ASR6601

Reference Manual

Version 1.3.0

Issue Date 2022-08-11

About This Document

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

Intended Readers

This document is mainly for engineers who use this chip to develop their own platform and products, for instance:

- Hardware Development Engineer
- Software Engineer
- Technical Support Engineer

Included Chip Models

The product models corresponding to this document are as follows.

Model	Flash	SRAM	Processor	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 Statement

ASR and ASR Microelectronics Co., Ltd. are trademarks of ASR Microelectronics Co., Ltd.

Other trade names, trademarks, and registered trademarks mentioned in this document are the property of their respective owners and are hereby declared.

Disclaimer

ASR does not give any warranty of any kind and may make improvements and/or changes in this document or in the product described in this document at any time.

This document is only used as a guide, and no contents in the document constitute any form of warranty. Information in this document is subject to change without notice.

All liability, including liability for infringement of any proprietary rights caused by using the information in this document, is disclaimed.

ASR Microelectronics Co., Ltd.

Address: 9F, Building 10, No. 399 Keyuan Road, Zhangjiang High-tech Park, Pudong New Area, Shanghai, 201203, China

Homepage: http://www.asrmicro.com/

Revision History

Date	Version	Release Notes
2022.08	V1.2.0	First Release.
2022.08	V1.3.0	● Updated some descriptions of the 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 Tree.

Table of Contents

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

1.1 General

ASR6601 is a general LPWAN Wireless Communication SoC, with integrated RF Transceiver, Modem and a 32-bit RISC MCU. The MCU uses Arm China STAR-MC1 Processor, with 48 MHz operation frequency. The RF Transceiver has continuous frequency coverage from 150 MHz to 960 MHz. The Modem supports LoRa modulation for LPWAN use cases and (G)FSK modulation for legacy use cases. The Modem also supports BPSK modulation in TX and (G)MSK modulation in TX and RX.

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. Among all the low-power modes, the Standby and Stop3 modes are commonly used. With a 3.3V power supply, the power consumption is down to 0.9 uA (without RF/MCU Retention, with RTC) in Standby mode and is down to 1.3 uA (with RF/MCU Retention and RTC for ASR6601CB) and 1.6 uA (with RF/MCU Retention and RTC for ASR6601SE) in Stop3 mode. The LPWAN Wireless Communication Module designed with ASR6601 provides ultra-long range and ultra-low power communication for LPWAN applications.

ASR6601 can achieve a high sensitivity to -148 dBm and the maximum transmit power is up to +22 dBm. This makes the chip suitable to be used in long-range LPWAN with high efficiency. The total chip package is of very small size, QFN 6 mm x 6 mm/QFN 8 mm x 8 mm.

1.2 Key Features

- Small footprint: QFN48, 6 mm x 6 mm or QFN68, 8 mm x 8 mm
- Frequency Range: 150 MHz ~ 960 MHz
- Maximum Power +22 dBm constant RF output
- High sensitivity: -148 dBm
- Programmable bit rate up to 62.5 Kbps in LoRa modulation mode
- Programmable bit rate up to 300 Kbps in (G)FSK modulation mode
- Preamble detection
- Embedded memories (up to 256 KB of Flash memory and 64 KB of SRAM).
- Up to 42 configurable GPIOs: 3 x I2C, 1 x I2S, 4 x UART, 1 x LPUART, 1 x SWD, 3 x SPI, 1 x QSPI and 2 x WDG
- 4 x GPtimer, 2 x Basic Timer, 2 x LP timer and 1 x Sys Ticker

- 48 MHz Arm China STAR-MC1 Processor
- 4-channel DMA engine x 2
- Support 37 IRQ interrupts with configurable 0~7 priority levels for each IRQ interrupt
- Embedded 12-bit 1 Msps SAR ADC
- Embedded 12-bit DAC
- 32.768 KHz External Watch Crystal Oscillator
- 32 MHz External Crystal Oscillator for RF Transceiver
- 24 MHz External Crystal Oscillator for SoC (optional)
- Embedded internal 4 MHz RC oscillator
- Embedded internal High frequency (48 MHz) RC oscillator
- Embedded internal Low frequency (32.768 KHz) RC oscillator
- Embedded internal PLL to generate 48 MHz clock
- Embedded 3 x OPA
- Embedded 2 x Low Power Comparator
- Embedded LCD driver
- Embedded LD, TD, VD and FD
- Supports AES, DES, RSA, ECC, SHA and SM2/3/4

3. ASR6601 Functions

3.1 ASR6601 SoC Diagram

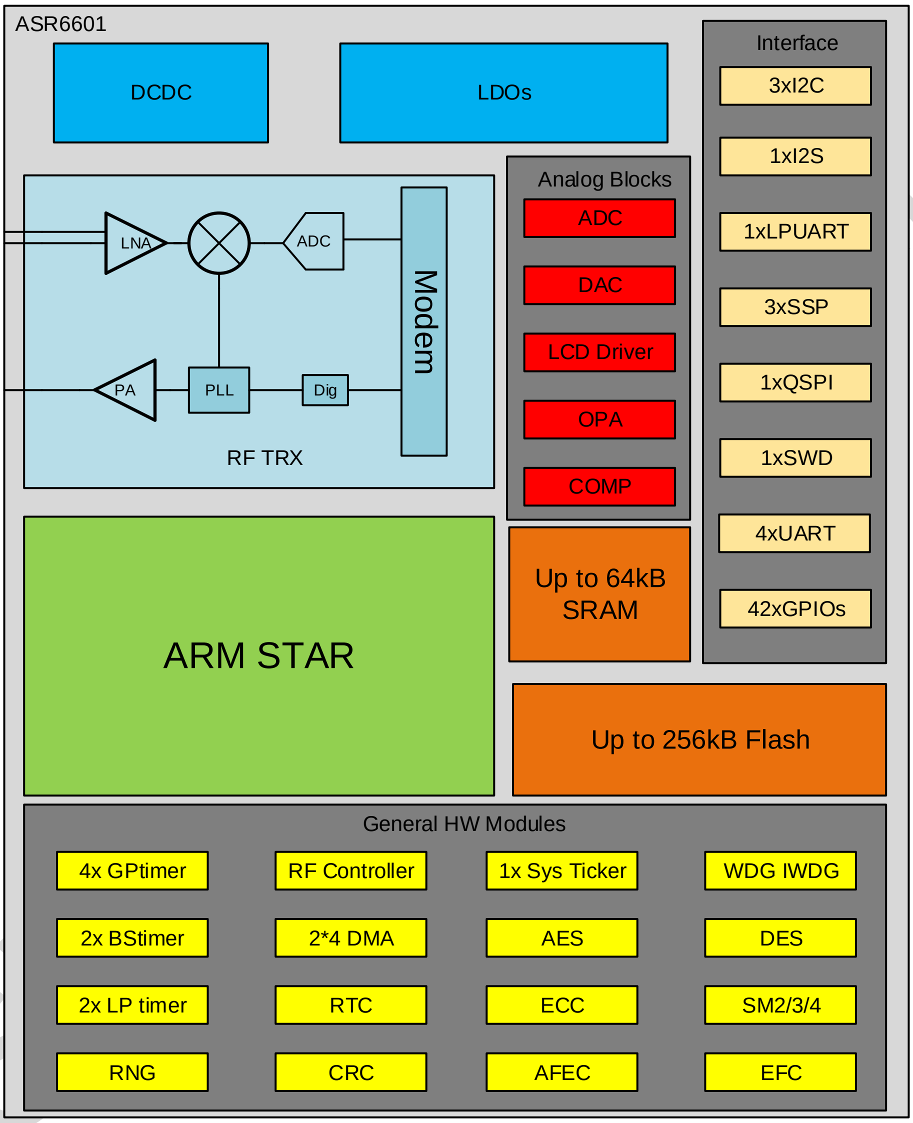

Figure 3-1 ASR6601 SoC Diagram

4. Power Management Unit

4.1 Power Supply

ASR6601 has several separated power supply pins. With these 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:

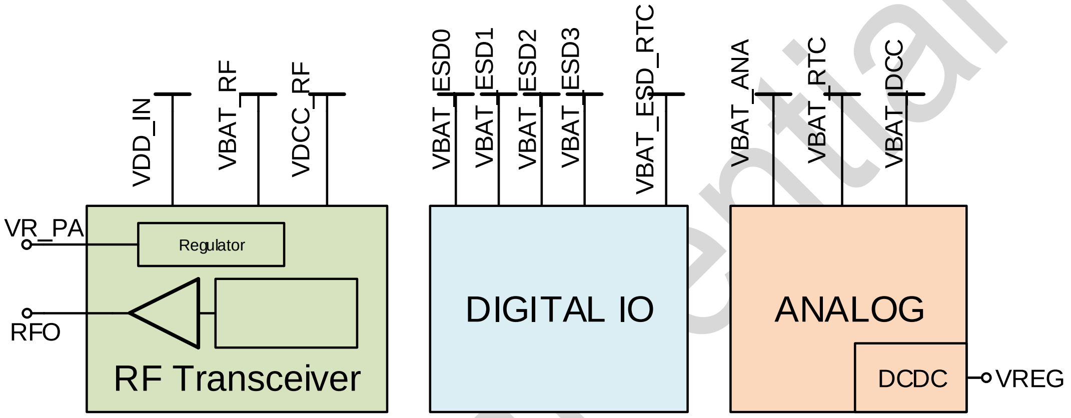

Figure 4-1 ASR6601 Power Grid

- **VDD_IN**: The power supply for the PA in the RF transmitter.
- **VBAT_RF**: The power supply for the RF TRX, excluding the PA.
- **VDCC_RF**: The low-power supply for RF TRX, which must be connected to VREG pin of SoC through the PCB.
- **VBAT_ESD0**: The power supply for digital IO.
- **VBAT_ESD1**: The power supply for digital IO.
- **VBAT_ESD2**: The power supply for digital IO.
- **VBAT_ESD3**: The power supply for digital IO.
- **VBAT_DCC**: The dedicated power supply for DCDC in analog circuit.
- **VBAT_ESD_RTC**: The power supply for IOs in RTC domain.
- **VBAT_RTC**: The power supply for analog blocks in RTC domain.
- **VBAT_ANA**: The power supply for analog blocks.

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** (always 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** (Always on and retention) 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.

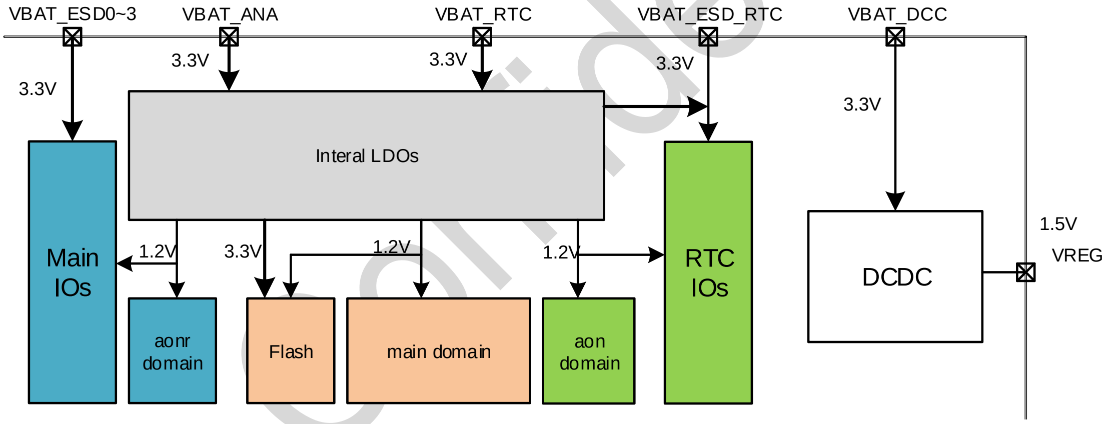

Figure 4-2 ASR6601 Power Supply Architecture

5. Access Control

5.1 Simple Configuration

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 strong.

5.2 Access Control

Based on debug_level rules, boot mode, exe-only access rules, write-protected area access rules, info area access rules, and secure areas access rules, the access rights of the four main interfaces (cpucode, cpuw, dmac0 and dmac1) are controlled.

5.2.1 Debug Level Rules

Debug_level mainly affects cpu_code (boot from SRAM and boot from bootloader), cpu_sw, dmac0 and dmac1 access to sensitive areas. Sensitive areas include the flash_main area, the OTP partition of flash_info area, and the retention SRAM.

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)

6. Operation Mode

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 a low-power mode, peripherals marked as “O” (excluding GPIO) are turned off by default. The functions used in low-power mode must be turned on before entering low-power mode.
2. When entering a low-power mode, developers need to configure below items to achieve specified power consumption:
   (1) Configure unused GPIOs to ANALOG mode (high impedance)
   (2) If the GPIOs are in input mode, users should configure them pull-up or pull-down
   (3) In output mode, configure the connected peripherals pull-up or pull-down according to the output level.
3. Use RCO48M/2 to enter or exit a low-power mode. If you use a clock other than RCO48M/2 before entering a low-power mode, you need to switch to RCO48M/2. After exiting the low-power mode, you can switch to the previously used clock.
4. RCO32K/XO32K and some other analog functions can retain active in low-power modes. If needed, turn on these functions before entering a low-power mode by software.
5. Clocks other than RCO48M/RCO32K/XO32K and the remaining analog function modules must be turned off by the software before entering a low-power mode.

