x0000

输出模式时结构如下：

15	14-12	11	10	9-8	7	6-4	3	2	1-0
OC1CE	OC1M	OC1PE	OC1FE	CC1S	OC0CE	OC0M	OC0PE	OC0FE	CC0S
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 15 OC1CE：通道 1 输出比较清除使能。

- 0：禁用清除功能
- 1：使能清除功能，ETRF 高电平可以清除通道输出

位 14-12 OC1M：通道 1 输出比较模式选择。

- 000：冻结模式，通道输出不随比较结果变化
- 001：有效模式，匹配后通道输出有效电平
- 010：失效模式，匹配后通道输出失效电平
- 011：翻转模式，匹配后将翻转通道输出
- 100：强制有效模式，选择该模式后，直接输出有效电平
- 101：强制失效模式，选择该模式后，直接输出失效电平
- 110：PWM1 模式，该模式下，向上计数时，CNT<CCR 时通道输出有效电平，否则输出失效电平；向下计数时，CNT>CCR 时通道输出失效电平，否则输出有效电平（向上计数时，若 CCRx>ARR，OCxREF 一直输出高电平，若 CCRx==0，OCxREF 一直输出低电平；向下计数时，若 CCRx>ARR，OCxREF 一直输出高电平，此时 0% PWM 不支持）
- 111：PWM2 模式，该模式下，向上计数时，CNT<CCR 时通道输出失效电平，否则输出有效电平；向下计数时，CNT>CCR 时通道输出有效电平，否则输出失效电平（0%与 100%波形与 PWM1 同理）

位 11 OC1PE：通道 1 输出比较影子寄存器使能。

- 0：禁用影子寄存器
- 1：使能影子寄存器

位 10 OC1FE：通道 1 快速输出使能。

- 0：禁用快速模式，输出仅在匹配时变化
- 1：使能快速模式，触发输入相当于匹配事件，直接影响通道输出，不受计数器与 CCR 的比较影响

位 9-8 CC1S：捕获比较选择。

- 00：通道配置为 输出模式
- 01：通道配置为输入模式，捕获通道输入映射至通道 1
- 10：通道配置为输入模式，捕获通道输入映射至通道 0
- 11：通道配置为输入模式，捕获通道输入映射至触发输入 TRC

位 7 OC0CE：通道 0 输出比较清除使能。

- 0：禁用清除功能
- 1：使能清除功能，ETRF 高电平可以清除通道输出

位 6-4 OC0M：通道 0 输出比较模式选择。

- 000：冻结模式，通道输出不随比较结果变化
- 001：有效模式，匹配后通道输出有效电平
- 010：失效模式，匹配后通道输出失效电平
- 011：翻转模式，匹配后将翻转通道输出
- 100：强制有效模式，选择该模式后，直接输出有效电平
- 101：强制失效模式，选择该模式后，直接输出失效电平
- 110：PWM1 模式，该模式下，向上计数时，CNT<CCR 时通道输出有效电平，否则输出失效电平；向下计数时，CNT>CCR 时通道输出失效电平，否则输出有效电平（向上计数时，若 CCRx>ARR，OCxREF 一直输出高电平，若 CCRx==0，OCxREF 一直输出低电平；向下计数时，若 CCRx>ARR，OCxREF 一直输出高电平，此时 0% PWM 不支持）
- 111：PWM2 模式，该模式下，向上计数时，CNT<CCR 时通道输出失效电平，否则输出有效电平；向下计数时，CNT>CCR 时通道输出有效电平，否则输出失效电平（0%与 100%波形与 PWM1 同理）

位 3 OC0PE：通道 0 输出比较影子寄存器使能。

- 0：禁用影子寄存器
- 1：使能影子寄存器

位 2 OC0FE：通道 0 快速输出使能。

- 0：禁用快速模式，输出仅在匹配时变化
- 1：使能快速模式，触发输入相当于匹配事件，直接影响通道输出，不受计数器与 CCR 的比较影响

位 1-0 CC0S：捕获比较选择。

- 00：通道配置为 输出模式
- 01：通道配置为输入模式，捕获通道输入映射至通道 0
- 10：通道配置为输入模式，捕获通道输入映射至通道 1
- 11：通道配置为输入模式，捕获通道输入映射至触发输入 TRC

