ASR6601

参考手册

文档版本 1.2.0

发布日期 2021-07-22

关于本文档

本文档主要提供 IoT LPWAN SoC 芯片 ASR6601 进行软硬件开发的参考信息。

读者对象

本文档主要适用于以下工程师:

- 单板硬件开发工程师
- 软件工程师
- 技术支持工程师

产品型号

与本文档相对应的产品型号如下:

型号	Flash	SRAM	内核	封装	频率
ASR6601SE	256 KB	64 KB	32-bit 48 MHz ARM ARM STAR	QFN68, 8*8 mm	150 ~ 960 MHz
ASR6601CB	128 KB	16 KB	32-bit 48 MHz ARM ARM STAR	QFN48, 6*6 mm	150 ~ 960 MHz

版权公告

商标声明

免责声明

翱捷科技股份有限公司对本文档内容不做任何形式的保证，并会对本文档内容或本文中介绍的产品进行不定期更新。

本文档仅作为使用指导，本文的所有内容不构成任何形式的担保。本文档中的信息如有变更，恕不另行通知。

本文档不负任何责任，包括使用本文档中的信息所产生的侵犯任何专有权行为的责任。

翱捷科技股份有限公司

地址：上海市浦东新区科苑路399号张江创新园10号楼9楼　邮编：201203

官网：http://www.asrmicro.com/asrweb/

文档修订历史

日期	版本号	发布说明
2021.03	1.0.0	首次发布。
2021.05	1.1.0	● 更新第 6 章的综述和表 6-1。 ● 更新 16.3 节、16.9 节和 16.14.1 节的部分描述。 ● 修正 12.4.3 节寄存器 LORAC_SR 的描述。
2021.07	1.2.0	更新了关于 CPU 的描述。

目录

表格

插图

1. 概述

ASR6601 是由翱捷科技股份有限公司研发的首颗支持 LoRa 调制方式的国产低功耗广域网无线通信 SoC。ASR6601 系列适用于智能表计、智能物流、智能建筑、智慧城市、智慧农业等诸多应用场景。在 ASR6601 的内部集成了 Sub 1GHz 的射频收发机、ARM STAR 微处理器、内置 Flash 存储、内置 SRAM 及众多功能的模拟模块。本参考手册主要是面向 SoC 产品应用开发人员，包含软件及硬件开发者。根据本手册中的信息配合手册最后附录的部分寄存器表格及 SDK 中的 API，可以满足工程师在开发过程中的绝大部分问题；如需其他帮助请联系翱捷科技的工程师。本手册会持续更新。

2. ASR6601 简介

ASR6601 SoC 是首颗国产的支持 LoRa 调制方式的 LPWAN SoC。ASR6601 芯片中集成的超低功耗收发机，配合片外匹配网络可以支持 150 MHz ~ 960 MHz 全频段工作。除了支持 LoRa 调制方式外，还可以支持 FSK 收发、MSK 收发和 BPSK 发射等。在 3.3 V 电源供电的情况下，通过高功率 PA，最大可发射 22 dBm 的输出功率。

ASR6601 SoC 主要有 Run、LpRun、Sleep、LpSleep、Stop0、Stop1、Stop2、Stop3、Standby 几种工作模式。每种模式支持的功能，工作的模块和功耗各不相同。终端用户可以根据自己的应用场景选择相应的工作模式。其中比较常用的两种低功耗模式为 Standby 模式和 Stop3 模式，在 3.3 V 电源供电的情况下 Standby 模式功耗低至 0.9 uA；Stop3 模式功耗低至 1.3 uA（ASR6601CB）和 1.6 uA（ASR6601SE）。

ASR6601 SoC 采用的 32 位 ARM STAR 内核，最高主频 48 MHz，支持 SWD 调试接口，支持 SysTick、MPU、FPU 功能，同时支持 37 个 IRQ，有 8 个中断优先级。

ASR6601 支持 UART，I2C，I2S，LPUART，SSP，QSPI 等接口，配合不同种类的对应接口的外设可以实现丰富的功能满足客户需求。ASR6601 除了支持丰富的数字功能外，也集成丰富的模拟功能，包含 ADC、DAC、OPA 以及 LCD 驱动等。

ASR6601 通过硬件方式实现 AES 加密，大大简化了加解密的效率。同时还支持国密 SM2/3/4。

3. 模块及功能

3.1 ASR6601 模块框图

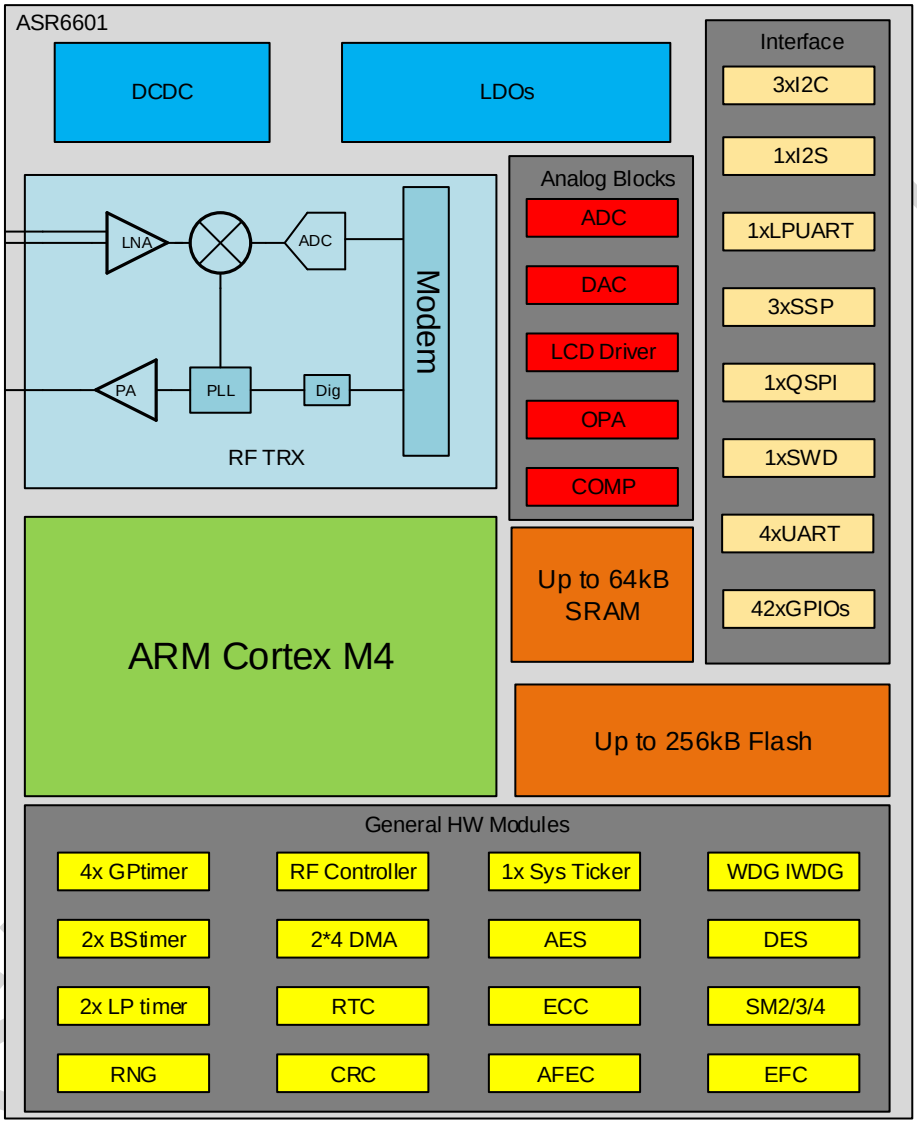

图 3-1 ASR6601 模块框图

3.2 ASR6601 功能点

表 3-1 ASR6601 功能点

模块名称	模块支持的功能点
rcc	时钟复位控制
syscfg	系统功能寄存器
pwr	1. 芯片低功耗模式控制 2. 支持中断信号产生
sec	1. 安全 IP 使能滤波 2. 安全 IP 报警信号滤波 3. 报警信号处理，支持产生状态、中断请求、复位请求
CPU	1. 支持 swd 调试接口 2. 支持 systick 功能 3. 支持 mpu 功能 4. 支持 fpu 功能 5. 支持 37 个 irq，8 个中断优先级
MPU	访问权限控制，包括 cpu、dma、swd 调试接口的 slave 访问操作
EFC	1. 上电芯片模式判断 2. flash info 区数据上电加载 3. flash 基本操作，包括 read、program、page erase、mass erase 4. flash 操作关键时序控制，包括读拍数、program 高压时间、erase 高压时间 5. flash 指令预取功能，1 个深度的预取 buffer 6. flash program 操作支持 single 与连续方式 7. flash info 区 option bytes 操作 8. 支持产生中断信号
I2S	1. Philips I2S 串行协议 2. 支持 Master 与 Slave 模式 3. 1 条接收通道，1 条发送通道，全双工 4. 接收 FIFO 深度为 4 5. 发送 FIFO 深度为 4 6. 接收支持 12，16，20，24，32 位分辨率 7. 发送支持 12，16，20，24，32 位分辨率 8. 支持可编程的 DMA 寄存器 9. 支持可编程的 FIFO Threshold 10. 1 支持中断信号产生
UART	1. 支持 IrDA，支持 3/16 与 low-power（1.41-2.23us）位宽 2. 支持 FIFO 收发，16x8bits 的发送 FIFO，16x10bits 的接收 FIFO 3. 支持 Buffer 收发，1 个深度的发送与接收 Buffer 4. 波特率产生，采用 16 倍过采样，支持 16 位整数分频与 6 位小数分频，最高支持接口时钟频率/16

模块名称	模块支持的功能点
	5. Uart 数据格式配置，包括 1-2 位 Stop，0-1 位 parity（odd，even，固定 0，固定 1，无校验），5-8 位数据 6. 支持 DMA 传输 7. 支持无效 start 检测 8. 支持 breakline 发送与检测 9. 支持硬件流控 CTS 与 RTS 10. 支持中断信号产生
lpuart	1. 支持低功耗唤醒 2. 波特率产生，不支持过采样，支持 4 位小数分频与 12 位整数分频，整数分频最小支持 3 3. 数据格式配置，包括 1-2 位 Stop，0-1 位 parity（奇校验、偶校验、固定 1、固定 0、无校验），5-8 位数据 4. 支持 cts 与 rts 流控 5. 支持 dma 请求 6. 支持中断信号产生
SSP	1. 支持 Master 与 Slave 2. 可编程波特率与预分频，Master 最高支持 1/2 接口时钟频率，Slave 最高支持 1/12 接口时钟频率 3. 支持 8*16 位的接收与发送 FIFO 4. 数据长度可配置，4-16 位 5. 支持 DMA 请求 6. 支持支持 Motorola、Microwire（NS）、TI 格式 7. Motorola 支持 4 种极性相位组合 8. 支持中断信号产生
I2C	1. 支持主模式与从模块 2. 支持多主仲裁 3. 支持 Standard Mode 与 Fast Mode 4. 支持 7 位地址模式 5. 支持 Clock Stretching 6. 支持产生中断信号 7. 支持 DMA 请求
AFEC	1. 模拟 IP 状态寄存器 2. 模拟 IP 控制寄存器 3. 部分寄存器支持安全锁定控制 4. 支持中断信号产生
lorac	1. LORA IP 控制寄存器 2. LORA 状态寄存器 3. LORA IP SPI 接口来源，支持 ssp master 控制与 reg 控制两种方式 4. 支持 DMA 请求与应答 5. 支持中断信号产生

模块名称	模块支持的功能点
RTC	1. 日历计数功能，采用 BCD 格式，支持秒、分、小时、日、月、年、星期 2. 支持 ppm 调整，调整步长 0.5ppm，支持+/-1024 ppm 调整 3. 支持低功耗唤醒 4. tamper/wakeup IO 检测功能，支持有效电平选择，滤波拍数可配置 5. 周期计数功能，32 位计数器 6. 闹钟功能，支持两个闹钟，支持 Mask 选择与日历匹配 7. tamper/wakeup 报警清除 retention sram 功能 8. 部分寄存器支持安全锁定控制 9. 内部信号 IO 输出，包括 alarm0 匹配脉冲，alarm1 匹配脉冲，周期计数配置脉冲，秒信号输出 10. 支持日历计数值读取 11. 支持 sub-second 计数值读取 12. 支持周期计数的计数值读取 13. 支持中断信号产生
iwdg	1. 看门狗计数功能，减法计数，计数时钟支持预分频（4-256 分频） 2. 看门狗异常状态，计数达到 0（喂狗过晚）或喂狗时计数值大于计数窗口值时（喂狗过早）产生 3. 支持产生中断信号 4. 支持喂狗窗口配置 5. 支持计数值读取 6. 支持低功耗唤醒
qspi	1. 仅支持主接口 2. 支持 1 线、2 线、4 线模式 3. 支持 3 种工作模式，包括间接访问、状态查询与 Memory-mapping 4. 支持波特率分频，最高支持接口时钟频率/2 5. 支持产生中断信号
crc	1. 可配置多项式位宽，支持 7、8、16、32 位 2. 支持不同的 hsize 访问，先算低 byte 3. 可编程 crc 初值 4. 支持输入数据 reverse，支持 byte，halfword 与 word 5. 支持输出数据 reverse，支持 word
DMA	1. 支持 1 个主接口 AHB 总线 2. AHB 接口仅支持小端结构 3. 支持中断信号产生 4. 传输模式，支持 M2M，P2M，M2P，P2P 5. 支持软件触发握手信号 6. 支持 4 组硬件握手信号，包括 burst 与 single 请求 7. 支持硬件握手信号来源，每组支持 64 个源头选择 8. 支持 4 个逻辑通道 9. 通道 0 配置如下： (1) 8 bytes 深度 FIFO

模块名称	模块支持的功能点
	(2) 最大 burst 长度为 8 (3) 最大传输长度为 2047 (4) 仅支持 dmac 流控 (5) 源地址数据位宽可配置 (6) 目的地址数据位宽可配置 (7) 地址支持递增，递减，不变 (8) 支持块传输，包括连续地址、自动加载与链表 (9) 支持 scatter 与 gather 10. 通道 1-3 配置如下: (1) 8 bytes 深度 FIFO (2) 最大 burst 长度为 8 (3) 最大传输长度为 2047 (4) 仅支持 dmac 流控 (5) 源地址数据位宽可配置 (6) 目的地址数据位宽可配置 (7) 地址支持递增，递减，不变 (8) 支持块传输，包括连续地址与自动加载，但不支持链表 (9) 不支持 scatter 与 gather
GPIO	1. IO 输出配置，支持 push-pull、open drain、输出高阻 2. IO 输入配置，支持 floating、输入上拉、输入下拉、模拟输入 3. IO 其它配置，上拉配置、下拉配置、驱动能力控制 4. 支持产生中断信号，包括上升沿中断、下降沿中断、双沿中断 5. 支持产生唤醒信号，包括高电平、低电平
SAE	1. 支持 AES128/192/256 2. 支持 DES 与 3DES 3. 支持 SM2、SM3、SM4（ASR6601SE） 4. 支持 RSA1024/2048 5. 支持 ECC224/256/384/512 6. 支持 SHA1，SHA-224，SHA256，SHA384，SHA512 7. 支持随机数发生器
Basic timer	1. 16bits 计数器，加法计数，支持自动加载 2. 16bits 的计数时钟预分频 3. 支持 DMA 请求 4. 支持产生中断信号
GP timer	1. 32bits 计数器，支持自动加载，支持加法、减法、加减计数 2. 16bits 的计数时钟预分频 3. gptimer0 与 gptimer1 支持 4 通道，gptimer2 与 gptimer3 支持 2 通道，每个通道可支持输入捕获、输出比较、PWM 产生、单脉冲输出 4. 支持正交解码 5. 支持产生中断信号 6. 支持 DMA 请求

模块名称	模块支持的功能点
LP timer	1. 支持选择内部时钟与 IO 时钟作为计数时钟 2. 16bits 计数器，加法计数，支持自动加载 3. 支持计数时钟预分频 4. 支持正交解码 5. 支持输入捕获、输出比较、PWM 产生、单脉冲输出 6. 支持产生中断信号 7. 支持 DMA 请求
ADC	1. 采样精度 12 bits 2. 采样速度可配置，最高支持 1 MHz 3. 支持单端与差分采样 4. 数据对齐方式仅支持右对齐 5. 支持 8 个外部通道 6. 支持 7 个内部通道，包括 DAC 输出、内部 Vref、VDD/3（电池电量）、Vts（内部温度传感器）、OPA 输出（3 个） 7. 触发方式，支持软件触发与硬件触发 8. 采样方式，支持采样序列、连续、单次、非连续 9. 支持模拟看门狗功能，共 3 路，可配置通道选择与高低阈值 10. 支持 DMA 请求 11. 支持产生中断信号
DAC	1. 输出精度 10 bits 2. 输出速度可配置，最高支持 1 MHz 3. 数据对齐方式仅支持右对齐 4. 特殊波形输出，支持三角波 5. 触发方式，支持软件触发与硬件触发 6. 支持 DMA 请求 7. 支持产生中断信号
lcdctrl	1. frame 速率分频控制 2. bias 控制，支持 static，1/2，1/3，1/4 3. duty 控制，支持 static（1comx27seg），1/2（2comx26seg），1/3（3comx25seg），1/4（4comx24seg），1/8（8comx20seg） 4. dead frame 控制，支持 0-7 拍的 dead frame，用于调节对比度 5. blink 控制，支持 1，2，3，4，8 或全部 pixel 的闪烁功能，闪烁频率可配置 6. 支持大小电流选择控制，包括状态机动态控制与寄存器静态控制，状态机动态控制时可配置大电流维持拍数 7. 支持中断信号产生

4. 电源供电

4.1 芯片供电管脚

ASR6601 有几个独立的电源供电管脚，通过将这些电源供电管脚分开，可以很好的避免 SoC 上各部分之间的相互影响，特别是 SoC 数字部分工作时对射频收发机性能的干扰。

ASRR6601 的供电接口如图 4-1 所示:

- **VDD_IN**：射频收发机中 PA 的供电电源。
- **VBAT_RF**：射频收发机供电电源。
- **VDCC_RF**：射频收发机中部分模块的供电电源，**须在 PCB 上连接到芯片 VREG 管脚**。
- **VBAT_ESD0**：数字 IO 的供电电源。
- **VBAT_ESD1**：数字 IO 的供电电源。
- **VBAT_ESD2**：数字 IO 的供电电源。
- **VBAT_ESD3**：数字 IO 的供电电源。
- **VBAT_DCC**：模拟电路中 DCDC 的单独供电电源。
- **VBAT_ESD_RTC**：RTC 域 IO 的供电电源。
- **VBAT_RTC**：模拟 RTC 的供电电源。
- **VBAT_ANA**：模拟电路的供电电源。

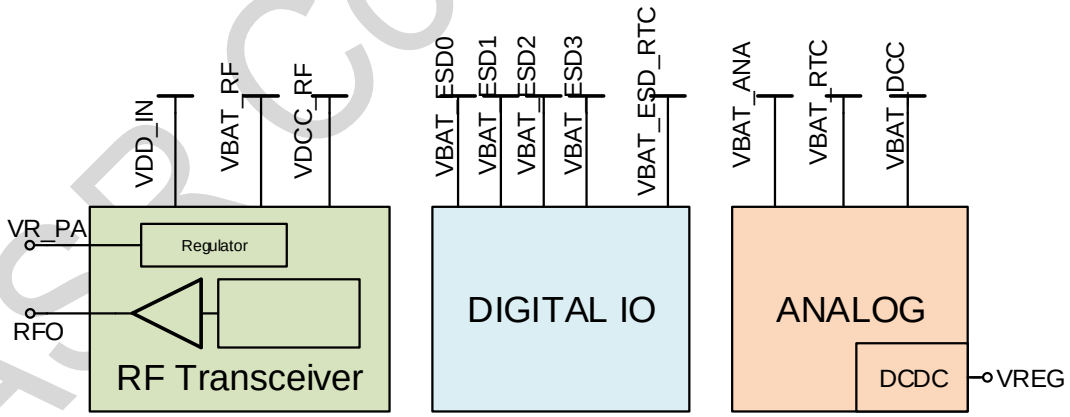

图 4-1 低功耗广域网 SoC 芯片供电接口

4.2 芯片内部电源架构

芯片的内部电源域主要分为 main 域、aon 域及 aonr 域。如图 4-2 所示，这里需要注意的是电源域是按照功能来划分的。

1. **main 域**包含了 SoC 的绝大部分数字逻辑电路，在常用的低功耗模式 Standby 和 Stop3 中，main 域的电源会断开。
2. **aon 域**的名字来源于 always on domain，顾名思义这部分的供电时一直有的，即使在低功耗模式下，这部分的供电也是一直有的。在 aon 域的大部分模块也是可以一直工作的。
3. **aonr 域**主要包含了在 Stop3 模式下需要保持的模块，这部分模块在 Standby 模式下是会断电的。在 aonr 域模块保持当前状态不掉电的情况下，系统可以快速恢复并继续执行。

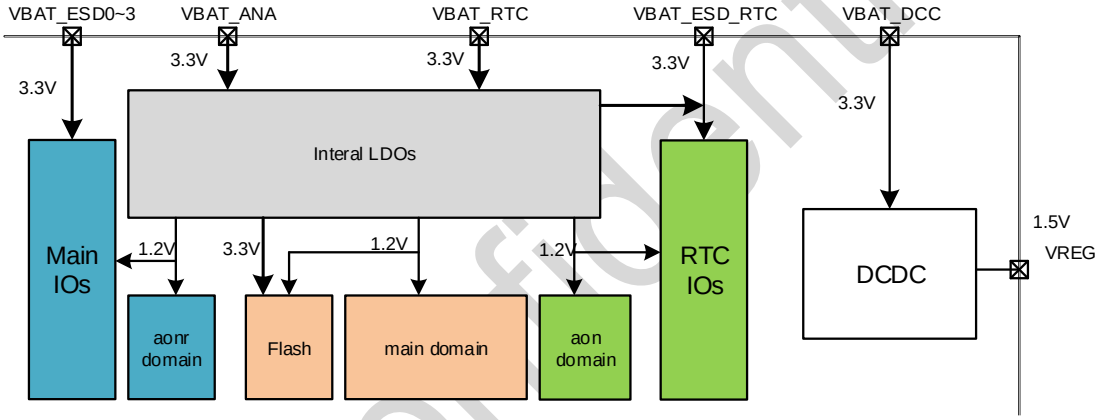

图 4-2 芯片内部的电源架构

5. 权限控制

5.1 简单配置

本节为客户提供常用的简单配置，以满足客户基本的安全需求。

5.1.1 可恢复安全配置

- 使能安全

  配置 OPTION1 的 Flashsecstart 为 0，Flashsecend 为 0x3F，配置整个 FLASHmain 为安全区域。

  该操作配置后，SWD 和非安全区域代码无法读写 FLASHmain 区域，以保护代码安全性。但需注意运行在非安全 SRAM 区域的代码或者非安全 DMA 将无法访问 FLASHmain。

- 去使能安全

  配置 OPTION1 的 Flashsecstart 为 0x3F，Flashsecend 为 0，配置整个 FLASHmain 为非安全区域。

  该操作将擦除整个 FLASHmain，之后可以重新烧录程序。

5.1.2 不可恢复安全配置

配置 Option0 的 debuglevel 为 2。该操作不可逆，需保证代码成熟可靠。

5.2 访问权限控制

基于 debuglevel 规则、boot 启动方式、exeonly 访问规则、wrprotect 访问规则、info 区访问规则、安全区域访问规则，控制来自 cpucode、cpusw、dmac0、dmac1 四个主接口的访问权限。

5.2.1 debuglevel 规则

debuglevel 主要影响 cpucode（bootfromsram 与 bootfrombootloader）、cpusw、dmac0、dmac1 对敏感区域的访问权限。敏感区域包括 flashmain，flashinfo 的 otp 区和 retentionsram。

详细信息请参见*《ASR6601 访问权限控制说明》*。

5.2.2 安全操作与非安全操作

- 安全操作

  由安全区域的代码发起的操作，包括：

  - 配置为安全区域的 dmac0 发起的操作
  - 配置为安全区域的 flashmain 区发起的操作（CPUCode）
  - 配置为安全区域的 systemsram 发起的操作（CPUCode）

- 非安全操作

  由非安全区域的代码发起的操作，包括：

  - 配置为非安全区域的 dmac0 发起的操作
  - dmac1 发起的操作
  - debug 接口发起的操作（CPUSW）
  - Bootloader 发起的操作（CPUCode）
  - 配置为非安全区域的 flashmain 区发起的操作（CPUCode）
  - 配置为非安全区域的 systemsram 发起的操作（CPUCode）

详细信息请参见*《ASR6601 访问权限控制说明》*。

6. 工作模式

ASR6601 低功耗广域网 SoC 主要有 Run、LpRun、Sleep、LpSleep、Stop0、Stop1、Stop2、Stop3、Standby 几种工作模式。每种模式支持的功能，工作的模块和功耗各不相同。终端用户可以根据自己的应用场景选择相应的工作模式。也可以在不同模式之间根据不同的占空比进行切换。每种模式如何进入（触发），哪些模块在工作状态及如何退出（唤醒），会在接下来的小节中作详细介绍。

另外有几点需要注意:

1. 进低功耗模式时标志为 O 的外设（不包含 GPIO）是默认关闭的，在低功耗模式下需要用到的功能需要在进低功耗之前将其打开。
2. 在进入低功耗模式时，为了能达到相应的设计功耗值，需注意以下几点:
   (1) 没有使用的 GPIO 需要配置为 ANALOG 模式（高阻）。
   (2) 如果有使用的 GPIO 是输入模式，需要配置上下拉。
   (3) 若是输出模式连接的外设的上下拉要根据输出的电平进行配置。
3. 进出低功耗模式使用的RCO48M/2，进低功耗模式之前如果使用的是非RCO48M/2时钟，则需要切换为 RCO48M/2，出低功耗模式之后可以再切换为之前使用的时钟。
4. RCO32K/XO32K 及其他低功耗下支持的模拟功能，如果在低功耗状态下需要用到，需要在进低功耗之前打开。
5. 除 RCO48M/RCO32K/XO32K 以外的时钟，及其余模拟功能模块需要在进低功耗之前软件关闭。

表 6-1 不同模块在各工作模式下的工作状态

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

	Run	LpRun	Sleep	LpSleep	Stop0	Stop1	Stop2	Stop3	Standby	Stop0-2 Wakeup	Stop3 Wakeup	Standby Wakeup
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			
lptim	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
RCO4M	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

表格中的注意事项和部分标号的意义：

- **stop0-2**：支持全部 gpio 唤醒，可配置；支持全部 gpio 状态保持。
- **stop3**：支持 56 个 main 域 gpio 唤醒，可配置；支持全部 gpio 状态保持。
- **standby**：支持 8 个 aon 域 gpio 状态保持，main 域 gpio 仅可用于模拟功能（lcd，comp）；不支持 gpio 唤醒。Standby/Stop3 时 lpuart 仅支持 rx。
- **Y**：正常工作。
- **O**：软件可配置。
- **O3**：不支持数据更新，但已有电平仍可输出。
- **Y1**：产生系统复位，间接唤醒系统。
- **Y2**：产生 bor 复位，间接唤醒系统。
- **Y3**：保持进低功耗之前的状态，支持唤醒功能。
- **Y4**: IO MUX Function=1 功能 NA；其余复用功能正常工作
- **NA1**：ret，alg 内容保留，sys 可配置是否内容保留。
- **NA2**：ret 内容保留。
- **NA3**: Analog Output Only

6.1 Run

6.1.1 进入与退出条件

上电或系统复位后的缺省工作模式。

可由 Run 进入 Sleep、LpRun、Stop0、Stop1、Stop2、Stop3、Standby。

可由 Sleep、LpRun、Stop0、Stop1、Stop2、Stop3、Standby 返回 Run。

具体模式切换条件，请见其它功耗模式的切换描述。

6.1.2 唤醒源

N/A

6.2 LpRun

6.2.1 进入与退出条件

可由 Run 进入 LpRun，进入条件如下，软件切换 LDO 的工作状态。在切换 LDO 工作之前，高速时钟要全部关闭，CPU 跑在 32K 的时钟上。

进入 LpRun 配置寄存器切换 LDO 的流程如下:

1：0x05 地址寄存器 [3:3] 配置为全 1，其他位保持不变

2：0x06 地址寄存器 [21:20] 配置为全 1，其他位保持不变

可由 LpRun 返回 Run，退出条件如下，软件切换 LDO 的工作状态。切换完成之后才可打开高速时钟。

退出 LpRun 返回 Run 的流程如下:

1：0x06 地址寄存器 [21:20] 配置为全 0，其他位保持不变

2：0x05 地址寄存器 [3:3] 配置为全 0，其他位保持不变

6.2.2 唤醒源

N/A

6.3 Sleep

6.3.1 进入与退出条件

可由 Run 进入 Sleep，进入条件如下，CPU 执行 wfi/wfe 指令（SLEEPDEEP=0），或者 isr 返回（SLEEPONEXIT=1 并且 SLEEPDEEP=0）。

可由 Sleep 返回 Run，退出条件如下，如果 wfi 进入则支持中断唤醒，如果 wfe 进入则支持事件唤醒。

注意：*因为无专用的事件唤醒信号，所以采用 SVONPEND=1 并且关闭对应 NVIC 的方式实现，此时中断信号用于事件唤醒。*

6.3.2 唤醒源

各个模块的中断信号。

6.4 LpSleep

6.4.1 进入与退出条件

可由 LpRun 进入 LpSleep，进入条件如下，CPU 执行 wfi/wfe 指令（SLEEPDEEP=0），或者 isr 返回（SLEEPONEXIT=1 并且 SLEEPDEEP=0）。

可由 LpSleep 返回 LpRun，退出条件如下，如果 wfi 进入则支持中断唤醒，如果 wfe 进入则支持事件唤醒。

注意：*无专用的事件唤醒信号，采用 SVONPEND=1 并且关闭对应 NVIC 的方式实现，此时中断信号用于事件唤醒。*

6.4.2 唤醒源

各个模块的中断信号。

6.5 Stop0

6.5.1 进入与退出条件

可由 Run 进入 Stop0，进入条件如下，配置 lp_mode 为 2'b00，CPU 执行 wfi/wfe 指令（SLEEPDEEP=1），或者 isr 返回（SLEEPONEXIT=1 同时 SLEEPDEEP=1）。

可由 Stop0 返回 Run，退出条件如下，如果 wfi 进入则支持中断唤醒，如果 wfe 进入则支持事件唤醒。

由 pwr 模块汇总唤醒源状态，并输出 *pwr_wakeup_int* 中断信号与 *pwr_wakeup_event* 事件信号给 CPU 用于唤醒。

6.5.2 唤醒源

- GPIO00-GPIO63 均可用于唤醒，4 个 IO 为一组，一组内每个 IO 均有唤醒使能配置，一组可产生一个唤醒信号，一组内每个 IO 均支持选择高电平唤醒或低电平唤醒。除 GPIO 外的唤醒源列举如下。