Table 6-1 Modules Working Status in Various Operation Modes

	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			

	Run	LpRun	Sleep	LpSleep	Stop0	Stop1	Stop2	Stop3	Standby	Stop0~2 Wakeup	Stop3 Wakeup	Standby Wakeup
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			
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			

	Run	LpRun	Sleep	LpSleep	Stop0	Stop1	Stop2	Stop3	Standby	Stop0-2 Wakeup	Stop3 Wakeup	Standby Wakeup
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			
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 above table:

- **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**: Work normally
- **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 Function**1** 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

Enter ***LpRun*** mode from Run mode in the following way:
Turn off all high-speed clocks to make CPU run at 32K clock frequency. Then switch the working state of LDO by software.

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.

Return to ***Run*** mode from LpRun mode in the following way:
Switch the working state of LDO by software. Then turn on the high-speed clock

LpRun config register is used to switch LDO working state:

(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

Enter ***Sleep*** mode from Run mode in the following way:
CPU executes WFI/WFE instruction SLEEPDEEP=0, or isr returns SLEEPONEXIT=1 and SLEEPDEEP=0.

Return to ***Run*** mode from Sleep mode in the following ways:

- If WFI instruction is used to enter Sleep mode, then the system is waked-up by interrupts.
- If WFE instruction is used to enter Sleep mode, then the system is waked-up by events.

Notice: *Since there is no dedicated wake-up event signal, the interrupt signal is used to generate wake-up events by instruction SVONPEND=1 and turning off the corresponding NVIC.*

6.3.2 Wakeup Source

Interrupt signal generated by each module

6.4 LpSleep

6.4.1 Enter and Exit

Enter ***LpSleep*** mode from LpRun mode in the following way:
CPU executes WFI/WFE instruction SLEEPDEEP=0, or isr returns SLEEPONEXIT=1 and SLEEPDEEP=0.

Return to ***LpRun*** mode from LpSleep mode in the following ways:

- If WFI instruction is used to enter LpSleep mode, then the system returns LpRun mode by interrupts.
- If WFE instruction is used to enter LpSleep mode, then the system returns LpRun mode by wake-up events.

Notice: *Since there is no dedicated event wake-up signal, the interrupt signal is used to generate wake-up events by instruction SVONPEND=1 and turning off the corresponding NVIC.*

6.4.2 Wakeup Source

Interrupt signal of each module

6.5 Stop0

6.5.1 Enter and Exit

Enter ***Stop0*** mode from Run mode in the following way:
Configure lp_mode to 2’b00, then CPU executes WFI/WFE instruction SLEEPDEEP=1, or isr returns SLEEPONEXIT=1 and SLEEPDEEP=1.

Return to ***Run*** mode from Stop0 mode in the following ways:

- If WFI instruction is used to enter Stop0 mode, then the system is waked-up by interrupts.
- If WFE instruction is used to enter Stop0 mode, then the system is waked-up by events.

The pwr module manages the status of the wake-up sources and outputs the *pwr_wakeup_int* signal and the *pwr_wakeup_event* signal to wake up the CPU.

6.5.2 Wakeup Source

- GPIO00-GPIO63 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
- 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

Enter ***Stop1*** mode from Run mode in the following way:
Configure lp_mode to 2’b01, then CPU executes WFI/WFE instruction SLEEPDEEP=1, or isr returns SLEEPONEXIT=1 and SLEEPDEEP=1.

Return to ***Run*** mode from Stop1 mode in the following ways:

- If WFI instruction is used to enter Stop1 mode, then the system is waked-up by interrupts.
- If WFE instruction is used to enter Stop1 mode, then the system is waked-up by events.

The pwr module manages the status of the wake-up sources and outputs the *pwr_wakeup_int* signal and the *pwr_wakeup_event* signal to wake up the CPU.

6.6.2 Wakeup Source

- GPIO00-GPIO63 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
- 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

Enter ***Stop2*** mode from Run mode in the following way:
Configure lp_mode to 2’b10, then CPU executes WFI/WFE instruction SLEEPDEEP=1, or isr returns SLEEPONEXIT=1 and SLEEPDEEP=1.

Return to ***Run*** mode from Stop2 mode in the following ways:

- If WFI instruction is used to enter Stop2 mode, then the system is waked-up by interrupts.
- If WFE instruction is used to enter Stop2 mode, then the system is waked-up by events.

The pwr module manages the status of the wake-up sources, and outputs *pwr_wakeup_int* signal and *pwr_wakeup_event* signal to wake up the CPU.

6.7.2 Wakeup Source

- GPIO00-GPIO63 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
- 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

Enter ***Stop3*** mode from Run mode in the following way:
Configure lp_mode to 2'b11 and lp_mode_ext to 1'b1, then CPU executes WFI/WFE instruction SLEEPDEEP=1, or isr returns SLEEPONEXIT=1 and SLEEPDEEP=1.

The system returns to ***Run*** mode if a wake-up event occurred in Stop3 mode.

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

Enter ***Standby*** mode from Run mode in the following way:
Configure lp_mode to 2'b11 and lp_mode_ext to 1'b0, then CPU executes WFI/WFE instruction SLEEPDEEP=1, or isr returns SLEEPONEXIT=1 and SLEEPDEEP=1.

The system returns to ***Run*** mode if a wake-up event occurred in Standby mode.

Notice:

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 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

7. System Configuration

7.1 System Architecture

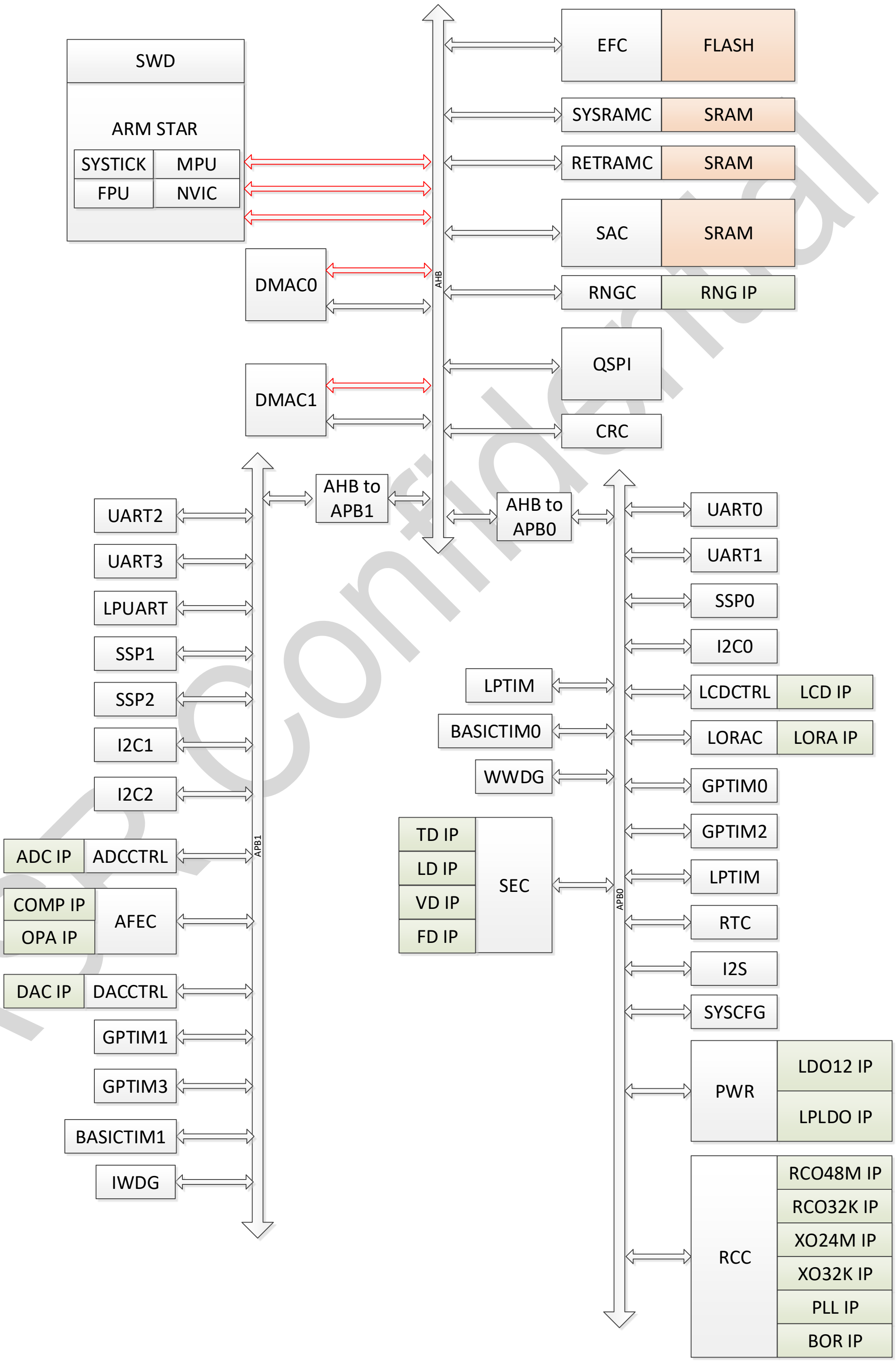

Figure 7-1 System Architecture Diagram

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 Scope

Start Address	End Address	Description	Executable	icode Access	dcode Access	system Access	DMAC0 Access	DMAC1 Access
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 Mapping

Category	Start Address	End Address	Description	Size
SYSTEM	0xE0100000	0xFFFFFFFF	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 Internal 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 Internal 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 Internal Address Mapping

Start Address	End Address	Description	Size
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

Start Address	End Address	Description	Size
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 Internal Address Mapping

Start Address	End Address	Description	Size
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 Mode

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 inner Bootloader
<2	0	X	1	1	X	Boot from inner 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 inner Bootloader
<2	1	0	X	1	X	Boot from inner 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

SYSCFG Base Address: 0x40001000

Table 7-8 SYSCFG Register Summary

Register Name	Address Offset	Description
SYSCFG_CR0	0x000	Control Register 0, DMA handshake
SYSCFG_CR1	0x004	Control Register 1, DMA handshake
SYSCFG_CR2	0x008	Control Register 2
SYSCFG_CR3	0x00C	Control Register 3
SYSCFG_CR4	0x010	Control Register 4
SYSCFG_CR5	0x014	Control Register 5
SYSCFG_CR6	0x018	Control Register 6, secure lock control
SYSCFG_CR7	0x01C	Control Register 7, secure lock control
SYSCFG_CR8	0x020	Control Register 8, QSPI memory encryption key
SYSCFG_CR9	0x024	Control Register 9, QSPI REMAP
SYSCFG_CR10	0x028	Control Register 10

7.5.1 SYSCFG_CR0

Address 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

Address 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

Address 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: Whether the LPTIM1 counter is stopped when 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

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

Bit 27 SYSCFG_HALTED_LPTIM0_EN: Whether the LPTIM0 counter is stopped when 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: Whether the independent watchdog counter is stopped when 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: Whether the window watchdog counter is stopped when 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: Whether the GPTIM0 counter is stopped when 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: Whether the GPTIM1 counter is stopped when 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: Whether the GPTIM2 counter is stopped when 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: Whether the GPTIM3 counter is stopped when 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: Whether the BASICTIM0 counter is stopped when 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: Whether the BASICTIM1 counter is stopped when 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

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

Bit 11 CPU_STCALIB_SKEW: CPU SysTick skew configuration

- 0: no skew
- 1: skew exist

Bit 10 SYSCFG_DBG_SLEEP: Whether to allow a debug connection in Deepsleep mode

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

- 0: not allowed
- 1: allowed

Bit 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

Address 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: Whether to allow a debug connection in Stop mode

It is only used in debug and it will affect the implementation of Stop mode.

- 0: not allowed
- 1: allowed

Bit 0 SYSCFG_DBG_STANDBY: Whether to allow a debug connection in Standby mode

It is only used in debug and it will affect the implementation of Standby mode.

- 0: not allowed
- 1: allowed

7.5.5 SYSCFG_CR4

Address 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

Address 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

Address 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: no security lock
- 1: security lock enabled

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

[5] Correspond to VD

- 0: no security lock
- 1: security lock enabled

[6] Correspond to TD

- 0: no security lock
- 1: security lock enabled

[7] Correspond to LD

- 0: no security lock
- 1: security lock enabled

[8] Correspond to FD24M

- 0: no security lock
- 1: security lock enabled

[9] Correspond to FD32M

- 0: no security lock
- 1: security lock enabled

[10] Correspond to RNG

- 0: no security lock
- 1: security lock enabled

[11] Correspond to TEST

- 0: no security lock
- 1: security lock enabled

[14:12]: Unused

- 0: no security lock
- 1: security lock enabled

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

Bit 3 SEC_SECURE_LOCK: SEC security lock

- 0: no security lock
- 1: security lock enabled

Bit 2 SAC_SECURE_LOCK: SAC security lock

- 0: no security lock
- 1: security lock enabled

Bit 1 DMAC0_SLAVE_SECURE_LOCK: DMAC0 slave interface security lock

- 0: no security lock
- 1: security lock enabled

Bit 0 DMAC0_MASTER_SECURE_LOCK: DMAC0 master interface security lock

- 0: no security lock
- 1: security lock enabled

7.5.8 SYSCFG_CR7

Address Offset: 0x01C

Reset Value: 0x00000000

This register is in the AON domain.