输入模式时结构如下：

15-12	11-10	9-8	7-4	3-2	1-0
IC1F	IC1PSC	CC1S	IC0F	IC0PSC	CC0S
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 15-12 IC1F：通道 1 输入滤波器配置（需配置 CCxS！=0x0 该功能才能生效）。

- 0000：禁用滤波器
- 0001：滤波器采样频率 fsampling=fpclk，滤波器长度 N=2
- 0010：滤波器采样频率 fsampling=fpclk，滤波器长度 N=4
- 0011：滤波器采样频率 fsampling=fpclk，滤波器长度 N=8
- 0100：滤波器采样频率 fsampling=fDTS/2，滤波器长度 N=6
- 0101：滤波器采样频率 fsampling=fDTS/2，滤波器长度 N=8
- 0110：滤波器采样频率 fsampling=fDTS/4，滤波器长度 N=6
- 0111：滤波器采样频率 fsampling=fDTS/4，滤波器长度 N=8
- 1000：滤波器采样频率 fsampling=fDTS/8，滤波器长度 N=6
- 1001：滤波器采样频率 fsampling=fDTS/8，滤波器长度 N=8
- 1010：滤波器采样频率 fsampling=fDTS/16，滤波器长度 N=5**（未完，接下页）**

- 1011：滤波器采样频率 fsampling=fDTS/16， 滤波器长度 N=6
- 1100：滤波器采样频率 fsampling=fDTS/16， 滤波器长度 N=8
- 1101：滤波器采样频率 fsampling=fDTS/32， 滤波器长度 N=5
- 1110：滤波器采样频率 fsampling=fDTS/32， 滤波器长度 N=6
- 1111：滤波器采样频率 fsampling=fDTS/32， 滤波器长度 N=8

位 11-10 IC1PSC：通道 1 分频（需配置 CCxS！=0x0 该功能才能生效）。

- 00：不分频
- 01：2 分频
- 10：4 分频
- 11：8 分频

位 9-8 CC1S：捕获比较选择。

- 00：通道配置为 输出模式
- 01：通道配置为输入模式，捕获通道输入映射至通道 1
- 10：通道配置为输入模式，捕获通道输入映射至通道 0
- 11：通道配置为输入模式，捕获通道输入映射至触发输入 TRC

位 7-4 IC0F：通道 0 输入滤波器配置（需配置 CCxS！=0x0 该功能才能生效）。

- 0000：禁用滤波器
- 0001：滤波器采样频率 fsampling=fpclk， 滤波器长度 N=2
- 0010：滤波器采样频率 fsampling=fpclk， 滤波器长度 N=4
- 0011：滤波器采样频率 fsampling=fpclk， 滤波器长度 N=8
- 0100：滤波器采样频率 fsampling=fDTS/2， 滤波器长度 N=6
- 0101：滤波器采样频率 fsampling=fDTS/2， 滤波器长度 N=8
- 0110：滤波器采样频率 fsampling=fDTS/4， 滤波器长度 N=6
- 0111：滤波器采样频率 fsampling=fDTS/4， 滤波器长度 N=8
- 1000：滤波器采样频率 fsampling=fDTS/8， 滤波器长度 N=6
- 1001：滤波器采样频率 fsampling=fDTS/8， 滤波器长度 N=8
- 1010：滤波器采样频率 fsampling=fDTS/16， 滤波器长度 N=5
- 1011：滤波器采样频率 fsampling=fDTS/16， 滤波器长度 N=6
- 1100：滤波器采样频率 fsampling=fDTS/16， 滤波器长度 N=8
- 1101：滤波器采样频率 fsampling=fDTS/32， 滤波器长度 N=5
- 1110：滤波器采样频率 fsampling=fDTS/32， 滤波器长度 N=6
- 1111：滤波器采样频率 fsampling=fDTS/32， 滤波器长度 N=8

位 3-2 IC0PSC：通道 0 分频（需配置 CCxS！=0x0 该功能才能生效）。

- 00：不分频
- 01：2 分频
- 10：4 分频
- 11：8 分频

位 1-0 CC0S：捕获比较选择。

- 00：通道配置为 输出模式
- 01：通道配置为输入模式，捕获通道输入映射至通道 0
- 10：通道配置为输入模式，捕获通道输入映射至通道 1
- 11：通道配置为输入模式，捕获通道输入映射至触发输入 TRC