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

6.6 Stop1

6.6.1 进入与退出条件

可由 Run 进入 Stop1，进入条件如下，配置 lp_mode 为 2'b01，CPU 执行 wfi/wfe 指令（SLEEPDEEP=1），或者 isr 返回（SLEEPONEXIT=1 同时 SLEEPDEEP=1）；

可由 Stop1 返回 Run，退出条件如下，如果 wfi 进入则支持中断唤醒，如果 wfe 进入则支持事件唤醒；

由 pwr 模块汇总唤醒源状态，并输出 *pwr_wakeup_int* 中断信号与 *pwr_wakeup_event* 事件信号给 CPU 用于唤醒。

6.6.2 唤醒源

- GPIO00-GPIO63 均可用于唤醒，4 个 IO 为一组，一组内每个 IO 均有唤醒使能配置，一组可产生一个唤醒信号，一组内每个 IO 均支持选择高电平唤醒或低电平唤醒。除 GPIO 外的唤醒源列举如下。

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

6.7 Stop2

6.7.1 进入与退出条件

可由 Run 进入 Stop2，进入条件如下，配置 lp_mode 为 2'b10，CPU 执行 wfi/wfe 指令（SLEEPDEEP=1），或者 isr 返回（SLEEPONEXIT=1 同时 SLEEPDEEP=1）；

可由 Stop2 返回 Run，退出条件如下，如果 wfi 进入则支持中断唤醒，如果 wfe 进入则支持事件唤醒；

由 pwr 模块汇总唤醒源状态，并输出 *pwr_wakeup_int* 中断信号与 *pwr_wakeup_event* 事件信号给 CPU 用于唤醒。

6.7.2 唤醒源

- GPIO00-GPIO63 均可用于唤醒，4 个 IO 为一组，一组内每个 IO 均有唤醒使能配置，一组可产生一个唤醒信号，一组内每个 IO 均支持选择高电平唤醒或低电平唤醒。除 GPIO 外的唤醒源列举如下。

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

6.8 Stop3

6.8.1 进入与退出条件

可由 Run 进入 Stop3，进入条件如下，配置 lp_mode 为 2'b11，lp_mode_ext 为 1'b1，CPU 执行 wfi/wfe 指令（SLEEPDEEP=1），或者 isr 返回（SLEEPONEXIT=1 同时 SLEEPDEEP=1）；

可由 Stop3 返回 Run，退出条件如下，有 Stop3 唤醒事件发生。

6.8.2 唤醒源

- GPIO00-GPIO55 均可用于唤醒，4 个 IO 为一组，每组可选择一个 IO 用于唤醒，每组产生一个唤醒信号，每组支持选择高电平唤醒或低电平唤醒。除 GPIO 外的唤醒源列举如下。

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

6.9 Standby

6.9.1 进入与退出条件

可由 Run 进入 Standby，进入条件如下，配置 lp_mode 为 2'b11，lp_mode_ext 为 1'b0，CPU 执行 wfi/wfe 指令（SLEEPDEEP=1），或者 isr 返回（SLEEPONEXIT=1 同时 SLEEPDEEP=1）；

可由 Standby 返回 Run，退出条件如下，有 Standby 唤醒事件发生。

注意：

1. *DCDC 与 VBAT 供电切换时，Standby 进入后会立即退出，不需要唤醒事件。*
2. *dbg_standby=1 时，不能进行 DCDC 与 VBAT 电源切换，如果两者均有效，以 dbg_standby 优先。*

6.9.2 唤醒源

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

7. 系统配置

7.1 系统架构

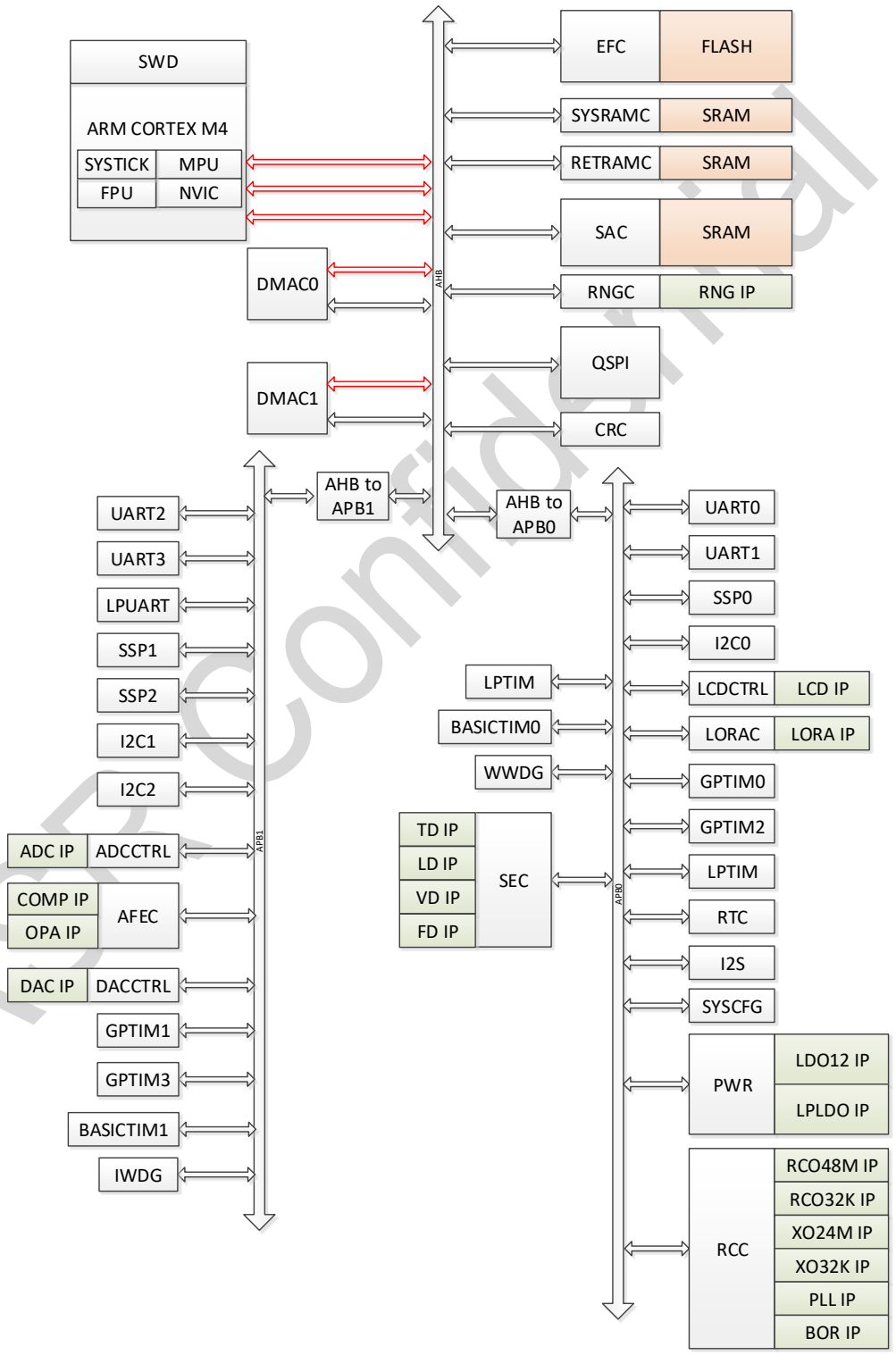

图 7-1 系统架构图

7.1.1 STAR CPU

STAR CPU 共包括三条 master 总线，包括 icode AHB bus，dcode AHB bus 以及 system AHB bus，用于执行程序访问、数据访问以及寄存器访问。

7.1.2 DMAC0

DMAC0 共包括一条 master 总线，可用于协助 CPU 完成数据搬移操作。

7.1.3 DMAC1

DMAC1 共包括一条 master 总线，可用于协助 CPU 完成数据搬移操作。

7.1.4 Master

各个 master 总线可访问的地址范围请见下表。*(1) 表示仅在 boot from bootloader 时可访问。*

表 7-1 master 总线访问范围

起始地址	结束地址	功能描述	可执行	STAR icode 访问	STAR dcode 访问	STAR system 访问	dmac0 访问	dmac1 访问
0xE0100000	0xFFFFFFFF	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

Memory Mapping 表如下所示，Bytes 数据使用小端格式存储，即低地址存储低位 Byte 数据。

表 7-2 Memory Mapping

区域	起始地址	结束地址	功能描述	地址范围
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) 由启动方式决定 0x00000000 地址对应的存储器。

7.2.1 AHB0 SFR

AHB0 SFR 内部的地址 mapping 请见下表。

表 7-3 AHB0 SFR 内部的地址 mapping

起始地址	结束地址	功能描述	地址范围
0x40025000	0x40030000	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

AHB0 SFR 内部的地址 mapping 请见下表。

表 7-4 AHB0 SFR 内部的地址 mapping

起始地址	结束地址	功能描述	地址范围
0x40034000	0x40040000	Reserved	
0x40033000	0x40033FFF	rngc	4KB
0x40030000	0x40032FFF	sac	12KB[(1)(2)]

(1) 低 8KB 为 ARAM 空间，高 4K 为寄存器空间。

(2) ARAM 空间仅支持 word 访问。

7.2.3 APB0 SFR

APB0 SFR 内部的地址 mapping 请见下表。

表 7-5 APB0 SFR 内部的地址 mapping

起始地址	结束地址	功能描述	地址范围
0x4000f000	0x4000ffff	sec	4KB
0x4000e000	0x4000efff	rtc	4KB
0x4000d800	0x4000dfff	lptim1	2KB
0x4000d000	0x4000d7ff	lptim	2KB
0x4000c000	0x4000cfff	basictim0	4KB
0x4000b000	0x4000bfff	gptim2	4KB
0x4000a000	0x4000afff	gptim0	4KB
0x40009000	0x40009fff	lorac	4KB
0x40008000	0x40008fff	afec	4KB

起始地址	结束地址	功能描述	地址范围
0x40007000	0x40007fff	i2c0	4KB
0x40006000	0x40006fff	ssp0	4KB
0x40005000	0x40005fff	lpuart	4KB
0x40004000	0x40004fff	uart1	4KB
0x40003000	0x40003fff	uart0	4KB
0x40002000	0x40002fff	i2s	4KB
0x40001800	0x40001fff	pwr	2KB
0x40001000	0x400017ff	syscfg	2KB
0x40000000	0x40000fff	rcc	4KB

7.2.4 APB0 SFR

APB1 SFR 内部的地址 mapping 请见下表。

表 7-6 APB1 SFR 内部的地址 mapping

起始地址	结束地址	功能描述	地址范围
0x4001fc00	0x4001ffff	portd	1KB
0x4001f800	0x4001fbff	portc	1KB
0x4001f400	0x4001f7ff	portb	1KB
0x4001f000	0x4001f3ff	porta	1KB
0x4001e000	0x4001efff	wwdg	4KB
0x4001d000	0x4001dfff	iwdg	4KB
0x4001c000	0x4001cfff	basictim1	4KB
0x4001b000	0x4001bfff	gptim3	4KB
0x4001a000	0x4001afff	gptim1	4KB
0x40019000	0x40019fff	dacctrl	4KB
0x40018000	0x40018fff	lcdctrl	4KB
0x40017000	0x40017fff	adcctrl	4KB
0x40016000	0x40016fff	Reserved	4KB
0x40015000	0x40015fff	i2c2	4KB
0x40014000	0x40014fff	i2c1	4KB
0x40013000	0x40013fff	ssp2	4KB
0x40012000	0x40012fff	ssp1	4KB
0x40011000	0x40011fff	uart3	4KB
0x40010000	0x40010fff	uart2	4KB

7.3 SRAM

系统中 SRAM 包括 system sram，retention sram 以及 sac sram。其中 sac sram 仅支持 word 访问，system sram 与 retention sram 支持 word、halfword、byte 访问。

7.4 启动方式

启动方式由 IO 电平以及 Flash 存储数据共同决定，具体见下表。

表 7-7 SoC 启动方式配置表

Debug_Level	Use_Flash_BOOT0	FLASH_Boot0	BOOT0 pin	FLASH_Boot1	Main_Flash_Empty	Boot Config
2	X	X	X	X	X	Boot from Flash Main
<2	0	X	0	X	0	Boot from Flash Main
<2	0	X	0	X	1	Boot from Flash Bootloader
<2	0	X	1	1	X	Boot from Flash Bootloader
<2	0	X	1	0	X	Boot from System SRAM
<2	1	1	X	X	0	Boot from Flash Main
<2	1	1	X	X	1	Boot from Flash Bootloader
<2	1	0	X	1	X	Boot from Flash Bootloader
<2	1	0	X	0	X	Boot from System SRAM

其中，DebugLevel、UseFlashBoot0、FlashBoot0、FlashBoot1 为 Flash info 区数据，可由应用根据需要修改。MainFlashEmpty 由 Flash Main 区的 0 地址数据决定，如果为 0xFFFFFFFF，则 MainFlashEmpty 为 1，否则为 0。BOOT0 pin 由 gpio2 电平决定。

启动方式的判断在芯片上电、退出 Standby、系统复位时发生，由硬件自动完成。

7.5 系统配置相关寄存器描述

寄存器基地址：0x40001000

表 7-8 SYSCFG 寄存器列表

寄存器	偏移量	描述
SYSCFG_CR0	0x000	控制寄存器 0，DMA 硬件握手控制
SYSCFG_CR1	0x004	控制寄存器 1，DMA 硬件握手控制
SYSCFG_CR2	0x008	控制寄存器 2
SYSCFG_CR3	0x00C	控制寄存器 3，低功耗 Debug 连接控制
SYSCFG_CR4	0x010	控制寄存器 4
SYSCFG_CR5	0x014	控制寄存器 5
SYSCFG_CR6	0x018	控制寄存器 6，安全锁定控制
SYSCFG_CR7	0x01C	控制寄存器 7，安全锁定控制
SYSCFG_CR8	0x020	控制寄存器 8，QSPI 存储密钥
SYSCFG_CR9	0x024	控制寄存器 9，QSPI REMAP 控制
SYSCFG_CR10	0x028	控制寄存器 10

7.5.1 SYSCFG_CR0

偏移量：0x000

复位值：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

位 31-30 RESERVED：保留，不能修改。

位 29-24 DMAC0_HANDSHAKE0_SEL：DMAC0 的 HANDSHAKE0 来源选择，请查询 DMA 请求 MUX 表格。

位 23-22 RESERVED：保留，不能修改。

位 21-16 DMAC0_HANDSHAKE1_SEL：DMAC0 的 HANDSHAKE1 来源选择。请查询 DMA 请求 MUX 表格。

位 15-14 RESERVED：保留，不能修改。

位 13-8 DMAC0_HANDSHAKE2_SEL：DMAC0 的 HANDSHAKE2 来源选择，请查询 DMA 请求 MUX 表格。

位 7-6 RESERVED：保留，不能修改。

位 5-0 DMAC0_HANDSHAKE3_SEL：DMAC0 的 HANDSHAKE3 来源选择，请查询 DMA 请求 MUX 表格。

7.5.2 SYSCFG_CR1

偏移量：0x004

复位值：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

位 31-30 RESERVED：保留，不能修改。

位 29-24 DMAC1_HANDSHAKE0_SEL：DMAC1 的 HANDSHAKE0 来源选择，请查询 DMA 请求 MUX 表格。

位 23-22 RESERVED：保留，不能修改。

位 21-16 DMAC1_HANDSHAKE1_SEL：DMAC1 的 HANDSHAKE1 来源选择。请查询 DMA 请求

MUX 表格。

位 15-14 RESERVED：保留，不能修改。

位 13-8 DMAC1_HANDSHAKE2_SEL：DMAC1 的 HANDSHAKE2 来源选择，请查询 DMA 请求 MUX 表格。

位 7-6 RESERVED：保留，不能修改。

位 5-0 DMAC1_HANDSHAKE3_SEL：DMAC1 的 HANDSHAKE3 来源选择，请查询 DMA 请求 MUX 表格。

7.5.3 SYSCFG_CR2

偏移量：0x008

复位值：0x00000000

31	30	29	28
RESERVED	SYSCFG_HALTED_IPTIM1_EN	SYSCFG_IPT_MODE_LOCK	SYSCFG_IPT_MODE
r	r/w	r/w	r/w
27	**26**	**25**	**24**
SYSCFG_HALTED_LPTIM_EN	SYSCFG_HALTED_IWDG_EN	SYSCFG_HALTED_WWDG_EN	SYSCFG_HALTED_GPTIM0_EN
r/w	r/w	r/w	r/w
23	**22**	**21**	**20**
SYSCFG_HALTED_GPTIM1_EN	SYSCFG_HALTED_GPTIM2_EN	SYSCFG_HALTED_GPTIM3_EN	SYSCFG_HALTED_BASICTIM0_EN
r/w	r/w	r/w	r/w
19	**18**	**17**	**16**
SYSCFG_HALTED_BASICTIM1_EN	QSPI_MEM_ENCRYPT_EN	QSPI_REMAP_ENABLE	RESERVED
r/w	r/w	r/w	r
15	**14**	**13**	**12**
RESERVED	RESERVED	RESERVED	BOUNDARY_SCAN_MODE
r	r	r	r/w
11	**10**	**9**	**8**
CPU_STCALIB_SKEW	SYSCFG_DBG_SLEEP	RESERVED	RESERVED
r/w	r/w	r	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

位 31 RESERVED：保留，不能修改。

位 30 SYSCFG_HALTED_LPTIM1_EN：CPU halted 时，IPTIM1 停止计数的使能控制。

- 0：不使能
- 1：使能

位 29 SYSCFG_IPT_MODE_LOCK：IPT mode 使能锁定。

- 0：不锁定
- 1：锁定

***注意：**此寄存器仅支持配置为 1，一旦配置为 1 后，无法配置为 0，除非发生系统复位。*

位 28 SYSCFG_IPT_MODE：IPT mode 使能。

- 0：不使能
- 1：使能

注意：

1. *此寄存器与上电判断出的 mode 信号共同决定是否进入 IPT mode。此寄存器用于 FT 时，通过 CPU 控制进入 IPT mode。*
2. *SYSCFG_IPT_MODE_LOCK 为 1 时，此寄存器始终为 0。*

位 27 SYSCFG_HALTED_LPTIM_EN：CPU halted 时，LPTIM 停止计数的使能控制。

- 0：不使能
- 1：使能

位 26 SYSCFG_HALTED_IWDG_EN：CPU halted 时，IWDG 停止计数的使能控制。

- 0：不使能
- 1：使能

位 25 SYSCFG_HALTED_WWDG_EN：CPU halted 时，WWDG 停止计数的使能控制。

- 0：不使能
- 1：使能

位 24 SYSCFG_HALTED_GPTIM0_EN：CPU halted 时，GPTIM0 停止计数的使能控制。

- 0：不使能
- 1：使能

位 23 SYSCFG_HALTED_GPTIM1_EN：CPU halted 时，GPTIM1 停止计数的使能控制。

- 0：不使能
- 1：使能

位 22 SYSCFG_HALTED_GPTIM2_EN：CPU halted 时，GPTIM2 停止计数的使能控制。

- 0：不使能
- 1：使能

位 21 SYSCFG_HALTED_GPTIM3_EN：CPU halted 时，GPTIM3 停止计数的使能控制。

- 0：不使能
- 1：使能

位 20 SYSCFG_HALTED_BASICTIM0_EN：CPU halted 时，BASICTIM0 停止计数的使能控制。

- 0：不使能
- 1：使能

位 19 SYSCFG_HALTED_BASICTIM1_EN：CPU halted 时，BASICTIM1 停止计数的使能控制。

- 0：不使能
- 1：使能

位 18 QSPI_MEM_ENCRYPT_EN：QSPI 的存储加密功能使能控制。

- 0：不使能
- 1：使能

位 17 QSPI_REMAP_ENABLE：QSPI 的 remap 功能使能控制。

- 0：不使能
- 1：使能

位 16 RESERVED：保留，不能修改。

位 15 RESERVED：保留，不能修改。

位 14 RESERVED：保留，不能修改。

位 13 RESERVED：保留，不能修改。

位 12 BOUNDARY_SCAN_MODE：BOUNDARY_SCAN_MODE 使能。

- 0：不使能
- 1：使能

注意：*配置为 1 后，所有 GPIO 均切换到 IO 测试模式，仅用于 IO 测试。*

位 11 CPU_STCALIB_SKEW：CPU SysTick 的 skew 配置。影响 CPU 的 STCALIB[24]位。

- 0：无 skew
- 1：有 skew

位 10 SYSCFG_DBG_SLEEP：Sleep 低功耗模式是否允许 debug 连接。

仅在 debug 时使用，会影响 sleep 低功耗模式的实现方式。

- 0：不允许
- 1：允许

位 9 RESERVED：保留，不能修改。

位 8 RESERVED：保留，不能修改。

位 7 UART0_DMA_CLR_SEL：UART0 的 DMA_CLR 信号选择。

建议配置为 1，提高 DMAC 搬数效率，UART IP 默认使用同步后的 DMA_CLR 信号。

- 0：使用 2 拍同步后的 DMA_CLR 信号
- 1：直接使用 dmac 输出的 DMA_CLR 信号

位 6 UART1_DMA_CLR_SEL：UART1 的 DMA_CLR 信号选择。

建议配置为 1，提高 DMAC 搬数效率，UART IP 默认使用同步后的 DMA_CLR 信号。

- 0：使用 2 拍同步后的 DMA_CLR 信号
- 1：直接使用 DMAC 输出的 DMA_CLR 信号

位 5 UART2_DMA_CLR_SEL：UART2 的 DMA_CLR 信号选择。

建议配置为 1，提高 DMAC 搬数效率，UART IP 默认使用同步后的 DMA_CLR 信号。

- 0：使用 2 拍同步后的 DMA_CLR 信号
- 1：直接使用 DMAC 输出的 DMA_CLR 信号

位 4 UART3_DMA_CLR_SEL：UART3 的 DMA_CLR 信号选择。

建议配置为 1，提高 DMAC 搬数效率，UART IP 默认使用同步后的 DMA_CLR 信号。

- 0：使用 2 拍同步后的 DMA_CLR 信号
- 1：直接使用 DMAC 输出的 DMA_CLR 信号

位 3 SSP0_DMA_CLR_SEL：SSP0 的 DMA_CLR 信号选择。

建议配置为 1，提高 DMAC 搬数效率，SSP IP 默认使用同步后的 DMA_CLR 信号。

- 0：使用 2 拍同步后的 DMA_CLR 信号
- 1：直接使用 DMAC 输出的 DMA_CLR 信号

位 2 SSP1_DMA_CLR_SEL：SSP1 的 DMA_CLR 信号选择。

建议配置为 1，提高 DMAC 搬数效率，SSP IP 默认使用同步后的 DMA_CLR 信号。

- 0：使用 2 拍同步后的 DMA_CLR 信号
- 1：直接使用 DMAC 输出的 DMA_CLR 信号

位 1 SSP2_DMA_CLR_SEL：SSP2 的 DMA_CLR 信号选择。

建议配置为 1，提高 DMAC 搬数效率，SSP IP 默认使用同步后的 DMA_CLR 信号。

- 0：使用 2 拍同步后的 DMA_CLR 信号
- 1：直接使用 DMAC 输出的 DMA_CLR 信号

位 0 SSP_AFEC_DMA_CLR_SEL：AFEC 中 SSP 的 DMA_CLR 信号选择。

建议配置为 1，提高 DMAC 搬数效率，SSP IP 默认使用同步后的 DMA_CLR 信号。

- 0：使用 2 拍同步后的 DMA_CLR 信号
- 1：直接使用 DMAC 输出的 DMA_CLR 信号

7.5.4 SYSCFG-CR3

偏移量：0x00C

复位值：0x00000000

对应寄存器在 AON 电源域。

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

位 31-2 RESERVED：保留，不能修改。

位 1 SYSCFG_DBG_STOP：Stop 低功耗模式是否允许 debug 连接。

仅在 debug 时使用，会影响 stop 低功耗模式的实现方式。

- 0：不允许
- 1：允许

位 0 SYSCFG_DBG_STANDBY：Standby 低功耗模式是否允许 debug 连接。

仅在 debug 时使用，会影响 standby 低功耗模式的实现方式。

- 0：不允许
- 1：允许

7.5.5 SYSCFG_CR4

偏移量：0x010

复位值：0x00000000

对应寄存器在 AON 电源域。

31-0
SYSCFG_CR4_REG
r/w

位 31：LPTIM1 的 IN2 重映射使能。

- 0：不使能，LPTIM1 的 IN2 由 GPIO 的 AFR 决定
- 1：使能，LPTIM1 的 IN2 来自 LPTIM 的 IN1

位 30-0：无特殊功能，可用于软件存储少量数据。

7.5.6 SYSCFG_CR5

偏移量：0x014

复位值：0x00000000

对应寄存器在 AON 电源域。

31-0
SYSCFG_CR5_REG
r/w

位 31-0 SYSCFG_CR5_REG：无特殊功能，可用于软件存储少量数据。

7.5.7 SYSCFG_CR6

偏移量：0x018

复位值：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

位 31-16 RESERVED：保留，不能修改。

位 15 RNGC_SECURE_LOCK：RNGC 安全锁定位。

- 0：非安全锁定
- 1：安全锁定

位 14-5 ANALOG_MAIN_SECURE_LOCK：AFEC 中 main 域配置的安全锁定位。

[0] 对应 VD

[1] 对应 TD

[2] 对应 LD

[3] 对应 FD24M

[4] 对应 FD32M

[5] 对应 RNG

[6] 对应 TEST

[9:7] 未使用

- 0：非安全锁定
- 1：安全锁定

位 4 RESERVED：保留，不能修改。

位 3 SEC_SECURE_LOCK：SEC 安全锁定位。

- 0：非安全锁定
- 1：安全锁定

位 2 SAC_SECURE_LOCK：SAC 安全锁定位。

- 0：非安全锁定
- 1：安全锁定

位 1 DMAC0_SLAVE_SECURE_LOCK：DMAC0 的从接口安全锁定位。

- 0：非安全锁定
- 1：安全锁定

位 0 DMAC0_MASTER_SECURE_LOCK：DMAC0 的主接口安全锁定位。

- 0：非安全锁定
- 1：安全锁定

7.5.8 SYSCFG_CR7

偏移量：0x01C

复位值：0x00000000

对应寄存器在 AON 电源域。

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

3	2	1	0
RTC_WAKEUP2_SEC	RTC_WAKEUP1_SECU	RTC_WAKEUP0_SECU	RTC_TAMPER_SECUR

URE_LOCK	RE_LOCK	RE_LOCK	E_LOCK
r/w	r/w	r/w	r/w

位 31-16 RESERVED：保留，不能修改。

位 14-5 ANALOG_AON_SECURE_LOCK：AFEC 中 aon 域配置的安全锁定位。

[0] 对应 LPLDO

[1] 对应 RCO4M

[2] 对应 PWRSW

[3] 对应 RCO32K

[4] 对应 XO32K

[5] 对应 LDO12

[6] 对应 FD32K

[9:7] 未使用

- 0：非安全锁定
- 1：安全锁定

位 4 RTC_CALENDAR_SECURE_LOCK：RTC 中 Calendar 配置的安全锁定位。

- 0：非安全锁定
- 1：安全锁定

位 3 RTC_WAKEUP2_SECURE_LOCK：RTC 中 Wakeup2 配置的安全锁定位。

- 0：非安全锁定
- 1：安全锁定

位 2 RTC_WAKEUP1_SECURE_LOCK：RTC 中 Wakeup1 配置的安全锁定位。

- 0：非安全锁定
- 1：安全锁定

位 1 RTC_WAKEUP0_SECURE_LOCK：RTC 中 Wakeup0 配置的安全锁定位。

- 0：非安全锁定
- 1：安全锁定

位 0 RTC_TAMPER_SECURE_LOCK：RTC 中 Tamper 配置的安全锁定位。

- 0：非安全锁定
- 1：安全锁定

7.5.9 SYSCFG_CR8

偏移量：0x020

复位值：0x00000000

31-0
QSPI_MEM_ENCRYPT_KEY
r/w

位 31-0 QSPI_MEM_ENCRYPT_KEY：QSPI 的存储加密密钥。

7.5.10 SYSCFG_CR9

偏移量：0x024

复位值：0x00000000

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

位 31-28 RESERVED：保留，不能修改。

位 27-14 QSPI_REMAP_SRC_ADDR：QSPI 的 remap 的原地址，单位为 1KB。

位 13-0 QSPI_REMAP_DST_ADDR：QSPI 的 remap 的目的地址，单位为 1KB。

7.5.11 SYSCFG_CR10

偏移量：0x028

复位值：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

位 31-24 RESERVED：保留，不能修改。

位 23 I2S_WS_SEL：I2S WS 输出延时使能。

- 0：不使能
- 1：使能

注意：*此寄存器仅在 I2S 主接口时有效，使能后，输出的 WS 信号比数据会延后一拍。*

位 22 I2S_WS_EN：I2S WS 使能。

- 0：不使能
- 1：使能

注意：*此寄存器仅在 I2S 主接口时有效，使能后会基于 I2S_WS_LEN 配置产生 WS 信号。*

位 21-15 I2S_WS_LEN：I2S 主接口的分辨率配置。

N：WS 频率=I2S 接口时钟频率/[(N+1)*2]

I2S 接口时钟频率由 RCC 的 I2S_CLK_DIV 与 I2S_CLK_SEL 共同决定。

位 14 I2S_MODE_SEL：I2S 主从选择。

- 0：从接口
- 1：主接口

注意：*除此寄存器外，还需要配置 RCC 的 I2S_CLK_DIV 与 I2S_CLK_SEL，以及 GPIO 的复用配置。*

位 13-0 QSPI_REMAP_SIZE：QSPI 的 REMAP 的空间大小，单位为 1KB。

7.6 DMA 请求 MUX

表 7-9 DMA 请求 MUX

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

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

8. RCC

8.1 复位

复位主要包括四种类型，外部复位、上电复位、系统复位、低功耗复位。

8.1.1 外部复位

外部复位由 RSTN IO 输入，低电平有效。

外部复位用于复位所有数字逻辑。

8.1.2 上电复位

上电复位由 BOR 电路产生。BOR 电路监控 VBAT 电压，保证当电压大于 1.8V 时内部复位释放。

上电复位用于复位所有数字逻辑。

8.1.3 系统复位

系统复位源头包括 iwdg 复位、wwdg 复位、option byte load 复位、software 复位、sec 复位、上电复位、外部复位。

- iwdg 复位：由 iwdg 模块产生，用于异常恢复。
- wwdg 复位：由 wwdg 模块产生，用于异常恢复。
- option byte load 复位：由 efc 模块产生，用于启动 option byte 重新加载。
- software 复位：由 cpu 模块产生。
- sec 复位：由 sec 模块产生，用于安全报警复位。

系统复位用于复位 main 域的大部分数据逻辑，但不包括复位源状态寄存器，该寄存器用于记录哪个系统复位源产生此次复位。

8.1.4 低功耗复位

低功耗复位由低功耗状态机产生，用于 standby 与 stop3 时复位 main 域的逻辑。

低功耗复位用于复位 main 域的所有数字逻辑。

8.2 时钟

时钟网络图如下所示:

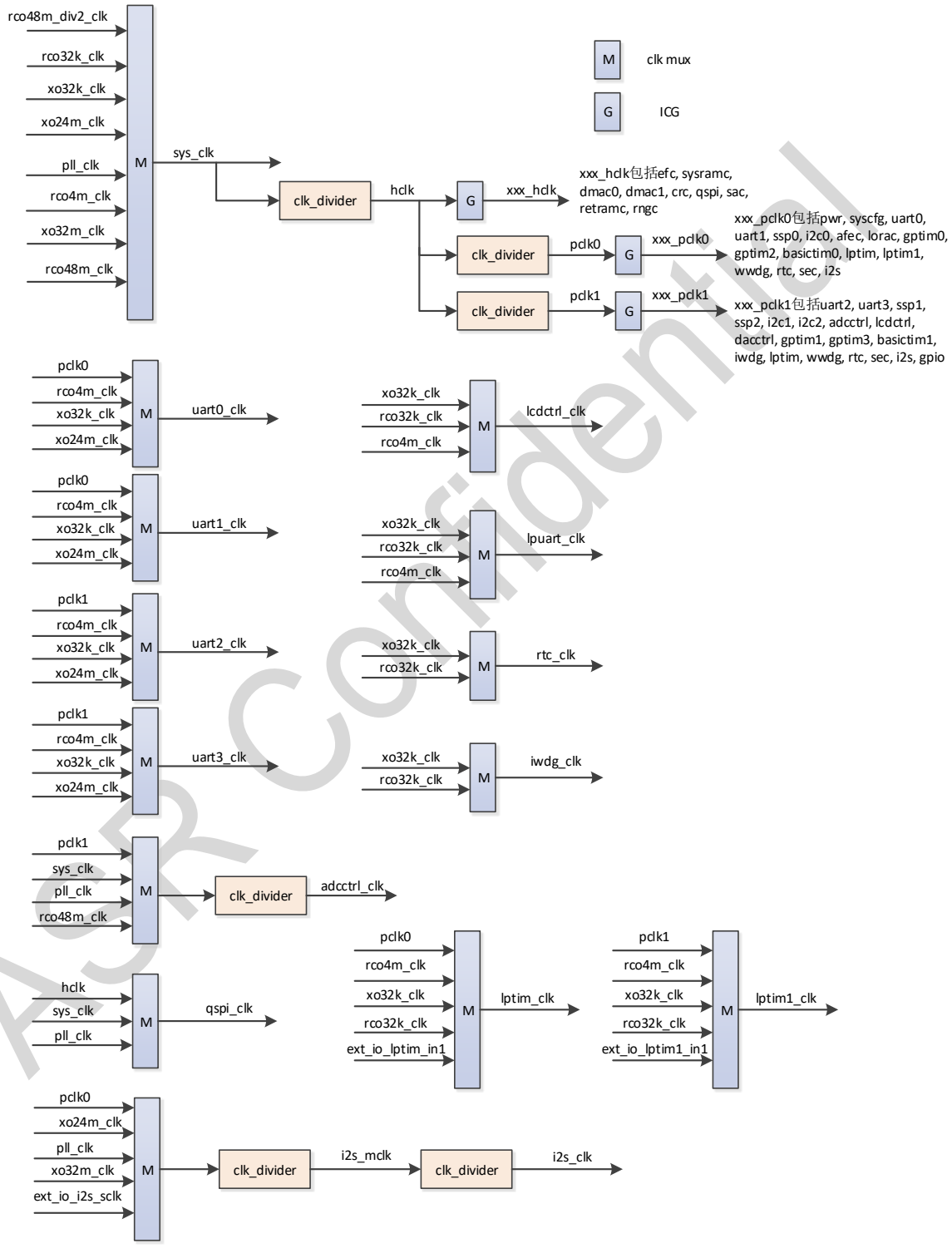

图 8-1 时钟网络图

8.2.1 系统时钟 sys_clk

系统时钟sys_clk来源包括rco48m的2分频，rco32k，xo32k，pll，xo24m，xo32m，rco4m，rco48m，缺省为 rco48m 的 2 分频。

- rco48m 为内部时钟电路产生，频率为 48MHz。
- rco32k 为内部时钟电路产生，频率为 32KHz。
- rco4m 为内部时钟电路产生，频率为 4MHz。
- xo32k 为外部晶振产生，频率为 32.768KHz。
- xo32m 为外部晶振产生，频率为 32MHz。
- xo24m 为外部晶振产生，频率为 24MHz。
- pll 为内部时钟电路，pll 的时钟源支持选择 rco48m、xo32m、xo24m、rco4m，pll 时钟输出最大支持 48MHz。

ahb 总线时钟 hclk 由 sys_clk 分频产生，分频系数支持 2^N，N 支持 0-9。

系统中包括两条apb总线，apb 总线时钟pclk1与pclk2由hclk分频产生，分频系数支持2^M，M 支持 0-4，两个 apb 总线分频系数可以独立配置。

8.2.2 模块时钟

模块时钟包括总线时钟与接口时钟。

总线时钟由 hclk 或 pclk 门控产生，用于模块的总线访问。

部分模块除总线时钟外，还有独立的接口时钟，与总线时钟异位，用于模块的功能实现。

模块的接口时钟来源如下:

- lptim 支持 pclk0/pclk1，rco4m，xo32k，rco32k，IO 输入时钟;
- lcdctrl 支持 xo32k，rco32k，rco4m;
- lpuart 支持 xo32k，rco32k，rco4m;
- rtc 支持 xo32k，rco32k;
- iwdg 支持 xo32k，rco32k;
- uart 支持 pclk0/pclk1，rco4m，xo32k，xo24m;
- adcctrl 支持 pclk1，sys_clk，pll，rco48m;
- i2s 支持 pclk0，xo24m，pll，xo32m，IO 输入时钟;
- qspi 支持 hclk，sys_clk，pll;
- adcctrl 与 i2s 还支持接口时钟分频功能，用于实现低频接口时钟产生。
- lptim，lcdctrl，lpuart，rtc，iwdg 支持 aon 域与 main 域时钟单独门控。

8.2.3 时钟输出 mco

支持通过 mco 信号将系统内部时钟通过 io 输出。

mco 时钟来源支持 rco32k，xo32k，rco4m，xo24m，xo32m，rco48m，pll，sys_clk。

时钟支持分频输出，分频系统可由软件配置。

8.3 RCC 相关寄存器描述

寄存器基地址：40000000

表 8-1 RCC 寄存器列表

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

8.3.1 RCC_CR0

偏移量：0x000

复位值：0x00000000

31	30-27	26	25	24-22	21-19	
RESERVED	RESERVED	RESERVED	STCLKEN_SEL	MCO_CLK_DIV_NUM	MCO_CLK_SEL	
r	r	r	r/w	r/w	r/w	
18	**17-15**	**14-12**	**11-8**	**7-5**	**4-1**	**0**
MCO_CLK_OUT_EN	PCLK1_DIV	SYS_CLK_SEL	HCLK_DIV	PCLK0_DIV	RESERVED	RESERVED
r/w	r/w	r/w	r/w	r/w	r	r

位 31 RESERVED：保留，不能修改。

位 30-27 RESERVED：保留，不能修改。

位 26 RESERVED：保留，不能修改。

位 25 STCLKEN_SEL：CPU SYSTICK 时钟来源选择。

- 0：XO32K
- 1：RCO32K

位 24-22 MCO_CLK_DIV_NUM：MCO 分频系数。

- <4：分频系数为 1
- 4：分频系数为 2
- 5：分频系数为 4
- 6：分频系数为 8
- 7：分频系数为 16

注意：*需要保证在 MCO_CLK_OUT_EN=0 时修改 mco 分频系数。如果 MCO_CLK_OUT_EN 已使能，则需要软件先关闭 MCO_CLK_OUT_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置 mco 分频系数。*

位 21-19 MCO_CLK_SEL：MCO 来源选择。

- 0：RCO32K
- 1：XO32K
- 2：RCO4M
- 3：XO24M
- 4：XO32M
- 5：RCO48M
- 6：PLL
- 7：SYS_CLK

注意：*需要保证在 MCO_CLK_OUT_EN=0 时修改 mco 来源选择。如果 MCO_CLK_OUT_EN 已使能，则需要软件先关闭 MCO_CLK_OUT_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置 mco 来源选择。*

位 18 MCO_CLK_OUT_EN：MCO 输出使能。

- 0：不使能
- 1：使能

位 17-15 PCLK1_DIV：PCLK1 分频系数。

- 0：pclk1 时钟频率等于 hclk 时钟频率
- 1：pclk1 时钟频率等于 hclk 时钟频率的 1/2
- 2：pclk1 时钟频率等于 hclk 时钟频率的 1/4
- 3：pclk1 时钟频率等于 hclk 时钟频率的 1/8
- >3：pclk1 时钟频率等于 hclk 时钟频率的 1/16

位 14-12 SYS_CLK_SEL：SYS_CLK 时钟源选择。

- 0：RCO48M 的 2 分频
- 1：RCO32K
- 2：XO32K
- 3：PLL
- 4：XO24M
- 5：XO32M
- 6：RCO4M
- 7：RCO48M

位 11-8 HCLK_DIV：HCLK 分频系数。

- 0：hclk 时钟频率等于 sys_clk 时钟频率
- 1：hclk 时钟频率等于 sys_clk 时钟频率的 1/2
- 2：hclk 时钟频率等于 sys_clk 时钟频率的 1/4
- 3：hclk 时钟频率等于 sys_clk 时钟频率的 1/8
- 4：hclk 时钟频率等于 sys_clk 时钟频率的 1/16
- 5：hclk 时钟频率等于 sys_clk 时钟频率的 1/32
- 6：hclk 时钟频率等于 sys_clk 时钟频率的 1/64
- 7：hclk 时钟频率等于 sys_clk 时钟频率的 1/128
- 8：hclk 时钟频率等于 sys_clk 时钟频率的 1/256
- >8：hclk 时钟频率等于 sys_clk 时钟频率的 1/512

位 7-5 PCLK0_DIV：PCLK0 分频系数。

- 0：pclk0 时钟频率等于 hclk 时钟频率
- 1：pclk0 时钟频率等于 hclk 时钟频率的 1/2
- 2：pclk0 时钟频率等于 hclk 时钟频率的 1/4
- 3：pclk0 时钟频率等于 hclk 时钟频率的 1/8
- >3：pclk0 时钟频率等于 hclk 时钟频率的 1/16

位 4-1 RESERVED：保留，不能修改。

位 0 RESERVED：保留，不能修改。

8.3.2 RCC_CR1

偏移量：0x004

复位值：0x00000000

对应寄存器在 AON 电源域。

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

位 31-12 RESERVED：保留，不能修改。

位 11 LPTIM1_EXT_CLK_SEL：LPTIM1 接口时钟选择。

- 0：由 LPTIM1_CLK_SEL 决定时钟来源
- 1：使用来自 IO 的 IN1 作为接口时钟

注意：

1. *需要保证在 LPTIM1_CLK_EN=0 时修改 LPTIM1 接口时钟选择。如果 LPTIM1_CLK_EN 已使能，则需要软件先关闭 LPTIM1_CLK_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置 LPTIM1 接口时钟选择。*
2. *与 LPTIM1_CLK_SEL 共同决定 LPTIM1 的接口时钟来源。*

位 10 LPTIM_EXT_CLK_SEL：LPTIM 接口时钟选择。

- 0：由 LPTIM_CLK_SEL 决定时钟来源
- 1：使用来自 IO 的 IN1 作为接口时钟

注意：

1. *需要保证在 LPTIM_CLK_EN=0 时修改 LPTIM 接口时钟选择。如果 LPTIM_CLK_EN 已使能，则需要软件先关闭 LPTIM_CLK_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置 LPTIM 接口时钟选择。*
2. *与 LPTIM_CLK_SEL 共同决定 LPTIM 的接口时钟来源。*

位 9-8 LPTIM1_CLK_SEL：LPTIM1 接口时钟选择。

- 0：PCLK0
- 1：RCO4M
- 2：XO32K
- 3：RCO32K

注意：

1. *需要保证在 LPTIM1_CLK_EN=0 时修改 LPTIM1 接口时钟选择。如果 LPTIM1_CLK_EN 已使能，则需要软件先关闭 LPTIM1_CLK_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置 LPTIM1 接口时钟选择。*
2. *与 LPTIM1 内部的时钟选择寄存器共同决定 LPTIM1 的接口时钟来源。*
3. *如果选择 PCLK0，则需要先配置 RCC_CGR1 的 LPTIM1_INF_CLK_EN 为 1。*

位 7-6 LPTIM_CLK_SEL：LPTIM 接口时钟选择。

- 0：PCLK0
- 1：RCO4M
- 2：XO32K
- 3：RCO32K

注意：

1. *需要保证在LPTIM_CLK_EN=0 时修改LPTIM 接口时钟选择。如果LPTIM_CLK_EN 已使能，则需要软件先关闭 LPTIM_CLK_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置LPTIM 接口时钟选择。*
2. *与LPTIM 内部的时钟选择寄存器共同决定LPTIM 的接口时钟来源。*
3. *如果选择PCLK0，则需要先配置RCC_CGR1 的LPTIM_INF_CLK_EN 为1。*

位 5-4 LCDCTRL_CLK_SEL：LCDCTRL 接口时钟选择。

- 0：使用 XO32K
- 1：使用 RCO32K
- >1：使用 RCO4M

位 3-2 LPUART_CLK_SEL：LPUART 接口时钟选择。

- 0：使用 XO32K
- 1：使用 RCO32K
- >1：使用 RCO4M

位 1 RTC_CLK_SEL：RTC 接口时钟选择。

- 0：使用 XO32K
- 1：使用 RCO32K

位 0 IWDG_CLK_SEL：IWDG 接口时钟选择。

- 0：使用 XO32K
- 1：使用 RCO32K

8.3.3 RCC_CR2

偏移量：0x008

复位值：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	SCC_CLK_SEL	ADCCTRL_CLK_SEL	I2S_CLK_SEL	QSPI_CLK_SEL
r/w	r/w	r/w	r/w	r/w

位 31-17 RESERVED：保留，不能修改。

位 16-15 UART0_CLK_SEL：UART0 接口时钟选择。

- 0：PCLK0
- 1：RCO4M

- 2：XO32K
- 3：XO24M

注意：*需要保证在 UART0_CLK_EN=0 时修改 UART0 接口时钟选择。如果 UART0_CLK_EN 已使能，则需要软件先关闭 UART0_CLK_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置 UART0 接口时钟选择。*

位 14-13 UART1_CLK_SEL：UART1 接口时钟选择。

- 0：PCLK0
- 1：RCO4M
- 2：XO32K
- 3：XO24M

注意：*需要保证在 UART1_CLK_EN=0 时修改 UART1 接口时钟选择。如果 UART1_CLK_EN 已使能，则需要软件先关闭 UART1_CLK_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置 UART1 接口时钟选择。*

位 12-11 UART2_CLK_SEL：UART2 接口时钟选择。

- 0：PCLK1
- 1：RCO4M
- 2：XO32K
- 3：XO24M

注意：*需要保证在 UART2_CLK_EN=0 时修改 UART2 接口时钟选择。如果 UART2_CLK_EN 已使能，则需要软件先关闭 UART2_CLK_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置 UART2 接口时钟选择。*

位 10-9 UART3_CLK_SEL：UART3 接口时钟选择。

- 0：PCLK1
- 1：RCO4M
- 2：XO32K
- 3：XO24M

注意：*需要保证在 UART3_CLK_EN=0 时修改 UART3 接口时钟选择。如果 UART3_CLK_EN 已使能，则需要软件先关闭 UART3_CLK_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置 UART3 接口时钟选择。*

位 8-7 SCC_CLK_SEL：SCC 接口时钟选择。

- 0：PCLK1
- 1：SYS_CLK
- 2：PLL
- 3：外部时钟 IOM_EXT_SCC_CLK

注意：*需要保证在 SCC_CLK_EN=0 时修改 SCC 接口时钟选择。如果 SCC_CLK_EN 已使能，则需要软件先关闭 SCC_CLK_EN，并且等待至少 2 拍切换前的时钟周期，然后再配置 SCC 接口时钟选择。*

位 6-5 ADCCTRL_CLK_SEL：ADCCTRL 接口时钟选择。

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

注意：*需要保证在 ADCCTRL_CLK_EN=0 时修改 ADCCTRL 接口时钟选择。如果 ADCCTRL_ CLK_EN 已使能，则需要软件先关闭 ADCCTRL_CLK_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置 ADCCTRL 接口时钟选择。*

位 4-2 I2S_CLK_SEL：I2S 接口时钟选择。

- 0：PCLK0
- 1：XO24M
- 2：PLL
- 3：XO32M
- >3：外部时钟 IOM_I2S_CLK

注意：

1. *需要保证在 I2S_CLK_EN=0 时修改 I2S 接口时钟选择。如果 I2S_CLK_EN 已使能，则需要软件先关闭 I2S_CLK_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置 I2S 接口时钟选择。*
2. *I2S 作为 slave 使用时，必须配置>3；I2S 作为 master 使用时，可根据功能需要进行选择。*

位 1-0 QSPI_CLK_SEL：QSPI 接口时钟选择。

- 0：HCLK
- 1：SYS_CLK
- >1：PLL

注意：*需要保证在 QSPI_CLK_EN=0 时修改 QSPI 接口时钟选择。如果 QSPI_CLK_EN 已使能，则需要软件先关闭 QSPI_CLK_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置 QSPI 接口时钟选择。*

8.3.4 RCC_CGR0

偏移量：0x00C

复位值：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	SCC_CLK_EN	ADCCTRL_CLK_EN
r/w	r/w	r/w	r/w	r/w	r/w	r/w	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

位 31 PWR_CLK_EN：PWR 时钟使能。

- 0：不使能
- 1：使能

位 30 DMAC0_CLK_EN：DMAC0 时钟使能。

- 0：不使能
- 1：使能

位 29 DMAC1_CLK_EN：DMAC1 时钟使能。

- 0：不使能
- 1：使能

位 28 CRC_CLK_EN：CRC 时钟使能。

- 0：不使能
- 1：使能

位 27 BASICTIM0_CLK_EN：BASICTIM0 时钟使能。

- 0：不使能
- 1：使能

位 26 BASICTIM1_CLK_EN：BASICTIM1 时钟使能。

- 0：不使能
- 1：使能

位 25 IOM0_CLK_EN：IOM0 时钟使能。

- 0：不使能
- 1：使能

位 24 IOM1_CLK_EN：IOM1 时钟使能。

- 0：不使能
- 1：使能

位 23 IOM2_CLK_EN：IOM2 时钟使能。

- 0：不使能
- 1：使能

位 22 IOM3_CLK_EN：IOM3 时钟使能。

- 0：不使能
- 1：使能

位 21 SYSCFG_CLK_EN：SYSCFG 时钟使能。

- 0：不使能
- 1：使能

位 20 UART0_CLK_EN：UART0 时钟使能。

- 0：不使能
- 1：使能

位 19 UART1_CLK_EN：UART1 时钟使能。

- 0：不使能

- 1：使能

位 18 UART2_CLK_EN：UART2 时钟使能。

- 0：不使能
- 1：使能

位 17 UART3_CLK_EN：UART3 时钟使能。

- 0：不使能
- 1：使能

位 16 LPUART_CLK_EN：LPUART 时钟使能。

- 0：不使能
- 1：使能

位 15 SSP0_CLK_EN：SSP0 时钟使能。

- 0：不使能
- 1：使能

位 14 SSP1_CLK_EN：SSP1 时钟使能。

- 0：不使能
- 1：使能

位 13 SSP2_CLK_EN：SSP2 时钟使能。

- 0：不使能
- 1：使能

位 12 I2C0_CLK_EN：I2C0 时钟使能。

- 0：不使能
- 1：使能

位 11 I2C1_CLK_EN：I2C1 时钟使能。

- 0：不使能
- 1：使能

位 10 I2C2_CLK_EN：I2C2 时钟使能。

- 0：不使能
- 1：使能

位 9 SCC_CLK_EN：SCC 时钟使能。

- 0：不使能
- 1：使能

位 8 ADCCTRL_CLK_EN：ADCCTRL 时钟使能。

- 0：不使能
- 1：使能

位 7 AFEC_CLK_EN：AFEC 时钟使能。

- 0：不使能
- 1：使能

位 6 LCDCTRL_CLK_EN：LCDCTRL 时钟使能。

- 0：不使能

- 1：使能

位 5 DACCTRL_CLK_EN：DACCTRL 时钟使能。

- 0：不使能
- 1：使能

位 4 LORAC_CLK_EN：LORAC 时钟使能。

- 0：不使能
- 1：使能

位 3 GPTIM0_CLK_EN：GPTIM0 时钟使能。

- 0：不使能
- 1：使能

位 2 GPTIM1_CLK_EN：GPTIM1 时钟使能。

- 0：不使能
- 1：使能

位 1 GPTIM2_CLK_EN：GPTIM2 时钟使能。

- 0：不使能
- 1：使能

位 0 GPTIM3_CLK_EN：GPTIM3 时钟使能。

- 0：不使能
- 1：使能

8.3.5 RCC_CGR1

偏移量：0x010

复位值：0x00000000

31-13	12	11	10	9	8	7
RESERVED	LPTIM1_INF_CLK_EN	LPTIM1_CLK_EN	RNGC_CLK_EN	LPTIM_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	LPTIM_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

位 31-13 RESERVED：保留，不能修改。

位 12 LPTIM1_INF_CLK_EN：LPTIM1 接口 PCLK 时钟使能。

- 0：不使能
- 1：使能

位 11 LPTIM1_CLK_EN：LPTIM1 时钟使能。

- 0：不使能
- 1：使能

注意：*打开 LPTIM1 时钟时，如果使用 PCLK，则要先打开 RCC_CGR1 中的 LPTIM_INF_CLK_ EN；关闭 LPTIM1 时钟时，如果使用 PCLK，要后关闭 RCC_CGR1 中的 LPTIM_INF_CLK_EN。*

位 10 RNGC_CLK_EN：RNGC 时钟使能。

- 0：不使能
- 1：使能

位 9 LPTIM_INF_CLK_EN：LPTIM 接口 PCLK 时钟使能。

- 0：不使能
- 1：使能

位 8 I2S_CLK_EN：I2S 时钟使能。

- 0：不使能
- 1：使能

位 7 SAC_CLK_EN：SAC 时钟使能。

- 0：不使能
- 1：使能

位 6 WWDG_CNT_CLK_EN：WWDG 计数时钟使能。

- 0：不使能
- 1：使能

位 5 QSPI_CLK_EN：QSPI 时钟使能。

- 0：不使能
- 1：使能

位 4 LPTIM_CLK_EN：LPTIM 时钟使能。

- 0：不使能
- 1：使能

注意：*打开 LPTIM 时钟时，如果使用 PCLK，则要先打开 RCC_CGR1 中的 LPTIM_INF_CLK_EN；关闭 LPTIM 时钟时，如果使用 PCLK，要后关闭 RCC_CGR1 中的 LPTIM_INF_CLK_EN。*

位 3 IWDG_CLK_EN：IWDG 时钟使能。

- 0：不使能
- 1：使能

位 2 WWDG_CLK_EN：WWDG 时钟使能。

- 0：不使能
- 1：使能

位 1 RTC_CLK_EN：RTC 时钟使能。

- 0：不使能
- 1：使能

位 0 SEC_CLK_EN：SEC 时钟使能。

- 0：不使能
- 1：使能

8.3.6 RCC_CGR2

偏移量：0x014

复位值：0x00000000

对应寄存器在 aon 电源域，对此寄存器进行操作前，需要读 rcc_sr 寄存器，等待全部的 done 为 1 后，才允许对此寄存器进行写操作，等待对应的 done 为 1 后，才允许对此寄存器进行读操作。

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

位 31-6 RESERVED：保留，不能修改。

位 5 LPTIM1_AON_CLK_EN：LPTIM1 的 AON 域接口时钟使能。

- 0：不使能
- 1：使能

位 4 LPTIM_AON_CLK_EN：LPTIM 的 AON 域接口时钟使能。

- 0：不使能
- 1：使能

位 3 LCDCTRL_AON_CLK_EN：LCDCTRL 的 AON 域接口时钟使能。

- 0：不使能
- 1：使能

位 2 LPUART_AON_CLK_EN：LPUART 的 AON 域接口时钟使能。

- 0：不使能
- 1：使能

位 1 RTC_AON_CLK_EN：RTC 的 AON 域接口时钟使能。

- 0：不使能
- 1：使能

位 0 IWDG_AON_CLK_EN：IWDG 的 AON 域接口时钟使能。

- 0：不使能
- 1：使能

8.3.7 RCC_RST0

偏移量：0x018

复位值：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	SCC_RST_N	ADCCTRL_RST_N	AFEC_RST_N	LCDCTRL_RST_N
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DACCTRL_RST_N	LORAC_RST_N	IOM_RST_N	GPTIM0_RST_N	GPTIM1_RST_N	GPTIM2_RST_N	GPTIM3_RST_N	BASICTIM0_RST_N
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BASICTIM1_RST_N	LPTIM_RST_N	IWDG_RST_N	WWDG_RST_N	RTC_RST_N	CRC_RST_N	SEC_RST_N	SAC_RST_N
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

位 31 UART0_RST_N：UART0 复位控制。

- 0：复位
- 1：不复位

位 30 UART1_RST_N：UART1 复位控制。

- 0：复位
- 1：不复位

位 29 UART2_RST_N：UART2 复位控制。

- 0：复位
- 1：不复位

位 28 UART3_RST_N：UART3 复位控制。

- 0：复位
- 1：不复位

位 27 LPUART_RST_N：LPUART 复位控制。

- 0：复位
- 1：不复位

位 26 SSP0_RST_N：SSP0 复位控制。

- 0：复位
- 1：不复位

位 25 SSP1_RST_N：SSP1 复位控制。

- 0：复位

- 1：不复位

位 24 SSP2_RST_N：SSP2 复位控制。

- 0：复位
- 1：不复位

位 23 QSPI_RST_N：QSPI 复位控制。

- 0：复位
- 1：不复位

位 22 I2C0_RST_N：I2C0 复位控制。

- 0：复位
- 1：不复位

位 21 I2C1_RST_N：I2C1 复位控制。

- 0：复位
- 1：不复位

位 20 I2C2_RST_N：I2C2 复位控制。

- 0：复位
- 1：不复位

位 19 SCC_RST_N：SCC 复位控制。

- 0：复位
- 1：不复位

位 18 ADCCTRL_RST_N：ADCCTRL 复位控制。

- 0：复位
- 1：不复位

位 17 AFEC_RST_N：AFEC 复位控制。

- 0：复位
- 1：不复位

位 16 LCDCTRL_RST_N：LCDCTRL 复位控制。

- 0：复位
- 1：不复位

位 15 DACCTRL_RST_N：DACCTRL 复位控制。

- 0：复位
- 1：不复位

位 14 LORAC_RST_N：LORAC 复位控制。

- 0：复位
- 1：不复位

位 13 IOM_RST_N：IOM 复位控制。

- 0：复位
- 1：不复位

位 12 GPTIM0_RST_N：GPTIM0 复位控制。

- 0：复位

- 1：不复位

位 11 GPTIM1_RST_N：GPTIM1 复位控制。

- 0：复位
- 1：不复位

位 10 GPTIM2_RST_N：GPTIM2 复位控制。

- 0：复位
- 1：不复位

位 9 GPTIM3_RST_N：GPTIM3 复位控制。

- 0：复位
- 1：不复位

位 8 BASICTIM0_RST_N：BASICTIM0 复位控制。

- 0：复位
- 1：不复位

位 7 BASICTIM1_RST_N：BASICTIM1 复位控制。

- 0：复位
- 1：不复位

位 6 LPTIM_RST_N：LPTIM 复位控制。

- 0：复位
- 1：不复位

位 5 IWDG_RST_N：IWDG 复位控制。

- 0：复位
- 1：不复位

位 4 WWDG_RST_N：WWDG 复位控制。

- 0：复位
- 1：不复位

位 3 RTC_RST_N：RTC 复位控制。

- 0：复位
- 1：不复位

位 2 CRC_RST_N：CRC 复位控制。

- 0：复位
- 1：不复位

位 1 SEC_RST_N：SEC 复位控制。

- 0：复位
- 1：不复位

位 0 SAC_RST_N：SAC 复位控制。

- 0：复位
- 1：不复位

8.3.8 RCC_RST1

偏移量：0x01C

复位值：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

位 31-5 RESERVED：保留，不能修改。

位 4 LPTIM1_RST_N：LPTIM1 复位控制。

- 0：复位
- 1：不复位

位 3 RNGC_RST_N：RNGC 复位控制。

- 0：复位
- 1：不复位

位 2 I2S_RST_N：I2S 复位控制。

- 0：复位
- 1：不复位

位 1 DMAC0_RST_N：DMAC0 复位控制。

- 0：复位
- 1：不复位

位 0 DMAC1_RST_N：DMAC1 复位控制。

- 0：复位
- 1：不复位

8.3.9 RCC_RST_SR

偏移量：0x020

复位值：0x00000040

注意：*BOR_RESET_SR 与 STANDBY_RESET_SR 的寄存器在 AON 域。*

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

位 31-7 RESERVED：保留，不能修改。

位 6 BOR_RESET_SR：BOR 系统复位标志。硬件置 1，软件写 1 清 0。

- 0：无 BOR 系统复位发生
- 1：有 BOR 系统复位发生

位 5 IWDG_RESET_SR：IWDG 系统复位标志。硬件置 1，软件写 1 清 0。

- 0：无 IWDG 系统复位发生
- 1：有 IWDG 系统复位发生

位 4 WWDG_RESET_SR：WWDG 系统复位标志。硬件置 1，软件写 1 清 0。

- 0：无 WWDG 系统复位发生
- 1：有 WWDG 系统复位发生

位 3 EFC_RESET_SR：EFC 系统复位标志。硬件置 1，软件写 1 清 0。

- 0：无 EFC 系统复位发生
- 1：有 EFC 系统复位发生

位 2 CPU_RESET_SR：CPU 系统复位标志。硬件置 1，软件写 1 清 0。

- 0：无 CPU 系统复位发生
- 1：有 CPU 系统复位发生

位 1 SEC_RESET_SR：SEC 系统复位标志。硬件置 1，软件写 1 清 0。

- 0：无 SEC 系统复位发生
- 1：有 SEC 系统复位发生

位 0 STANDBY_RESET_SR：Standby 系统复位标志。硬件置 1，软件写 1 清 0。

- 0：无 MPU 系统复位发生
- 1：有 MPU 系统复位发生

8.3.10 RCC_RST_CR

偏移量：0x024

复位值：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

位 31-6 RESERVED：保留，不能修改。

位 5 IWDG_RESET_REQ_EN：IWDG 系统复位使能。

- 0：不使能
- 1：使能

位 4 WWDG_RESET_REQ_EN：WWDG 系统复位使能。

- 0：不使能
- 1：使能

位 3 EFC_RESET_REQ_EN：EFC 系统复位使能。

- 0：不使能
- 1：使能

位 2 CPU_RESET_REQ_EN：CPU 系统复位使能。

- 0：不使能
- 1：使能

位 1 SEC_RESET_REQ_EN：SEC 系统复位使能。

- 0：不使能
- 1：使能

位 0 RESERVED：保留，不能修改。

8.3.11 RCC_SR

偏移量：0x028

复位值：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

位 31-6 RESERVED：保留，不能修改。

位 5 SET_LPTIM1_AON_CLK_EN_DONE：LPTIM1_AON_CLK_EN 配置完成状态。硬件控制置 1 与清 0。

- 0：配置正在进行中
- 1：配置已完成

位 4 SET_LPTIM_AON_CLK_EN_DONE：LPTIM_AON_CLK_EN 配置完成状态。硬件控制置 1 与清 0。

- 0：配置正在进行中
- 1：配置已完成

位 3 SET_LCDCTRL_AON_CLK_EN_DONE：LCDCTRL_AON_CLK_EN 配置完成状态。硬件控制置 1 与清 0。

- 0：配置正在进行中
- 1：配置已完成

位 2 SET_LPUART_AON_CLK_EN_DONE：LPUART_AON_CLK_EN 配置完成状态。硬件控制置 1 与清 0。

- 0：配置正在进行中
- 1：配置已完成

位 1 SET_RTC_AON_CLK_EN_DONE：RTC_AON_CLK_EN 配置完成状态。硬件控制置 1 与清 0。

- 0：配置正在进行中
- 1：配置已完成

位 0 SET_IWDG_AON_CLK_EN_DONE：IWDG_AON_CLK_EN 配置完成状态。硬件控制置 1 与清 0。

- 0：配置正在进行中
- 1：配置已完成

8.3.12 RCC_SR1

偏移量：0x02C

复位值：0x00000000

为避免时钟来源切换或分频变化时出现毛刺，在切换配置前先关闭时钟使能，可通过此寄存器确定使能是否关闭。

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	SCC_CLK_EN_SYNC	ADCCTRL_CLK_EN_SYNC	LPTIM_CLK_EN_SYNC	QSPI_CLK_EN_SYNC	LPUART_CLK_EN_SYNC
r	r	r	r	r	r
9	**8**	**7**	**6**	**5**	**4**
LCDCTRL_CLK_EN_SYNC	IWDG_CLK_EN_SYNC	RTC_CLK_EN_SYNC	MCO_CLK_EN_SYNC	I2S_CLK_EN_SYNC	LPTIM_AON_CLK_EN_SYNC
r	r	r	r	r	r