<table>
<tr><th>31-15</th><th colspan="2">14-5</th><th>4</th></tr>
<tr><td>RESERVED</td><td colspan="2">ANALOG_AON_SECURE_LOCK</td><td>RTC_CALENDAR_SECURE_LOCK</td></tr>
<tr><td>r</td><td colspan="2">r/w</td><td>r/w</td></tr>
<tr><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>RTC_WAKEUP2_SECURE_LOCK</td><td>RTC_WAKEUP1_SECURE_LOCK</td><td>RTC_WAKEUP0_SECURE_LOCK</td><td>RTC_TAMPER_SECURE_LOCK</td></tr>
<tr><td>r/w</td><td>r/w</td><td>r/w</td><td>r/w</td></tr>
</table>

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: no security lock
- 1: security lock enabled

[6] Correspond to RCO3.6M

- 0: no security lock
- 1: security lock enabled

[7] Correspond to PWRSW

- 0: no security lock
- 1: security lock enabled

[8] Correspond to RCO32K

- 0: no security lock
- 1: security lock enabled

[9] Correspond to XO32K

- 0: no security lock
- 1: security lock enabled

[10] Correspond to LDO12

- 0: no security lock
- 1: security lock enabled

[11] Correspond to FD32K

- 0: no security lock
- 1: security lock enabled

[14:12] Unused

- 0: no security lock
- 1: security lock enabled

Bit 4 RTC_CALENDAR_SECURE_LOCK: Calendar configuration security lock in RTC

- 0: no security lock

- 1: security lock enabled

Bit 3 RTC_WAKEUP2_SECURE_LOCK: Wakeup2 configuration security lock in RTC

- 0: no security lock
- 1: security lock enabled

Bit 2 RTC_WAKEUP1_SECURE_LOCK: Wakeup1 configuration security lock in RTC

- 0: no security lock
- 1: security lock enabled

Bit 1 RTC_WAKEUP0_SECURE_LOCK: Wakeup0 configuration security lock in RTC

- 0: no security lock
- 1: security lock enabled

Bit 0 RTC_TAMPER_SECURE_LOCK: Tamper configuration security lock in RTC

- 0: no security lock
- 1: security lock enabled

7.5.9 SYSCFG_CR8

Address 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

Address 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

Address 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

Notice: *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

Notice: *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

Notice: *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
26	uart2_tx

No.	Source
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

The system reset is used to reset most of the data logic in the Main domain excluding the Reset Source Status Register (*RCC_RST_SR*) which is used to record the latest system reset source.

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.

The low-power reset is used to reset all digital logic in the main domain.

8.2 Clock

Figure 8-1 shows the system clock structure.

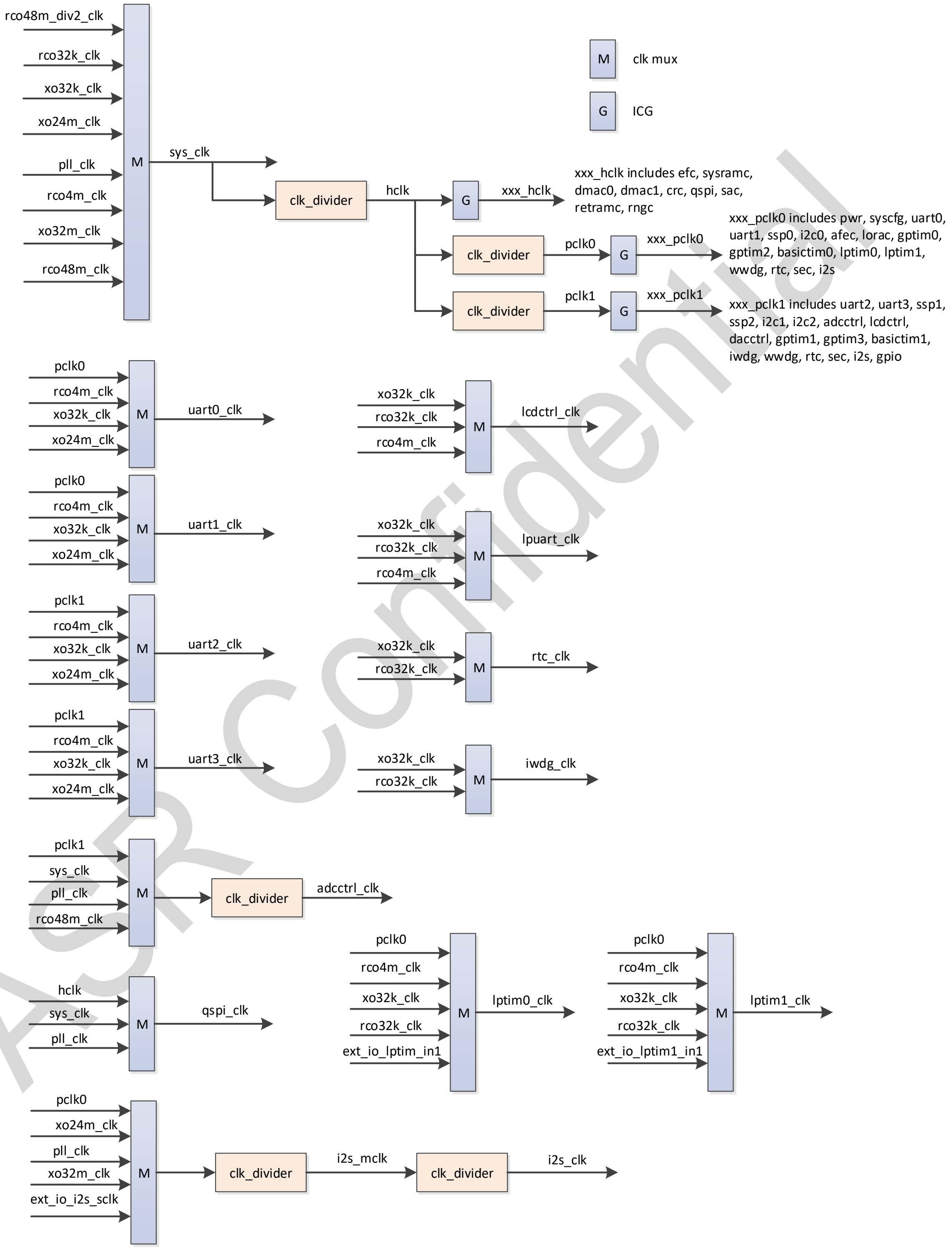

Figure 8-1 Clock Tree

8.2.1 SYS_CLK

RCO48M divided by 2, RCO32K, XO32K, PLL, XO24M, XO32M, RCO3.6M or RCO48M can be selected as the SYS_CLK clock source. 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 ranges 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, IO input clock
- 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 Clock-out Capability

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

RCC Base Address: 0x40000000

Table 8-1 RCC Register Summary

Register Name	Address Offset	Description
RCC_CR0	0x000	Control register 0
RCC_CR1	0x004	Control register 1, interface clock source selection
RCC_CR2	0x008	Control register 2, interface clock source selection
RCC_CGR0	0x00C	Module clock configure register 0
RCC_CGR1	0x010	Module clock configure register 1
RCC_CGR2	0x014	Module clock configure register 2
RCC_RST0	0x018	Module reset control register 0
RCC_RST1	0x01C	Module reset control register 1
RCC_RST_SR	0x020	System reset source Status register
RCC_RST_CR	0x024	System reset source control register
RCC_SR	0x028	Status register, RCC_CGR2 configuration status
RCC_SR1	0x02C	Status register 1, module clock configuration status
RCC_CR3	0x030	Control register 3, interface clock division

8.3.1 RCC_CR0

Address 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

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

Notice: *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

Notice: *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.*

Bits 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

Address Offset: 0x004

Reset Value: 0x00000000

This register is in AON power 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

Notice:

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

Notice:

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

Notice:

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*

have been enabled.

Bits 7-6 LPTIM0_CLK_SEL: LPTIM0 interface clock source selection

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

Notice:

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 have been 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

Address 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

Notice: *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

Notice: *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

Notice: *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

Notice: *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

Notice: *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: external IOM_I2S_CLK

Notice:

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

Notice: *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

Address 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: enabled

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

Address 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

Notice: *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

Notice: *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

Address 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

Address 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	LPTIM0_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: not reset

Bit 30 UART1_RST_N: UART1 reset control

- 0: reset
- 1: not reset

Bit 29 UART2_RST_N: UART2 reset control

- 0: reset
- 1: not reset

Bit 28 UART3_RST_N: UART3 reset control

- 0: reset
- 1: not reset

Bit 27 LPUART_RST_N: LPUART reset control

- 0: reset
- 1: not reset

Bit 26 SSP0_RST_N: SSP0 reset control

- 0: reset
- 1: not reset

Bit 25 SSP1_RST_N: SSP1 reset control

- 0: reset
- 1: not reset

Bit 24 SSP2_RST_N: SSP2 reset control

- 0: reset
- 1: not reset

Bit 23 QSPI_RST_N: QSPI reset control

- 0: reset
- 1: not reset

Bit 22 I2C0_RST_N: I2C0 reset control

- 0: reset
- 1: not reset

Bit 21 I2C1_RST_N: I2C1 reset control

- 0: reset
- 1: not reset

Bit 20 I2C2_RST_N: I2C2 reset control

- 0: reset
- 1: not reset

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

Bit 18 ADCCTRL_RST_N: ADCCTRL reset control

- 0: reset
- 1: not reset

Bit 17 AFEC_RST_N: AFEC reset control

- 0: reset
- 1: not reset

Bit 16 LCDCTRL_RST_N: LCDCTRL reset control

- 0: reset
- 1: not reset

Bit 15 DACCTRL_RST_N: DACCTRL reset control

- 0: reset
- 1: not reset

Bit 14 LORAC_RST_N: LORAC reset control

- 0: reset
- 1: not reset

Bit 13 IOM_RST_N: IOM reset control

- 0: reset
- 1: not reset

Bit 12 GPTIM0_RST_N: GPTIM0 reset control

- 0: reset
- 1: not reset

Bit 11 GPTIM1_RST_N: GPTIM1 reset control

- 0: reset
- 1: not reset

Bit 10 GPTIM2_RST_N: GPTIM2 reset control

- 0: reset
- 1: not reset

Bit 9 GPTIM3_RST_N: GPTIM3 reset control

- 0: reset
- 1: not reset

Bit 8 BASICTIM0_RST_N: BASICTIM0 reset control

- 0: reset
- 1: not reset

Bit 7 BASICTIM1_RST_N: BASICTIM1 reset control

- 0: reset
- 1: not reset

Bit 6 LPTIM0_RST_N: LPTIM0 reset control

- 0: reset
- 1: not reset

Bit 5 IWDG_RST_N: IWDG reset control

- 0: reset
- 1: not reset

Bit 4 WWDG_RST_N: WWDG reset control

- 0: reset
- 1: not reset

Bit 3 RTC_RST_N: RTC reset control

- 0: reset
- 1: not reset

Bit 2 CRC_RST_N: CRC reset control

- 0: reset
- 1: not reset

Bit 1 SEC_RST_N: SEC reset control

- 0: reset
- 1: not reset

Bit 0 SAC_RST_N: SAC reset control

- 0: reset
- 1: not reset

8.3.8 RCC_RST1

Address 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: not reset

Bit 3 RNGC_RST_N: RNGC reset control

- 0: reset
- 1: not reset

Bit 2 I2S_RST_N: I2S reset control

- 0: reset
- 1: not reset

Bit 1 DMAC0_RST_N: DMAC0 reset control

- 0: reset
- 1: not reset

Bit 0 DMAC1_RST_N: DMAC1 reset control

- 0: reset
- 1: not reset

8.3.9 RCC_RST_SR

Address Offset: 0x020

Reset Value: 0x00000040

Notice: 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. This bit is set by hardware and cleared by software writing 1 to it.

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

Bit 5 IWDG_RESET_SR: IWDG reset status. This bit is set by hardware and cleared by software writing 1 to it.

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

Bit 4 WWDG_RESET_SR: WWDG reset status. This bit is set by hardware and cleared by software writing 1 to it.

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

Bit 3 EFC_RESET_SR: EFC reset status. This bit is set by hardware and cleared by software writing 1 to it.

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

Bit 2 CPU_RESET_SR: CPU reset status. This bit is set by hardware and cleared by software writing 1 to it.

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

Bit 1 SEC_RESET_SR: SEC reset status. This bit is set by hardware and cleared by software writing 1 to it.

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

Bit 0 STANDBY_RESET_SR: Standby reset status. This bit is set by hardware and cleared by software writing 1 to it.

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

8.3.10 RCC_RST_CR

Address 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

Address 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 is in progress
- 1: configuration is complete

Bit 4 SET_LPTIM0_AON_CLK_EN_DONE: LPTIM0_AON_CLK_EN configuration status

This bit is set and cleared by hardware.

- 0: configuration is in progress
- 1: configuration is complete

Bit 3 SET_LCDCTRL_AON_CLK_EN_DONE: LCDCTRL_AON_CLK_EN configuration status

This bit is set and cleared by hardware.

- 0: configuration is in progress
- 1: configuration is complete

Bit 2 SET_LPUART_AON_CLK_EN_DONE: LPUART_AON_CLK_EN configuration status

This bit is set and cleared by hardware.

- 0: configuration is in progress
- 1: configuration is complete

Bit 1 SET_RTC_AON_CLK_EN_DONE: RTC_AON_CLK_EN configuration status

This bit is set and cleared by hardware.

- 0: configuration is in progress
- 1: configuration is complete

Bit 0 SET_IWDG_AON_CLK_EN_DONE: IWDG_AON_CLK_EN configuration status

This bit is set and cleared by hardware.

- 0: configuration is in progress
- 1: configuration is complete

8.3.12 RCC_SR1

Address 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 whether the clock is disabled.