22.17.8 GPTIM_CCMR2

偏移量：0x1C

复位值：0x0000

输出模式时结构如下：

15	14-12	11	10	9-8	7	6-4	3	2	1-0
OC3CE	OC3M	OC3PE	OC3FE	CC3S	OC2CE	OC2M	OC2PE	OC2FE	CC2S
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 15 OC3CE：通道 3 输出比较清除使能。

- 0：禁用清除功能
- 1：使能清除功能，ETRF 高电平可以清除通道输出

位 14-12 OC3M：通道 3 输出比较模式选择。

- 000：冻结模式，通道输出不随比较结果变化
- 001：有效模式，匹配后通道输出有效电平
- 010：失效模式，匹配后通道输出失效电平
- 011：翻转模式，匹配后将翻转通道输出
- 100：强制有效模式，选择该模式后，直接输出有效电平
- 101：强制失效模式，选择该模式后，直接输出失效电平
- 110：PWM1 模式，该模式下，向上计数时，CNT<CCR 时通道输出有效电平，否则输出失效电平；向下计数时，CNT>CCR 时通道输出失效电平，否则输出有效电平（向上计数时，若 CCRx>ARR，OCxREF 一直输出高电平，若 CCRx==0，OCxREF 一直输出低电平；向下计数时，若 CCRx>ARR，OCxREF 一直输出高电平，此时 0% PWM 不支持）
- 111：PWM2 模式，该模式下，向上计数时，CNT<CCR 时通道输出失效电平，否则输出有效电平；向下计数时，CNT>CCR 时通道输出有效电平，否则输出失效电平（0%与 100%波形与 PWM1 同理）

位 11 OC3PE：通道 3 输出比较影子寄存器使能。

- 0：禁用影子寄存器
- 1：使能影子寄存器

位 10 OC3FE：通道 3 快速输出使能。

- 0：禁用快速模式，输出仅在匹配时变化
- 1：使能快速模式，触发输入相当于匹配事件，直接影响通道输出，不受计数器与 CCR 的比较影响

位 9-8 CC3S：捕获比较选择。

- 00：通道配置为 输出模式
- 01：通道配置为输入模式，捕获通道输入映射至通道 3
- 10：通道配置为输入模式，捕获通道输入映射至通道 2
- 11：通道配置为输入模式，捕获通道输入映射至触发输入 TRC

位 7 OC2CE：通道 2 输出比较清除使能。

- 0：禁用清除功能
- 1：使能清除功能，ETRF 高电平可以清除通道输出

位 6-4 OC2M：通道 2 输出比较模式选择。

- 000：冻结模式，通道输出不随比较结果变化
- 001：有效模式，匹配后通道输出有效电平
- 010：失效模式，匹配后通道输出失效电平
- 011：翻转模式，匹配后将翻转通道输出
- 100：强制有效模式，选择该模式后，直接输出有效电平
- 101：强制失效模式，选择该模式后，直接输出失效电平
- 110：PWM1 模式，该模式下，向上计数时，CNT<CCR 时通道输出有效电平，否则输出失效电平；向下计数时，CNT>CCR 时通道输出失效电平，否则输出有效电平（向上计数时，若 CCRx>ARR，OCxREF 一直输出高电平，若 CCRx==0，OCxREF 一直输出低电平；向下计数时，若 CCRx>ARR，OCxREF 一直输出高电平，此时 0% PWM 不支持）
- 111：PWM2 模式，该模式下，向上计数时，CNT<CCR 时通道输出失效电平，否则输出有效电平；向下计数时，CNT>CCR 时通道输出有效电平，否则输出失效电平（0%与 100%波形与 PWM1 同理）

位 3 OC2PE：通道 2 输出比较影子寄存器使能。

- 0：禁用影子寄存器
- 1：使能影子寄存器

位 2 OC2FE：通道 2 快速输出使能。

- 0：禁用快速模式，输出仅在匹配时变化
- 1：使能快速模式，触发输入相当于匹配事件，直接影响通道输出，不受计数器与 CCR 的比较影响

位 1-0 CC2S：捕获比较选择。

- 00：通道配置为输出模式
- 01：通道配置为输入模式，捕获通道输入映射至通道 2
- 10：通道配置为输入模式，捕获通道输入映射至通道 3
- 11：通道配置为输入模式，捕获通道输入映射至触发输入 TRC

输入模式时结构如下：

15-12	11-10	9-8	7-4	3-2	1-0
IC3F	IC3PSC	CC3S	IC2F	IC2PSC	CC2S
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 15-12 IC3F：通道 3 输入滤波器配置（需配置 CCxS！=0x0 该功能才能生效）。