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

位 31-21 RESERVED：保留，不能修改。

位 20 LPTIM1_CLK_EN_SYNC：LPTIM1_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 19 LPTIM1_AON_CLK_EN_SYNC：LPTIM1_AON_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 18 UART0_CLK_EN_SYNC：UART0_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 17 UART1_CLK_EN_SYNC：UART1_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 16 UART2_CLK_EN_SYNC：UART2_CLK_EN 状态。

- 0：门控关闭

- 1：门控打开

位 15 UART3_CLK_EN_SYNC：UART3_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 14 SCC_CLK_EN_SYNC：SCC_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 13 ADCCTRL_CLK_EN_SYNC：ADCCTRL_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 12 LPTIM_CLK_EN_SYNC：LPTIM_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 11 QSPI_CLK_EN_SYNC：QSPI_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 10 LPUART_CLK_EN_SYNC：LPUART_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 9 LCDCTRL_CLK_EN_SYNC：LCDCTRL_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 8 IWDG_CLK_EN_SYNC：IWDG_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 7 RTC_CLK_EN_SYNC：RTC_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 6 MCO_CLK_EN_SYNC：MCO_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 5 I2S_CLK_EN_SYNC：I2S_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 4 LPTIM_AON_CLK_EN_SYNC：LPTIM_AON_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 3 LCDCTRL_AON_CLK_EN_SYNC：LCDCTRL_AON_CLK_EN 状态。

- 0：门控关闭

- 1：门控打开

位 2 LPUART_AON_CLK_EN_SYNC：LPUART_AON_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 1 RTC_AON_CLK_EN_SYNC：RTC_AON_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

位 0 IWDG_AON_CLK_EN_SYNC：IWDG_AON_CLK_EN 状态。

- 0：门控关闭
- 1：门控打开

8.3.13 RCC_CR3

偏移量：0x030

复位值：0x00000000

31-28	27-16	15-8	7-0
RESERVED	SCC_CLK_DIV	I2S_MCLK_DIV	I2S_SCLK_DIV
r	r/w	r/w	r/w

位 31-28 RESERVED：保留，不能修改。

位 27-16 SCC_CLK_DIV：SCC 接口时钟分频控制。

- 0：不分频
- 1：不分频
- 2：2 分频
- 3：3 分频
- N：N 分频

注意：

1. *需要保证在 SCC_CLK_EN=0 时修改 SCC 接口时钟分频。如果 SCC_CLK_EN 已使能，则需要软件先关闭 SCC_CLK_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置 SCC 接口时钟分频。*
2. *输出时钟的占空比为 50%。*

位 15-8 I2S_MCLK_DIV：I2S 接口时钟 MCLK 分频控制。

- 0：不分频
- 1：不分频
- 2：2 分频
- 3：3 分频
- N：N 分频

注意：

1. *需要保证在 I2S_CLK_EN=0 时修改 I2S 接口时钟分频。如果 I2S_CLK_EN 已使能，则需要软件先关闭 I2S_CLK_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置 I2S 接口时钟分频。*

2. *I2S 作为 slave 使用时，必须配置为 0 或 1；I2S 作为 master 使用时，可根据功能需要进行选择。*
3. *输出时钟的占空比为 50%。*

位 7-0 I2S_SCLK_DIV：I2S 接口时钟 SCLK 分频控制。

- 0：不分频
- 1：不分频
- 2：2 分频
- 3：3 分频
- N：N 分频

注意：

1. *需要保证在 I2S_CLK_EN=0 时修改 I2S 接口时钟分频。如果 I2S_CLK_EN 已使能，则需要软件先关闭 I2S_CLK_EN，并且等待至少 2 拍切换前的时钟周期或者查询 RCC_SR1，然后再配置 I2S 接口时钟分频。*
2. *I2S 作为 slave 使用时，必须配置为 0 或 1；I2S 作为 master 使用时，可根据功能需要进行选择。*
3. *输出时钟的占空比为 50%。*

9. Interrupt

9.1 主要功能

- 共支持 37 个 IRQ 中断。
- 每个 IRQ 中断支持 8 个中断优先级配置。

9.2 SysTick 功能

SysTick calibration 值为 0x147，使用 32.768KHz 时钟用于 SysTick 计数时可产生精确的 10ms 间隔。

9.3 中断向量表

中断向量表如下所示:

表 9-1 中断向量表

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

Position	Priority	Type of priority	Acronym	Description	Address
2	9	settable	wwdg		0x0000_0048
3	10	settable	efc		0x0000_004C
4	11	settable	uart3		0x0000_0050
5	12	settable	i2c2		0x0000_0054
6	13	settable	uart0		0x0000_0058
7	14	settable	uart1		0x0000_005C
8	15	settable	uart2		0x0000_0060
9	16	settable	lpuart		0x0000_0064
10	17	settable	ssp0		0x0000_0068
11	18	settable	ssp1		0x0000_006C
12	19	settable	qspi		0x0000_0070
13	20	settable	i2c0		0x0000_0074
14	21	settable	i2c1		0x0000_0078
15	22	settable	scc		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	lptim		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. On-Chip Flash

10.1 简介

- Flash 组成：整个 Flash 分为 Flash info 区和 Flash main 区
- Flash 容量:
  - Flash info 区：共 16 KB
  - Flash main 区：ASR6601SE 为 256 KB，ASR6601CB 为 128 KB
- Page erase (4KB), Mass erase (all flash main)

10.2 主要特性

- Flash 基本操作，包括 read, program, page erase, mass erase
- Flash 读拍数
- Flash 读加速
- Flash 指令预取，1 个深度的预取 buffer
- Flash program 操作支持 single 和连续方式
- Flash info 区的 option bytes 操作
- 支持产生中断信号

10.3 功能描述

10.3.1 Flash info 区划分

Flash info 区主要包含 Option Bytes，Factory Bytes，OTP 和 BootLoader 四个部分，具体划分见如下表格。

表 10-1 Flash info 区划分

起始地址	功能描述	地址范围
0x10003000	Option Bytes	4KB
0x10002000	Factory Bytes	4KB
0x10001C00	OTP	1KB
0x10000000	BootLoader	7KB

10.3.2 EFC_CR 保护

默认情况下，EFC_CR不能被修改，软件在操作EFC_CR前需要按照以下顺序配置保护序列，才能够正确操作 EFC_CR 寄存器，中间插入了错误序列，则操作无效，需要重新配置；

(1) 先写 0x8C9DAEBF 到 EFC_PROTECT_SEQ 寄存器

(2) 再写 0x13141516 到 EFC_PROTECT_SEQ 寄存器

10.3.3 读拍数

为提高 Flash 读性能，Flash 读拍数需根据 SYS_CLK 的频率做相应的配置，可以通过配置 EFC_TIMING_CFG 的 READ_NUM 来配置 Flash 读拍数，Flash 读拍数为（READ_NUM+1）个 SYS_CLK 时钟周期。不同的 SYS_CLK 频率下，READ_NUM 的配置如下：

- SYS_CLK 频率为 48MHz 时，READ_NUM 要大于等于 2。
- SYS_CLK 频率为 32MHz 时，READ_NUM 要大于等于 1。
- SYS_CLK 频率为 24MHz 时，READ_NUM 要大于等于 1。
- SYS_CLK 频率为 4MHz 时，READ_NUM 要大于等于 0。
- SYS_CLK 频率为 32KHz 时，READ_NUM 要大于等于 0。

将 SYS_CLK 切换到快时钟源时的操作：

(1) 修改 EFC_TIMING_CFG 寄存器的 READ_NUM 值，以匹配切换后的 SYS_CLK；

(2) 等待 EFC_SR 寄存器中的 READ_NUM_DONE 状态位置位；

(3) 修改 RCC_CR0 寄存器中的 SYS_CLK_SEL 值，切换到新的时钟源。

将 SYS_CLK 切换到慢时钟源时的操作：

(1) 修改 RCC_CR0 寄存器中的 SYS_CLK_SEL 值，切换到新的时钟源；

(2) 修改 EFC_TIMING_CFG 寄存器的 READ_NUM 值，以匹配切换后的 SYS_CLK；

(3) 等待 EFC_SR 寄存器中的 READ_NUM_DONE 状态位置位。

注意：*当修改 SYS_CLK 时钟源选择时，若切为快时钟源，则先配置 READ_NUM 变大，再配置时钟源选择；反之若切为慢时钟源时，先配置时钟源选择，再配置 READ_NUM 变小。*

10.3.4 读加速

默认不开启，若当前 READ_NUM < (2^HCLK_DIV)时可以配置开启读加速功能，以达到最大的总线访问效率；配置读加速使能，需要在 READ_NUM 和 HCLK_DIV 配置完成后进行。

注意：*读加速功能与指令预取功能不能同时开启。*

10.3.5 指令预取

默认不开启，若 READ_NUM ≥ (2^HCLK_DIV)时，读加速功能不能开启，可以选择开启指令预取功能以提高读取效率。

注意：*读加速功能与指令预取功能不能同时开启。*

10.3.6 Flash Program

Flash 的 program 操作主要分为 single program 和连续 program 方式。

- **Single program**

  Single program 每次烧写 2 个 word（8byte）。

- **连续 program**

  连续 program 每次烧写一整条 word line（512byte），连续 program 时无法从 flash 读取和执行程序，所以连续 program 代码必须执行在 RAM 中。

Single program 的操作如下：

(1) 将 EFC_CR 寄存器的 PROG_EN 位置 1;
(2) 将要写入数据低 4 个字节写入 EFC_PROG_DATA0 寄存器;
(3) 将要写入数据高 4 个字节写入 EFC_PROG_DATA1 寄存器;
(4) 向将要写入的 Flash 地址写入任意值;
(5) 等待 EFC_SR 寄存器的 OPERATION_DONE 位置 1;
(6) 向 EFC_SR 寄存器的 OPERATION_DONE 位写 1，清除该标志位。

连续 program 的操作如下：

(1) 将 EFC_CR 寄存器的 PROG_EN，WRITE_RELEASE_EN 和 PROG_MODE 位置 1;
(2) 等待 EFC_SR 寄存器的 PROG_DATA_WAIT 位置 1;
(3) 将要写入数据低 4 个字节写入 EFC_PROG_DATA0 寄存器;
(4) 将要写入数据高 4 个字节写入 EFC_PROG_DATA1 寄存器;
(5) 向将要写入的 Flash 地址写入任意值;
(6) 等待 EFC_SR 寄存器的 PROG_DATA_WAIT 位置 1;
(7) 继续将数据写入 EFC_PROG_DATA0 和 EFC_PROG_DATA1 寄存器;
(8) 循环执行 6）和 7），直到 512 字节写完
(9) 等待 EFC_SR 寄存器的 OPERATION_DONE 位置 1;
(10) 向 EFC_SR 寄存器的 OPERATION_DONE 位写 1，清除该标志位。

10.3.7 Flash Erase

Flash 的 erase 操作主要分为 page erase 和 mass erase 两种。

- **Page erase**

  Page erase 的单位为 4KB。

- **Mass erase**

  Mass erase 后，整个 Flash main 区域被擦除为 0xFF。

Page erase 的操作如下：

(1) 将 EFC_CR 寄存器的 PAGE_ERASE_EN 位置 1；
(2) 向将要擦除的 Flash 地址写入任意值；
(3) 等待 EFC_SR 寄存器的 OPERATION_DONE 位置 1；
(4) 向 EFC_SR 寄存器的 OPERATION_DONE 位写 1，清除该标志位。

Mass erase 的操作如下：

(1) 将 EFC_CR 寄存器的 MASS_ERASE_EN 位置 1；
(2) 向 0x08000000 地址写入任意值；
(3) 等待 EFC_SR 寄存器的 OPERATION_DONE 位置 1；
(4) 向 EFC_SR 寄存器的 OPERATION_DONE 位写 1，清除该标志位。

10.4 Flash Option Bytes

Flash option bytes 主要分为 option0 和 option1 两部分。

10.4.1 Flash Option0

Option0 共 64 位，其格式如下：

表 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

位 63-50 Reserved：保留，不能修改。

位 49-44 WR_PROTECT_END：写保护区域的结束 Page 地址。

当 *WR_PROTECT_START* > *WR_PROTECT_END* 时，写保护区域不使能；默认不使能。

位 43-38 WR_PROTECT_START：写保护区域的开始 Page 地址。

当 *WR_PROTECT_START* > *WR_PROTECT_END* 时，写保护区域不使能；默认不使能。

位 37-32 EXE_ONLY2_END：ExeOnly 区域 2 的结束 Page 地址。

当 *EXE_ONLY2_START* > *EXE_ONLY2_END* 时，ExeOnly 区域 2 不使能；

默认不使能，修改时只能使能或扩大区域，不能减少或关闭。

位 31-26 EXE_ONLY2_START：ExeOnly 区域 2 的开始 Page 地址。

当 *EXE_ONLY2_START* > *EXE_ONLY2_END* 时，ExeOnly 区域 2 不使能；

默认不使能，修改时只能使能或扩大区域，不能减少或关闭。

位 25 EXE_ONLY_KEEP：ExeOnly 区域在 DebugLevel 由 1 变为 0 时，是否保留：

- 0：不保留 ExeOnly 区域
- 1：保留 ExeOnly 区域

支持修改，但只能写 0；DebugLevel 从 1 变为 0 时，EXE_ONLY_KEEP 自动变 0。

位 24-19 EXE_ONLY1_END：ExeOnly 区域 1 的结束 Page 地址。

当 *EXE_ONLY1_START* > *EXE_ONLY1_END* 时，ExeOnly 区域 1 不使能；

默认不使能，修改时只能使能或扩大区域，不能减少或关闭。

位 18-13 EXE_ONLY1_START：ExeOnly 区域 1 的开始 Page 地址。

当 *EXE_ONLY1_START* > *EXE_ONLY1_END* 时，ExeOnly 区域 1 不使能；

默认不使能，修改时只能使能或扩大区域，不能减少或关闭。

位 12-5 DEBUG_LEVEL：debug_level 配置。

- AA：Level0
- CC：Level2
- 其它值：Level1

位 4 Reserved：保留，不能修改。

位 3 SYS_SRAM_RESET：System 复位后 Startup 过程中是否清除系统 SRAM:

- 1：清除系统 SRAM
- 0：不清除系统 SRAM

位 2 FLASH_BOOT1：用于启动模式判断。

位 1 USE_FLASH_BOOT0：用于启动模式判断。

位 0 FLASH_BOOT0：用于启动模式判断。

启动模式判断如下:

表 10-3 SoC 启动方式配置表

Debug_Level	Use_Flash_BOOT0	FLASH_Boot0	BOOT0 pin	FLASH_Boot1	Main_Flash_Empty	Boot Config
2	X	X	X	X	X	Boot from Flash Main
<2	0	X	0	X	0	Boot from Flash Main
<2	0	X	0	X	1	Boot from Flash Bootloader
<2	0	X	1	1	X	Boot from Flash Bootloader
<2	0	X	1	0	X	Boot from System SRAM
<2	1	1	X	X	0	Boot from Flash Main
<2	1	1	X	X	1	Boot from Flash Bootloader
<2	1	0	X	1	X	Boot from Flash Bootloader
<2	1	0	X	0	X	Boot from System SRAM

10.4.2 Flash Option1

Option1 共 64 位，其格式如下：

表 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

位 63-56 Reserved：保留，不能修改。

位 55 SYSRAM_HIDE_EN：SysSramHide 区域使能控制。

- 0：SysSramHide 区域使能
- 1：SysSramHide 区域不使能

仅当 FlashSecure 区域使能时才有效。

位 54-49 SYSRAM_HIDE_START：SysSramHide 区域的开始 Page 地址。

必须在 SysSramSecure 区域内才使能，区域范围由 SysSramHideStart 开始一直到 SysSramSecureEnd；

仅当 FlashSecure 区域使能时才有效。

位 48 FLASH_HIDE_EN：FlashHide 区域使能控制。

- 0：FlashHide 区域使能
- 1：FlashHide 区域不使能

仅当 FlashSecure 区域使能时才有效。

位 47-42 FLASH_HIDE_START：FlashHide 区域的开始 Page 地址。

必须在 FlashSecure 区域内才使能，区域范围由 FlashHideStart 开始一直到 FlashSecureEnd；

仅当 FlashSecure 区域使能时才有效。

位 41-37 RETRAM_SECURE_END：RetRam Secure 区域的结束地址。

当 *RETRAM_SECURE_START* > *RETRAM_SECURE_END* 时，则该区域安全不使能；

仅当 FlashSecure 区域使能时才有效。

位 36-32 RETRAM_SECURE_START：RetRam Secure 区域的开始地址。

当 *RETRAM_SECURE_START* > *RETRAM_SECURE_END* 时，则该区域安全不使能；

仅当 FlashSecure 区域使能时才有效。

位 31-24 Reserved：保留，不能修改。

位 23-18 SYSRAM_SECURE_END：SysRam Secure 区域的结束地址。

当 *SYSRAM_SECURE_START* > *SYSRAM_SECURE_END* 时，则该区域安全不使能；

仅当 FlashSecure 区域使能时才有效。

位 17-12 SYSRAM_SECURE_START：SysRam Secure 区域的开始地址。

当 *SYSRAM_SECURE_START* > *SYSRAM_SECURE_END* 时，则该区域安全不使能；

仅当 FlashSecure 区域使能时才有效。

位 11-6 FLASH_SECURE_END：Flash Secure 区域的结束地址。

当 *FLASH _SECURE_START > FLASH _SECURE_END* 时，则该区域安全不使能；

FlashSecure 区域使能也是安全区域总使能开关，当 FlashSecure 去使能时会触发擦除操作。

位 5-0 FLASH _SECURE_START：Flash Secure 区域的开始地址。

当 *FLASH_SECURE_START > FLASH _SECURE_END* 时，则该区域安全不使能；

FlashSecure 区域使能也是安全区域总使能开关，当 FlashSecure 去使能时会触发擦除操作。

10.5 On-Chip Flash 相关寄存器描述

寄存器基地址：0x40020000

表 10-5 On-Chip Flash 寄存器列表

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

10.5.1 EFC_CR

偏移量：0x00

复位值：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

位 31 INFO_BYTE_LOAD：Info byte load 复位请求。

- 写 0：无效
- 写 1：将引起系统复位，重新加载 Flash info 区中的信息，如 option 信息等，该位由硬件自动清零

位 30-10 Reserved：保留，不能修改。

位 9 ECC_DIS：ECC 编码除能控制。

位 8 OPTION_OPR_EN：Option 操作使能。

- 0：Option 操作不使能
- 1：Option 操作使能

注意：

1. *OPTION_OPR_EN, PROG_EN 和 PAGE_ERASE_EN 任意两个不能同时配置使能。*
2. *每执行完一次 option 操作后，需要执行一次复位操作，配置才能真正生效。*

位 7 Reserved：保留，不能修改。

位 6 WRITE_RELEASE_EN：Flash 执行编程、擦除（包括 Mass）、Option 操作时，AHB 总线 hold/release 模式选择。

- 0：hold 模式
- 1：release 模式

注意：*一旦配置为 release 模式，则编程/擦除过程中不能读 Flash 区域，也不能在 Flash 中执行程序，否则操作会被拦截，返回数据未知，并且会起 FLASHBUSY_ERR 错误标志；但可以访问 SR 寄存器，等待操作完成。*

位 5 PREFETCH_EN：Flash 指令预取功能使能。

- 0：不使能预取
- 1：使能预取

注意：*预取使能和读加速使能控制不能同时开启。*

位 4 READ_ACC_EN：Flash 读操作加速使能控制。

- 0：不使能读加速（hold 模式）
- 1：使能读加速（release 模式）

注意：

1. *当 READ_NUM < (2^HCLK_DIV)时，可以配置为 1，开启读加速功能；配置读加速使能，需要在 READ_NUM 和 HCLK_DIV 配置完成后进行。*
2. *预取使能和读加速使能控制不要同时开启。*

位 3 PROG_MODE：flash program 模式控制。

- 0：单次编程模式，每次 program 将 EFC_PROG_DATA1 与 EFC_PROG_DATA0 写入指定地址。
- 1：WL 连续编程模式，该模式下会自动将 1 个 WL（512byte）中的连续地址均编程，软件需要查询 PROG_DATA_WAIT 标志来决定是否要将新的数据写入 EFC_PROG_DATA1 与 EFC_PROG_DATA0，以便于连续编程。

注意：

1. *Flash 中的 ECC 编码格式为 64+8，因此每次编程为偶数个 Word。*
2. *WL 连续编程模式下需要将 WRITE_RELEASE_EN 配置为 1，并且编程过程中只能读写 EFC_SR, EFC_PROG_DATA1 与 EFC_PROG_DATA0，不能读取 Flash 区域，也不能在 Flash 中执行程序。*

位 2 PROG_EN：flash program 使能控制。

- 0：对 Flash 存储空间的写操作不产生 flash program
- 1：对 Flash 存储空间的写操作产生 flash program

注意：

1. *单次编程模式下通过向 Flash 地址写入数据启动编程，EFC_PROG_DATA0 的数据会写入该地址 8 字节对齐的低地址空间，EFC_PROG_DATA1 写入该地址 8 字节对齐的高地址空间。*
2. *WL 连续编程模式下通过向 Flash 地址写入数据启动编程，编程地址依次累加，直到一条 WL 编程结束。*

位 1 PAGE_ERASE_EN：flash page erase 使能控制。

- 0：对 Flash 存储空间的写操作不产生 flash page erase
- 1：对 Flash 存储空间的写操作产生 flash page erase，page 地址为写入地址所在的 PAGE

位 0 MASS_ERASE_EN：flash mass erase 使能控制。

- 0：对 Flash 存储空间的写操作不产生 flash mass erase
- 1：对 Flash 存储空间的写操作产生 flash mass erase

注意：

1. *写操作的地址为 flash main 区地址则仅执行 main 区的 mass erase；若写操作的地址为 flash info 区，则 main+info 区均执行 mass erase。*
2. *请不要对 info 区进行 mass erase，否则芯片将被毁掉。*

10.5.2 EFC_INT_EN

偏移量：0x04

复位值：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	OPTION_DONE_INT_EN
r/w	r/w	r/w	r	r/w

位 31-9 Reserved：保留，不能修改。

位 8 TWO_BIT_ERROR_INT_EN：ecc two_bit_error 中断使能。

- 0：不使能
- 1：使能

位 7 ONE_BIT_CORRECT_INT_EN：ecc one_bit_correct 中断使能。

- 0：不使能
- 1：使能

位 6 PROG_ERR_INT_EN：prog_err 中断使能。

- 0：不使能
- 1：使能

位 5 PAGE_ERASE_ERR_INT_EN：page_erase_err 中断使能。

- 0：不使能
- 1：使能

位 4 OPTION_WR_ERR_INT_EN：option_wr_err 中断使能。

- 0：不使能
- 1：使能

位 3 FLASHBUSY_ERR_INT_EN：flashbusy_err 中断使能。

- 0：不使能
- 1：使能

位 2 PROG_DATA_WAIT_INT_EN：prog_data_wait 中断使能。

- 0：不使能
- 1：使能

位 1 Reserved：保留，不能修改。

位 0 OPTION_DONE_INT_EN：operation_done 中断使能。

- 0：不使能
- 1：使能

10.5.3 EFC_SR

偏移量：0x08

复位值：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	OPTION_DONE
r/w	r/w	r/w	r	r/w

位 31-9 Reserved：保留，不能修改。

位 8 TWO_BIT_ERROR：Flash 读数据 two bit ECC 错误标志。

- 0：无 two bit 错误发生
- 1：Flash 读数据发生 two bit 错误，ECC 不纠正

位 7 ONE_BIT_CORRECT：Flash 读数据单 bit ECC 错误纠正标志。

- 0：无错误发生
- 1：Flash 读数据发生单 bit 错误，ECC 进行了纠正

位 6 PROG_ERR：Flash info 部分区域不支持编程操作（PROG_EN），对这些区域的编程操作会被拦截，并且起该标志，硬件置 1，软件写 1 清 0。

- 0：无 Program 错误产生
- 1：产生 Program 错误

注意：option 区域不能通过直接编程的方式实现写，需要选择对应操作；bootloader 区域不能编程。

位 5 PAGE_ERASE_ERR：Flash info 区域不支持擦除操作，对 info 的擦除操作会被拦截，并且起该标志，硬件置 1，软件写 1 清 0。

- 0：无 Page erase 错误
- 1：发生了 Page erase 错误

位 4 OPTION_WR_ERR：Option 区域的配置需要满足一定条件，不满足的 option 操作会被拦截，并且起该标志，硬件置 1，软件写 1 清 0。

- 0：无 Option byte 写权限错误
- 1：发生 Option byte 写权限错误

Option 区域配置约束条件为：

1. Flash exe_only1/2 区域使能时不能关闭或缩小；
2. exe_only_keep 不能由 0 写 1
3. secure_area_en 使能时非安全操作只能配置 option 将 secure_area_en 关闭

位 3 FLASHBUSY_ERR：Flash 在执行编程、擦除（包括 mass）、option 操作时，软件发起了 Flash 空间读操作，则读操作会拦截，总线返回数据不确定，为异常状态，硬件置 1，软件写 1 清 0。

- 0：无错误发生
- 1：Flash 操作期间发生读操作错误

位 2 PROG_DATA_WAIT：Flash 连续编程模式下等待数据写入状态，硬件置 1，软件写 EFC_PROG_DATA0 和 EFC_PROG_DATA1 后硬件自动清零，或者软件写 1 清零。

- 0：EFC_PROG_DATA0 和 EFC_PROG_DATA1 数据写入完毕
- 1：等待 EFC_PROG_DATA0 和 EFC_PROG_DATA1 数据写入

位 1 READ_NUM_DONE：READ_NUM_DONE 状态，指示 READ_NUM 配置是否已完成。硬件控制置 1 与清 0。

- 0：未完成
- 1：完成

位 0 OPTION_DONE：operation_done 状态，用于指示 mass erase、page erase、program、option 操作是否完成。硬件置 1，软件写 1 清 0。

- 0：未完成
- 1：完成

10.5.4 EFC_PROG_DATA0

偏移量：0x0C

复位值：0x00000000

31-0
PROG_DATA0
r/w

位 31-0 PROG_DATA0：program 编程数据 0。

注意：*program 时先写入 EFC_PROG_DATA0，再写入 EFC_PROG_DATA1*

10.5.5 EFC_PROG_DATA1

偏移量：0x10

复位值：0x00000000

31-0
PROG_DATA1
r/w

位 31-0 PROG_DATA1：program 编程数据 1。

注意：*program 时先写入 EFC_PROG_DATA0，再写入 EFC_PROG_DATA1*

10.5.6 EFC_TIMING_CFG

偏移量：0x14

复位值：0x00031D1D

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

位 31-20 Reserved：保留，不能修改。

位 19-16 READ_NUM：flash 读操作读等待拍数控制，读等待拍数等于（READ_NUM+1）个 SYS_CLK 时钟周期。

- SYS_CLK 频率为 48 MHz 时，READ_NUM 要大于等于 2
- SYS_CLK 频率为 32 MHz 时，READ_NUM 要大于等于 1
- SYS_CLK 频率为 24 MHz 时，READ_NUM 要大于等于 1
- SYS_CLK 频率为 4 MHz 时，READ_NUM 要大于等于 0
- SYS_CLK 频率为 32 KHz 时，READ_NUM 要大于等于 0

注意：*当修改 RCC_CR0 中的 SYS_CLK 时钟源选择时，需要注意操作顺序：若切为快时钟，则先配置 READ_NUM 变大，再配置时钟源选择；反之若切为慢时钟时，先配置时钟源选择，再配置 READ_NUM 变小。*