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	LPTIM0_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	LPTIM0_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: Indicate LPTIM1_CLK_EN actual status

- 0: LPTIM1 clock is disabled
- 1: LPTIM1 clock is enabled

Bit 19 LPTIM1_AON_CLK_EN_SYNC: Indicate LPTIM1_AON_CLK_EN actual status

- 0: LPTIM1 clock is disabled in AON domain
- 1: LPTIM1 clock is enabled in AON domain

Bit 18 UART0_CLK_EN_SYNC: Indicate UART0_CLK_EN actual status

- 0: UART0 clock is disabled
- 1: UART0 clock is enabled

Bit 17 UART1_CLK_EN_SYNC: Indicate UART1_CLK_EN actual status

- 0: UART1 clock is disabled
- 1: UART1 clock is enabled

Bit 16 UART2_CLK_EN_SYNC: Indicate UART2_CLK_EN actual status

- 0: UART2 clock is disabled
- 1: UART2 clock is enabled

Bit 15 UART3_CLK_EN_SYNC: Indicate UART3_CLK_EN actual status

- 0: UART3 clock is disabled

- 1: UART3 clock is enabled

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

Bit 13 ADCCTRL_CLK_EN_SYNC: Indicate ADCCTRL_CLK_EN actual status

- 0: ADCCTRL clock is disabled
- 1: ADCCTRL clock is enabled

Bit 12 LPTIM0_CLK_EN_SYNC: Indicate LPTIM0_CLK_EN actual status

- 0: LPTIM0 clock is disabled
- 1: LPTIM0 clock is enabled

Bit 11 QSPI_CLK_EN_SYNC: Indicate QSPI_CLK_EN actual status

- 0: QSPI clock is disabled
- 1: QSPI clock is enabled

Bit 10 LPUART_CLK_EN_SYNC: Indicate LPUART_CLK_EN actual status

- 0: LPUART clock is disabled
- 1: LPUART clock is enabled

Bit 9 LCDCTRL_CLK_EN_SYNC: Indicate LCDCTRL_CLK_EN actual status

- 0: LCDCTRL clock is disabled
- 1: LCDCTRL clock is enabled

Bit 8 IWDG_CLK_EN_SYNC: Indicate IWDG_CLK_EN actual status

- 0: IWDG clock is disabled
- 1: IWDG clock is enabled

Bit 7 RTC_CLK_EN_SYNC: Indicate RTC_CLK_EN actual status

- 0: RTC clock is disabled
- 1: RTC clock is enabled

Bit 6 MCO_CLK_EN_SYNC: Indicate MCO_CLK_EN actual status

- 0: MCO clock is disabled
- 1: MCO clock is enabled

Bit 5 I2S_CLK_EN_SYNC: Indicate I2S_CLK_EN actual status

- 0: I2S clock is disabled
- 1: I2S clock is enabled

Bit 4 LPTIM0_AON_CLK_EN_SYNC: Indicate LPTIM0_AON_CLK_EN actual status

- 0: LPTIM0 clock in AON domain is disabled
- 1: LPTIM0 clock in AON domain is enabled

Bit 3 LCDCTRL_AON_CLK_EN_SYNC: Indicate LCDCTRL_AON_CLK_EN actual status

- 0: LCDCTRL clock is disabled in AON domain
- 1: LCDCTRL clock is enabled in AON domain

Bit 2 LPUART_AON_CLK_EN_SYNC: Indicate LPUART_AON_CLK_EN actual status

- 0: LPUART clock is disabled in AON domain
- 1: LPUART clock is enabled in AON domain

Bit 1 RTC_AON_CLK_EN_SYNC: Indicate RTC_AON_CLK_EN actual status

- 0: RTC clock is disabled in AON domain
- 1: RTC clock is enabled in AON domain

Bit 0 IWDG_AON_CLK_EN_SYNC: Indicate IWDG_AON_CLK_EN actual status

- 0: IWDG clock is disabled in AON domain
- 1: IWDG clock is enabled in AON domain

8.3.13 RCC_CR3

Address 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
- 1: not divided
- 2: divided by 2
- 3: divided by 3
- N: divided by N

Notice:

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

Notice:

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

Table 9-1 Interrupt Vector

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
2	9	settable	wwdg		0x0000_0048
3	10	settable	efc		0x0000_004C

Position	Priority	Type of priority	Acronym	Description	Address
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 in total
  - Flash main area: 256 KB for ASR6601SE, 128 KB for ASR6601CB
- Page erase (4 KB) and Mass erase (all flash main)

10.2 Main Features

- Flash operations include read, program, page erase and mass erase
- Read access latency
- Acceleration for accessing the Flash memory
- Support instruction prefetch with 1 deep buffer
- Flash program operation supports single and continuous mode
- Option bytes in Flash info area
- Can be used to generate interrupt signals

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 Address	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. Details are as follows:

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

The 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.

The 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.

Notice: *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 Acceleration for Accessing the Flash Memory

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.

Notice: *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.

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

10.3.6 Flash Program Operation

There are two modes for Flash programming:

- **Single Programming Mode**

  In single mode, it programs 2 words (8 Bytes) each 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 Operation

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 below table 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 inner Bootloader
<2	0	X	1	1	X	Boot from inner 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 inner Bootloader
<2	1	0	X	1	X	Boot from inner 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

The configuration is only valid when the FlashSecure area is enabled by bits[11:0].

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

The configuration is only valid when the FlashSecure area is enabled by bits[11:0].

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

Embedded Flash Base Address: 0x40020000

Table 10-5 Embedded Flash Register Summary

Register Name	Address Offset	Description
EFC_CR	0x00	Control register
EFC_INT_EN	0x04	Interrupt enable register
EFC_SR	0x08	Status register
EFC_PROG_DATA0	0x0C	Program Data 0
EFC_PROG_DATA1	0x10	Program Data 1
EFC_TIMING_CFG	0x14	Timing configuration register
EFC_PROTECT_SEQ	0x18	Protection Sequence
RESERVED	0x1C-0x28	Reserved
SERIAL_NUM_LOW	0x2C	Less Significant 32 bits of the Chip Serial Number
SERIAL_NUM_HIGH	0x30	More Significant 32 bits of the Chip Serial Number
RESERVED	0x34-0x38	Reserved
OPTION_CSR_BYTES	0x3C	OPTION control and status data
OPTION_EXE_ONLY_BYTES	0x40	OPTION Execution-only data
OPTION_WR_PROTECT_BYTES	0x44	OPTION Write-protection data
OPTION_SECURE_BYTES0	0x48	OPTION Secure Data 0
OPTION_SECURE_BYTES1	0x4C	OPTION Secure Data 0

10.5.1 EFC_CR

Address 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

- 0: invalid
- 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

Notice:

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

Notice: *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

Notice: *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 (in hold mode)
- 1: read acceleration enabled (in release mode)

Notice:

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.

Notice:

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

Notice:

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

Bit 0 MASS_ERASE_EN: Flash mass erasing enable

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

Notice:

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

Address 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

Address 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: 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: a programming error occurred

Notice: *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: a 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 *EFC_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: not complete
- 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: not complete
- 1: complete

10.5.4 EFC_PROG_DATA0

Address Offset: 0x0C

Reset Value: 0x00000000

31-0
PROG_DATA0
r/w

Bits 31-0 PROG_DATA0: programming data 0

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

10.5.5 EFC_PROG_DATA1

Address Offset: 0x10

Reset Value: 0x00000000

31-0
PROG_DATA1
r/w

Bits 31-0 PROG_DATA1: programming data 1

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

10.5.6 EFC_TIMING_CFG

Address 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.

Bits 19-16 READ_NUM: Program the number of wait states.

The READ_NUM equals to (READ_NUM+1) multiplied by SYS_CLK clock period.

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

***Notice:** 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

Address 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 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
2. Then write 0x13141516

10.5.8 SERIAL_NUM_LOW

Address 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

Address Offset: 0x30

31-0
SERIAL_NUM_HIGH
r

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

10.5.10 OPTION_CSR_BYTES

Address 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 is disabled
- 1: secure area is enabled

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

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

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

- 0: 0
- 1: 1

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

- 0: 0
- 1: 1

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

- 0: 0
- 1: 1

10.5.11 OPTION_EXE_ONLY_BYTES

Address 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: Whether Exe_Only area is kept when the Debug_Level changes from 1 to 0

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

This bit can only be set to 0 by software.

Bits 23-18 EXE_ONLY2_END: Exe_Only2 area end offset

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

Bits 17-12 EXE_ONLY2_START: Exe_Only2 area start offset

When *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

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

Bits 5-0 EXE_ONLY1_START: Exe_Only1 area start offset

When *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

Address 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

When *WRPROTECT_START* > *WRPROTECT_END*, the write-protected area is disabled.

Bits 5-0 WRPROTECT_START: Write-protected area start offset

When *WRPROTECT_START* > *WRPROTECT_END*, the write-protected area is disabled.

10.5.13 OPTION_SECURE_BYTES0

Address 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 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 SECURE_AREA_EN=1.

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 SECURE_AREA_EN=1.

Bits 11-6 FLASH_SECURE_END: Flash Secure area end

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

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 enabled, the SECURE_AREA_EN bit is set, which means all the other secure areas can be enabled.

When the Flash Secure area is disabled, the SECURE_AREA_EN bit is cleared, which triggers the erase operation.

10.5.14 OPTION_SECURE_BYTES1

Address 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 cannot 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

When *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

When *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 struct, 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

For GPIO00~GPIO55 in the Main domain, every 4 IO MUX outputs a wake-up signal, thus a total of 14 wake-up signals.

In Stop3 mode, wakeup enabling and wakeup at high or low level are configured by the corresponding bits in registers *GPIOA_STOP3_WU_CR*, *GPIOx_STOP3_WU_CR (x=B, C)* and *GPIOD_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.

Configure the function of Portx Pin[7:0] by register *GPIOx_AFRL*, and configure the function of Portx Pin[15:8] by registers *GPIOx_AFRH (x=A, B, C)* and *GPIOD_AFRH*.

11.9 Clock 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 Domain

Main Domain:

For all pins except PortD Pin8~Pin15, the corresponding PADs are 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 SWC pull-down (PortA Pin7) and SWD pull-up (PortA Pin6) are default.

Seal Control: IO status is determined by the default value of the *GPIOx_AFRL* register at power-on until the DebugLevel decision is carried out. If sealing is needed, then the SWD interface will be sealed (disabled) eternally, otherwise, it is still controlled by the register.

Software Configuration: The SWD port can be disabled by the *GPIOx_AFRL* register. Note that it is one-way sealing, which means it cannot be enabled after being disabled.

11.13 BOOT0 Control

Default Control: Since all IOs except the SWC and SWD IOs are analog IOs by default, the BOOT0, SWC and SWD 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 Register Summary

Register Name	Address Offset	Description
GPIOx_OER	0x00	General output enable register
GPIOx_OTYPER	0x04	General output type control register
GPIOx_IER	0x08	General input enable register
GPIOx_PER	0x0C	Pull-up/pull-down enable register
GPIOx_PSR	0x10	Pull-up/pull-down selection register
GPIOx_IDR	0x14	Input data register
GPIOx_ODR	0x18	Output data register
GPIOx_BRR	0x1C	Bit reset register
GPIOx_BSRR	0x20	Bit set or reset register
GPIOx_DSR	0x24	Output drive strength register
GPIOx_INT_CR	0x28	Interrupt enable register
GPIOx_FR	0x2C	Interrupt edge flag register
GPIOx_WU_EN	0x30	Wake-up enable resigter for Sleep/Stop0~2 mode
GPIOx_WU_LVL	0x34	Wake-up level control register for Sleep/Stop0~2 mode
GPIOx_AFRL	0x38	GPIO alternate function low register
GPIOx_AFRH	0x3C	GPIO alternate function high register
GPIOx_STOP3_WU_CR	0x40	Wake-up enable control register in Stop3 mode

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

Address 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)

Address 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)

Address 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
- 1: input enabled

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

Address 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)

Address 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)

Address 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)

Address 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)

Address 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: invalid
- 1: clear the corresponding bit of the GPIOx_ODR register

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

Address 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: disabled
- 1: clear the corresponding bit of the GPIOx_ODR register

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

- 0: disabled
- 1: set the corresponding bit of the GPIOx_ODR register

***Note:** If both the BSR and BR bits are enabled, the BSR bit has a higher priority.*

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

Address Offset: 0x24

Reset 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)

Address Offset: 0x28

Reset Value: 0x00000000

2*n + 1 (n≤15)	2*n (n≤15)
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)

Address Offset: 0x2C

Reset Value: 0x00000000

2*n + 1 (n≤15)	2*n (n≤15)
NEG_F	POS_F
rw1c-0h	rw1c-0h

Bits 2*n + 1 NEG_INT_EN: 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_EN: 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)

Address 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 mode

- 0: disabled
- 1: enabled

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

Address 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 the Portx Pin[15:0] to wakeup CPU from Sleep/Stop0~2 mode in 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)

Address 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

Bits 27-24 AF6: Portx Pin6 function selection

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

Bits 23-20 AF5: Portx Pin5 function selection

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

Bits 19-16 AF4: Portx Pin4 function selection

- 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)

Address 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

Address 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: PortD 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

Address 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: wake-up at low level
- 1: wake-up at 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: wake-up at low level
- 1: wake-up at 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: wake-up at low level
- 1: wake-up at 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: wake-up at low level
- 1: wake-up at 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)

Address 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: wake-up at low level
- 1: wake-up at 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: wake-up at low level
- 1: wake-up at 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: wake-up at low level
- 1: wake-up at 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: wake-up at low level
- 1: wake-up at 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

Address 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: wake-up at low level
- 1: wake-up at 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: wake-up at low level
- 1: wake-up at 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 mainly used to control the internal RF TRX to transmit and reception LoRa signal.

12.2 Main Features

- Support SPI interface to connect with RF TRX
- Support 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 whether 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. Write the REG_NSS bit in register *LORAC_NSS_CR* to 0.
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. Write the REG_NSS bit in register *LORAC_NSS_CR* to 1.