- 0000：禁用滤波器
- 0001：滤波器采样频率 fsampling=fpclk， 滤波器长度 N=2
- 0010：滤波器采样频率 fsampling=fpclk， 滤波器长度 N=4
- 0011：滤波器采样频率 fsampling=fpclk， 滤波器长度 N=8
- 0100：滤波器采样频率 fsampling=fDTS/2， 滤波器长度 N=6
- 0101：滤波器采样频率 fsampling=fDTS/2， 滤波器长度 N=8
- 0110：滤波器采样频率 fsampling=fDTS/4， 滤波器长度 N=6
- 0111：滤波器采样频率 fsampling=fDTS/4， 滤波器长度 N=8
- 1000：滤波器采样频率 fsampling=fDTS/8， 滤波器长度 N=6
- 1001：滤波器采样频率 fsampling=fDTS/8， 滤波器长度 N=8
- 1010：滤波器采样频率 fsampling=fDTS/16， 滤波器长度 N=5 **（未完，接下页）**

- 1011：滤波器采样频率 fsampling=fDTS/16， 滤波器长度 N=6
- 1100：滤波器采样频率 fsampling=fDTS/16， 滤波器长度 N=8
- 1101：滤波器采样频率 fsampling=fDTS/32， 滤波器长度 N=5
- 1110：滤波器采样频率 fsampling=fDTS/32， 滤波器长度 N=6
- 1111：滤波器采样频率 fsampling=fDTS/32， 滤波器长度 N=8

位 11-10 IC3PSC：通道 3 分频（需配置 CCxS！=0x0 该功能才能生效）。

- 00：不分频
- 01：2 分频
- 10：4 分频
- 11：8 分频

位 9-8 CC3S：捕获比较选择。

- 00：通道配置为 输出模式
- 01：通道配置为输入模式，捕获通道输入映射至通道 3
- 10：通道配置为输入模式，捕获通道输入映射至通道 2
- 11：通道配置为输入模式，捕获通道输入映射至触发输入 TRC

位 7-4 IC2F：通道 2 输入滤波器配置（需配置 CCxS！=0x0 该功能才能生效）。

- 0000：禁用滤波器
- 0001：滤波器采样频率 fsampling=fpclk， 滤波器长度 N=2
- 0010：滤波器采样频率 fsampling=fpclk， 滤波器长度 N=4
- 0011：滤波器采样频率 fsampling=fpclk， 滤波器长度 N=8
- 0100：滤波器采样频率 fsampling=fDTS/2， 滤波器长度 N=6
- 0101：滤波器采样频率 fsampling=fDTS/2， 滤波器长度 N=8
- 0110：滤波器采样频率 fsampling=fDTS/4， 滤波器长度 N=6
- 0111：滤波器采样频率 fsampling=fDTS/4， 滤波器长度 N=8
- 1000：滤波器采样频率 fsampling=fDTS/8， 滤波器长度 N=6
- 1001：滤波器采样频率 fsampling=fDTS/8， 滤波器长度 N=8
- 1010：滤波器采样频率 fsampling=fDTS/16， 滤波器长度 N=5
- 1011：滤波器采样频率 fsampling=fDTS/16， 滤波器长度 N=6
- 1100：滤波器采样频率 fsampling=fDTS/16， 滤波器长度 N=8
- 1101：滤波器采样频率 fsampling=fDTS/32， 滤波器长度 N=5
- 1110：滤波器采样频率 fsampling=fDTS/32， 滤波器长度 N=6
- 1111：滤波器采样频率 fsampling=fDTS/32， 滤波器长度 N=8

位 3-2 IC2PSC：通道 2 分频（需配置 CCxS！=0x0 该功能才能生效）。

- 00：不分频
- 01：2 分频
- 10：4 分频
- 11：8 分频

位 1-0 CC2S：捕获比较选择。

- 00：通道配置为 输出模式
- 01：通道配置为输入模式，捕获通道输入映射至通道 2
- 10：通道配置为输入模式，捕获通道输入映射至通道 3
- 11：通道配置为输入模式，捕获通道输入映射至触发输入 TRC