位 15-0 Reserved：保留，不能修改。

10.5.7 EFC_PROTECT_SEQ

偏移量：0x18

复位值：0x00000000

31-0
PROTECT_SEQ
w

位 31-0 PROTECT_SEQ：EFC_CR 配置的保护序列。

软件在操作 EFC_CR 前需要按照以下顺序配置保护序列，才能够正确操作 EFC_CR 寄存器，中间插入了错误序列，则操作无效，需要按以下步骤重新配置：

1. 先写 0x8C9DAEBF
2. 再写 0x13141516
3. 然后才能操作 EFC_CR

10.5.8 SERIAL_NUM_LOW

偏移量：0x2C

31-0
SERIAL_NUM_LOW
r

位 31-0 SERIAL_NUM_LOW：芯片序列号的低 32 位。

10.5.9 SERIAL_NUM_HIGH

偏移量：0x30

31-0
SERIAL_NUM_HIGH
r

位 31-0 SERIAL_NUM_HIGH：芯片序列号的高 32 位。

10.5.10 OPTION_CSR_BYTES

偏移量：0x3C

复位值：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

位 31-7 Reserved：保留，不能修改。

位 6-5 DEBUG_LEVEL：Debug level 配置。

- 0：Level0
- 1：Level1
- 2：Level2

位 4 SECURE_AREA_EN：安全区域状态指示寄存器。

- 0：不使能
- 1：使能

位 3 SYS_SRAM_RST：System 复位后 Startup 过程中是否清除系统 SRAM。

- 1：清除系统 SRAM
- 0：不清除系统 SRAM

位 2 FLASH_BOOT1：用于 Boot 方式判断。

- 1：BootLoader 启动
- 0：系统 SRAM 启动

位 1 USE_FLASH_BOOT0：控制使用 option bit 或 IO 进行 Boot 方式判断。

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

位 0 FLASH_BOOT0：USE_FLASH_BOOT0=1 时生效，用于 Boot 方式判断。

- 0：由 FLASH_BOOT1 控制启动方式
- 1：Flash Main 启动

10.5.11 OPTION_EXE_ONLY_BYTES

偏移量：0x40

复位值：0x00FC0FC0

31-25	24	23-18	17-12	11-6	5-0
RESERVED	EXEONLY_KEEP	EXEONLY2_END	EXEONLY2_START	EXEONLY1_END	EXEONLY1_START
r	r	r	r	r	r

位 31-25 Reserved：保留，不能修改。

位 24 EXEONLY_KEEP：ExeOnly 区域在 Debug_Level 由 1 变为 0 时，是否保留。

- 0：不保留 ExeOnly 区域
- 1：保留 ExeOnly 区域

支持修改，但只能写 0。

位 23-18 EXEONLY2_END：ExeOnly 区域 2 的结束 Page 地址。

当 *EXEONLY2_START* > *EXEONLY2_END*，ExeOnly 区域 2 不使能。

位 17-12 EXEONLY2_START：ExeOnly 区域 2 的开始 Page 地址。

当 *EXEONLY2_START* > *EXEONLY2_END*，ExeOnly 区域 2 不使能。

ExeOnly 区域 2 支持修改，但只能使能或扩大区域，不能减少或关闭。

位 11-6 EXEONLY1_END：ExeOnly 区域 1 的结束 Page 地址。

当 *EXEONLY1_START* > *EXEONLY1_END*，ExeOnly 区域 1 不使能。

位 5-0 EXEONLY1_START：ExeOnly 区域 1 的开始 Page 地址。

当 *EXEONLY1_START* > *EXEONLY1_END*，ExeOnly 区域 1 不使能。

ExeOnly 区域 1 支持修改，但只能使能或扩大区域，不能减少或关闭。

10.5.12 OPTION_WR_PROTECT_BYTES

偏移量：0x44

复位值：0x0003F03F

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

位 31-12 Reserved：保留，不能修改。

位 11-6 WRPROTECT_END：WrProtect 区域的结束 Page 地址。

当 *WRPROTECT_START* > *WRPROTECT_END*，WrProtect 区域不使能。

位 5-0 WRPROTECT_START：WrProtect 区域的开始 Page 地址。

当 *WRPROTECT_START* > *WRPROTECT_END*，WrProtect 区域不使能。

10.5.13 OPTION_SECURE_BYTES0

偏移量：0x48

复位值：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

位 31-24 Reserved：保留，不能修改。

位 23-18 SYSRAM_SECURE_END：SysRam Secure 区域的结束地址。

当 *SYSRAM_SECURE_START* > *SYSRAM_SECURE_END* 时，该区域安全不使能；且仅在 SECURE_AREA_EN=1 才有效。

位 17-12 SYSRAM_SECURE_START：SysRam Secure 区域的起始地址。

当 *SYSRAM_SECURE_START* > *SYSRAM_SECURE_END* 时，该区域安全不使能；且仅在 SECURE_AREA_EN=1 才有效。

位 11-6 FLASH_SECURE_END：Flash Secure 区域的结束地址。

当 *FLASH_SECURE_START* > *FLASH_SECURE_END* 时，该区域安全不使能。

位 5-0 FLASH_SECURE_START：FLASH Secure 区域的起始地址。

当 *FLASH_SECURE_START* > *FLASH_SECURE_END* 时，该区域安全不使能。

FLASH 区域安全使能为整个安全区域的总使能：

当 FLASH 区域安全使能时，SECURE_AREA_EN 为 1，使能整个安全区域；

当 FLASH 区域安全去使能时，SECURE_AREA_EN 由 1 变为 0，将触发擦除操作。

10.5.14 OPTION_SECURE_BYTES1

偏移量：0x4C

复位值：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

位 31-24 Reserved：保留，不能修改。

位 23 SYSRAM_HIDE_ENABLE：SysSramHide 区域使能控制。

- 0：SysSramHide 区域使能
- 1：SysSramHide 区域不使能

仅当 SECURE_AREA_EN=1 时才有效。

位 22-17 SYSRAM_HIDE_START：SysSramHide 区域的开始 Page 地址。

必须在 SysSramSecure 区域内才使能，区域范围由 SYSSRAM_HIDE_START 开始一直到 SYSSRAM_SECURE_END;

仅当 SECURE_AREA_EN=1 时才有效。

位 16 FLASH_HIDE_ENABLE：FlashHide 区域使能控制。

- 0：FlashHide 区域使能
- 1：FlashHide 区域不使能

仅当 SECURE_AREA_EN=1 时才有效。

位 15-10 FLASH_HIDE_START：FlashHide 区域的开始 Page 地址。

必须在 FlashSecure 区域内才使能，区域范围由 FLASH_HIDE_START 开始一直到 FLASH_SECURE_END;

仅当 SECURE_AREA_EN=1 时才有效。

位 9-5 RETRAM_SECURE_END：RetRam Secure 区域的结束地址。

当 *RETRAM_SECURE_START > RETRAM_SECURE_END* 时，该区域安全不使能;

仅当 SECURE_AREA_EN=1 时才有效。

位 4-0 RETRAM_SECURE_START：RetRam Secure 区域的开始地址。

当 *RETRAM_SECURE_START > RETRAM_SECURE_END* 时，该区域安全不使能;

仅当 SECURE_AREA_EN=1 时才有效。

11. GPIO

11.1 简介

共 A、B、C、D 四组 GPIO，每组的 SFR 寄存器分配一致，利用不同基地址进行区分；PortD Pin8~PortD Pin15 位于 AlwayOn 域，其他 IO 位于 Main 域。

支持输入输出功能，支持上下拉功能，支持推挽输出和开漏输出，可以配置输出驱动电流为 4mA/8mA，所有 IO 都支持中断功能，上升沿、下降沿或双沿可配，Sleep/Stop0~2 模式所有 IO 都可以唤醒，Stop3 模式部分 IO 可唤醒，支持功能复用配置。

11.2 输出配置

支持输出数据配置，通过配置输出使能寄存器 GPIOx_OER 和输出数据寄存器 GPIOx_ODR 实现。

支持输出数据置位和清零两种位操作，通过向输出数据清零寄存器 GPIOx_BRR 写 1 来清除 GPIOx_ODR 的相应位，或者向输出数据置位清零寄存器 GPIOx_BSRR 低 16 位写 1 置位 GPIOx_ODR 的相应位，向 GPIOx_BSRR 高 16 位写 1 清零 GPIOx_ODR 的相应位，只操作指定位，避免读后写。

支持推挽输出，通过配置输出类型寄存器 GPIOx_OTYPER 实现。支持开漏输出，其中 PortD Pin8~PortD Pin15 通过配置 GPIOx_IER/GPIOx_OER/GPIOx_PSR 实现，其他的 IO 通过配置 GPIOx_OER/GPIOx_IER/GPIOx_OTYPER 实现。不支持真的 Open Drain 结构，通过数据控制 OE 的方式实现。

支持配置为模拟输出。

11.3 输入配置

支持输入数据配置，通过配置输入使能寄存器 GPIOx_IER 使能输入功能，读取输入数据寄存器 GPIOx_IDR 获取输入状态。

支持 Floating 输入方式，通过配置上下拉使能寄存器 GPIOx_PER 关闭上下拉功能实现。

支持上拉输入和下拉输入，配置 GPIOx_PER 使能上下拉功能，配置上下拉选择寄存器 GPIOx_PSR 选择上拉或者下拉功能。

支持配置为模拟输入。

11.4 输出驱动能力

可配为低驱动能力 4mA 或高驱动能力 8mA，通过配置输出驱动能力寄存器 GPIOx_DSR 实现。

11.5 中断

所有 IO 都支持上升沿、下降沿和双沿中断，中断使能可配，通过配置中断使能寄存器 GPIOx_INT_CR 实现。

11.6 Sleep/Stop0~2 唤醒请求

支持高电平与低电平，输出唤醒信号为高电平。GPIO00~GPIO63 均可用于唤醒，每 4 个 IO 为一组，一组可以产生一个唤醒信号，同组内的每个 IO 均支持配置高电平或者低电平唤醒。通过配置 Sleep/Stop0~2 唤醒使能寄存器 GPIOx_WU_EN 使能唤醒功能，通过配置 Sleep/Stop0~2 唤醒电平控制寄存器选择高电平或者低电平唤醒。

11.7 Stop3 唤醒请求

Main 域的 GPIO00~GPIO55，每 4 个 IO MUX 出一个唤醒信号，共 14 个唤醒信号。支持高电平唤醒或低电平唤醒和唤醒使能控制。通过配置 Stop3 唤醒使能寄存器 GPIOx_STOP3_WU_CR 实现。

11.8 复用功能控制

可在 GPIO 以及不同外设之间选择，GPIO 下的输入输出使能由 GPIO 寄存器控制，外设下的输入输出控制由外设实现，上下拉由 GPIO 寄存器实现。

每个 IO 均有 4bit 的复用功能控制（PortD Pin8~PortD Pin15 为每个 3bit），除 PortA Pin6~PortA Pin7 默认选择 SWD 外，其余 IO 均默认选择 GPIO 功能。

通过低 8 Pin 功能 MUX 选择寄存器 GPIOx_AFRL 配置 Portx Pin[7:0] 的功能，通过高 8 Pin 功能 MUX 选择寄存器 GPIOx_AFRH 配置 Portx Pin[15:8] 的功能。

11.9 时钟复位

APB 总线时钟，共 4 组，每组均有独立的总线时钟。APB 总线复位，共 4 组，每组均有独立的总线复位。

11.10 电源域

Main 域：

除了 PortD Pin8~PortD Pin15 外相应的 PAD 均在 Main 域。

AlwayOn 域：

PortD Pin8~PortD Pin15 对应的 PAD 位于 AlwaysOn 域，如果配置为 Function 模式，那么直接由外设去控制，如果没有配置为 Function 模式，需要由 AlwaysOn 的 GPIO 寄存器控制。

11.11 低功耗工作与唤醒

1. Sleep 下所有 GPIO 均可工作，可输出唤醒信号。
2. Stop0~2 下所有 GPIO 均可工作，可输出唤醒信号。
3. Stop3 下 GPIO00~GPIO55 工作状态可保持，并且均可配置为唤醒信号。
4. Stop3 下 AlwaysOn 域的 PortD Pin8~PortD Pin15 可保持，也可以通过 RTC 唤醒。
5. Standby 下 PortD Pin8~PortD Pin15 可工作，其他 IO 不可工作。

11.12 SWD IO

默认控制：控制 GPIO MUX 的寄存器默认应选择 SWD，并且上下拉默认值应为 SWC-下拉（PortA Pin7），SWD-上拉（PortA Pin6）。

封口控制：上电时由寄存器默认态控制 IO 状态，直到 DebugLevel 判断完毕，若发现需要封口，则执行永久封口；否则，继续由寄存器控制。

软件配置：软件运行过程中，可通过复用寄存器控制关闭 SWD，注意是单向封口的，即只能控制关闭，不能关闭后再使能。

11.13 BOOT0 的控制

默认控制：由于除了 SWC 和 SWD 两个 IO 外，其他 IO 默认都是模拟 IO，因此上电时这三个 IO 需要特殊控制。

BOOT0：BOOT0 在 io_lock 前为输入下拉，待 EFC 锁定后，切换为 GPIO 控制。

11.14 GPIO 相关寄存器描述

GPIO Port A 基地址：0x4001F000

GPIO Port B 基地址：0x4001F400

GPIO Port C 基地址：0x4001F800

GPIO Port D 基地址：0x4001FC00

表 11-1 GPIO 寄存器列表

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

11.14.1 GPIOx_OER(x=A、B、C、D)

偏移量：0x00

复位值：0x0000FFFF

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

位 31-16 RESERVED：保留，不可更改。

位 15-0 OEN：Portx Pin[15:0] 输出使能。

- 1：输出使能无效
- 0：输出使能有效

11.14.2 GPIOx_OTYPER(x=A、B、C、D)

偏移量：0x04

复位值：0x00000000

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

位 31-16 RESERVED：保留，不可更改。

位 15-0 OTYPE：Portx Pin[15:0] 输出类型控制。

- 1：开漏输出
- 0：推挽输出

说明：*AON 的 PAD（PortD_Pin[15:8]）不支持该寄存器的配置，而是通过 GPIOx_OER 直接控制 PAD 的 OEN 端，开漏模式可以通过配置 GPIOx_ODR/GPIOx_IE/GPIOx_OE 的组合去实现。*

11.14.3 GPIOx_IER(x=A、B、C、D)

偏移量：0x08

复位值：0x00000000

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

位 31-16 RESERVED：保留，不可更改。

位 15-0 IE：Portx Pin[15:0] 输入使能。

- 1：输入使能有效
- 0：输入使能无效

11.14.4 GPIOx_PER(x=A、B、C、D)

偏移量：0x0C

复位值：0x00000000

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

位 31-16 RESERVED：保留，不可更改。

位 15-0 PE：Portx Pin[15:0] 上下拉使能。

- 1：上下拉使能有效
- 0：上下拉使能无效

上下拉直接由寄存器控制。默认上下拉不使能，IO 处于模拟模式，PortA_Pin[7:6] 例外，为 SWD 接口。

11.14.5 GPIOx_PSR(x=A、B、C、D)

偏移量：0x10

复位值：0x00000000

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

位 31-16 RESERVED：保留，不可更改。

位 15-0 PS：Portx Pin[15:0] 上下拉选择。

- 1：选择上拉
- 0：选择下拉

11.14.6 GPIOx_IDR(x=A、B、C、D)

偏移量：0x14

复位值：0x00000000

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

位 31-16 RESERVED：保留，不可更改。

位 15-0 ID：Portx Pin[15:0]输入数据。

- 1：输入高
- 0：输入低

11.14.7 GPIOx_ODR(x=A、B、C、D)

偏移量：0x18

复位值：0x00000000

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

位 31-16 RESERVED：保留，不可更改。

位 15-0 OD：Portx Pin[15:0]输出数据。

- 1：输出高
- 0：输出低

11.14.8 GPIOx_BRR(x=A、B、C、D)

偏移量：0x1C

复位值：0x00000000

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

位 31-16 RESERVED：保留，不可更改。

位 15-0 BR：Portx Pin[15:0] 输出数据清零。

- 写 1：清除 GPIOx_ODR 相应位
- 写 0：无效

11.14.9 GPIOx_BSRR(x=A、B、C、D)

偏移量：0x20

复位值：0x00000000

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

位 31-16 BR：Portx Pin[15:0] 输出数据清零。

- 写 1：清除 GPIOx_ODR 相应位
- 写 0：无效

说明：*若 BSR 和 BR 同时有效，则 BSR 优先级高。*

位 15-0 BSR：Portx Pin[15:0] 输出数据置位。

- 写 1：置位 GPIOx_ODR 相应位
- 写 0：无效

说明：*若 BSR 和 BR 同时有效，则 BSR 优先级高。*

11.14.10 GPIOx_DSR(x=A、B、C、D)

偏移量：0x24

复位值：0x00000000

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

位 31-16 RESERVED：保留，不可更改。

位 15-0 DS：Portx Pin[15:0] 输出驱动能力配置。

- 写 1：高驱动能力，8mA
- 写 0：低驱动能力，4mA

11.14.11 GPIOx_INT_CR(x=A、B、C、D)

偏移量：0x28

复位值：0x00000000

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

位 2*n + 1 NEG_INT_EN：Portx Pin[15:0] 下降沿中断使能。

- 1：使能下降沿中断
- 0：不使能下降沿中断

位 2*n POS_INT_EN：Portx Pin[15:0] 上升沿中断使能。

- 1：使能上升沿中断
- 0：不使能上升沿中断

11.14.12 GPIOx_FR(x=A、B、C、D)

偏移量：0x2C

复位值：0x00000000

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

位 2*n + 1 NEG_INT_EN：Portx Pin[15:0] 下降沿中断标志。

- 1：发生下降沿中断
- 0：未发生下降沿中断

位 2*n POS_INT_EN：Portx Pin[15:0] 上升沿中断标志。

- 1：发生上升沿中断
- 0：未发生上升沿中断

11.14.13 GPIOx_WU_EN(x=A、B、C、D)

偏移量：0x30

复位值：0x00000000

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

位 31-16 RESERVED：保留，不可更改。

位 15-0 WU_EN：Portx Pin[15:0] Sleep/Stop0~2 唤醒使能。

- 1：使能 Sleep/Stop0~2 唤醒
- 0：不使能 Sleep/Stop0~2 唤醒

11.14.14 GPIOx_WU_LVL(x=A、B、C、D)

偏移量：0x34

复位值：0x00000000

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

位 31-16 RESERVED：保留，不可更改。

位 15-0 WU_LVL：Portx Pin[15:0] Sleep/Stop0~2 电平控制。

- 1：高电平唤醒
- 0：低电平唤醒

11.14.15 GPIOx_AFRL(x=A、B、C、D)

偏移量：0x38

复位值：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

位 31-28 AF7：Portx Pin7 功能 MUX 选择。

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

位 27-24 AF6：Portx Pin6 功能 MUX 选择。

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

位 23-20 AF5：Portx Pin5 功能 MUX 选择。

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

位 19-16 AF4：Portx Pin4 功能 MUX 选择。

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

位 15-12 AF3：Portx Pin3 功能 MUX 选择。

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

位 11-8 AF2：Portx Pin2 功能 MUX 选择。

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

位 7-4 AF1：Portx Pin1 功能 MUX 选择。

- 0000：Function0
- 0001：Function1

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

位 3-0 AF0：Portx Pin0 功能 MUX 选择。

- 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)

偏移量：0x3C

复位值：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

位 31-28 AF15：Portx Pin15 功能 MUX 选择。

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

位 27-24 AF14：Portx Pin14 功能 MUX 选择。

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

- 0101：Function5
- 0110：Function6
- 0111：Function7
- others：Reserved

位 23-20 AF13：Portx Pin13 功能 MUX 选择。

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

位 19-16 AF12：Portx Pin12 功能 MUX 选择。

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

位 15-12 AF11：Portx Pin11 功能 MUX 选择。

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

位 11-8 AF10：Portx Pin10 功能 MUX 选择。

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

- others：Reserved

位 7-4 AF9：Portx Pin9 功能 MUX 选择。

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

位 3-0 AF8：Portx Pin8 功能 MUX 选择。

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

11.14.17 GPIOD_AFRH

偏移量：0x3C

复位值：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

位 31-24 RESERVED：保留，不可更改。

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

位 23-21 AF15：PortD Pin15 功能 MUX 选择。

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

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

位 20-18 AF14：PortD Pin14 功能 MUX 选择。

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

位 17-15 AF13：PortD Pin13 功能 MUX 选择。

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

位 14-12 AF12：PortD Pin12 功能 MUX 选择。

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

位 11-9 AF11：PortD Pin11 功能 MUX 选择。

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

位 8-6 AF10：PortD Pin10 功能 MUX 选择。

- 001：Function1
- 010：Function2
- 011：Function3
- 100：Function4
- 101：Function5

- 110：Function6
- 111：Function7

位 5-3 AF9：PortD Pin9 功能 MUX 选择。

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

位 2-0 AF8：PortD Pin8 功能 MUX 选择。

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

11.14.18 GPIOA_STOP3_WU_CR

偏移量：0x40

复位值：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

位 31-16 RESERVED：保留，不可更改。

位 15 STOP3_WU_EN_G3：PortA Pin Group3 Stop3 唤醒使能控制。

- 0：PortA Pin[15/14/7/6] 唤醒不使能
- 1：PortA Pin[15/14/7/6] 唤醒使能

位 14 STOP3_WU_LVL_G3：PortA Pin Group3 Stop3 唤醒电平选择。

- 0：PortA Pin[15/14/7/6] 低电平唤醒
- 1：PortA Pin[15/14/7/6] 高电平唤醒

位 13-12 STOP3_WU_SEL_G3：PortA Pin Group3 Stop3 唤醒源选择。

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

位 11 STOP3_WU_EN_G2：PortA Pin Group2 Stop3 唤醒使能控制。

- 0：PortA Pin[11:8] 唤醒不使能
- 1：PortA Pin[11:8] 唤醒使能

位 10 STOP3_WU_LVL_G2：PortA Pin Group2 Stop3 唤醒电平选择。

- 0：PortA Pin[11:8] 低电平唤醒
- 1：PortA Pin[11:8] 高电平唤醒

位 9-8 STOP3_WU_SEL_G2：PortA Pin Group2 Stop3 唤醒源选择。

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

位 7 STOP3_WU_EN_G1：PortA Pin Group1 Stop3 唤醒使能控制。

- 0：PortA Pin[13/12/5/4] 唤醒不使能
- 1：PortA Pin[13/12/5/4] 唤醒使能

位 6 STOP3_WU_LVL_G1：PortA Pin Group1 Stop3 唤醒电平选择。

- 0：PortA Pin[13/12/5/4] 低电平唤醒
- 1：PortA Pin[13/12/5/4] 高电平唤醒

位 5-4 STOP3_WU_SEL_G1：PortA Pin Group1 Stop3 唤醒源选择。

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

位 3 STOP3_WU_EN_G0：PortA Pin Group0 Stop3 唤醒使能控制。

- 0：PortA Pin[3:0] 唤醒不使能
- 1：PortA Pin[3:0] 唤醒使能

位 2 STOP3_WU_LVL_G0：PortA Pin Group0 Stop3 唤醒电平选择。

- 0：PortA Pin[3:0] 低电平唤醒
- 1：PortA Pin[3:0] 高电平唤醒

位 1-0 STOP3_WU_SEL_G0：PortA Pin Group0 Stop3 唤醒源选择。

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

11.14.19 GPIOx_STOP3_WU_CR(x=B、C)

偏移量：0x40

复位值：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

位 31-16 RESERVED：保留，不可更改。

位 15 STOP3_WU_EN_G3：Portx Pin Group3 Stop3 唤醒使能控制。

- 0：Portx Pin[15:12] 唤醒不使能
- 1：Portx Pin[15:12] 唤醒使能

位 14 STOP3_WU_LVL_G3：Portx Pin Group3 Stop3 唤醒电平选择。

- 0：Portx Pin[15:12] 低电平唤醒
- 1：Portx Pin[15:12] 高电平唤醒

位 13-12 STOP3_WU_SEL_G3：Portx Pin Group3 Stop3 唤醒源选择。

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

位 11 STOP3_WU_EN_G2：Portx Pin Group2 Stop3 唤醒使能控制。

- 0：Portx Pin[11:8] 唤醒不使能
- 1：Portx Pin[11:8] 唤醒使能

位 10 STOP3_WU_LVL_G2：Portx Pin Group2 Stop3 唤醒电平选择。

- 0：Portx Pin[11:8] 低电平唤醒
- 1：Portx Pin[11:8] 高电平唤醒

位 9-8 STOP3_WU_SEL_G2：Portx Pin Group2 Stop3 唤醒源选择。

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

位 7 STOP3_WU_EN_G1：Portx Pin Group1 Stop3 唤醒使能控制。

- 0：Portx Pin[7:4] 唤醒不使能
- 1：Portx Pin[7:4] 唤醒使能

位 6 STOP3_WU_LVL_G1：Portx Pin Group1 Stop3 唤醒电平选择。

- 0：Portx Pin[7:4] 低电平唤醒
- 1：Portx Pin[7:4] 高电平唤醒

位 5-4 STOP3_WU_SEL_G1：Portx Pin Group1 Stop3 唤醒源选择。

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

位 3 STOP3_WU_EN_G0：Portx Pin Group0 Stop3 唤醒使能控制。

- 0：Portx Pin[3:0] 唤醒不使能
- 1：Portx Pin[3:0] 唤醒使能

位 2 STOP3_WU_LVL_G0：Portx Pin Group0 Stop3 唤醒电平选择。

- 0：Portx Pin[3:0] 低电平唤醒
- 1：Portx Pin[3:0] 高电平唤醒

位 1-0 STOP3_WU_SEL_G0：Portx Pin Group0 Stop3 唤醒源选择。

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

11.14.20 GPIOD_STOP3_WU_CR

偏移量：0x40

复位值：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

位 31-8 RESERVED：保留，不可更改。

位 7 STOP3_WU_EN_G1：PortD Pin Group1 Stop3 唤醒使能控制。

- 0：PortD Pin[7:4] 唤醒不使能
- 1：PortD Pin[7:4] 唤醒使能

位 6 STOP3_WU_LVL_G1：PortD Pin Group1 Stop3 唤醒电平选择。

- 0：PortD Pin[7:4] 低电平唤醒
- 1：PortD Pin[7:4] 高电平唤醒

位 5-4 STOP3_WU_SEL_G1：PortD Pin Group1 Stop3 唤醒源选择。

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

位 3 STOP3_WU_EN_G0：PortD Pin Group0 Stop3 唤醒使能控制。

- 0：PortD Pin[3:0] 唤醒不使能
- 1：PortD Pin[3:0] 唤醒使能

位 2 STOP3_WU_LVL_G0：PortD Pin Group0 Stop3 唤醒电平选择。

- 0：PortD Pin[3:0] 低电平唤醒
- 1：PortD Pin[3:0] 高电平唤醒

位 1-0 STOP3_WU_SEL_G0：PortD Pin Group0 Stop3 唤醒源选择。

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

12. LoRa Controller (LoRaC)

12.1 简介

LoRa Controller 主要用来控制内部的 RF TRX 实现 LoRa 的发送和接收。

12.2 主要特性

- 支持 SPI 接口连接 RF TRX
- 支持中断信号产生

12.3 功能描述

12.3.1 内部 SPI 接口

LoRa Controller 拥有一路内部 SPI 接口，可以通过寄存器直接控制 RF TRX。与 RF TRX 之间的通信如下:

(1) 初始化 LoRa Controller 内部 SSP。

(2) 检查 LORAC_SR 寄存器的 BUSY_DIG_SR 是否为 0，若为 0 则说明当前 RF TRX 空闲，可以进行通信。

(3) 将寄存器 LORAC_NSS_CR 写 0。

(4) 将数据写入 LoRa Controller 内部 SSP 的寄存器 SSP_DR。

(5) 等待发送完成。

(6) 通过寄存器 SSP_DR 读回数据。

(7) 根据需求重复执行步骤(4) - (6)。

(8) 将寄存器 LORAC_NSS_CR 写 1。

12.3.2 上电初始化

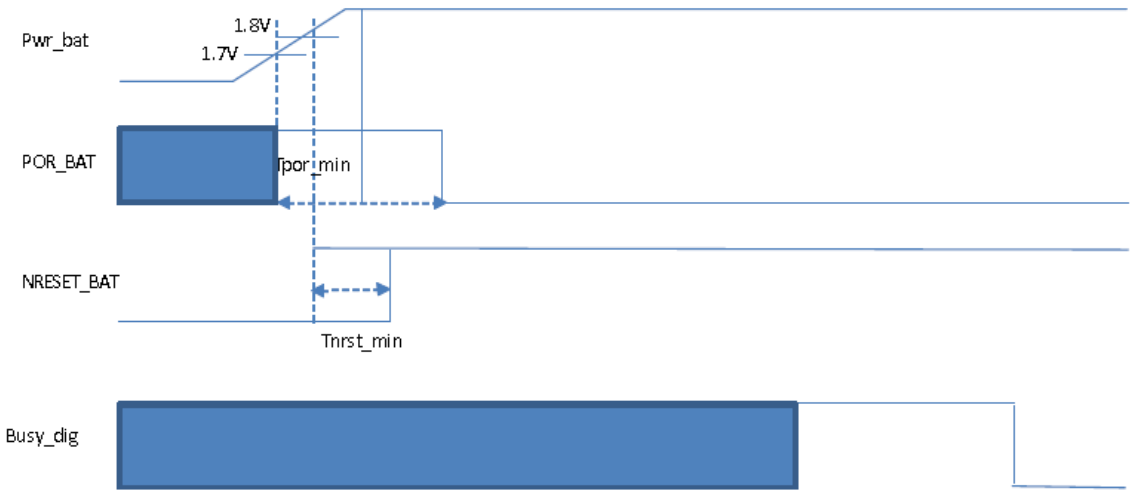

图 12-1 上电初始化时序

如上图所示，上电初始化的流程为：

(1) 将寄存器 LORAC_CR1 的 NRESET_BAT 置 1。

(2) 将寄存器 LORAC_CR1 的 POR_BAT 置 0。

(3) 等待 BUSY_DIG_SR 为 0。

其中，Tpor_min 为 100us，Tnrst_min 为 50us。

12.3.3 中断信号

LoRa Controller 的中断信号主要是透传 RF TRX 的中断请求，需注意的是当 LoRa Controller 的中断请求触发时，需要向 RF TRX 发送 ClearIrqStatus 命令清除中断，不然中断请求会一直触发。

12.4 LoRaC 相关寄存器描述

LORAC 寄存器基地址：0x40009000

表 12-1 LORAC 寄存器列表

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

其中 SSP 相关寄存器可以参考 *SSP* 章节的说明。

12.4.1 LORAC_CR0

偏移量：0x100

复位值：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

位 31-11 RESERVED：保留，不能修改。

位 10 NSS_SEL：RF TRX 的 nss 来源选择。

- 0：来自寄存器 LORAC_NSS_CR
- 1：来自 LORAC 的内部 SSP

位 9 SCK_MOSI_SEL：RF TRX 的 sck/mosi/miso 来源选择。

- 0：来自 LORAC_SCK_CR，LORAC_MOSI_CR，LORA_MISO_SR
- 1：来自 LORAC 的内部 SSP

位 8 RESERVED：保留，不能修改。

位 7-5 IRQ_DIG_INT_EN：IRQ_DIG_INT 高电平中断使能，[0] 对应 irq_dig[0]，[1] 对应 irq_dig[1]，[2] 对应 irq_dig[2]。