12.3.2 Timing Sequence of Power-on

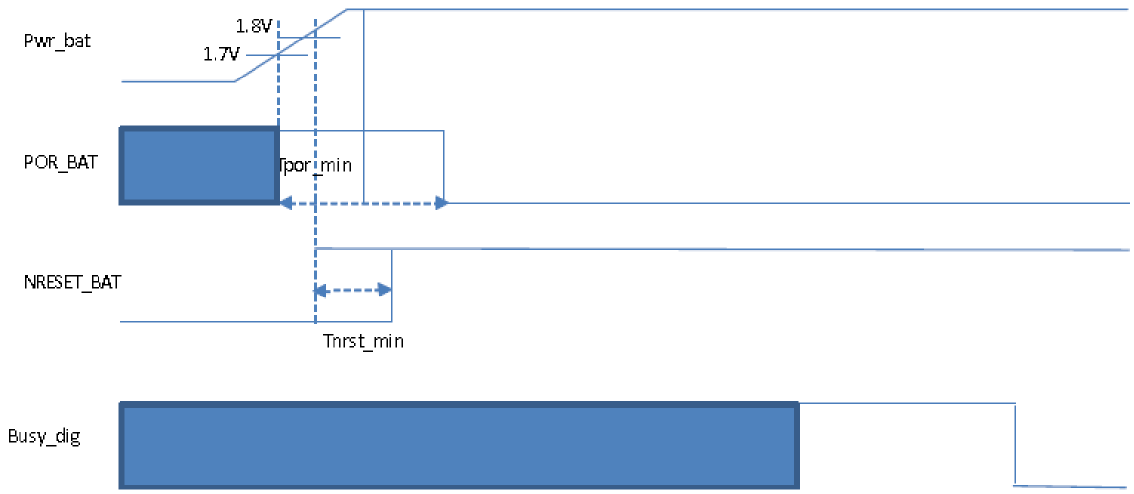

Figure 12-1 Timing Sequence of Power-on

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

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

(2) Set the POR_BAT bit in register *LORAC_CR1* to 0.

(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 this 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 all the time.

12.4 LoRaC Registers

LORAC Base Address: 0x40009000

Table 12-1 LORAC Register Summary

Register Name	Address Offset	Description
SSP_CR0	0x00	LORAC Internal SSP Control Register 0
SSP_CR1	0x04	LORAC Internal SSP Control Register 1
SSP_DR	0x08	LORAC Internal SSP Data Register
SSP_SR	0x0C	LORAC Internal SSP Status Register
SSP_CPSR	0x10	LORAC Internal SSP Clock Prescaler Register
SSP_IMSC	0x14	LORAC Internal SSP Interrupt Mask Set/Clear Register
SSP_RIS	0x18	LORAC Internal SSP Raw Interrupt Status register
SSP_MIS	0x1C	LORAC Internal SSP Masked Interrupt Status register
SSP_ICR	0x20	LORAC Internal SSP Interrupt Clear Register
SSP_DMACR	0x24	LORAC Internal SSP DMA Control Register
RESERVED	0x28-0xFC	Must be kept, and cannot be modified.
LORAC_CR0	0x100	LORAC Control Register 0
LORAC_CR1	0x104	LORAC Control Register 1
LORAC_SR	0x108	LORAC Status Register
LORAC_NSS_CR	0x10C	LORAC NSS Control Register
LORAC_SCK_CR	0x110	LORAC SCK Control Register
LORAC_MOSI_CR	0x114	LORAC MOSI Control Register
LORAC_MISO_SR	0x118	LORAC MISO Status Register

12.4.1 SSP_CR0

Address Offset: 0x00

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 bit
- 4: 5 bit
- 5: 6 bit
- 6: 7 bit
- 7: 8 bit
- 8: 9 bit
- 9: 10 bit
- 10: 11 bit
- 11: 12 bit
- 12: 13 bit
- 13: 14 bit
- 14: 15 bit
- 15: 16 bit

12.4.2 SSP_CR1

Address Offset: 0x04

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

Address Offset: 0x08

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

Address 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 is on going

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

Address Offset: 0x0C

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

Address Offset: 0x00

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

Address Offset: 0x00

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

Address Offset: 0x00

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

Address Offset: 0x00

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, while writing 0 has no effect.

Bit 0 RORIC: RX overrun interrupt clear. This bit is cleared by software writing 1 to it, while writing 0 has no effect.

12.4.10 SSP_DMACR

Address Offset: 0x00

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

Address 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

Address 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: not reset
- 1: reset

Bit 6 RESERVED: Must be kept, and cannot be modified.

Bit 5 NRESET_BAT: NRESET_BAT control

- 0: reset
- 1: not 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

Address 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 all the time.

- 0: no interrupt
- 1: an 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

Address 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

Address 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

Address 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

Address 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.

- **UART mode:**
  - Independent Receive FIFO and transmit FIFO
  - FIFO enable (16 deep) or disable (1 deep)
  - Programmable FIFO trigger levels: $^1/_8$, $^1/_4$, $^1/_2$, $^3/_4$, $^7/_8$
  - Baud rate divisor: 16-bit integer part and 6-bit fractional part
  - Standard asynchronous communication bits: support 5, 6, 7 or 8 data bits, the parity bit and 1 or 2 stop bits
  - Support DMA
  - Support false start bit detection
  - Support line break generation and detection
  - Support hardware flow control
- **IrDA mode:**
  - Support the maximum baud rates (460800 bps) in IrDA mode, and the maximum baud rates (115200 bps) in Low-power IrDA mode (half-duplex)
  - Support normal $^3/_{16}$ and low-power (1.41~2.23 μs) bit durations.
  - Appropriate bit duration generated by the UARTCLK reference clock division in low-power IrDA mode

Each UART port can be uniquely identified by the ID register.

13.2 Clock Reset

Each UART has independent APB bus clock and independent APB bus reset.

13.3 Reference Clock

The frequency of UARTCLK must meet the required range of baud rates:

$$F_{UARTCLK(min)} >= 16 \times baudrate_{(max)}$$

$$F_{UARTCLK(max)} <= 16 \times 65535 \times baudrate_{(min)}$$

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 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 indicator
10	Break error

FIFO Bit	Function
9	Parity error
8	Framing error
7:0	Received data

13.6 UART Operation

13.6.1 Baud Rate Divisor

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:

$$\textbf{\textit{Baud Rate Divisor = UARTCLK / (16 x BautRate) = BRD}}_I \textbf{\textit{ + BRD}}_F$$

where $\mathrm{BRD_I}$ is the integer part and $\mathrm{BRD_F}$ is the fractional part separated by a decimal point as shown below.

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:

$$\textbf{\textit{Fractional Part = integer(BRD}}_F \textbf{\textit{ x 2}}^n \textbf{\textit{ + 0.5)}}$$

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 Data Register (*UARTx_CR*) and configure the data bit, 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 electronics might become biased.

13.7.1 Low-Power Divisor

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*.

$$\textit{Low-Power Divisor} = (F_{UARTCLK} / F_{IrLPBAUD16})$$

$F_{IrLPBAUD16}$ is nominally 1.8432 MHz, which meets the requirement of $\mathbf{1.42MHz < F_{IrLPBAUD16} < 2.12MHz}$.

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.

A 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.

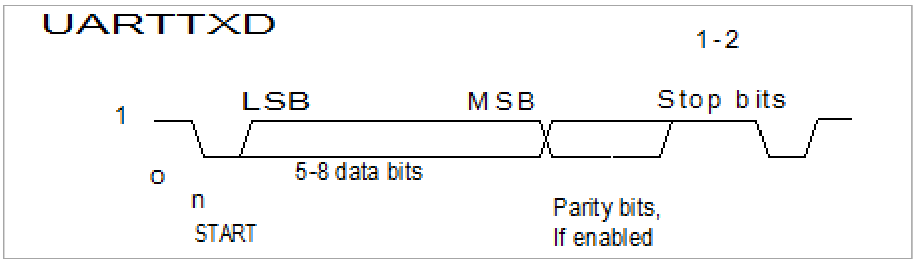

Figure 13-1 UART Character Frame

13.9 IrDA Data Modulation

The effect of IrDA $^{3}/_{16}$ data modulation is shown below.

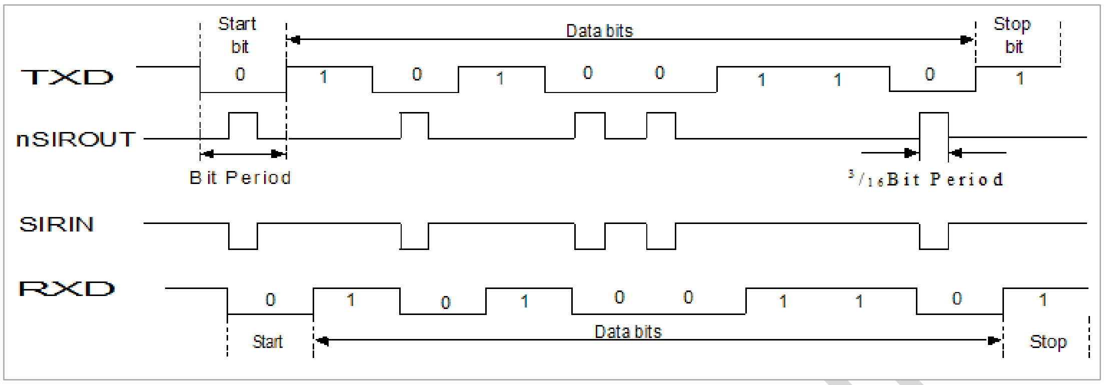

Figure 13-2 IrDA Data Modulation ($^{3}/_{16}$)

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 Register Summary

Register Name	Address Offset	Description
UARTx_DR	0x00	Data Register
UARTx_RSR_ECR	0x04	Receive Status Register/Error Clear Register
UARTx_RSV0[4]	0x08	4 x 4 Bytes reserved
UARTx_FR	0x18	Flag Register
UARTx_RSV1	0x1C	4 Bytes reserved
UARTx_ILPR	0x20	IrDA Low-Power Counter Register
UARTx_IBRD	0x24	Integer Baud Rate Register
UARTx_FBRD	0x28	Fractional Baud Rate Register
UARTx_LCR_H	0x2C	Line Control Register
UARTx_CR	0x30	Control Register
UARTx_IFLS	0x34	Interrupt FIFO Level Selection Register
UARTx_IMSC	0x38	Interrupt Mask Set/Clear Register
UARTx_RIS	0x3C	Raw Interrupt Status Register
UARTx_MIS	0x40	Masked Interrupt Status Register
UARTx_ICR	0x44	Interrupt Clear Register
UARTx_DMACR	0x48	DMA Control Register
UARTx_RSV2[997]	0x4C	4 x 997 Bytes reserved

13.13.1 UARTx_DR (x=0, 1, 2, 3)

Address Offset: 0x00

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)

Address Offset: 0x04

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)

Address Offset: 0x18

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 transmission
- 1: transmission on going

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)

Address Offset: 0x20

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)

Address Offset: 0x24

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)

Address Offset: 0x28

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)

Address Offset: 0x2C

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: Even parity selection

- 0: odd parity
- 1: even parity

This bit has no effect when 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)

Address Offset: 0x30

Reset Value: 0x00000000

31-24		23-16	15	14	13-10
RESERVED		RESERVED	CTSEn	RTSEn	RESERVED
r-0h		r-0h	rw-0h	rw-0h	r-0h
9	**8**	**7-3**	**2**	**1**	**0**
RXE	TXE	RESERVED	SIRLP	SIREN	UARTEN
rw-0h	rw-0h	r-0h	rw-0h	rw-0h	rw-0h

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: Receive enable

- 0: reception disabled. If the UART is disabled in the middle of reception, then it completes the current character before stopping.
- 1: reception enabled

Bit 8 TXE: Transmit enable

- 0: transmission disabled. If the UART is disabled in the middle of transmission, then it completes the current character before stopping.
- 1: transmission 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 in the middle of transmission or reception, then it completes the current character before stopping.

- 1: UART enabled

13.13.9 UARTx_IFLS (x=0, 1, 2, 3)

Address Offset: 0x34

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)

Address Offset: 0x38

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: reception completion interrupt enabled

Bits 3-0 RESERVED: Must be kept, and cannot be modified.

13.13.11 UARTx_RIS (x=0, 1, 2, 3)

Address Offset: 0x3C

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)

Address Offset: 0x40

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)

Address Offset: 0x44

Reset Value: 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: Overrun error interrupt clear

- 0: a write of 0 has no effect
- 1: write 1 to clear overrun error interrupt

Bit 9 BEIC: Break error interrupt clear

- 0: a write of 0 has no effect
- 1: write 1 to clear break error interrupt

Bit 8 PEIC: Parity error interrupt clear

- 0: a write of 0 has no effect
- 1: write 1 to clear parity error interrupt

Bit 7 FEIC: Framing error interrupt clear

- 0: a write of 0 has no effect
- 1: write 1 to clear framing error interrupt

Bit 6 RTIC: Receive timeout interrupt clear

- 0: a write of 0 has no effect
- 1: write 1 to clear receive timeout interrupt

Bit 5 TXIC: Transmission completion interrupt clear

- 0: a write of 0 has no effect
- 1: write 1 to clear transmission completion interrupt

Bit 4 RXIC: Reception completion interrupt clear

- 0: a write of 0 has no effect
- 1: write 1 to 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)

Address Offset: 0x48

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

Address 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

Address 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

Address 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

Address 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

Address 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

Address 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

Address 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

Address 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 transmission rate.

14.2 Main Features

- Master or slave operation
- Support up to 16 MHz output
- Support 16-bit wide TX/RX FIFO with a depth of 8
- Support multiple frame formats
- Support 4-bit to 16-bit data width
- Support DMA
- Support 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 acts as the clock input in slave mode.