22.17.9 GPTIM_CCER

偏移量：0x20

复位值：0x0000

15	14	13	12	11	10	9	8
CC3NP	RESERVED	CC3P	CC3E	CC2NP	RESERVED	CC2P	CC2E
rw-0h	r-0h	rw-0h	rw-0h	rw-0h	r-0h	rw-0h	rw-0h
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CC1NP	RESERVED	CC1P	CC1E	CC0NP	RESERVED	CC0P	CC0E
rw-0h	r-0h	rw-0h	rw-0h	rw-0h	r-0h	rw-0h	rw-0h

位 15 CC3NP：输出反相极性，输出模式时该位必须位 0，输入模式时参看 CC3P。

位 14 RESERVED：保留。

位 13 CC3P：输出极性，须与 CC3NP 共同作用（配置极性最好在模式选择之前，防止内部信号翻转触发未知错误）。

输出模式：

- 0：输出有效极性位高电平
- 1：输出有效极性位低电平

输入模式，{CC3NP，CC3P}：

- 00：通道输入上升沿有效（捕获、触发、复位、时钟模式），高电平有效（门控、编码模式）
- 01：通道输入下降沿有效（捕获、触发、复位、时钟模式），低电平有效（门控、编码模式）
- 10：保留
- 11：通道输入上升沿和下降沿均有效（捕获、触发、复位、时钟模式），高电平有效（门控、编码模式）

位 12 CC3E：通道使能。

输入模式：

- 0：禁用捕获
- 1：使能捕获

输出模式：

- 0：禁用输出
- 1：使能输出

位 11 CC2NP：输出反相极性，输出模式时该位必须位 0，输入模式时参看 CC2P。

位 10 RESERVED：保留。

位 9 CC2P：输出极性，须与 CC2NP 共同作用（配置极性最好在模式选择之前，防止内部信号翻转触发未知错误）。

输出模式：

- 0：输出有效极性位高电平
- 1：输出有效极性位低电平

输入模式，{CC2NP，CC2P}：

- 00：通道输入上升沿有效（捕获、触发、复位、时钟模式），高电平有效（门控、编码模式）
- 01：通道输入下降沿有效（捕获、触发、复位、时钟模式），低电平有效（门控、编码模式）
- 10：保留
- 11：通道输入上升沿和下降沿均有效（捕获、触发、复位、时钟模式），高电平有效（门控、编码模式）

位 8 CC2E：通道使能。

输入模式：

- 0：禁用捕获
- 1：使能捕获

输出模式：

- 0：禁用输出
- 1：使能输出

位 7 CC1NP：输出反相极性，输出模式时该位必须位 0，输入模式时参看 CC1P。

位 6 RESERVED：保留。

位 5 CC1P：输出极性，须与 CC3NP 共同作用（配置极性最好在模式选择之前，防止内部信号翻转触发未知错误）。

输出模式：

- 0：输出有效极性位高电平
- 1：输出有效极性位低电平

输入模式，{CC1NP，CC1P}：

- 00：通道输入上升沿有效（捕获、触发、复位、时钟模式），高电平有效（门控、编码模式）
- 01：通道输入下降沿有效（捕获、触发、复位、时钟模式），低电平有效（门控、编码模式）
- 10：保留
- 11：通道输入上升沿和下降沿均有效（捕获、触发、复位、时钟模式），高电平有效（门控、编码模式）

位 4 CC1E：通道使能。

输入模式：

- 0：禁用捕获
- 1：使能捕获

输出模式：

- 0：禁用输出
- 1：使能输出

位 3 CC0NP：输出反相极性，输出模式时该位必须位 0，输入模式时参看 CC0P。

位 2 RESERVED：保留。

位 1 CC0P：输出极性，须与 CC0NP 共同作用（配置极性最好在模式选择之前，防止内部信号翻转触发未知错误）。

输出模式：

- 0：输出有效极性位高电平
- 1：输出有效极性位低电平

输入模式，{CC0NP，CC0P}：

- 00：通道输入上升沿有效（捕获、触发、复位、时钟模式），高电平有效（门控、编码模式）
- 01：通道输入下降沿有效（捕获、触发、复位、时钟模式），低电平有效（门控、编码模式）
- 10：保留
- 11：通道输入上升沿和下降沿均有效（捕获、触发、复位、时钟模式），高电平有效（门控、编码模式）