- 0：不使能
- 1：使能

位 4-0 RESERVED：保留，不能修改。

12.4.2 LORAC_CR1

偏移量：0x104

复位值：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

位 31-8 RESERVED：保留，不能修改。

位 7 POR_BAT：RF TRX 的 POR_BAT 控制。

- 0：不复位
- 1：复位

位 6 RESERVED：保留，不能修改。

位 5 NRESET_BAT：RF TRX 的 NRESET_BAT 控制。

- 0：复位
- 1：不复位

位 4-3 RESERVED：保留，不能修改。

位 2 CLK_32M_EN_BAT：RF TRX 的 CLK_32M_EN_BAT 控制。

- 0：不使能
- 1：使能

位 1 TCXO_EN_BAT：RF TRX 的 TCXO_EN_BAT 控制。

- 0：不使能
- 1：使能

位 0 PWRTCXO_EN_BAT：RF TRX 的 PWRTCXO_EN_BAT 控制。

- 0：不使能
- 1：使能

12.4.3 LORAC_SR

偏移量：0x108

复位值：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

位 31-9 RESERVED：保留，不能修改。

位 8 BUSY_DIG_SR：BUSY_DIG 状态位，用于指示 LoRa IP 是否处于命令处理状态。硬件置 1 与清 0。

- 0：未进行命令处理
- 1：正在进行命令处理

位 7-5 IRQ_DIG_SR：IRQ_DIG 状态位，用于指示 RF TRX 的中断请求。硬件置 1 与清 0。需注意的是，需要向 RF TRX 发送 ClearIrqStatus 命令清除中断，不然中断请求会一直触发。

- 0：无中断
- 1：有中断

位 4-2 RESERVED：保留，不能修改。

位 1 CLK_32M_RDY_BAT_SR：CLK_32M_RDY_BAT 状态位，用于指示 RF TRX 的 XO32M 时钟是否建立。硬件置 1 与清 0。

- 0：未建立
- 1：已建立

位 0 RESERVED：保留，不能修改。

12.4.4 LORAC_NSS_CR

偏移量：0x10C

复位值：0x00000001

31-1	0
RESERVED	REG_NSS
r	r/w

位 31-1 RESERVED：保留，不能修改。

位 0 REG_NSS：nss 的寄存器控制位。

- 0：nss 拉低
- 1：nss 拉高

12.4.5 LORAC_SCK_CR

偏移量：0x110

复位值：0x00000000

31-1	0
RESERVED	REG_SCK
r	r/w

位 31-1 RESERVED：保留，不能修改。

位 0 REG_SCK：sck 的寄存器控制位。

- 0：sck 拉低
- 1：sck 拉高

12.4.6 LORAC_MOSI_CR

偏移量：0x114

复位值：0x00000000

31-1	0
RESERVED	REG_MOSI
r	r/w

位 31-1 RESERVED：保留，不能修改。

位 0 REG_MOSI：mosi 的寄存器控制位。

- 0：mosi 拉低
- 1：mosi 拉高

12.4.7 LORAC_MISO_SR

偏移量：0x118

复位值：0x00000000

31-1	0
RESERVED	REG_MISO
r	r

位 31-1 RESERVED：保留，不能修改。

位 0 REG_MISO：miso 状态位，指示 RF TRX 输出的 miso 状态。硬件置 1 与清 0。

- 0：低电平
- 1：高电平

13. UART

13.1 简介

支持 UART 和 IrDA 模式。

发送与接收的 FIFO 独立，深度 16，FIFO 水位可配置为 1/8、1/4、1/2、3/4 和 7/8，禁用 FIFO 相当于 1 字符深度。16 位波特率除数整数部和 6 位波特率除数小数部。标准异步通信位，支持 5、6、7 和 8 位数据，支持奇偶校验，支持 1 个或者 2 个停止位。支持 DMA，支持假开始位检测，支持 Line Break 产生与检测，支持硬件流控。

IrDA 模式码率最高 460800，Low-Power IrDA 模式最高码率 115200，半双工。支持 3/16 和 Low-Power (1.41~2.23uS)位长度。Low-Power IrDA 模式，通过 UARTCLK 分频近似产生位长度。

可通过 ID 寄存器唯一地识别每个 UART 端口。

13.2 时钟复位

每个 UART 都有独立的 APB 总线时钟和独立的 APB 总线复位。

13.3 参考时钟

UARTCLK 的频率必须要满足波特率产生的要求：

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

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

例如，要产生介于 110 到 460800 之间的波特率，UARTCLK 的频率必须要介于 7.3728MHz 与 115.34MHz 之间。

同时，UARTCLK 不能大于 5/3 倍 PCLK：$F_{UARTCLK} <= 5/3 * F_{PCLK}$

例如，UART 模式下，UARTCLK 为 14.7456MHz，要产生 921600 的波特率，那么 PCLK 必须大于等于 8.85276MHz。这保证了 UART 有足够的时间将接收的数据写入接收 FIFO。

13.4 波特率发生器

波特率发生器包含产生内部 16 倍时钟的自由运行计数器，Baud16 和 IrLPBaud16。Baud16 为 UART 发送和接收控制提供时序信息，是一个宽度为 1 个 UARTCLK 周期的脉冲流，频率为 16 倍波特率。当运行在 Low-Power IrDA 模式时，IrLPBaud16 为 IrDA 编码发送比特流的脉冲宽度的产生提供时序信息。

13.5 FIFO

发送与接收的 FIFO 独立，通过线控寄存器 UARTx_LCR_H{FEN} 选择开启或者关闭。发送 16 x 8，接收 16 x 12，接收 FIFO 每个字符有 4 个位的状态码，FIFO 水位可通过 FIFO 中断水位选择寄存器 UARTx_IFLS 配置为 1/8、1/4、1/2、3/4 和 7/8，当 FIFO 禁用时相当于深度 1。FIFO 状态通过查询标志寄存器 UARTx_FR 获取。

接收 FIFO 的[10:8]位为错误位，指示相应的错误。第 11 位为 Overrun 位，指示 Overrun 错误。

表 13-1 接收 FIFO 位功能描述

FIFO Bit	功能
11	Overrun 错误
10	Break 错误
9	奇偶校验错误
8	帧错误
7:0	被接收数据

13.6 UART 方式

13.6.1 波特率除数

波特率除数由 16 位的整数部和 6 位的小数部组成，整数部存入寄存器 UARTx_IBRD，小数部存入寄存器 UARTx_FBRD，这允许使用任何 > 3.6864MHz 的时钟作为 UARTCLK 以支持产生所有标准波特率，满足以下公式：

$$BaudRateDivisor = UARTCLK / (16 \times BautRate) = BRD_I + BRD_F$$

其中 BRD_I 为整数部，BRD_F 为小数部，以小数点隔开，如下所示：

16 bit Integer Part	.	6 bit Fractional Part

6 位小数部的计算方法是，将波特率除数的小数部分乘以 64 （ $=2^n$, n 为 UARTx_FBRD 寄存器的有效位宽度 6 ），考虑到舍入误差，再加上 0.5，公式如下：

$$FractionalPart = BRD_F \times 2^n + 0.5$$

13.6.2 数据发送

发送与接收的数据都保存在 16 字节的 FIFO 里面，接收 FIFO 每个字符另有 4 个状态标志位。

发送时，数据通过数据寄存器 UARTx_DR 写入发送 FIFO。通过 UARTx_CR{UARTEN} 使能 UART，通过线控寄存器 UARTx_LCR_H 的配置数据位、停止位、奇偶校验等参数，数据开始发送，直到 FIFO 为空。一旦数据写入发送 FIFO，BUSY 信号变高，并且当数据在发送的时候

一直保持为高。只有当发送 FIFO 为空，并且位移寄存器的最后一个字符的 stop 位发送完，BUSY 信号才会变低。当 UART 没有使能的时候，BUSY 信号也为高。

13.6.3 数据接收

通过 UARTx_CR{UARTEN}使能 UART，通过线控寄存器 UARTx_LCR_H 的配置数据位、停止位、奇偶校验等参数。

当接收为空闲，UARTRXD 拉低，Baud16 使能接收计数器开始计数，UART 模式在第 8 个计数周期开始采样。IrDA 模式在第 4 个计数周期开始采样，以允许更短的逻辑 0 脉冲。

若 UARTRXD 在第 8 个计数周期仍保持为低，那么有效的 start 位被检测到，否则判定为假 start，并且被忽略。

若 start 位有效，接着每 16 个 Baud16 周期进行一次数据采样，长度由 UARTx_LCR_H{WLEN}决定。如果使能了奇偶校验，会进行奇偶校验位的比对。

最后，当 UARTRXD 变高，有效的 stop 位被确认到，否则发生帧错误。完整接收的字符与错误位一起被存入接收 FIFO。

13.7 IrDA SIR 方式

IrDA SIR ENDEC 提供了在 UART 数据流和半双工串行 SIR 接口之间转换的功能，将数据从 UART 编码输出和解码输入到 UART，有两种模式：

- **IrDA 模式**，逻辑 0 电平被转换为高电平脉冲，宽度为 nSIROUT 波特率比特周期的 $^{3}/_{16}$，逻辑 1 电平被转换为低电平。
- **Low-Power IrDA 模式**，发送的高电平脉冲宽度为内部 IrLPBaud16 周期的 3 倍（1.63us，假定名义频率为 1.842MHz）。

IrDA SIR 物理层为半双工的通信链接，发送与接收之间切换至少要保持 10ms 的延时。这个延时必须由软件完成，因为 UART 不支持自动延时。这是必须的，因为红外接收端可能会有偏差。

13.7.1 低功耗除数

IrLPBAUD16 由 UARTCLK 分频产生，分频系数通过 UARTx_ILPR{ ILPDVSR } 配置。

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

$F_{IrLPBAUD16}$ 名义上为 1.8432MHz，满足 **1.42MHz < $F_{IrLPBAUD16}$ < 2.12MHz** 的要求。

13.7.2 IrDA SIR 发送编码器

SIR 发送编码器将 UART 输出的 NRZ（Non Return-to-Zero）发送比特流进行调制。IrDA 物理层使用 RZI（Return-to-Zero Inverted）调制模式，把逻辑 0 转换为一个红外脉冲。调制的输出脉冲流被传送到外部的输出驱动和红外发光二极管。

在 IrDA 模式，发送的脉冲宽度为 Baud16 周期的 3 倍，即 $^{3}/_{16}$ 比特周期。

在 Low-Power IrDA 模式，被传输的脉冲宽度为 115200 波特率比特周期的 $^{3}/_{16}$。即为 1.842MHz 时钟 IrLPBaud16 周期的三倍。

如果使用了小数波特率除数，发送的 SIR 脉冲流会包含更多的抖动。因为使用了小数波特率除数使得不能产生一个规律间隔的 Baud16 脉冲，这些 Baud16 周期包含了不同的 UARTCLK 周期数。最坏的情况下，导致 SIR 脉冲流的抖动可以达到 3 个 UARTCLK 周期。只要 UARTCLK > 3.6864MHz，且 IrDA 的波特率小于 115200，抖动 < 9%，这满足 SIR IrDA 抖动小于 13%的要求。

13.7.3 IrDA SIR 接收解码器

SIR 接收解码器从红外接收器解调 Return-to-Zero 比特流，输出接收到的 NRZ 串行比特流到 UART 接收输入。正常情况下，在空闲状态接收的解码器的输入为高，发送的编码器的输出与解码器输入的极性相反。

当接收的解码器输入为低则 start 位被检测到。

为防止 UART 响应接收数据输入的毛刺，在 IrDA 模式，SIRIN 上小于 Buad16 的 $^{3}/_{16}$ 的脉冲会被忽略；在 Low-Power IrDA 模式，SIRIN 上小于 IrLPBuad16 的 $^{3}/_{16}$ 的脉冲会被忽略。

13.8 UART 字符帧结构

UART 字符帧结构如下图所示：

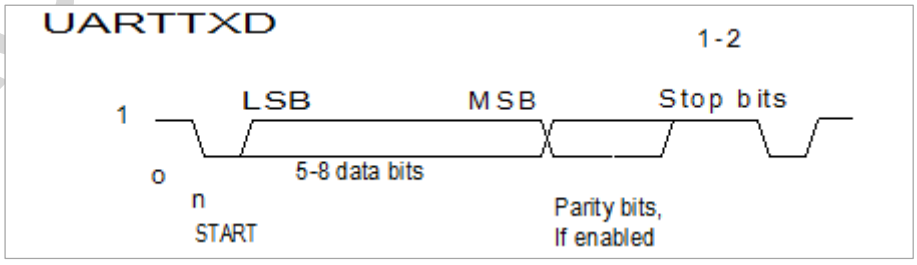

图 13-1 UART 字符帧

13.9 IrDA 数据调制

下图展示了 IrDA $^{3}/_{16}$ 数据调制的效果：

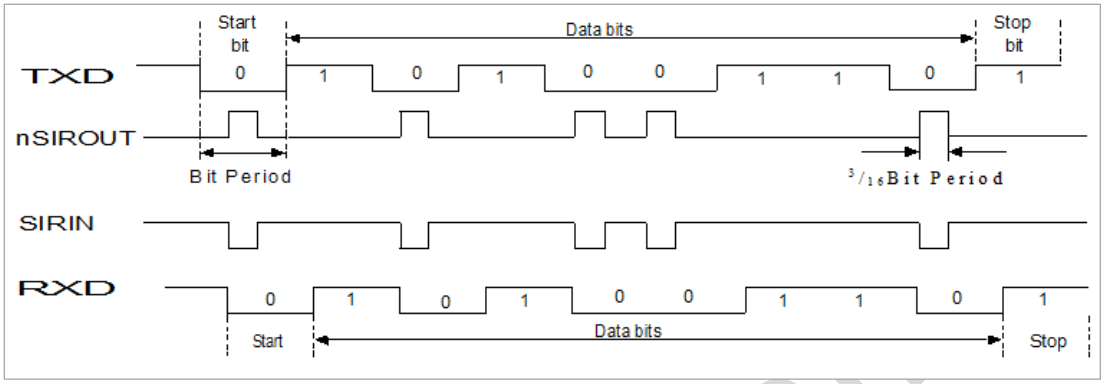

图 13-2 IrDA 数据调制（3/16）

13.10 硬件流控

硬件流控可以通过 UARTx_CR{CTSEn}和 UARTx_CR{RTSEn}配置。

当 RTS 功能被使能，且接收 FIFO 水位未满，则 nUARTRTS 信号一直有效。

当 CTS 功能被使能，只有 nUARTCTS 信号有效，且发送 FIFO 不为空才会进行传输。

13.11 中断

支持 Tx Done、Rx Done、Rx Timeout、Frame Error、Break Error、Parity Error 和 Overrun Error 中断，可以通过中断掩码设置清除寄存器 UARTx_IMSC 配置。所有中断信号可以通过原始中断状态寄存器 UARTx_RIS 查询到，包括未使能的中断位。通过掩码中断寄存器 UARTx_MIS 查询当前已使能的中断，通过中断清除寄存器 UARTx_ICR 写 1 清除相应中断。

13.12 DMA

支持 DMA 发送与接收，通过 UARTx_DMACR 开启或者关闭。

13.13 UART 相关寄存器描述

UART0 基地址：0x40003000

UART1 基地址：0x40004000

UART2 基地址：0x40010000

UART3 基地址：0x40011000

表 13-2 UART 寄存器列表

寄存器	偏移量	描述
UARTx_DR	0x00	数据寄存器
UARTx_RSR_ECR	0x04	接收状态寄存器/错误清除寄存器
UARTx_RSV0[4]	0x08	4 x 4 字节保留
UARTx_FR	0x18	标志寄存器
UARTx_RSV1	0x1C	4 字节保留
UARTx_ILPR	0x20	红外低功耗计数寄存器
UARTx_IBRD	0x24	波特率整数寄存器
UARTx_FBRD	0x28	波特率小数寄存器
UARTx_LCR_H	0x2C	线控寄存器
UARTx_CR	0x30	控制寄存器
UARTx_IFLS	0x34	中断 FIFO 水位选择寄存器
UARTx_IMSC	0x38	中断掩码设置清除寄存器
UARTx_RIS	0x3C	原始中断状态寄存器
UARTx_MIS	0x40	被掩中断状态寄存器
UARTx_ICR	0x44	中断清除寄存器
UARTx_DMACR	0x48	DMA 控制寄存器
UARTx_RSV2[997]	0x4C	4 x 997 字节保留

13.13.1 UARTx_DR(x=0、1、2、3)

偏移量：0x00

复位值：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

位 31-12 RESERVED：保留，不可更改。

位 11 OE：溢出错误标志。

- 1：发生溢出
- 0：无溢出

位 10 BE：Break 错误标志。

- 1：发生 Break 错误
- 0：未发生 Break 错误

接收数据的输入被拉低超过 1 个整字（=开始位+数据+奇偶校验位+停止位）的传输时间长度为 Break 错误。

FIFO 模式，该错误与 FIFO 顶部的字符相关。当 Break 错误产生时，只有一个 0 字符会被写入 FIFO。

位 9 PE：奇偶校验错误标志。

- 1：发生奇偶校验错误
- 0：未发生奇偶校验错误

接收字符的奇偶校验位与 UARTx_LCR_H {EPS} 和 UARTx_LCR_H {SPS} 不匹配则产生就校验错误。

FIFO 模式，该错误与 FIFO 顶部的字符相关。

位 8 FE：帧错误标志。

- 1：发生帧错误
- 0：未发生帧错误帧

错误表示收到的字符停止位无效。

FIFO 模式，该错误与 FIFO 顶部的字符相关。

位 7-0 DATA：发送数据字符/接收数据字符。

13.13.2 UARTx_RSR_ECR (x=0、1、2、3)

偏移量：0x04

复位值：0x00000000

31-4	3	2	1	0
RESERVED	OE	BE	PE	FE
r-0h	r-0h	r-0h	r-0h	r-0h

位 31-4 RESERVED：保留，不可更改。

位 3 OE：溢出错误标志。

- 1：发生溢出。
- 0：无溢出

位 2 BE：Break 错误标志。

- 1：发生 Break 错误
- 0：未发生 Break 错误

接收数据的输入被拉低超过 1 个整字（=开始位+数据+奇偶校验位+停止位）的传输时间长度为 Break 错误。

FIFO 模式，该错误与 FIFO 顶部的字符相关。当 Break 错误产生时，只有一个 0 字符会被写入 FIFO。

位 1 PE：奇偶校验错误标志。

- 1：发生奇偶校验错误
- 0：未发生奇偶校验错误

接收字符的奇偶校验位与 UARTx_LCR_H {EPS} 和 UARTx_LCR_H {SPS} 不匹配则产生就校验错误。

FIFO 模式，该错误与 FIFO 顶部的字符相关。

位 0 FE：帧错误标志。

- 1：发生帧错误
- 0：未发生帧错误帧

错误表示收到的字符停止位无效。

FIFO 模式，该错误与 FIFO 顶部的字符相关。

13.13.3 UARTx_FR (x=0、1、2、3)

偏移量：0x18

复位值：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

位 31-8 RESERVED：保留，不可更改。

位 7 TXFE：发送 FIFO 空。

- 1：发送 FIFO/数据寄存器为空
- 0：发送 FIFO/数据寄存器不为空

与 UARTx_CR_H{FEN} 位相关，该位不能用于指示发送位移寄存器中是否有数据。

位 6 RXFF：接收 FIFO 满。

- 1：接收 FIFO/数据寄存器满
- 0：接收 FIFO/数据寄存器未满

与 UARTx_CR_H{FEN} 位相关。

位 5 TXFF：发送 FIFO 满。

- 1：发送 FIFO/数据寄存器满
- 0：发送 FIFO/数据寄存器未满

与 UARTx_CR_H{FEN} 位相关。

位 4 RXFE：接收 FIFO 空标志。

- 1：接收 FIFO/数据寄存器为空
- 0：接收 FIFO/数据寄存器不为空

与 UARTx_CR_H{FEN} 位相关。

位 3 BUSY：忙标志。

- 1：正在发送数据
- 0：无数据发送

该位在发送 FIFO 变为非空状态马上置 1，不论 UART 是否使能。

位 2-0 RESERVED：保留，不可更改。

13.13.4 UARTx_ILPR (x=0、1、2、3)

偏移量：0x20

复位值：0x00000000

31-8	7-0
RESERVED	ILPDVSR
r-0h	rw-0h

位 31-8 RESERVED：保留，不可更改。

位 7-0 ILPDVSR：低功耗除数值，0 为非法值，写入 0 将导致无 IrLPBaud16 产生。

13.13.5 UARTx_IBRD (x=0、1、2、3)

偏移量：0x24

复位值：0x00000000

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

位 31-16 RESERVED：保留，不可更改。

位 15-0 BAUD_DIVINT：波特率除数整数部。

13.13.6 UARTx_FBRD (x=0、1、2、3)

偏移量：0x28

复位值：0x00000000

31-6	5-0
RESERVED	BAUD_DIVFRAC
r-0h	rw-0h

位 31-6 RESERVED：保留，不可更改。

位 5-0 BAUD_DIVFRAC：波特率除数小数部。

13.13.7 UARTx_LCR_H (x=0、1、2、3)

偏移量：0x2C

复位值：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

位 31-7 RESERVED：保留，不可更改。

位 6-5 WLEN：数据位长度。

- 11：8 位
- 10：7 位
- 01：6 位
- 00：5 位

位 4 FEN：FIFO 使能。

- 1：使能 FIFO 模式
- 0：禁用 FIFO 模式

位 3 STP2：停止位选择。

- 1：2 个停止位
- 0：1 个停止位

位 2 EPS：偶校验位选择。

- 1：偶校验
- 0：奇校验

当 PEN 为 0 时该位无效。

位 1 PEN：奇偶校验使能。

- 1：使能奇偶校验功能，发送产生奇偶校验位，接收检查奇偶校验位
- 0：禁用奇偶校验功能

位 0 BRK：发送 Break。

- 写 1：当前发送字符完成后，UART_TXD 引脚一直拉低
- 写 0：结束 Break 命令

为确保 Break 命令的执行，软件必须至少保持该位被设置超过 2 个完整的帧长度。

13.13.8 UARTx_CR (x=0、1、2、3)

偏移量：0x30

复位值：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

位 31-16 RESERVED：保留，不可更改。

位 15 CTSEn：硬件 CTS 流控使能。

- 1：开启硬件 CTS 流控
- 0：关闭硬件 CTS 流控

位 14 RTSEn：硬件 RTS 流控使能。

- 1：开启硬件 RTS 流控
- 0：关闭硬件 RTS 流控

位 13-10 RESERVED：保留，不可更改。

位 9 RXE：接收使能。

- 写 1：使能接收
- 写 0：禁止接收，若当前正在接收数据，将在这一帧数据接收完成后停止

位 8 TXE：发送使能。

- 写 1：使能发送
- 写 0：禁止发送，若当前有数据在发送，将在这一帧数据发送完成后停止

位 7-3 RESERVED：保留，不可更改。

位 2 SIRLP：Low-Power IrDA 使能。

- 1：低电平位以 3 倍 IrLPBAUD16 周期的脉冲宽度来传输，有利于降低功耗，但是也会缩短传输距离。
- 0：低电平位以 $^{3}/_{16}$ 比特周期的脉冲宽度传输。

位 1 SIRE：IrDA 使能。

- 1：开启 IrDA SIR ENDEC，UART_TXD 保持高，UART_RXD 被忽略。数据在 SIR_OUT 和 SIR_IN 上传输。
- 0：关闭 IrDA SIR ENDEC，SIR_OUT 保持低，SIR_IN 被忽略。数据在 UART_TXD 和 UART_RXD 上传输。

若 UARTEN 位为 0，该位无效。

位 0 UARTEN：串口使能。

- 写 1：使能 UART 功能
- 写 0：关闭 UART 功能，若当前有数据在发送或者接收，将在这一帧数据传输完成后再关闭 UART。

13.13.9 UARTx_IFLS (x=0、1、2、3)

偏移量：0x34

复位值：0x00000000

31-6	5-3	2-0
RESERVED	RXIFLSEL	TXIFLSEL
r-0h	rw-0h	rw-0h

位 31-6 RESERVED：保留，不可更改。

位 5-3 RXIFLSEL：接收 FIFO 中断水位选择。

- 000：接收 FIFO 水位超 1/8
- 001：接收 FIFO 水位超 1/4
- 010：接收 FIFO 水位超 1/2
- 011：接收 FIFO 水位超 3/4
- 100：接收 FIFO 水位超 7/8
- 101~111：保留

位 2-0 TXIFLSEL：发送 FIFO 中断水位选择。

- 000：发送 FIFO 水位超 1/8
- 001：发送 FIFO 水位超 1/4
- 010：发送 FIFO 水位超 1/2
- 011：发送 FIFO 水位超 3/4
- 100：发送 FIFO 水位超 7/8
- 101~111：保留

13.13.10 UARTx_IMSC (x=0、1、2、3)

偏移量：0x38

复位值：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

位 31-11 RESERVED：保留，不可更改。

位 10 OEIM：Overrun 错误中断掩码。

- 1：使能 Overrun 错误中断
- 0：禁用 Overrun 错误中断

位 9 BEIM：Break 错误中断掩码。

- 1：使能 Break 错误中断
- 0：禁用 Break 错误中断

位 8 PEIM：校验错误中断掩码。

- 1：使能校验错误中断
- 0：禁用校验错误中断

位 7 FEIM：帧错误中断掩码。

- 1：使能帧错误中断
- 0：禁用帧错误中断

位 6 RTIM：接收超时中断掩码。

- 1：使能接收超时中断
- 0：禁用接收超时中断

位 5 TXIM：发送完成中断掩码。

- 1：使能发送完成中断
- 0：禁用发送完成中断

位 4 RXIM：接收完成中断掩码。

- 1：使能接收完成中断
- 0：禁用接收完成中断

位 3-0 RESERVED：保留，不可更改。

13.13.11 UARTx_RIS (x=0、1、2、3)

偏移量：0x3C

复位值：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

位 31-11 RESERVED：保留，不可更改。

位 10 OERIS：Overrun 错误中断原始状态。

位 9 BERIS：Break 错误中断原始状态。

位 8 PERIS：校验错误中断原始状态。

位 7 FERIS：帧错误中断原始状态。

位 6 RTRIS：接收超时中断原始状态。

位 5 TXRIS：发送完成中断原始状态。

位 4 RXRIS：接收完成中断原始状态。

位 3-0 RESERVED：保留，不可更改。

13.13.12 UARTx_MIS (x=0、1、2、3)

偏移量：0x40

复位值：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

位 31-11 RESERVED：保留，不可更改。

位 10 OEMIS：Overrun 错误中断状态。

位 9 BEMIS：Break 错误中断状态。

位 8 PEMIS：校验错误中断状态。

位 7 FEMIS：帧错误中断状态。

位 6 RTMIS：接收超时中断状态。

位 5 TXMIS：发送完成中断状态。

位 4 RXMIS：接收完成中断状态。

位 3-0 RESERVED：保留，不可更改。

13.13.13 UARTx_ICR (x=0、1、2、3)

偏移量：0x44

复位值：0x00000000

31-16	15-11	10	9	8	7	6	5	4	3-0
RESERVED	RESERVED	OEIC	BEIC	PEIC	FEIC	RTIC	TXIC	RXIC	RESERVED
r-0h	r-0h	w-0h	w-0h	w-0h	w-0h	w-0h	w-0h	w-0h	r-0h

位 31-11 RESERVED：保留，不可更改。

位 10 OEMIS：Overrun 错误中断清除。

- 写 1：清除 overrun 中断
- 写 0：无效

位 9 BEMIS：Break 错误中断清除。

- 写 1：清除 Break 错误中断
- 写 0：无效

位 8 PEMIS：校验错误中断清除。

- 写 1：清除校验错误中断
- 写 0：无效

位 7 FEMIS：帧错误中断清除。

- 写 1：清除帧错误中断
- 写 0：无效

位 6 RTMIS：接收超时中断清除。

- 写 1：清除接收超时中断
- 写 0：无效

位 5 TXMIS：发送完成中断清除。

- 写 1：清除发送完成中断
- 写 0：无效

位 4 RXMIS：接收完成中断清除。

- 写 1：清除接收完成中断
- 写 0：无效

位 3-0 RESERVED：保留，不可更改。

13.13.14 UARTx_DMACR (x=0、1、2、3)

偏移量：0x48

复位值：0x00000000

31-3	2	1	0
RESERVED	DMAONERR	TXDMAE	RXDMAE
r-0h	rw-0h	rw-0h	rw-0h

位 31-3 RESERVED：保留，不可更改。

位 2 DMAONERR：DMA 错误。

位 1 TXDMAE：发送 DMA 使能。

- 1：开启发送 DMA 功能
- 0：关闭发送 DMA 功能

位 0 RXDMAE：接收 DMA 使能。

- 1：开启接收 DMA 功能
- 0：关闭接收 DMA 功能

13.13.15 UARTx_ID[8] (x=0、1、2、3)

13.13.15.1 PeriphID0

偏移量：0x0FE0

复位值：0x00000000

31-8	7-0
RESERVED	PARTNUMBER0
r-0h	r-11h

位 31-8 RESERVED：保留，不可更改。

位 7-0 PARTNUMBER0：=0x11

13.13.15.2 PeriphID1

偏移量：0x0FE4

复位值：0x00000000

31-8	7-4	3-0
RESERVED	DESIGNER0	PARTNUMBER1
r-0h	r-1h	r-0h

位 31-8 RESERVED：保留，不可更改。

位 7-4 DESIGNER0：=0x1

位 3-0 PARTNUMBER1：=0x0

13.13.15.3 PeriphID2

偏移量：0x0FE8

复位值：0x00000000

31-8	7-4	3-0
RESERVED	REVISION0	DESIGNER1
r-0h	r-xh	r-0h

位 31-8 RESERVED：保留，不可更改

位 7-4 REVISION0：

- 0x0：r1p0
- 0x1：r1p1
- 0x2：r1p3/r1p4
- 0x3：r1p5

位 3-0 DESIGNER1：=0x0

13.13.15.4 PeriphID3

偏移量：0x0FEC

复位值：0x00000000

31-8	7-0
RESERVED	CONFIGURATION
r-0h	r-0h

位 31-8 RESERVED：保留，不可更改。

位 7-0 CONFIGURATION：=0x00

13.13.15.5 PCellID0

偏移量：0x0FD0

复位值：0x00000000

31-8	7-0
RESERVED	CellID0
r-0h	r-dh

位 31-8 RESERVED：保留，不可更改。

位 7-0 CellID0：=0x0d

13.13.15.6 PCellID1

偏移量：0x0FD4

复位值：0x00000000

31-8	7-0
RESERVED	CellID1
r-0h	r-f0h

位 31-8 RESERVED：保留，不可更改。

位 7-0 CellID1：=0xf0

13.13.15.7 PCellID2

偏移量：0x0FD8

复位值：0x00000000

31-8	7-0
RESERVED	CellID2
r-0h	r-5h

位 31-8 RESERVED：保留，不可更改。

位 7-0 CellID2：=0x05

13.13.15.8 PCellID3

偏移量：0x0FDC

复位值：0x00000000

31-8	7-0
RESERVED	CellID3
r-0h	r-b1h

位 31-8 RESERVED：保留，不可更改。

位 7-0 CellID3：=0xb1

14. SSP

14.1 简介

SSP（Synchronous serial port）是一种同步串行接口，支持 MASTER 和 SLAVE 模式。

SSP 支持多种帧格式，并且可以根据需要配置数据宽度和输出速率。

14.2 主要特性

- 支持 MASTER 和 SLAVE 模式的配置
- 最大支持 16MHz 输出
- 支持 16bit 深度为 8 的 TX/RX FIFO
- 支持多种帧格式
- 支持 4-16bit 数据宽度
- 支持 DMA 请求
- 支持中断请求