3. **SSP_TX**

   The SSP TX pin is used to transmit data in both master and slave modes.

4. **SSP_RX**

   The 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.

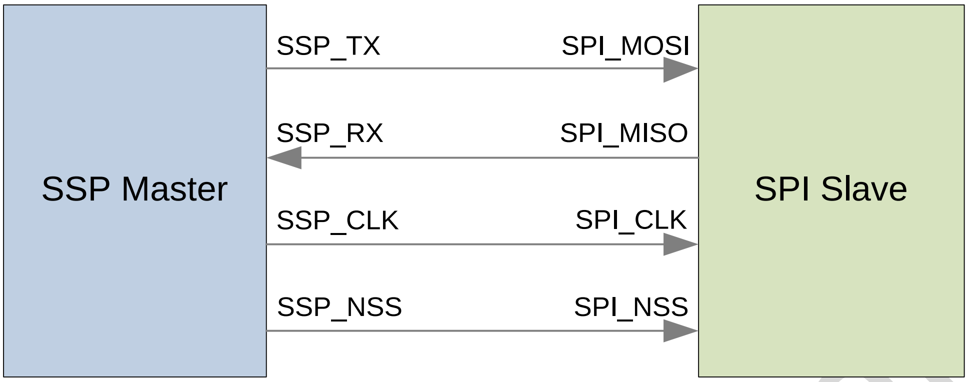

Figure 14-1 Connection between a SSP Master and a SPI Slave

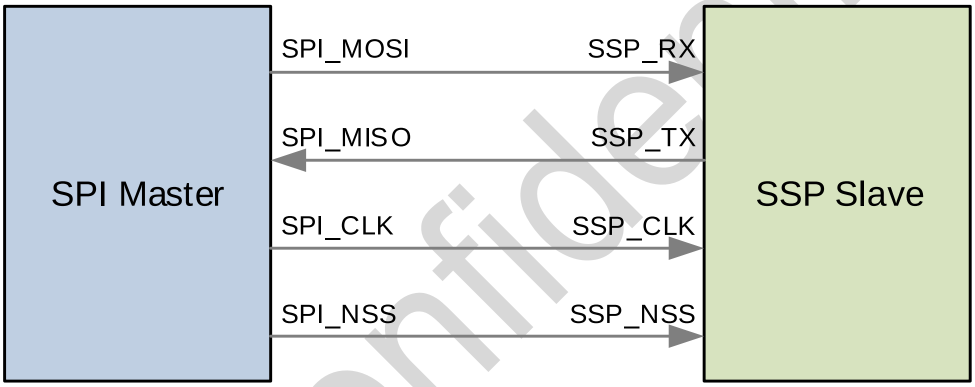

Figure 14-2 Connection between a SPI Master and a SSP Slave

14.3.2 Clock Division

SSP clock should meet below requirements:

(1) Support 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 as follows:

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

Figure 14-3 The Formula to Calculate Clock Output in Master Mode

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 support three frame formats:

- Motorola SPI
- Texas Instruments SPI
- National Semiconductor Microwire

14.3.4 DMA Transaction

SSP DMA Transmission Process

(1) Enable the TXDMAE bit in register *SSP_DMACR*.

(2) Configure register *SSP_DR* 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 SSP Interrupts

There are four SSP interrupt signals.

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 Register Summary

Register Name	Address Offset	Description
SSP_CR0	0x00	SSP Control register 0
SSP_CR1	0x04	SSP Control register 1
SSP_DR	0x08	SSP Data register
SSP_SR	0x0C	SSP Status register
SSP_CPSR	0x10	SSP Clock Prescaler Register
SSP_IMSC	0x14	SSP Interrupt Mask Set/Clear Register
SSP_RIS	0x18	SSP Raw Interrupt Status register
SSP_MIS	0x1C	SSP Masked Interrupt Status register
SSP_ICR	0x20	SSP Interrupt Clear Register
SSP_DMACR	0x24	SSP DMA Control Register

14.4.1 SSP_CR0

Address Offset: 0x00

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 bit
- 4: 5 bit
- 5: 6 bit
- 6: 7 bit
- 7: 8 bit
- 8: 9 bit
- 9: 10 bit
- 10: 11 bit
- 11: 12 bit
- 12: 13 bit
- 13: 14 bit
- 14: 15 bit
- 15: 16 bit

14.4.2 SSP_CR1

Address Offset: 0x04

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

Address Offset: 0x08

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

Address 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 is on going

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

Address Offset: 0x0C

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

Address Offset: 0x00

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

14.4.7 SSP_RIS

Address Offset: 0x00

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

Address Offset: 0x00

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

Address Offset: 0x00

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, while writing 0 has no effect.

Bit 0 RORIC: RX overrun interrupt clear. This bit is cleared by software writing 1 to it, while writing 0 has no effect.

14.4.10 SSP_DMACR

Address Offset: 0x00

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 main features of I2C (inter-integrated circuit) bus interface are as follows:

- Supports master mode and slave mode.
- SDA is the data transmission line, and SCL is the reference clock line
- Supports multi-master and bus arbitration
- Support standard mode (up to 100 Kbps) and fast mode (up to 400 Kbps)
- Support FIFO mode with configurable read and write pointers, TxFIFO depth is 8, and RxFIFO depth is 16

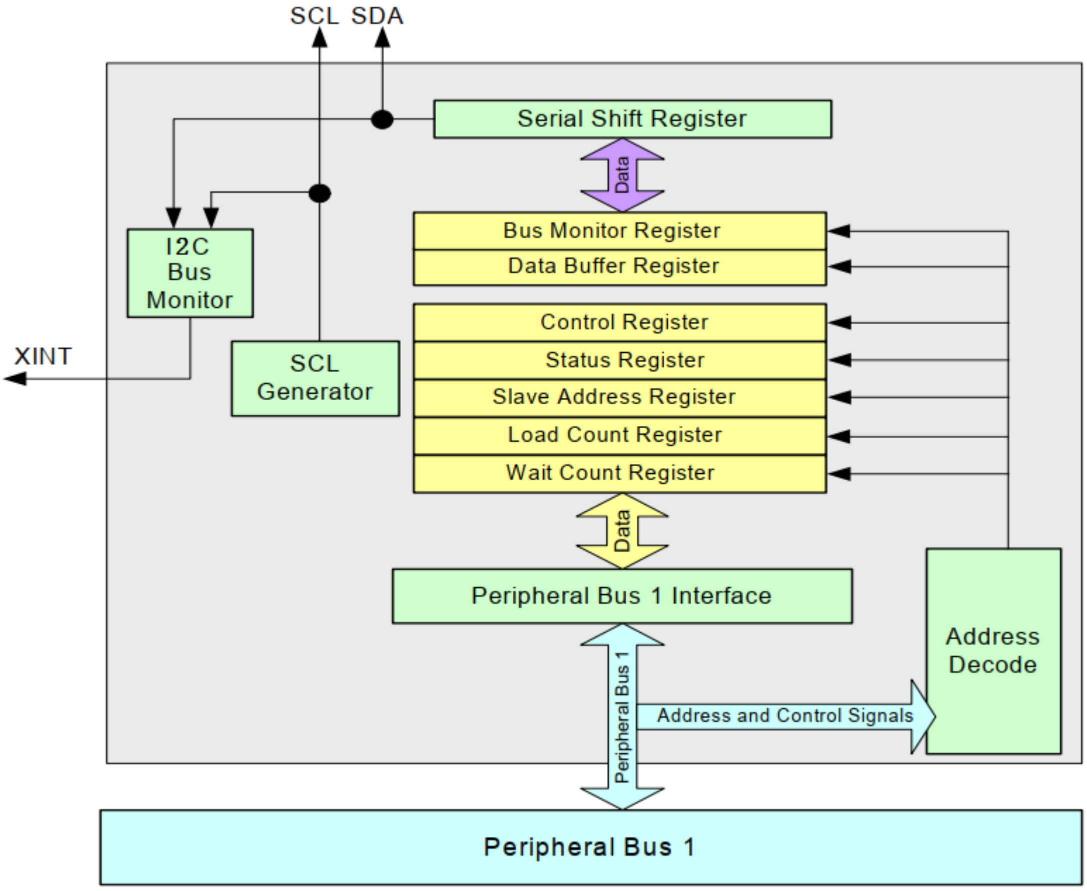

Figure 15-1 I2C Block Diagram

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.

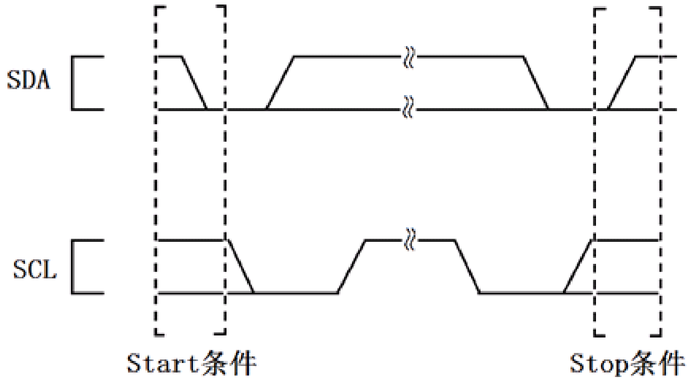

Figure 15-2 SDA and SCL Signals During Start and Stop Conditions

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 Bit	Stop Bit	Condition Type	Description
0	0	No Start or Stop	When multiple data bytes are to be transmitted, I2C will not send a Start or Stop condition.
0	1	Start or Repeated Start	I2C sends a Start condition and then transmits the 8-bit data in the *I2Cx_DBR* register. Before a Start condition is sent, register *I2Cx_DBR* must contain the 7-bit slave address and the R/nW bit. For a Repeated Start, the *I2Cx_DBR* register must contain the target slave address and the R/nW bit, which allows a master to perform multiple transfers without freeing the bus. The interface stays in master transmit mode for writes, and switches to master receive mode for reads.
1	x	Stop	In master transmit mode, a Stop condition is sent on the I2C bus after the 8-bit data in the *I2Cx_DBR* register has been transferred. In master receive mode, bit ACKNAK in the *I2Cx_CR* register must be set to send a NAK. The received data is stored in the *I2Cx_DBR* register, and then a Stop condition is sent on the I2C bus.

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 Transmission Sequence

I2C transmits data in 1-byte increments and follows below sequence:

1. Start
2. 7-bit slave address
3. R/nW bit
4. Acknowledge pulse
5. 8 bits of data
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. **When the 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.

   Notice: *In FIFO mode, software writes to the TX FIFO instead of the I2Cx_DBR register.*

2. **When the 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.

   Notice: *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.

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	Mode of Operation	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 target 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.

I2C Master Action	Mode of Operation	Definition
Write one data byte to I2Cx_DBR	Master transmit only	● When the ITE bit in the I2Cx_SR register is set and the TB bit in the I2Cx_CR register is cleared, if enabled, the I2Cx_DBR transmit-empty interrupt is generated. ● The CPU writes a data byte to the I2Cx_DBR register, sets the appropriate Start/Stop condition combination as required, and sets the TB bit in register I2Cx_CR to send data. The 8 bits of data are transferred from the shift register to the serial bus. If the STOP bit in register I2Cx_CR is set before the transfer, then the 8 bits of data is followed by a Stop condition.
Wait for ACK from slave receiver	Master transmit only	As a master transmitter, the I2C generates the ACK clock, and releases the SDA line for the slave receiver to transmit the ACK.
Read one byte from I2Cx_DBR	Master receive only	After the ACKNAK bit in register I2Cx_CR is read, the 8 bits of data in the shift register is transferred to the I2Cx_DBR register, ● The CPU reads the I2Cx_DBR register when the IRF bit in register I2Cx_SR is set and the TB bit in register I2Cx_CR is cleared. If the I2Cx_DBR receive-full interrupt is enabled, it is signalled to the CPU. ● When the I2Cx_DBR register is read, if the ACKNAK bit in register I2Cx_SR is cleared (indicating ACK), the software must clear the ACKNAK bit and set the TB bit in register I2Cx_CR to start the next byte Read. ● If the ACKNAK bit in I2Cx_SR is set (indicating NAK), the TB bit in I2Cx_CR is cleared, the STOP bit in I2Cx_CR is set, and the UB bit in I2Cx_SR is set, then the last byte has been read from the I2Cx_DBR register, and the I2C is sending the Stop. ● If the ACKNAK bit in I2Cx_SR is set (indicating NAK) and the TB bit in I2Cx_CR is cleared, but the STOP bit in I2Cx_CR is cleared, then the software has two options: 1. Set the START bit in I2Cx_CR, write a new target slave address to the I2Cx_DBR register, set the TB bit in I2Cx_CR, and send a Repeated Start. 2. Set the MA bit in I2Cx_CR and keep the TB bit as 0 in I2Cx_CR, and only send a Stop.
Transmit ACK to slave transmitter	Master receive only	● As a master receiver, the I2C generates the ACK clock and drives the SDA line during the ACK cycle. ● If the next data byte is the last transaction, the user software sets the ACKNAK bit in register I2Cx_CR to generate NAK.
Generate a Repeated Start	Master transmit Master receive	Use a Repeated Start condition instead of a Stop condition to initiate a new transaction without releasing the bus. ● The Repeated Start is generated after the last data byte has been transmitted. ● Software must write the next 7-bit target slave address and the R/nW bit to the I2Cx_DBR register, set the START bit in register I2Cx_CR, and set the TB bit in register I2Cx_CR.