位 0 CC0E：通道使能。

输入模式：

- 0：禁用捕获
- 1：使能捕获

输出模式：

- 0：禁用输出
- 1：使能输出

22.17.10 GPTIM_CNT

偏移量：0x24

复位值：0x0000

15-0
CNT
rw-0h

位 15-0 CNT：计数器计数值。

22.17.11 GPTIM_PSC

偏移量：0x28

复位值：0x0000

15-0
PSC
rw-0h

位 15-0 PSC：时钟分频值为 PSC+1。

22.17.12 GPTIM_ARR

偏移量：0x2C

复位值：0xFFFF

15-0
PSC
rw-FFFFh

位 15-0 ARR：计数器重装载值。

22.17.13 GPTIM_CCR0

偏移量：0x34

复位值：0x0000

15-0
CCR0
rw-0h

位 15-0 CCR0：输出模式时，该寄存器保存用户写入的比较值，用于与 CNT 进行比较；输入模式时，该寄存器保存捕获的值，只读。

22.17.14 GPTIM_CCR1

偏移量：0x38

复位值：0x0000

15-0
CCR1
rw-0h

位 15-0 CCR1：输出模式时，该寄存器保存用户写入的比较值，用于与 CNT 进行比较；输入模式时，该寄存器保存捕获的值，只读。

22.17.15 GPTIM_CCR2

偏移量：0x3C

复位值：0x0000

15-0
CCR2
rw-0h

位 15-0 CCR2：输出模式时，该寄存器保存用户写入的比较值，用于与 CNT 进行比较；输入模式时，该寄存器保存捕获的值，只读。

22.17.16 GPTIM_CCR3

偏移量：0x40

复位值：0x0000

15-0
CCR3
rw-0h

位 15-0 CCR3：输出模式时，该寄存器保存用户写入的比较值，用于与 CNT 进行比较；输入模式时，该寄存器保存捕获的值，只读。

22.17.17 GPTIM_DCR

偏移量：0x48

复位值：0x0000

15-13	12-8	7-5	4-0
RESERVED	DBL	RESERVED	DBA
r-0h	rw-0h	r-0h	rw-0h

位 15-13 RESERVED：保留。

位 12-8 DBL：DMA 连续读写长度。

- 00000：1 个传输
- 00001：2 个传输
- 00010：3 个传输
- 00011：4 个传输
- 00100：5 个传输
- 00101：6 个传输
- 00110：7 个传输
- 00111：8 个传输
- 01000：9 个传输**（未完，接下页）**

- 01001：10 个传输
- 01010：11 个传输
- 01011：12 个传输
- 01100：13 个传输
- 01101：14 个传输
- 01110：15 个传输
- 01111：16 个传输
- 10000：17 个传输
- 10001：18 个传输

位 7-5 RESERVED：保留。

位 4-0 DBA：DMA 连续读写基地址。

- 00000：CR1 寄存器
- 00001：CR2 寄存器
- 00010：SMCR 寄存器
- 00011：DIER 寄存器
- 00100：SR 寄存器
- 00101：EGR 寄存器
- 00110：CCMR1 寄存器
- 00111：CCMR2 寄存器
- 01000：CCER 寄存器
- 01001：CNT 寄存器
- 01010：PSC 寄存器
- 01011：ARR 寄存器
- 01100：偏移地址为 0X30 的保留寄存器
- 01101：CCR0 寄存器
- 01110：CCR1 寄存器
- 01111：CCR2 寄存器
- 10000：CCR3 寄存器
- 10001：偏移地址为 0X44 的保留寄存器
- 10010：DCR 寄存器
- 10011：DMAR 寄存器
- 10100：OR 寄存器
- 10101：reserved
- 10110：reserved
- 10111：reserved
- 11000：reserved
- 11001：reserved
- 11010：reserved
- 11011：reserved
- 11100：reserved
- 11101：reserved
- 11110：reserved
- 11111：reserved

22.17.18 GPTIM_DMAR

偏移量：0x4C

复位值：0x0000

15-0
DMAR
rw-0h

位 15-0 DMAR：该寄存器保存当前 DMA 操作的寄存器的值，例如当前 DMA 需要操作 TIM_CR2 寄存器，那么直接操作该地址，便相当于操作 TIM_CR2 寄存器。具体代表那个寄存器需要参考 DSTEP、DBL 和 DBA 的值。