14.3 功能描述

14.3.1 基础说明

SSP 主要有 4 个 pin：SSP_NSS，SSP_CLK，SSP_TX 和 SSP_RX。

1. **SSP_NSS**
   SSP 片选信号，低有效。
2. **SSP_CLK**
   SSP 时钟信号，对 MASTER 模式来说是时钟输出，对 SLAVE 模式来说是时钟输入。
3. **SSP_TX**
   SSP 发送信号，无论 MASTER 模式还是 SLAVE 模式，均为发送 pin。
4. **SSP_RX**
   SSP 接收信号，无论 MASTER 模式还是 SLAVE 模式，均为接收 pin。

SSP 与 SPI 设备的连接如下图，需注意 SSP_TX/SSP_RX 与 SPI_MOSI/SPI_MISO 的不同。

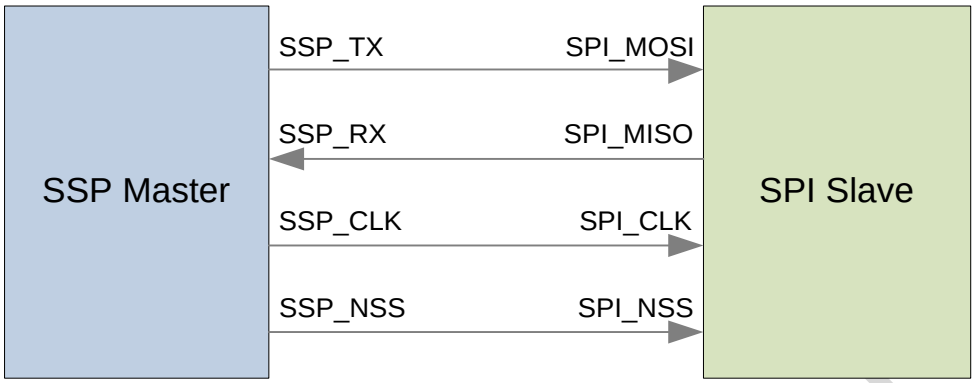

图 14-1 SSP master 与 SPI slave 之间的连接

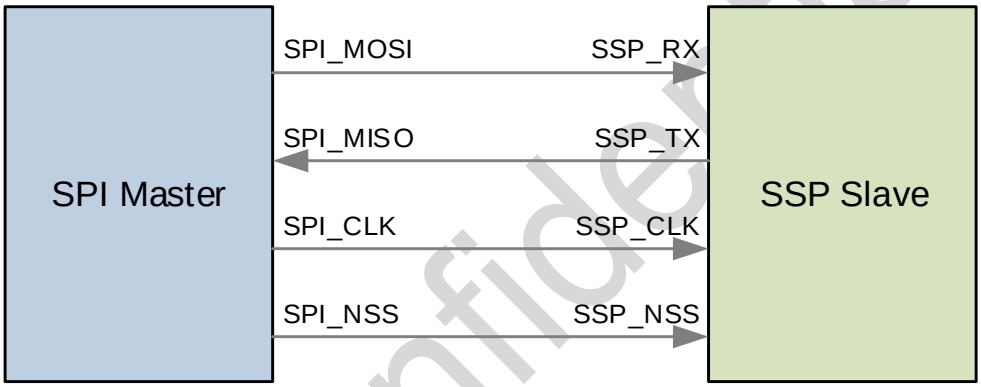

图 14-2 SPI Master 与 SSP Slave 之间的连接

14.3.2 时钟分频

SSP 时钟约束条件:

(1) 最大支持的输出时钟为 16MHz

(2) MASTER 模式下时钟最大为 PCLK 的 1/2

(3) SLAVE 模式下时钟最大为 PCLK 的 1/12

MASTER 模式下时钟输出的公式如下:

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

图 14-3 MASTER 模式下时钟输出的公式

SSPCLK 为 SSP 的接口时钟，SSPCLKOUT 为 SSP 的输出时钟。以默认 24MHz 为例，如果要输出 1MHz 的时钟，设置 CPSDVR 为 2，则 SCR 为 11。

14.3.3 数据格式

SSP 支持 3 种帧格式:

- Motorola SPI
- Texas Instruments SPI
- National Semiconductor Microwire

14.3.4 DMA 传输

SSP DMA 发送过程：

(1) 将寄存器 SSP_DMACR 中的 TXDMAE 位配置为使能;
(2) 将寄存器 SSP_DR 地址配置为 DMA 的目的地址;
(3) 将发送数据的内存地址配置为 DMA 的源地址;
(4) 配置 DMA 的 DATA_WIDTH 为 0（数据位宽为 8bit）;
(5) 配置 DMA 的 SRC_MSIZE 和 DEST_MSIZE 为 1（burst length 为 4）;
(6) 配置 DMA 的数据传输总长度;
(7) 配置 DMA 的 handshake 类型为对应 SSP 的 TX 类型（如 SSP0 为 DMA_HANDSHAKE_SSP_0_TX）;
(8) 激活 DMA 通道。

当 DMA 传输完成后，会将 DMA CHENREG 寄存器的 CH_EN 位清 0。

SSP DMA 接收过程：

(1) 将寄存器 SSP_DMACR 中的 RXDMAE 位配置为使能;
(2) 将寄存器 SSP_DR 地址配置为 DMA 的源地址;
(3) 将数据接收的内存地址配置为 DMA 的目的地址;
(4) 配置 DMA 的 DATA_WIDTH 为 0（数据位宽为 8bit）;
(5) 配置 DMA 的 SRC_MSIZE 和 DEST_MSIZE 为 1（burst length 为 4）;
(6) 配置 DMA 的数据传输总长度;
(7) 配置 DMA 的 handshake 类型为对应 SSP 的 RX 类型（如 SSP0 为 DMA_HANDSHAKE_SSP_0_RX）;
(8) 激活 DMA 通道。

当 DMA 传输完成后，会将 DMA CHENREG 寄存器的 CH_EN 位清 0。

14.3.5 中断信号

SSP 主要有四个中断：SSP RX 中断，SSP TX 中断，SSP RX OVERRUN 中断和 SSP RX TIMEOUT。

1. **SSP RX 中断**

   当 SSP RX FIFO 中有大于等于 4 个数据时触发。

2. **SSP TX 中断**

   当 SSP TX FIFO 中有小于等于 4 个数据时触发。

3. **SSP RX Overrun 中断**

   当 SSP RX FIFO 已满，继续收到数据时触发。

4. **SSP RX Timeout 中断**

   当 SSP RX FIFO 不为空，但是 SSP 在 32bit 传输周期中未继续收到数据时触发。

14.4 SSP 相关寄存器描述

SSP0 寄存器基地址：0x40006000

SSP1 寄存器基地址：0x40012000

SSP2 寄存器基地址：0x40013000

表 14-1 SSP 寄存器列表

寄存器	偏移量	描述
SSP_CR0	0x00	控制寄存器 0
SSP_CR1	0x04	控制寄存器 1
SSP_DR	0x08	数据寄存器
SSP_SR	0x0C	状态寄存器
SSP_CPSR	0x10	时钟分频寄存器
SSP_IMSC	0x14	中断设置寄存器
SSP_RIS	0x18	原始中断状态寄存器
SSP_MIS	0x1C	屏蔽中断状态寄存器
SSP_ICR	0x20	中断清除寄存器
SSP_DMACR	0x24	DMA 控制寄存器

14.4.1 SSP_CR0

偏移量：0x00

复位值：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

位 31-16 RESERVED：保留，不能修改。

位 15-8 SCR：串行时钟速率，用于设置 SSP 传输的数据速率。

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

SSP 的数据速率计算公式如上，其中 CPSDVR 是取值 2 到 254 的偶数。

位 7 SPH：SSP 相位设置，仅应用于 Motorola SPI 格式。

位 6 SPO：SSP 极性设置，仅应用于 Motorola SPI 格式。

位 5-4 FRF：SSP 帧格式设置。

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

位 3-0 DSS：数据位宽设置。

- 0：保留
- 1：保留
- 2：保留
- 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

偏移量：0x04

复位值：0x00000000

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

位 31-4 RESERVED：保留，不能修改。

位 3 SOD：从模式输出禁止。

- 0：从模式下，SSP 可以输出
- 1：从模式下，SSP 不可输出

位 2 MS：主从模式选择。

- 0：主模式
- 1：从模式

位 1 SSE：SSP 使能。

- 0：不使能
- 1：使能

位 0 LBM：回环模式。

- 0：正常模式
- 1：回环模式

14.4.3 SSP_DR

偏移量：0x08

复位值：0x00000000

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

位 31-16 RESERVED：保留，不能修改。

位 15-0 DATA：SSP TX/RX 数据。

14.4.4 SSP_SR

偏移量：0x0C

复位值：0x00000003

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

位 31-5 RESERVED：保留，不能修改。

位 4 BSY：SSP 忙标识。

- 0：SSP 空闲
- 1：SSP 正在传输中

位 3 RFF：RX FIFO 满标识。

- 0：RX FIFO 未满
- 1：RX FIFO 满

位 2 RNE：RX FIFO 非空标识。

- 0：RX FIFO 为空
- 1：RX FIFO 不为空

位 1 TNF：TX FIFO 非满标识。

- 0：TX FIFO 未满
- 1：TX FIFO 满

位 0 TFE：TX FIFO 空标识。

- 0：TX FIFO 不为空
- 1：TX FIFO 为空

14.4.5 SSP_CPSR

偏移量：0x0C

复位值：0x00000000

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

位 31-8 RESERVED：保留，不能修改。

位 7-0 CPSDVSR：时钟分频因子，必须为 2-254 之间的偶数。

14.4.6 SSP_IMSC

偏移量：0x00

复位值：0x00000000

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

位 31-4 RESERVED：保留，不能修改。

位 3 TXIM：TX 中断屏蔽位。

- 0：不允许产生 TX 中断
- 1：允许产生 TX 中断

位 2 RXIM：RX 中断屏蔽位。

- 0：不允许产生 RX 中断
- 1：允许产生 RX 中断

位 1 RTIM：RX TIMEOUT 中断屏蔽位。

- 0：不允许产生 RX TIMEOUT 中断
- 1：允许产生 RX TIMEOUT 中断

位 0 RORIM：RX OVERRUN 中断屏蔽位。

- 0：不允许产生 RX OVERRUN 中断
- 1：允许产生 RX OVERRUN 中断

14.4.7 SSP_RIS

偏移量：0x00

复位值：0x00000008

31-4	3	2	1	0
RESERVED	TXRIS	RXRIS	RTRIS	RORRIS
r	r	r	r	r

位 31-4 RESERVED：保留，不能修改。

位 3 TXRIS：TX 原始中断状态。

位 2 RXRIS：RX 原始中断状态。

位 1 RTRIS：RX TIMEOUT 原始中断状态。

位 0 RORRIS：RX OVERRUN 原始中断状态。

14.4.8 SSP_MIS

偏移量：0x00

复位值：0x00000000

31-4	3	2	1	0
RESERVED	TXMIS	RXMIS	RTMIS	RORMIS
r	r	r	r	r

位 31-4 RESERVED：保留，不能修改。

位 3 TXMIS：TX 屏蔽中断状态。

位 2 RXMIS：RX 屏蔽中断状态。

位 1 RTMIS：RX TIMEOUT 屏蔽中断状态。

位 0 RORMIS：RX OVERRUN 屏蔽中断状态。

14.4.9 SSP_ICR

偏移量：0x00

复位值：0x00000000

31-2	1	0
RESERVED	RTIC	RORIC
r	w	w

位 31-2 RESERVED：保留，不能修改。

位 1 RTIC：RX TIMEOUT 中断清除，写 1 清除，写 0 无效。

位 0 RORIC：RX OVERRUN 中断清除，写 1 清除，写 0 无效。

14.4.10 SSP_DMACR

偏移量：0x00

复位值：0x00000000

31-2	1	0
RESERVED	TXDMAE	RXDMAE
r	r/w	r/w

位 31-2 RESERVED：保留，不能修改。

位 1 TXDMAE：DMA TX 使能。

- 0：关闭 DMA TX
- 1：使能 DMA TX

位 0 RXDMAE：DMA RX 使能。

- 0：关闭 DMA RX
- 1：使能 DMA RX

15. I2C

15.1 简介

I2C 总线接口单元支持主机模式和从机模式。SDA 为数据传输线，SCL 为参考时钟线。支持多主机和总线仲裁功能。支持 100Kbps 标准速率模式，和 400Kbps 快速模式。支持 FIFO 模式，发送 FIFO 深度 8，接收 FIFO 深度 16，FIFO 的读写指针可配。

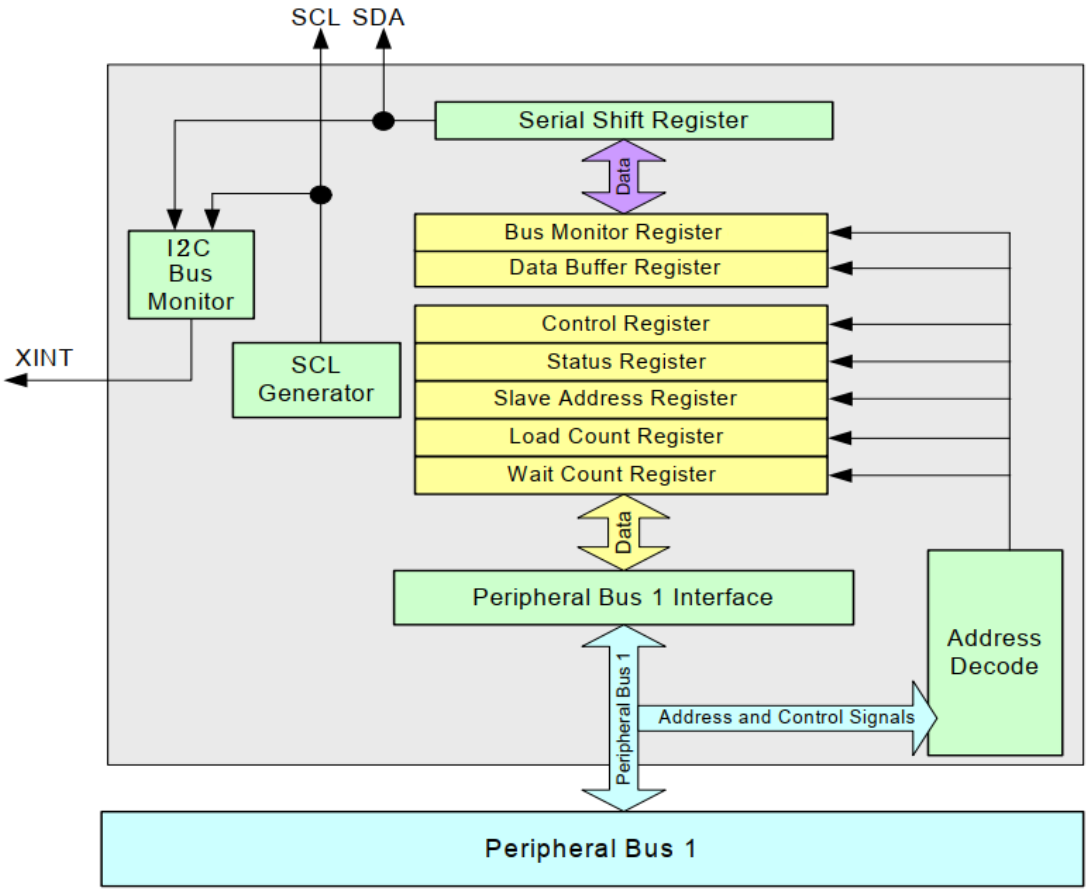

图 15-1 I2C 框图

15.2 Start 与 Stop 条件

Start 条件：当 SCL 为高时，SDA 从高跳变为低则产生 Start 条件。

Stop 条件：当 SCL 为高时，SDA 从低跳变为高则产生 Stop 条件。

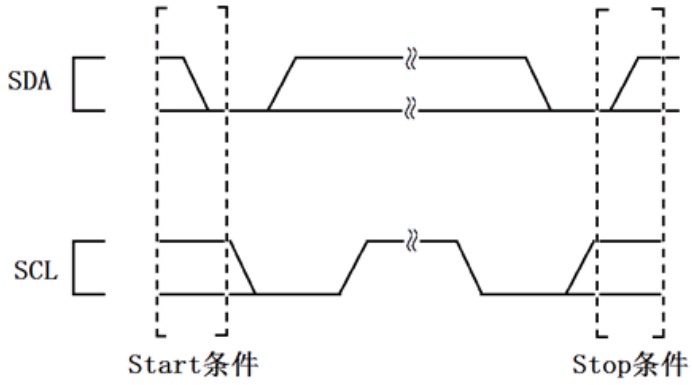

图 15-2 Start 与 Stop 条件的 SDA 与 SCL 信号

通过配置 I2Cx_CR{START} 和 I2Cx_CR{STOP} 来开始一个字节的传输，或者产生 Start、Repeat Start 和 Stop 条件。

表 15-1 Start 和 Stop 条件定义

Start 位	Stop 位	条件	描述
0	0	无 Start 和 Stop	当有多个数据字节将要被传输的时候，I2C 不会发送 Start 或者 Stop 条件
0	1	Start 或 Repeat Start	I2C 发送一个 Start 条件然后再发送 I2Cx_DBR 内的 8 位数据。Start 发送前，I2Cx_DBR 必须要包含 7 位的从地址和 1 位的 R/nW。 Repeat Start 条件，I2Cx_DBR 包含目标从设备地址和 R/nW 位，这允许主机在不释放总线的情况下进行多次传输。 接口停留在主机发送模式用于写，切换到主机接收模式用于读。
1	x	Stop 条件	在主机发送模式，I2Cx_DBR 内的 8 位数据发送完成之后在总线上发送一个 Stop 条件。 在主机接收模式，I2Cx_CR{ACKNAK}必需置 1 用来发送一个 NAK 脉冲，接收的数据被存入 I2Cx_DBR 寄存器，然后在总线上发送一个 Stop 条件。

1. **Start 条件**

Start 条件和 I2Cx_DBR 内的数据在 I2Cx_CR{TB}被置 1 后开始发送。写请求，I2C 总线停留在主机发送模式，读请求将停留在主机接收模式。Repeat Start 条件，改变读写或者目

标从设备地址，I2Cx_DBR 将包含更新的从设备地址和 R/nW 位。

I2C 不会 I2Cx_CR{START}。如果在开始发送 Start 条件的时候丢失总线仲裁，I2C 会在总线空闲的时候重新尝试发送一个 Start 条件。

2. **无 Start 或者 Stop 条件**

当 I2C 在发送多个数据字节的时候，I2Cx_CR{START}=0，I2Cx_CR{STOP}=0，此时无 Start 或者 Stop 条件。软件写数据字节，I2C 将 I2Cx_SR{ITE}置 1 并且清除 I2Cx_CR{TB}。软件继续写一个新的字节到 I2Cx_DBR 寄存器，并且把 I2Cx_CR{TB}置 1，开始一个新的字节发送。这个过程一直继续，直到软件把 I2Cx_CR{START}或者 I2Cx_CR{STOP}置 1，I2C 在发送完一个 Start、Stop 或 Repeat Start 条件后，I2Cx_CR{START} 和 I2Cx_CR{STOP}不会被自动清 0。

在每个字节与 ACK/NAK 被发送完成后，I2C 一直将 SCL 拉低等待，直到 I2Cx_CR{TB}被置 1。

3. **停止条件**

停止条件结束一次数据传输。在主机发送模式，I2Cx_CR{STOP}和 I2Cx_CR{TB}必须置 1 来开始最后一个字节的传输。在主机接收模式，I2Cx_CR{ACKNAK}、I2Cx_CR{STOP}和 I2Cx_CR{TB}必须置 1 来开最后一个字节的接收。Stop 条件发送完成之后，软件必须把 I2Cx_CR{STOP}清 0。

15.3 数据传输顺序

I2C 以 1 字节递增的方式传输数据，遵循以下顺序:

1. Start
2. 7 位从机地址
3. R/nW 位
4. ACK
5. 8 位数据
6. Acknowlegement
7. 重复步骤 5 和步骤 6
8. Repeat Start（重复步骤 1）或 Stop

15.4 数据与寻址

I2C 数据 Buffer 寄存器 I2Cx_DBR 和 I2C 从地址寄存器 I2Cx_SAR 管理数据和从机寻址。I2Cx_DBR 包含 1 字节数据或者 7 位目标从机地址和 1 位 R/nW。I2Cx_SAR 包含 I2C 单元自身的从机地址。I2C 接收完一个完整字节数据和 ACK 后将数据存入 I2Cx_DBR。发送时，CPU 将数据写入 I2Cx_DBR，当 I2Cx_CR{TB}置 1 后把数据发送到总线上。

1. **主机或者从机发送模式：**

   (1) 将数据写入 I2Cx_DBR 寄存器开始一次主机事务，或者在 I2Cx_SR{ITE} 置 1 后发送下一个字节。

   (2) 当 I2Cx_CR{TB} 置 1 后发送 I2Cx_DBR 中的数据。

   (3) 如果使能了 I2Cx_CR{ITEIE}，在发送完一个字节和 ACK 后会触发 I2Cx_DBR 空中断。

   (4) 在 CPU 写 I2Cx_DBR 寄存器前，当 I2C 已准备好发送下一个字节，且无 Stop 条件，I2C 处于等待状态，直到 CPU 写 I2Cx_DBR 寄存器并且把 I2Cx_CR{TB} 置 1。

   注意：*在 FIFO 模式，以 TX FIFO 替代 I2Cx_DBR。*

2. **主机或者从机接收模式：**

   (1) 当一个完整字节数据被接收后（使能 I2Cx_CR{DRFIE}，触发 I2Cx_DBR 接收满中断，I2Cx_SR{IRF}被置 1），CPU 读 I2Cx_DBR 寄存器取回数据。

   (2) 当 ACK 完成后，I2C 将数据从位移寄存器传输到 I2Cx_DBR 寄存器。

   (3) I2C 处于等待模式，直到 I2Cx_DBR 寄存器被 CPU 读取。

   (4) CPU 读取 I2Cx_DBR 寄存器之后，I2C 更新 I2Cx_CR{ACKNAK}位和 I2Cx_CR{TB}位，允许下一字节的传输。

   注意：*在 FIFO 模式，以 RX FIFO 替代 I2Cx_DBR。*

3. **从机寻址：**

   作为主机设备，I2C 必须要构建和发送一次事务的第一个字节。这个字节由 7 位的从机地址和 1 位的 R/nW 组成。第一个字节的发送必须要得到从设备的 ACK 响应。如果是写事务，I2C 保持在主机发送模式，同时从机保持在接收模式。如果是读事务，I2C 在收到 ACK 后马上切换到主机接收模式，同时从机切换到发送模式。如果收到 NAK，I2C 自动发送 Stop 条件并且把 I2Cx_SR{BED}置 1 来中止当前事务。

15.5 ACK

每一个字节的传输必须伴随 ACK，由接收的主机或者从机产生。发送方必须释放 SDA 线给接收方传输 ACK 脉冲。

在主机发送模式，如果目标接收从机未产生 ACK，SDA 线保持高电平指示一个 NAK。缺少 ACK 导致 I2C 将 I2Cx_SR{BED} 置 1 并产生中断，I2C 自动产生 Stop 条件并且中止传输。

在主机接收模式，I2C 发送 NAK 给发送从机通知从机停止发送数据，I2Cx_CR{ACKNAK} 控制总线上 ACK/NAK 的产生。按照 I2C 协议的规定，主机接收模式 NAK 不会将 I2Cx_SR{BED} 置 1。I2C 从总线上每接收一个字节会自动发送 ACK，在接收到最后一个字节之前软件必须将 I2Cx_{ACKNAK} 置 1 来发送 NAK。NAK 在最后一个字节被传输后发送，告知最后一个字节被发送完成。

在从机接收模式，I2C 自动对自身从机地址进行 ACK 响应，不论 I2Cx_{ACKNAK} 是否被置 1。在从机模式，I2C 自动对接收到的每一个字节数据进行 ACK 响应，不论 I2Cx_{ACKNAK} 是否被置 1。I2C 在接收到每个字节的 8 位数据之后发送 ACK。

在从机发送模式，接收 NAK 意味着当前这次传输的最后一个字节被发送完成。主机接着发送一个 Stop 条件或者 Repeat Start 条件。I2Cx_SR{UB} 保持为 1 直到一个 Stop 条件或者 Repeat Start 条件被接收到。

15.6 仲裁

为兼容多主机，需要总线仲裁功能。总线仲裁用于在最小 I2C Start 条件时间内有 2 个或者更多主机同时产生 Start 条件的情形。

仲裁可以持续一段长的时间。如果从机地址和 R/nW 位一直的话，仲裁移到数据阶段。由于 I2C 总线的线与属性，如果 2 个或者所有主机输出同样的总线状态。如果地址，或者 R/nW 位，或者数据不同，转变到高状态的主机（主机数据与 SDA 线不同）丢失仲裁，并且结束数据传输，将 I2Cx_SR{ALD} 置 1，返回空闲状态。

在 FIFO 模式，丢失仲裁的时候软件必须清空 FIFO。这可以通过清空发送和接收 FIFO 的读写指针寄存器来实现。

15.7 主机模式

当软件开始执行读或者写操作，I2C 从默认的从机接收模式切换到主机发送模式。Start 条件之后跟随着 7 位的从机地址和 1 位的 R/nW。

当接收到 ACK 后，I2C 进入以下两种模式之一:

- 主机发送模式-写数据
- 主机接收模式-读数据

CPU 写 I2Cx_CR 寄存器来开始一次主机事务。

表 15-2 主机事务

主机动作	主机操作	定义
产生时钟输出	主机发送 主机接收	● 主机驱动 SCL 线 ● I2Cx_CR{SCLE} 和 I2Cx_CR{IUE} 必须置 1
写目标从机地址到 I2Cx_DBR	主机发送 主机接收	● CPU 在使能 Start 条件前写 I2Cx_DBR[7:1] ● 前 7 个位在 Start 条件之后发送
写 R/nW 位到 I2Cx_DBR	主机发送 主机接收	● CPU 把 R/nW 控制位写入 I2Cx_DBR 的最低位 ● 若 R/nW 为低则主机保持发送模式，若 R/nW 为高则主机切换到接收模式
发送 Start 条件	主机发送 主机接收	在 7 位目标从机地址和 1 位 R/nW 写入 I2Cx_DBR 寄存器之后， ● 软件把 I2Cx_CR{START} 置 1 ● 软件把 I2Cx_CR{TB}置 1 开始发送 Start 条件
开始第一个字节传输	主机发送 主机接收	● CPU 写一个字节到 I2Cx_DBR 寄存器 ● 软件把 I2Cx_CR{TB} 置 1，开始这个字节的发送 ● 发送完成之后，I2Cx_CR{TB} 被清 0，I2Cx_SR{ITE} 被置 1
总线仲裁	主机发送 主机接收	如果在同一个时钟周期内有多个主机在总线上发送了 Start 条件，那么总线仲裁必须产生， ● 只要有需要，I2C 仲裁就会产生。总线仲裁发生在目标从机地址和 R/nW 位，以及数据传输阶段，直到除 1 个主机之外的主机都丢失总线。数据不会丢失。 ● 如果丢失仲裁，I2Cx_SR{ALD} 会被置 1，I2C 切换到从机接收模式。 ● 如果在发送目标从机地址的时候丢失仲裁，I2C 会在总线空闲的时候再次尝试重发。
写一个字节到 I2Cx_DBR	仅主机发送	● 如果 I2Cx_SR{ITE} 被置 1 并且 I2Cx_CR{TB} 被清 0，当 I2Cx_DBR 空中断被使能，那么中断产生。 ● CPU 写一个字节到 I2Cx_DBR 寄存器，并且根据需要设置合适的 Start/Stop 条件组合，然后把 I2Cx_CR{TB}

		置 1 发送数据。数据的 8 个位被从位移寄存器搬到串行总线上。若发送前 I2Cx_CR{STOP} 置 1，那么在数据的 8 个位传输完成之后会跟随一个 Stop 条件。
等待接收从机 ACK	仅主机发送	作为发送方，主机产生 ACK 的时钟，并且将 SDA 线释放给接收的从机发送 ACK。
从 I2Cx_DBR 读取一个字节	仅主机接收	在 I2Cx_CR{ACKNAK} 被读取之后，位移寄存器内的 8 位数据被搬到 I2Cx_DBR 寄存器， ● 当 I2Cx_SR{IRF} 被置 1 且 I2Cx_CR{TB} 被清 0 时，CPU 读取 I2Cx_DBR 寄存器。可以使能 I2Cx_DBR 寄存器接收满中断通知 CPU。 ● 当 I2Cx_DBR 被读取完，如果 I2Cx_SR{ACKNAK} 被清 0（代表 ACK），软件必须把 I2Cx_CR{ACKNAK} 清 0 并且把 I2Cx_CR{TB} 置 1 来开始下一个字节的读取。 ● 如果 I2Cx_SR{ACKNAK} 被置 1（代表 NAK），I2Cx_CR{TB} 被清除，I2Cx_CR{STOP} 被置 1，且 I2Cx_SR{UB} 被置 1，最后一个字节已经被读取到 I2Cx_DBR 寄存器，I2C 正在发送 Stop 条件。 ● 如果 I2Cx_SR{ACKNAK} 被置 1（代表 NAK），并且 I2Cx_CR{TB} 被清 0，但是 I2Cx_CR{STOP} 被清 0，软件有两个选择: 1. 把 I2Cx_CR{START} 置 1，将新的目标从机地址写入 I2Cx_DBR，把 I2Cx_CR{TB} 置 1，发送一个 Repeat Start 条件。 2. 把 I2Cx_CR{MA} 置，并且保持 I2Cx_CR{TB}为 0，仅发送一个 Stop 条件。
发送 ACK 到发送从机	仅主机接收	● 作为接收主机一方，在 ACK 期间，产生 ACK 时钟，并且驱动 SDA 线 ● 如果下一个字节为最后一个事务，软件需要把 I2Cx_CR{ACKNAK} 置 1 来产生 NAK。
产生 Repeat Start 条件	主机发送 主机接收	使用 Repeat Start 代替 Stop 条件可以在不释放总线的情况下继续新的传输 ● Repeat Start 条件在最后一个字节数据被传输后产生 ● 软件必须把 7 位的目标从机地址和 1 位的 R/nW 位写入 I2Cx_DBR 寄存器，然后把 I2Cx_CR{START} 置 1，再把 I2Cx_CR{TB} 置 1
产生 Stop 条件	主机发送 主机接收	● Stop 条件在最后一个字节数据被传输后产生 ● I2Cx_CR{STOP} 要在最后一个字节被传输前置 1