I2C Master Action	Mode of Operation	Definition
Generate a Stop	Master transmit Master receive	● A Stop is generated after the last data byte has been transmitted. ● The STOP bit in register *I2Cx_CR* must be set before the transmission of the last byte.

15.8 FIFO Mode

The FIFO mode can only be used when the I2C is in *Master Mode*.

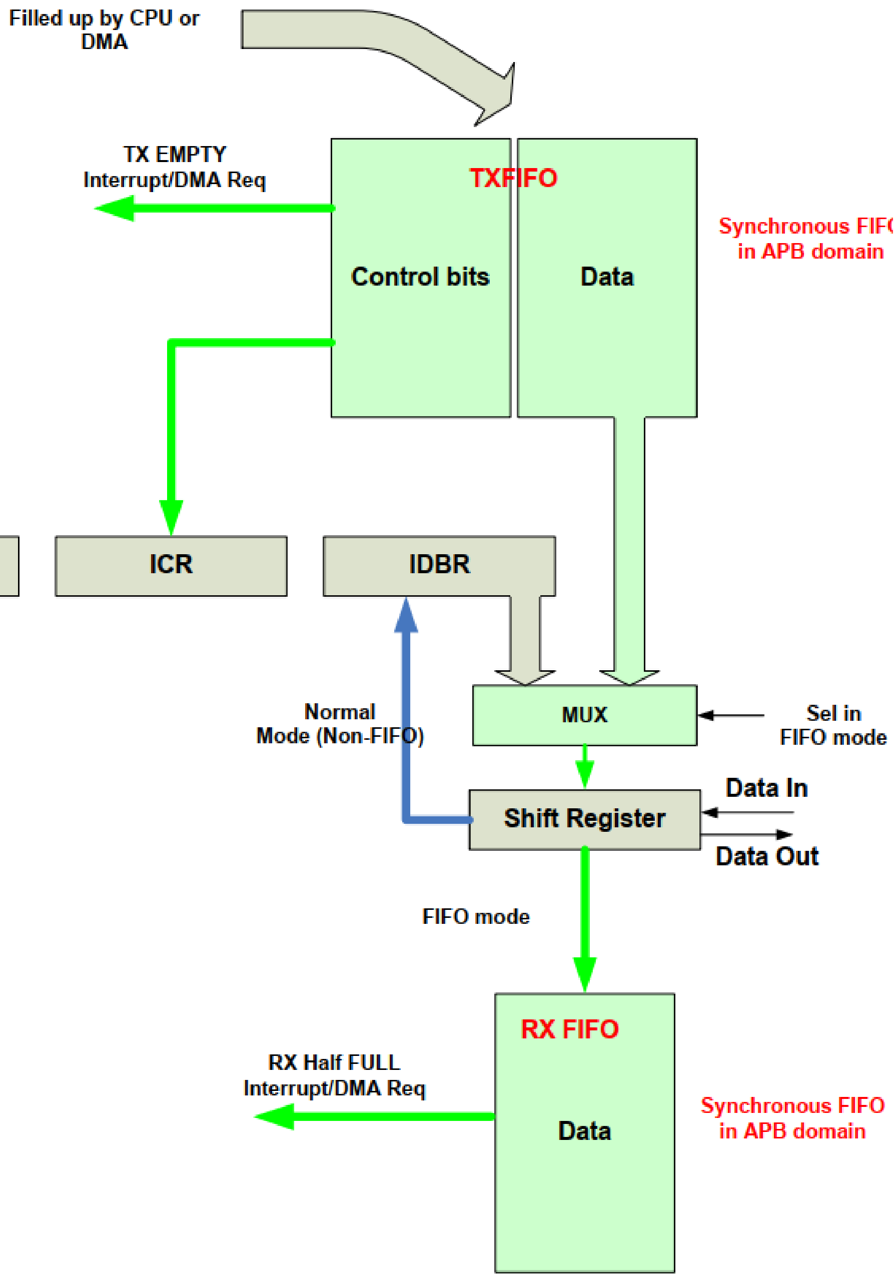

Figure 15-3 FIFO Mode Block Diagram

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 Transaction

Slave Operation	Type	Description
Slave receive mode (default)	Slave receive only	● The I2C monitors all slave address transactions. ● The UE bit in register *I2Cx_CR* must be set. ● I2C monitors the Start conditions on the bus. When a Start condition is detected, the interface reads the first 8 bits of data and compares the most significant 7 bits with those in the *I2Cx_SAR* register. If they match, the I2C sends an ACK. ● If the 8th bit (R/nW bit) of the first byte is low, then I2C stays in slave receive mode, and the SAD bit in register *I2Cx_SR* is cleared. If the R/nW bit is high, I2C switches to slave transmit mode and sets the SAD bit in register *I2Cx_SR*.
Set the slave address detection bit	Slave receive Slave transmit	● Indicates that the interface has defected the matching slave address ● If enabled, an slave address detection interrupt is generated after the matching slave address is received and acknowledged, and the SAD bit in register *I2Cx_SR* is set.
Read one byte from *I2Cx_DBR*	Slave receive only	● Eight bits are read from the serial bus into the shift register. When a full byte has been received and the ACK/NAK bit is completed, the byte in the shift register is transferred to the I2Cx_DBR register. ● When the IRF bit in register *I2Cx_SR* is set, and the TB bit in register *I2Cx_CR* is cleared, if enabled, the I2Cx_DBR receive-full interrupt is generated. ● Software reads one data byte from the I2Cx_DBR register, then configures the ACKNAK bit in register *I2Cx_CR* as required and sets the TB bit in register *I2Cx_CR*. This makes the I2C exit from the wait state and continue to receive data from the master transmitter.
Transmit ACK to master transmitter	Slave receive only	● As a slave receiver, the I2C pulls the SDA line low to generate the ACK when SCL is high. ● ACK/NAK is controlled by bit ACKNAK in register *I2Cx_CR*.
Write one byte to *I2Cx_DBR*	Slave transmit only	● When the ITE bit in register *I2Cx_SR* is set and the TB bit in register *I2Cx_CR* is cleared, if enabled, the I2Cx_DBR transmit-empty interrupt is generated. ● Software writes a byte into register *I2Cx_DBR* and then sets the TB bit in register *I2Cx_CR* to start the transmission.

Waiting for ACK from master receiver	Slave transmit only	As a slave transmitter, the I2C releases the SDA line for the master receiver to pull the line low to transmit the ACK.

15.10 I2C Clock 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 I2C 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 Requests

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 Register Summary

Register Name	Address Offset	Description
I2Cx_CR	0x00	Control Register
I2Cx_SR	0x04	Status Register
I2Cx_SAR	0x08	Slave Address Register
I2Cx_DBR	0x0C	Data Buffer Register
I2Cx_LCR	0x10	Load Count Register
I2Cx_WCR	0x14	Wait Count Register
I2Cx_RST_CYCL	0x18	Reset SCL Cycle
I2Cx_BMR	0x1C	Bus Monitor Register
I2Cx_WFIF0	0x20	Write FIFO Register
I2Cx_WFIFO_WPTR	0x24	Write FIFO Write Pointer Register
I2Cx_WFIFO_RPTR	0x28	Write FIFO Read Pointer Register
I2Cx_RFIFO	0x2C	Read FIFO Register
I2Cx_RFIFO_WPTR	0x30	Read FIFO Write Pointer Register
I2Cx_RFIFO_RPTR	0x34	Read FIFO Read Pointer Register
I2Cx_RESV[2]	0x38	Reserved
I2Cx_WFIFO_STATUS	0x40	Write FIFO Status Register
I2Cx_RFIFO_STATUS	0x44	Read FIFO Status Register

15.13.1 I2Cx_CR (x=0, 1, 2)

Address Offset: 0x00

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 enable

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: a Stop condition is generated when the STOP bit in this register is set
- 1: a 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: invalid
- 1: I2C bus reset, and this bit is cleared automatically

Bit 10 UR: Unit reset

- 0: no reset
- 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 transaction begins
- 1: a 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 a ACK after receiving a data byte
- 1: send a 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 Stop condition is generated
- 1: generate a 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 Start condition is generated
- 1: generate a Start condition

This bit is used to generate a Start condition on the I2C bus in master mode.

15.13.2 I2Cx_SR (x=0, 1, 2)

Address Offset: 0x04

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 is not done
- 1: transaction is done, and it is cleared by software writing 1 to it.

Bit 26 MSD: Master Stop detection flag (only effective in master mode)

- 0: no master Stop was detected
- 1: a master Stop was 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 was detected
- 1: a 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 was detected
- 1: the matching slave address was detected, and it is cleared by software writing 1 to it.

Bit 22 BED: Bus error detection flag

- 0: no bus error was detected
- 1: a bus error was 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 is still being transmitted.
- 1: 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: the I2C wins the arbitration or no arbitration took place
- 1: the 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: the I2C bus is idle, or the ASR6601 I2C is using the bus
- 1: the I2C bus is busy but not used by the ASR6601 I2C

Bit 15 UB: I2C unit busy flag

- 0: the I2C unit is idle
- 1: the I2C unit is busy

Bit 14 ACKNAK: ACK/NAK status flag

- 0: the I2C received or sent an ACK
- 1: the 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)

Address Offset: 0x08

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)

Address Offset: 0x0C

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)

Address Offset: 0x10

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)

Address 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)

Address Offset: 0x18

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)

Address Offset: 0x1C

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)

Address Offset: 0x20

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)

Address Offset: 0x24

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)

Address Offset: 0x28

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)

Address Offset: 0x2C

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)

Address Offset: 0x30

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)

Address Offset: 0x34

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)

Address Offset: 0x40

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)

Address Offset: 0x44

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.

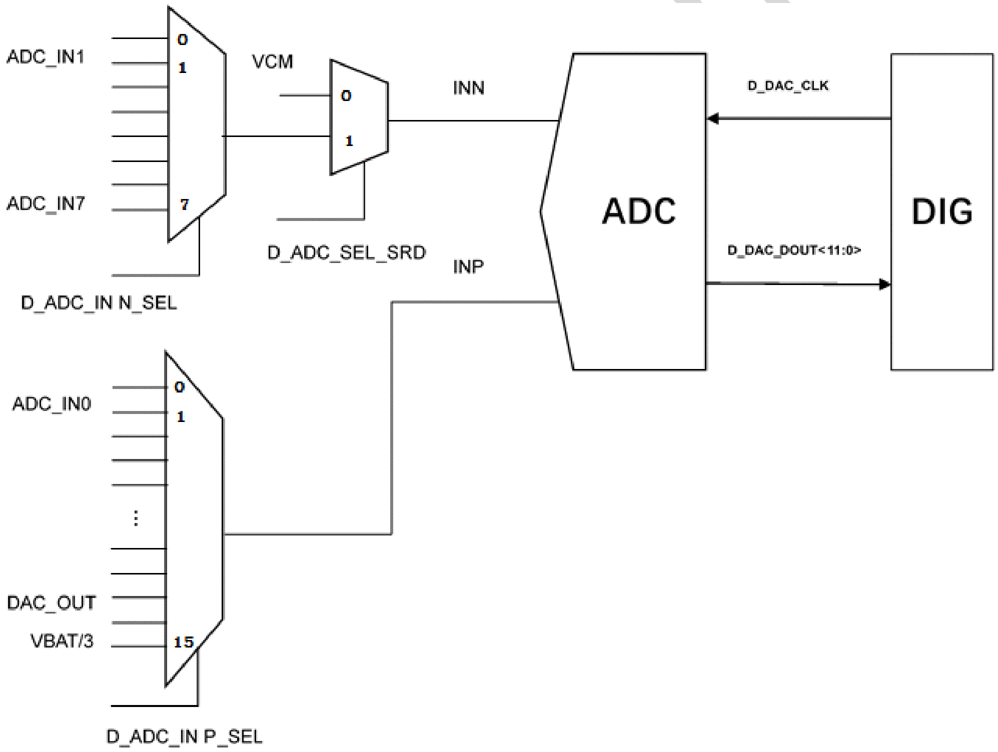

Figure 16-1 ADC Diagram

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

Sampling Channel No.	Signal	Remark
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	Internal channel
10	OPA1_ADC_OUT	Internal channel
11	OPA2_ADC_OUT	Internal channel
12	DCTEST_OUT	Internal channel
13	TD_OUT_TEST	Internal channel
14	DAC_CORE_AOUT	Internal channel
15	VBAT31	Internal channel

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

- **By software:** The conversion starts immediately when a rising edge on the START bit of ADC_CR is detected.

- **By hardware:** The conversion is triggered by Timer or IO, containing 10 selectable trigger sources with a configurable level.

The trigger mode is selected through the TRIG_SEL bit in register *ADC_CFGR* and the external trigger source is selected through the EXT_TRIG_SEL bit in register *ADC_CFGR*.

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 ADC 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 re-starts 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 read from the data buffer.

16.10 DMA Request

When the 12-bit data buffer is full, the DMA request is genetated if the DMA_EN bit in register ADC_CFGR is set. And the DMA request is disabled by writing '0' to the DMA_EN bit.

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 ADC 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 Register Summary

Register Name	Address Offset	Description
ADC_CR	0x00	ADC Control Register
ADC_CFGR	0x04	ADC Configuration Register
ADC_SEQR0	0x08	ADC Sequence Register 0
ADC_SEQR1	0x0C	ADC Sequence Register 1
ADC_DIFFSEL	0x10	ADC Differential Mode Selection Register
ADC_ISR	0x14	ADC Interrupt and Status Register
ADC_IER	0x18	ADC Interrupt Enable Register
ADC_DR	0x1C	ADC Data Register

16.14.1 ADC_CR

Address 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: A write of 0 has no effect.
- 1: Write 1 to stop the ADC. Read 1 means that the STOP command is under execution.