22.17.19 GPTIM_OR

偏移量：0x50

复位值：0x0000

GPTIMER0 时此寄存器的结构如下：

15-11	10-7	6-4	3-0
RESERVED	ETR_RMP	TI3_RMP	TI0_RMP
r-0h	rw-0h	rw-0h	rw-0h

位 15-11 RESERVED：保留。

位 10-7 ETR_RMP：ETR 重映射。

- 0000：iom
- 0001：comp0
- 0010：comp1
- 0011：xo32k
- 0100：rco48m
- 0101：adcctrl_awd0
- 0110：adcctrl_awd1
- 0111：adcctrl_awd2
- 1000：uart_rx[0]
- 1001：uart_rx[1]
- 1010：uart_rx[2]
- 1011：uart_rx[3]
- 1100：uart_rx[4]
- 1101：reserved
- 1110：reserved
- 1111：reserved

位 6-4 TI3_RMP：通道 3 重映射。

- 000：iom
- 001：comp0
- 010：comp1
- 011：reserved
- 100：reserved
- 101：reserved
- 110：reserved
- 111：reserved

位 3-0 TI0_RMP：通道 0 重映射。

- 0000：iom
- 0001：uart_rx[0]
- 0010：uart_rx[1]
- 0011：uart_rx[2]
- 0100：uart_rx[3]
- 0101：uart_rx[4]
- 0110：reserved
- 0111：reserved
- 1000：reserved
- 1001：reserved
- 1010：reserved
- 1011：reserved
- 1100：reserved
- 1101：reserved
- 1110：reserved
- 1111：reserved

GPTIMER1 时此寄存器的结构如下：

15-2	1-0
RESERVED	TI2_RMP
r-0h	rw-0h

位 15-2 RESERVED：保留。

位 1-0 TI2_RMP：通道 2 重映射。

- 00：iom
- 01：TIM3_CH1
- 10：reserved
- 11：reserved

GPTIMER2 时此寄存器的结构如下：

15-10	9-7	6-5	4-0
RESERVED	ETR_RMP	TI1_RMP	TI0_RMP
r-0h	rw-0h	rw-0h	rw-0h

位 15-10 RESERVED：保留。

位 9-7 ETR_RMP：ETR 重映射。

- 0000：iom
- 0001：comp0
- 0010：comp1
- 0011：xo32k
- 0100：reserved
- 0101：reserved
- 0110：reserved
- 0111：reserved
- 1000：reserved
- 1001：reserved
- 1010：reserved
- 1011：reserved
- 1100：reserved
- 1101：reserved
- 1110：reserved
- 1111：reserved

位 6-5 TI1_RMP：通道 1 重映射。

- 00：iom
- 01：comp1
- 10：reserved
- 11：reserved

位 4-0 TI0_RMP：通道 0 重映射。

- 00000：iom
- 00001：xo24m
- 00010：xo32m
- 00011：rco48m
- 00100：xo32k
- 00101：rco32k
- 00110：mco
- 00111：comp0
- 01000：rco3.6m
- 01001：rtc_alarm1_happen_pulse
- 01010：rtc_alarm0_happen_pulse
- 01011：rtc_cyc_counter_pulse
- 01100：reserved**（未完，接下页）**
- 01101：reserved

- 01110：reserved
- 01111：reserved
- 10000：reserved
- 10001：reserved
- 10010：reserved
- 10011：reserved
- 10100：reserved
- 10101：reserved
- 10110：reserved
- 10111：reserved
- 11000：reserved
- 11001：reserved
- 11010：reserved
- 11011：reserved
- 11100：reserved
- 11101：reserved
- 11110：reserved
- 11111：reserved

GPTIMER3 时此寄存器的结构如下：

15-7	6-3	2-0
RESERVED	ETR_RMP	TI0_RMP
r-0h	rw-0h	rw-0h

位 15-7 RESERVED：保留。

位 6-3 ETR_RMP：ETR 重映射。

- 0000：iom
- 0001：comp0
- 0010：comp1
- 0011：xo32k
- 0100：uart_rx[0]
- 0101：uart_rx[1]
- 0110：uart_rx[2]
- 0111：uart_rx[3]
- 1000：uart_rx[4]
- 1001：reserved
- 1010：reserved
- 1011：reserved
- 1100：reserved
- 1101：reserved
- 1110：reserved
- 1111：reserved

位 2-0 TI0_RMP：通道 0 重映射。

- 000：iom
- 001：comp0
- 010：comp1
- 011：uart_rx[0]
- 100：uart_rx[1]
- 101：uart_rx[2]
- 110：uart_rx[3]
- 111：uart_rx[4]