Notice:

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: A write of 0 has no effect.
- 1: Write 1 to 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:

- 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.
- In discontinuous conversion mode, if software trigger is selected, the START bit is cleared when the EOC flag in register *ADC_ISR* is set.
- 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: A write of 0 has no effect.
- 1: Write 1 to disable the ADC

Software is allowed to set this bit only when EN=1 and START=0 (no conversion is ongoing).

Bit 0 EN: ADC enable

- 0: A write of 0 has no effect.
- 1: Write 1 to 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

Address Offset: 0x04

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*).

Notice:

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: hardware trigger detection on the falling edge
- 11: hardware trigger detection on both edges

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

Notice:

(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

Notice:

(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

Address Offset: 0x08

Reset Value: 0x00000000

Notice: *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
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

Address Offset: 0x0C

Reset Value: 0x00000000

Notice: *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

Address Offset: 0x10

Reset Value: 0x00000000

Notice: *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.

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

16.14.6 ADC_ISR

Address Offset: 0x14

Reset Value: 0x00000000

Notice: *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 occurred
- 1: overrun has 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 is not complete
- 1: conversion sequence is complete

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 is not completed
- 1: channel conversion is 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

Address Offset: 0x18

Reset Value: 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

Address Offset: 0x1C

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. RTC

17.1 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.

17.2 Main Features

The main features of RTC are as follows:

1. 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)
2. Support RTC frequency calibration with a resolution of about 0.5 ppm with a range from -1024 ppm to +1024 ppm.
3. Support wake-up from low-power mode
4. Tamper/wakeup IO detection is activated at high or low level with configurable filter
5. 32-bit counter for periodic count
6. 2 Alarms support calendar matching
7. Retention SRAM is cleared once tamper/wakeup alarm occurs
8. Internal signal is output by GPIO, including Alarm0 pulse, Alarm1 pulse, periodic counter pulse and second signal
9. Support reading calendar values
10. Support reading the sub-seconds value
11. Support reading the period counting value
12. Support interrupt signal generation

17.3 Interface Clock

Both XO32K and RCO32K can be RTC clock source and XO32K accuracy is higher than RCO32K.

See Chapter *RCC* for clock configuration details.

17.4 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* and *RTC_SYNDATA_H* are **asynchronous 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.

17.4.1 Reading the Calendar

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

17.4.2 Setting the Calendar

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 sub-second 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.

17.5 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.

17.6 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 17-1 RTC Wake-up Source

Mode	Description
Sleep	RTC interrupts can wake up the device from the Sleep mode.
Stop0/Stop1/Stop2/Stop3	RTC wakeup, RTC tamper event, RTC alarm, and periodic count signal can wake up the device from Stop mode.
Standby	RTC wakeup, RTC tamper event, RTC alarm, and periodic count signal can wake up the device from Standby mode.

Enable the wakeup/tamper IO, RTC alarm and periodic count signal for wake-up through the corresponding bit in register *RTC_CR*:

Table 17-2 Bits to Enable Wake-up Signals

Function	Bits in Register RTC_CR
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

17.7 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

17.7.1 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.

17.7.2 Erase Operation on Retention SRAM

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.

17.8 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.

17.9 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*.

17.10 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.

17.11 RTC Interrupts

Table 17-3 RTC Interrupts

Interrupt	Description
Alarm 0 interrupt	Interrupt is generated when the interval set by Alarm 0 is reached.
Alarm 1 interrupt	Interrupt is generated when the interval set by Alarm 1 is reached.
Periodic wakeup interrupt	Interrupt is generated at regular intervals.
Tamper interrupt	Interrupt is generated when an input event is detected by tamper IO.
Wakeup IO0 interrupt	Interrupt is generated when an input event is detected by Wakeup IO0 IO.
Wakeup IO1 interrupt	Interrupt is generated when an input event is detected by Wakeup IO1 IO.
Wakeup IO2 interrupt	Interrupt is generated when an input event is detected by Wakeup IO2 IO.
Second signal interrupt	Interrupt is generated by the second signal every second.

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.

17.12 RTC Registers

Base Address: 0x4000E000

Table 17-4 RTC Register Summary

Register Name	Address Offset	Description
RTC_CR	0x00	RTC Control Register 1
RTC_ALARM0	0x04	RTC Alarm 0 Register
RTC_ALARM1	0x08	RTC Alarm 1 Register
RTC_PPMADJUST	0x0c	RTC PPMADJUST Register
RTC_CALENDAR	0x10	RTC Calendar Configuration Register (second, minute, hour)
RTC_CALENDAR_H	0x14	RTC Calendar Configuration Register (date/day of week, month, year)
RTC_CYC_MAX_VALUE	0x18	RTC Periodic Counter Value Configuration Register
RTC_SR	0x1c	RTC Status Register
RTC_ASYNDATA	0x20	RTC Calendar ASYNDATA Register (second, minute, hour)
RTC_ASYNDATA_H	0x24	RTC Calendar ASYNDATA Register (date/day of week, month, year)
RTC_CR1	0x28	RTC Control register 1 (interrupt enable)
RTC_SR1	0x2c	RTC Status Register 1
RTC_CR2	0x30	RTC Control register 2
RTC_SUB_SECOND	0x34	RTC Sub-second Register
RTC_CYC_CNT_VALUE	0x38	RTC Periodic Counter Value Register (read-only)
RTC_ALARM0_SUB	0x3c	RTC Alarm 0 Sub-second Register
RTC_ALARM1_SUB	0x40	RTC Alarm 1 Sub-second Register
RTC_CALENDAR_R	0x44	RTC Calendar SYNDATA Register (second, minute, hour)
RTC_CALENDAR_R_H	0x48	RTC Calendar SYNDATA Register (date/day of week, month, year)

17.12.1 RTC_CR

Address Offset: 0x00

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: Reserved.

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: active at low level
- 1: active at 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: no filter
- 1: the filter length is 1 RTC interface clock cycle
- 2: the filter length is 3 RTC interface clock cycles
- 3: the 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: active at low level
- 1: active at 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: no filter
- 1: the filter length is 1 RTC interface clock cycle
- 2: the filter length is 3 RTC interface clock cycles
- 3: the 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: active at low level
- 1: active at 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: no filter
- 1: the filter length is 1 RTC interface clock cycle
- 2: the filter length is 3 RTC interface clock cycles
- 3: the 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: active at low level
- 1: active at 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: no filter
- 1: the filter length is 1 RTC interface clock cycle
- 2: the filter length is 3 RTC interface clock cycles
- 3: the filter length is 7 RTC interface clock cycles

17.12.2 RTC_ALARM0

Address Offset: 0x04

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
- 1: 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

17.12.3 RTC_ALARM1

Address Offset: 0x08

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

17.12.4 RTC_PPMADJUST

Address Offset: 0x0c

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

17.12.5 RTC_CALENDAR

Address Offset: 0x10

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

17.12.6 RTC_CALENDAR_H

Address Offset: 0x14

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

[21:18]: year tens

17.12.7 RTC_CYC_MAX_VALUE

Address Offset: 0x18

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.

17.12.8 RTC_SR

Address Offset: 0x1c

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: the Alarm 0 values doesn't match the Calendar
- 1: the 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: the Alarm 1 values doesn't match the Calendar
- 1: the 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: the CYC_MAX_VALUE is not reached
- 1: the 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: the tamper pin active level is not detected
- 1: the 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: the Wakeup1 active level is not detected
- 1: the 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: the Wakeup2 active level is not detected
- 1: the Wakeup2 active level is detected

17.12.9 RTC_ASYNDATA

Address Offset: 0x20

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

17.12.10 RTC_ASYNDATA_H

Address Offset: 0x24

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

17.12.11 RTC_CR1

Address Offset: 0x28

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

17.12.12 RTC_SR1

Address Offset: 0x2c

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: a write operation is ongoing
- 1: a write operation is 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: a write operation is ongoing
- 1: a write operation is 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 is generated
- 1: a second signal interrupt is generated

Bit 8 WRITE_RTCCR2_DONE: The complete flag of the write operation to register *RTC_CR2*. This bit is set and cleared by hardware.

- 0: a write operation is ongoing
- 1: a write operation is 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: a write operation is ongoing
- 1: a write operation is 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: a write operation is ongoing

- 1: a write operation is 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: a write operation is ongoing
- 1: a write operation is 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: a write operation is ongoing
- 1: a write operation is 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: a write operation is ongoing
- 1: a write operation is 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: a write operation is ongoing
- 1: a write operation is 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: a write operation is ongoing
- 1: a write operation is 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: a write operation is ongoing
- 1: a write operation is completed

17.12.13 RTC_CR2

Address Offset: 0x30

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

17.12.14 RTC_SUB_SECOND

Address Offset: 0x34

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.

17.12.15 RTC_CYC_CNT_VALUE

Address Offset: 0x38

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.

17.12.16 RTC_ALARM0_SUB

Address Offset: 0x3c

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.

…

- 14: RTC_ALARM0_SUB_VALUE [14] are not involved in Alarm 0 comparison. Bits[13:0] are compared.
- 15: All 15 RTC_ALARM0_SUB_VALUE bits are compared.

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.

17.12.17 RTC_ALARM1_SUB

Address Offset: 0x40

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.

…

- 14: RTC_ALARM1_SUB_VALUE [14] are not involved in Alarm 1 comparison. Bits[13:0] are compared.

- 15: All 15 RTC_ALARM1_SUB_VALUE bits are compared.

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.

17.12.18 RTC_CALENDAR_R

Address Offset: 0x44

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.

17.12.19 RTC_CALENDAR_R_H

Address Offset: 0x48

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.

18. LPUART

18.1 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.

In addition, LPUART supports CTS (Clear To Send)/RTS (Require To Send) flow control.

DMA (direct memory access) can be used for data transmission and reception.

18.2 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)
- Support DMA request
- 1-deep TX FIFO/RX FIFO
- Support CTS/RTS flow control
- Support LPUART interrupt generation
- Wakeup CPU from low-power modes

18.3 Functional Description

18.3.1 Data Format

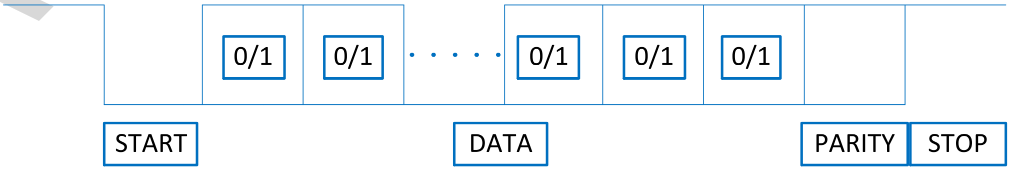

Figure 18-1 LPUART Data Format

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.

18.3.2 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.

18.3.3 CTS/RTS Flow Control

The connection between two LPUART devices is shown in the following figure:

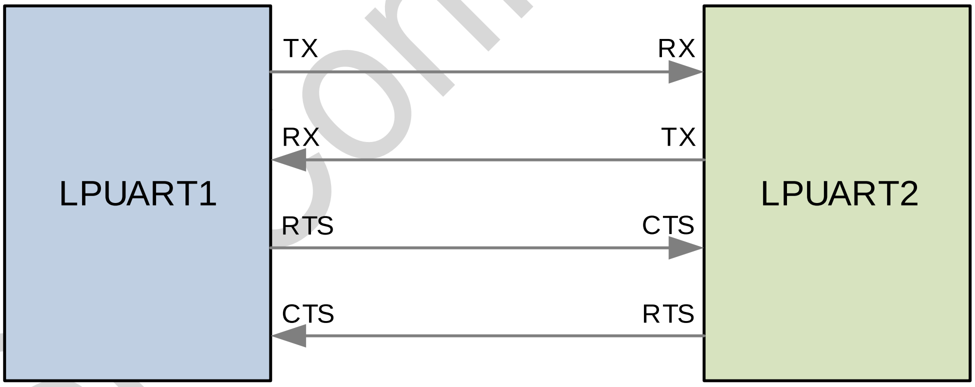

Figure 18-2 Connection between Two LPUART Devices

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.

18.3.4 DMA Transaction

LPUART DMA Transmitter 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 Reception 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.

18.3.5 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

18.3.6 CPU Wakeup from Low-power Mode

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*.

18.4 LPUART Registers

LPUART Base Address: 0x40005000

Table 18-1 LPUART Register Summary

Register Name	Address Offset	Description
LPUART_CR0	0x00	LPUART Control Register 0
LPUART_CR1	0x04	LPUART Control Register 1
LPUART_SR0	0x08	LPUART Status Register 0
LPUART_SR1	0x0C	LPUART Status Register 1
LPUART_DATA	0x10	LPUART Data Register

18.4.1 LPUART_CR0

Address 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

Baud rate divisor=UART interface clock frequency/Baud rate

Taking an LPUART interface clock frequency of 32.768 kHz 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 **6 or 7** (based on 0.413*16=6.608) through the LPUART_BAUD_RATE_FRA bit.

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

18.4.2 LPUART_CR1

Address Offset: 0x04

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

18.4.3 LPUART_SR0

Address Offset: 0x08

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 occurred
- 1: an 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 occurred
- 1: a 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 occurred
- 1: a 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: an 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 is not completed
- 1: data reception is 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: a valid Start bit has been received

18.4.4 LPUART_SR1

Address Offset: 0x0C

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 is on-going
- 1: data transmission is 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: a write operation to the *LPUART_CR0* register is on-going
- 1: a write operation to the *LPUART_CR0* register has been 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: a write operation to the *LPUART_SR0* register is on-going
- 1: a write operation to the *LPUART_SR0* register has been completed

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

18.4.5 LPUART_DATA

Address Offset: 0x10

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

Notice:

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.*