15.8 FIFO 模式

FIFO 模式只能在主机模式被使用。

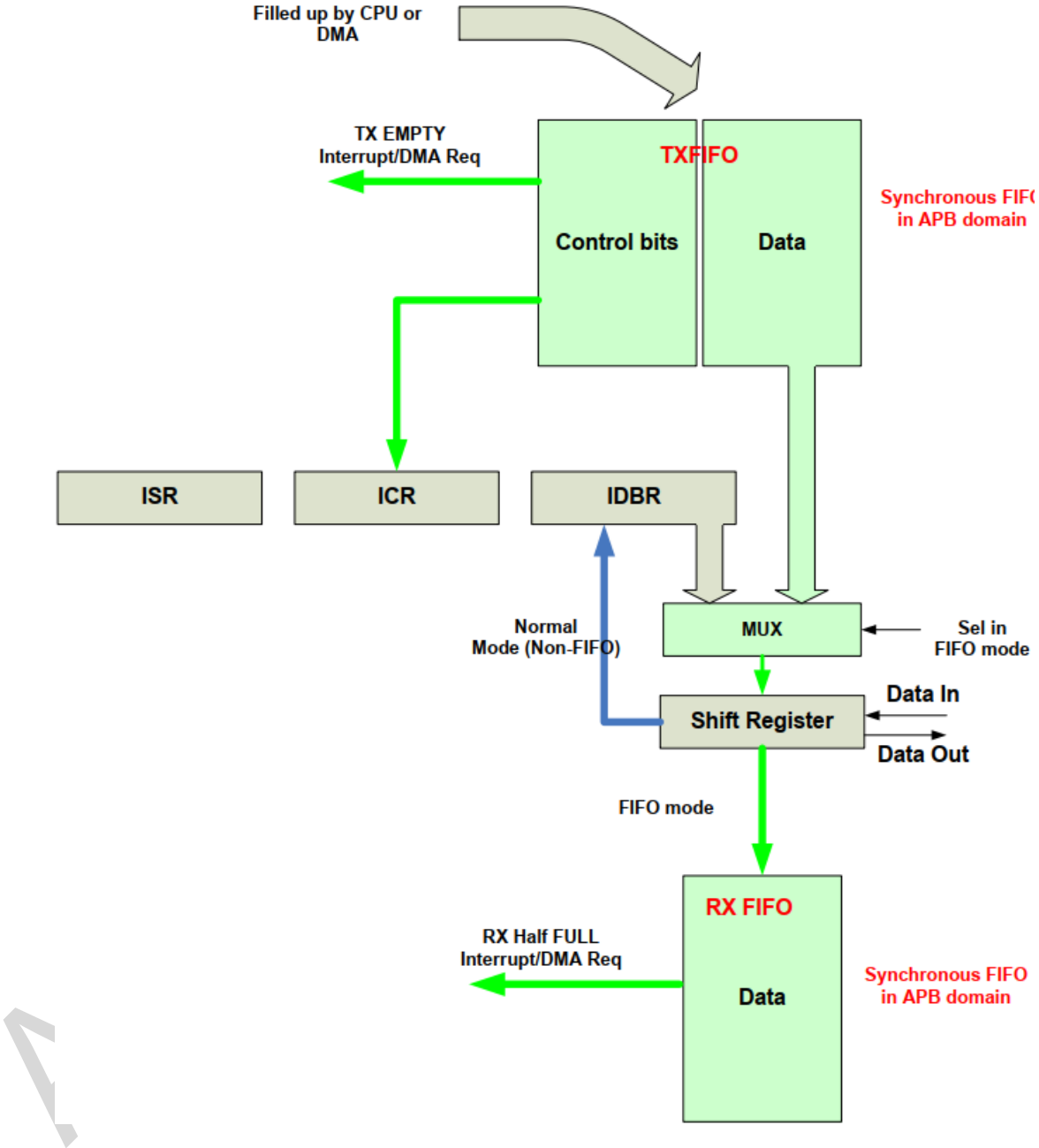

图 15-3 FIFO 模式示意图

FIFO 模式可以用于发送和接收，以帮助减少 I2Cx_DBR 寄存器空中断和满中断，FIFO 允许读取和写入多个字节而不需要在每个字节操作之后中断 CPU。

DMA 被用于改善传输长度超过 8 个字节数据的 I2C 事务，整个事务可用 DMA 的方式完成，而不用产生多次 FIFO 中断。

FIFO 模式向下兼容普通模式，通过把 I2Cx_CR{FIFO_EN}清 0 来禁用 FIFO 模式。

发送 FIFO 的宽度为 12 位，4 个控制位，8 个数据位，深度为 8。4 个控制位为 I2Cx_CR[3:0]，这是发送每个字节所必须的控制位。当一个字节被传输之后，新的字节从 TX FIFO 拷贝到位移寄存器，控制位被拷贝到 I2Cx_CR[3:0]。这个字节现在被传输，持续循环直到 Stop 条件产生。

接收 FIFO 为宽度为 8 位，用于保存接收到的数据，深度为 16。每个字节的控制位与一个空数据被保存到 TX FIFO 的相应位置。当接收 FIFO 半满，会产生 FIFO 半满中断或者 DMA 请求，将 FIFO 中的数据读出。

为支持 FIFO 功能，同时完整利用 FIFO 的容量，需要配置以下状态和控制位:

(1) I2Cx_CR{FIFO_EN} 置 1 使能 FIFO 模式

(2) I2Cx_CR{TXBEGIN} 置 1 开始事务

(3) 使能 I2Cx_CR[31:27] 的 FIFO 相关中断位，可通过 I2Cx_SR[31:27] 查询相应中断状态

(4) 每次事务完成之后（Stop 条件发送完）触发 TXDONE 中断

(5) I2Cx_CR{DMA_EN} 用来使能/禁用 DMA 模式

在 DMA 模式，I2Cx_CR[31:28] 相关的 FIFO 中断必须禁用，同时 I2Cx_CR{DMA_EN} 置 1。这样，所有 DMA 请求被发送到 DMA，而不是 CPU。I2Cx_CR{TXDONE_IE} 在 FIFO 模式和 DMA 模式都需要置 1，用来通知 CPU 事务的结束。

15.9 从机模式

表 15-3 从机事务

从机动作	从机操作	定义
从机接收（默认模式）	仅从机接收	● I2C 监视所有从机地址事务 ● I2Cx_CR{IUE} 必须置 1 ● I2C 监视总线上的 Start 条件。若检测到 Start 条件，接口读取前 8 位数据，并把前 7 位与自身从机地址做比较，若匹配则响应 ACK ● 若首字节的第 8 位（R/nW）为低，那么 I2C 保持在从机接收模式，并把 I2Cx_SR{SAD} 清 0。若 R/nW 为高，I2C 切换到从机发送模式，并把 I2Cx_SR{SAD} 置 1
设置从机地址检测位	从机接收 从机发送	● 用来指示接口检测到匹配的 I2C 寻址 ● 若使能 I2Cx_CR{SADIE}，在匹配的从机地址被接收和 ACK 响应之后，中断产生，I2Cx_SR{SAD} 置 1
从 I2Cx_DBR 读取 1 个字节	仅从机接收	● 8 位数据从总线上读取到位移寄存器，在整个字节被接收完成和 ACK/NAK 完成之后，位移寄存器内的数据被搬到 I2Cx_DBR 寄存器 ● 当 I2Cx_SR{IRF} 置 1，且 I2Cx_CR{TB} 清 0，若使能 I2Cx_CR{DRFIE}，I2Cx_DBR 接收满中断产生 ● 软件从 I2Cx_DBR 读取数据，并根据需要配置 I2Cx_CR{ACKNAK}，把 I2Cx_CR{TB} 置 1，这个操作使从机退出等待模式，接续接收主机的数据
响应 ACK 到发送主机	仅从机接收	● 作为接收从机，I2C 在 SCL 为高的时候将 SDA 线拉低产生 ACK ● ACK/NAK 由 I2Cx_CR{ACKNAK} 控制
写 1 个字节到 I2Cx_DBR	仅从机发送	● I2Cx_SR{ITE} 置 1，I2Cx_CR{TB} 清 0，若使能 I2Cx_CR{ITEIE} 中断，I2Cx_DBR 发送空中断产生 ● 软件把数据写入 I2Cx_DBR 寄存器，然后把 I2Cx_CR{TB} 置 1 开始数据的发送
等待接收主机的 ACK	仅从机发送	● 作为发送从机，I2C 释放 SDA 线等待接收主机拉低响应 ACK

15.10 时钟复位

每个 I2C 接口都有独立的 APB 总线时钟和独立的 APB 总线复位。

复位前软件必须保证 I2Cx_CR{UE}为 0，并且复位后保证总线在空闲状态（I2Cx_SR{IBB}为 0）。Reset 时，除 I2Cx_SAR 寄存器之外的所有寄存器均恢复到默认的复位状态，I2Cx_SAR 不受复位影响。

复位操作顺序:

1. 将 I2Cx_CR{UR} 置 1，并把 I2Cx_CR 寄存器的其余位清 0
2. 将 I2Cx_SR 寄存器清 0
3. 将 I2Cx_CR{UR} 清 0

15.11 中断请求

通过 I2Cx_CR 配置中断使能，查询 I2Cx_SR 相应位可以获取中断状态。

15.12 DMA 请求

通过 I2Cx_CR{DMA_EN} 使能 DMA，支持发送和接收。

15.13 I2C 相关寄存器描述

I2C0 基地址：0x40007000

I2C1 基地址：0x40014000

I2C2 基地址：0x40015000

表 15-4 I2C 寄存器列表

寄存器	偏移量	描述
I2Cx_CR	0x00	控制寄存器
I2Cx_SR	0x04	状态寄存器
I2Cx_SAR	0x08	从地址寄存器
I2Cx_DBR	0x0C	数据 Buffer 寄存器
I2Cx_LCR	0x10	加载计数寄存器
I2Cx_WCR	0x14	等待计数寄存器
I2Cx_RST_CYCL	0x18	复位周期寄存器
I2Cx_BMR	0x1C	总线监视寄存器
I2Cx_WFIF0	0x20	发送 FIFO 寄存器
I2Cx_WFIFO_WPTR	0x24	发送 FIFO 写指针寄存器
I2Cx_WFIFO_RPTR	0x28	发送 FIFO 读指针寄存器
I2Cx_RFIFO	0x2C	接收 FIFO 寄存器
I2Cx_RFIFO_WPTR	0x30	接收 FIFO 写指针寄存器
I2Cx_RFIFO_RPTR	0x34	接收 FIFO 读指针寄存器
I2Cx_RESV[2]	0x38	4 x 2 字节保留
I2Cx_WFIFO_STATUS	0x40	写 FIFO 状态寄存器
I2Cx_RFIFO_STATUS	0x44	读 FIFO 状态寄存器

15.13.1 I2Cx_CR(x=0、1、2)

偏移量：0x00

复位值：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	IDRFIE	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	IUE	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

位 31 RXOV_IE：接收 FIFO overrun 中断使能。

- 1：使能接收 FIFO overrun 中断
- 0：禁用接收 FIFO overrun 中断

位 30 RXF_IE：接收 FIFO 满中断使能。

- 1：使能接收 FIFO 满中断
- 0：禁用接收 FIFO 满中断

位 29 RXHF_IE：接收 FIFO 半满中断使能。

- 1：使能接收 FIFO 半满中断
- 0：禁用接收 FIFO 半满中断

位 28 TXE_IE：发送 FIFO 空中断使能。

- 1：使能发送 FIFO 空中断
- 0：禁用发送 FIFO 空中断

位 27 TXDONE_IE：事务完成中断使能。

- 1：使能事务完成中断
- 0：禁用事务完成中断

位 26 MSDE：主机停止检测使能。

- 1：使能主机停止检测功能
- 0：禁用主机停止检测功能

位 25 MSDIE：主机停止检测中断使能。

- 1：使能主机停止检测中断
- 0：禁用主机停止检测中断

位 24 SSDIE：从机停止检测中断使能。

- 1：使能从机停止检测中断

- 0：禁用从机停止检测中断

位 23 SADIE：从机地址检测中断使能。

- 1：使能从机地址检测中断
- 0：禁用从机地址检测中断

位 22 BEIE：总线错误中断使能。

- 1：使能总线错误中断
- 0：禁用总线错误中断

位 21 RESERVED：保留，不可更改。

位 20 IDRFIE：I2Cx_DBR 接收满中断使能。

- 1：使能 I2Cx_DBR 接收满中断
- 0：禁用 I2Cx_DBR 接收满中断

位 19 ITEIE：I2Cx_DBR 发送空中断使能。

- 1：使能 I2Cx_DBR 发送空中断
- 0：禁用 I2Cx_DBR 发送空中断

位 18 ALDIE：仲裁丢失检测中断使能。

- 1：使能仲裁丢失检测中断
- 0：禁用仲裁丢失检测中断

位 17-15 RESERVED：保留，不可更改。

位 14 IUE：I2C 接口单元使能。

- 1：使能 I2C 接口单元，默认为从机接收模式
- 0：禁用 I2C 接口单元

软件必须保证在使能 I2C 接口单元前总线为空闲状态，并且在置 1 或者清 0 该位前使能 I2C 内部时钟。

位 13 SCLE：SCL 使能。

- 1：使能主模式 I2C 时钟输出
- 0：禁用 SCL 线

位 12 MA：主机中止。

用于 Master 模式产生 Stop 条件。

- 1：产生 Stop 条件无需发送数据
- 0：STOP 为 1 产生 Stop 条件

在主机发送模式，当一个数据字节发送完成，TB 被清除，I2Cx_SR{ITE} 被置 1，若没有更多的数据需要被发送，可将 MA 置 1 产生 Stop 条件释放总线。在主机接收模式，当 STOP 为 0，发送 NAK 后，且没有发送 Repeat Start 条件，可将 MA 置 1 产生 Stop 条件释放总线。TB 必须保持为 0。

位 11 IBRR：总线复位请求。

- 1：总线复位，该位自动清 0
- 0：无效

位 10 UR：单元复位。

- 1：I2C 单元复位
- 0：无效

位 9-8 MODE：主机总线时钟模式。

- 01：快速模式，400Kbps
- 00：标准模式，100Kbps

位 7 DMA_EN：DMA 使能。

- 1：使能 DMA 请求
- 0：禁用 DMA 请求

位 6 RESERVED：保留，不可更改。

位 5 FIFOEN：FIFO 模式使能。

- 1：使能 FIFO 模式
- 0：禁用 FIFO 模式

位 4 TXBEGIN：事务开始。

- 1：新的事务开始
- 0：无事务开始

该位在产生 Stop 条件后被硬件清 0，软件需要在开始新的事务时置 1。

位 3 TB：传输字节，用来在总线上发送或者接收一个字节。

- 1：发送或者接收一个字节
- 0：一个字节收发完成后被清 0

I2C 单元会监视这个位来确定该字节是否收发完成。从机模式，在一个字节包括 ACK 收发完成后，I2C 会一直将 SCL 拉低直到 TB 被置 1。

位 2 ACKNAK：主机接收模式 ACK/NAK 控制位。

- 1：接收完成一个字节后发送 NAK
- 0：接收完成一个字节后发送 ACK

从机模式，从地址匹配或者接收完成时，I2C 单元自动发送一个 ACK，不论 ACKNAK 是否置 1。

位 1 STOP：产生 Stop 条件。

- 1：产生 Stop 条件
- 0：不产生 Stop 条件

用来在主机模式传输完下一个字节后在总线上产生 Stop 条件。在主机接收模式，ACKNAK 必须与 STOP 位同时置 1。

位 0 START：产生 Start 条件。

- 1：产生 Start 条件
- 0：不产生 Start 条件

用来在主机模式在总线上产生 Start 条件。

15.13.2 I2Cx_SR(x=0、1、2)

偏移量：0x04

复位值：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	IDRF	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	

位 31 RXOV：接收 FIFO overrun 标志。

- 1：接收 FIFO 发生 overrun，写 1 清 0
- 0：接收 FIFO 未发生 overrun 中断

位 30 RXF：接收 FIFO 满标志。

- 1：接收 FIFO 满，写 1 清 0
- 0：接收 FIFO 未满

位 29 RXHF：接收 FIFO 半满标志。

- 1：接收 FIFO 半满，写 1 清 0
- 0：接收 FIFO 未半满

位 28 TXE：发送 FIFO 空标志。

- 1：发送 FIFO 空，写 1 清 0
- 0：发送 FIFO 不为空

位 27 TXDONE：事务完成标志，FIFO 模式使用。

- 1：事务完成，写 1 清 0
- 0：事务未完成

位 26 MSD：主机停止检测标志（主模式有效）。

- 1：检测到主机停止，写 1 清 0
- 0：未检测到主机停止

位 25 RESERVED：保留，不可更改。

位 24 SSDIE：从机停止检测标志。

- 1：检测到从机停止，写 1 清 0
- 0：未检测到从机停止

位 23 SAD：从机地址检测标志。

- 1：检测到匹配的从机地址，写 1 清 0
- 0：无匹配的从机地址被检测到

位 22 BED：总线错误标志。

- 1：检测到总线错误，写 1 清 0
- 0：未检测到总线错误

两种情况下回产生该标志，主机发送一个字节后未收到 ACK，或者从机接收产生一个 NAK 脉冲。

位 21 RESERVED：保留，不可更改。

位 20 IDRF：I2Cx_DBR 接收满标志。

- 1：I2Cx_DBR 收到新的一个数据字节，写 1 清 0
- 0：I2Cx_DBR 没有收到新的数据字节或者 I2C 总线处于空闲状态

位 19 ITE：I2Cx_DBR 发送空。

- 1：总线发送完成一个数据字节，写 1 清 0
- 0：数据仍在发送中

位 18 ALD：仲裁丢失标志，多主机场景使用。

- 1：丢失仲裁，写 1 清 0
- 0：获得仲裁，或者未发生仲裁

位 17 RESERVED：保留，不可更改。

位 16 IBB：总线忙标志。

- 1：总线忙但是未被 I2C 接口使用
- 0：总线空闲或者总线正在被 I2C 接口使用

位 15 UB：I2C 接口单元忙标志。

- 1：I2C 接口单元忙
- 0：I2C 接口单元空闲

位 14 ACKNAK：ACK/NAK 状态标志。

- 1：收到或发送完一个 NAK
- 0：收到或发送完一个 ACK

在从机发送模式，该位用于确定被发送的字节是否是最后一个。该位在每个字节的 ACK/NAK 信息被收到后都会更新。

位 13-0 RESERVED：保留，不可更改。

15.13.3 I2Cx_SAR(x=0、1、2)

偏移量：0x08

复位值：0x00000000

31-7	6-0
RESERVED	SLAVE_ADDRESS
r-0h	rw-0h

位 31-7 RESERVED：保留，不可更改。

位 6-0 SLAVE_ADDRESS：从地址，从机模式使用。

15.13.4 I2Cx_DBR(x=0、1、2)

偏移量：0x0C

复位值：0x00000000

31-8	7-0
RESERVED	DATA_BUFFER
r-0h	rw-0h

位 31-8 RESERVED：保留，不可更改。

位 7-0 DATA_BUFFER：收发数据 Buffer。

15.13.5 I2Cx_LCR(x=0、1、2)

偏移量：0x10

复位值：0x18183a7e

31-18	17-9	8-0
RESERVED	FLV	SLV
r-1818h	rw-1dh	rw-7eh

位 31-18 RESERVED：保留，不可更改。

位 17-9 FLV：主机快速时钟模式相位减幅器装载值。

位 8-0 SLV：主机标准时钟模式相位减幅器装载值。

15.13.6 I2Cx_WCR(x=0、1、2)

偏移量：0x14

复位值：0x0000143a

31-5	4-0
RESERVED	COUNT
r-a1h	rw-1ah

位 31-5 RESERVED：保留，不可更改。

位 4-0 COUNT：快速与标准时钟模式 setup 与 hold 次数计数值。

15.13.7 I2Cx_RST_CYCL(x=0、1、2)

偏移量：0x18

复位值：0x00000000

31-4	3-0
RESERVED	RST_CYC
r-0h	rw-0h

位 31-4 RESERVED：保留，不可更改。

位 3-0 RST_CYC：总线复位 SCL 周期计数。

15.13.8 I2Cx_BMR(x=0、1、2)

偏移量：0x1C

复位值：0x00000003

31-2	1	0
RESERVED	SCL	SDA
r-0h	r-1h	r-1h

位 31-2 RESERVED：保留，不可更改。

位 1 SCL：SCL 引脚状态。

位 0 SDA：SDA 引脚状态。

15.13.9 I2Cx_WFIF0(x=0、1、2)

偏移量：0x20

复位值：0x00000000

31-12	11-8	7-0
RESERVED	CONTROL	DATA
r-0h	w-0h	w-0h

位 31-12 RESERVED：保留，不可更改。

位 11-8 CONTROL：收发数据控制位。

位 7-0 DATA：写事务发送数据和读事务空数据。

15.13.10 I2Cx_WFIFO_WPTR(x=0、1、2)

偏移量：0x24

复位值：0x00000000

31-4	3-0
RESERVED	DATA
r-0h	rw-0h

位 31-4 RESERVED：保留，不可更改。

位 3-0 DATA：发送 FIFO 软件写入位置指针。

15.13.11 I2Cx_WFIFO_RPTR(x=0、1、2)

偏移量：0x28

复位值：0x00000000

31-4	3-0
RESERVED	DATA
r-0h	rw-0h

位 31-4 RESERVED：保留，不可更改。

位 3-0 DATA：发送 FIFO 硬件读取位置指针。

15.13.12 I2Cx_RFIFO(x=0、1、2)

偏移量：0x2C

复位值：0x00000000

31-8	7-0
RESERVED	DATA
r-0h	r-0h

位 31-8 RESERVED：保留，不可更改。

位 7-0 DATA：读事务接收数据。

15.13.13 I2Cx_RFIFO_WPTR(x=0、1、2)

偏移量：0x30

复位值：0x00000000

31-4	3-0
RESERVED	DATA
r-0h	r-0h

位 31-4 RESERVED：保留，不可更改。

位 3-0 DATA：接收 FIFO 硬件写入位置指针。

15.13.14 I2Cx_RFIFO_RPTR(x=0、1、2)

偏移量：0x34

复位值：0x00000000

31-4	3-0
RESERVED	DATA
r-0h	r-0h

位 31-4 RESERVED：保留，不可更改。

位 3-0 DATA：接收 FIFO 软件读取位置指针。

15.13.15 I2Cx_WFIFO_STATUS(x=0、1、2)

偏移量：0x40

复位值：0x00000000

31-16	15-9	8-1	0
RESERVED	WFIFO_SIZE	WFIFO_EMPTY	WFIFO_FULL
r-0h	r-0h	r-0h	r-0h

位 31-6 RESERVED：保留，不可更改。

位 5-2 WFIFO_SIZE：发送 FIFO 空间。

位 1 WFIFO_EMPTY：发送 FIFO 空。

位 0 WFIFO_FULL：发送 FIFO 满。

15.13.16 I2Cx_RFIFO_STATUS(x=0、1、2)

偏移量：0x44

复位值：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

位 31-8 RESERVED：保留，不可更改。

位 7-4 RFIFO_SIZE：接收 FIFO 空间。

位 3 RFIFO_EMPTY：接收 FIFO 空。

位 2 RFIFO_FULL：接收 FIFO 满。

位 1 RFIFO_HALFFULL：接收 FIFO 半满。

位 0 RFIFO_OVERRUN：接收 FIFO overrun。

16. ADC

16.1 简介

12 位模数转换器（Analog to Digital Converter），支持 8 个外部通道，7 个内部通道，内部通道可采集 VBAT/3，最高支持 1M 采样率。支持单端和差分两种模式，单端量程 0.1V~1.1V，差分量程 -1.0~1.0V。可配置 16 个采样序列，支持连续、单次、非连续采样方式。支持软件触发和硬件触发，触发源可配。支持 DMA 请求和中断请求。

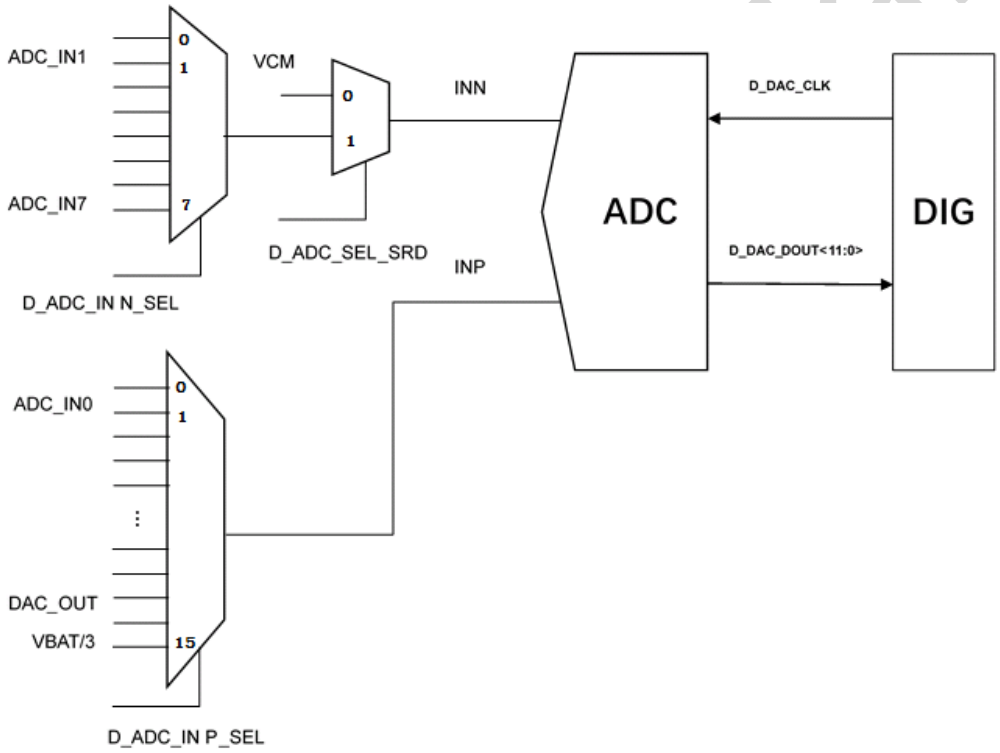

图 16-1 ADC 框图

16.2 输入模式

支持配置为单端与差分模式。外部通道支持单端与差分模式，内部通道只支持单端模式。差分为固定组合，不支持随意配对，其中 0/1 通道为一组，2/3 通道为一组，4/5 通道为一组，6/7 通道为一组。单端和差分仅在采样阶段控制不同，保持阶段没有区别，最后的数据中差分输入最高位为符号位（11bit 数据位，1 个符号位），单端输入为 12bit 数据位，没有符号位。通过采样通道差分/单端选择寄存器 ADC_DIFFSEL 配置输入模式。

16.3 采样通道

- **外部通道：**8 个，单端模式各通道独立，差分模式则每两个通道为一组，不可拆分。
- **内部通道：**7 个，包括 DAC 输出、内部 VRef、VDD/3（电池电量）、Vts（内部温度传感器）、内部测试专用。内部通道不支持差分模式。

表 16-1 ADC 采样通道

采样通道号	采样内容	备注
1	ADC_PAD_IN<0>	gpio11
2	ADC_PAD_IN<1>	gpio08
3	ADC_PAD_IN<2>	gpio05
4	ADC_PAD_IN<3>	gpio04
5	ADC_PAD_IN<4>	gpio50
6	ADC_PAD_IN<5>	gpio49
7	ADC_PAD_IN<6>	gpio48
8	ADC_PAD_IN<7>	gpio47
9	OPA0_ADC_OUT	
10	OPA1_ADC_OUT	
11	OPA2_ADC_OUT	
12	DCTEST_OUT	
13	TD_OUT_TEST	
14	DAC_CORE_AOUT	
15	VBAT31	

VBAT31 需要通过模拟部分 RESV1 寄存器的 D_VBAT_DIV3_EN 置位使能 VBAT/3 分压。这个通道名义为 VBAT 的 1/3 分压，精确值为 1/3.06。

16.4 触发方式

- **软件方式：**ADC_START 上升沿决定转换立即开始。
- **硬件方式：**支持 Timer 与 IO 触发，触发源可选择，16 个，可配置触发电平。

通过配置寄存器 ADC_CFGR{TRIG_SEL} 选择触发方式以及 ADC_CFGR{EXT_TRIG_SEL} 触发源。

16.5 低功耗运行

ADC_DR 数据被读走或者 EOC 标志被清除后才能接收新的触发请求，可以防止 overrun，但可能 bypass 触发请求。

16.6 溢出控制

控制 overrun 发生时，数据寄存器采样新数据或者保持。

16.7 采样模式

通过 ADC_CFGR{CONV_MODE} 配置采样模式：

- **采样序列：**支持采样序列配置，采样序列最多 16 个通道，单端和差分通道都可以配置。差分模式，采样序列仅配置 P 端即可。采样通道可以重复配置相同通道以决定每次序列多次采样该通道。通过通道采样序列控制寄存器 ADC_SEQR0 和 ADC_SEQR1 寄存器配置采样序列，每 4 位配置 1 个采样通道，两个 32 位寄存器共 64 位，最多可以配置 16 个采样通道。
- **连续采样：**一旦触发有效，则开始连续地转化选定的输入序列，每轮循环完成后自动开始新一轮循环，直到软件配置 stop。
- **单次采样：**每次触发执行一次采样序列循环，采样完成自动结束。
- **非连续采样：**序列中的每一次 ADC 转化都需要硬件或软件触发，如果一个序列完后，再次触发又从该序列的开头开始；而连续和单次模式，每次触发都会完成一个完整序列。

16.8 参考电压

模拟部分 RST 寄存器的 D_ADC_SEL_VREF 位配置参考电压，清 0 配置为外部参考电压，置 1 为内部参考电压。

- **内部参考电压：**VRef，1.2V
- **外部参考电压：**VREFP/3，VREFP 不可超过 3.6V，48PIN 内部 VREFP 与 VDDA 有连接

16.9 数据 Buffer

1 个 12bit 的数据 buffer，差分模式下最高位为符号位。

ADC 编码	差分模式意义	单端模式意义
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) *为校准前从数据 buffer 读到的值。*

差分模式的量程为 -1.0~1.0V，单端模式的量程为 0.1~1.1V。为了纠正 ADC 模拟电路上的误差，ASR6601 在出厂前都会做一个校准，校准数据 Offet 和 Gain 存储在 Flash 中，用户需要将从 ADC_DR 读到的数据做一个转换才能得到最终的 AD 值，公式如下:

$$V = (Vout – Offset) / Gain$$

其中 Vout 为从数据 buffer 中读到的值。

16.10 DMA 请求

采用 Req 与 Clear 方式，数据 buf 为 1 个 12bit，因此 Buffer 数据满则产生请求。通过 ADC_CFGR{DMA_EN} 配置。

16.11 中断请求

采用电平方式，包括单次转换完成 EOC，序列转换完成 EOS，溢出 OVERRUN。通过中断使能寄存器 ADC_IER 使能中断，通过中断状态寄存器 ADC_ISR 查询中断。

16.12 低功耗工作与唤醒

支持 Sleep 模式中断唤醒与事件唤醒。

16.13 时钟复位

总线复位和工作时钟复位独立，支持 APB 总线时钟，可配置内部分频，接口时钟来源，包括 sys_clk、apb_x_pclk、pll_clk、rco48m_clk。两者异步，部分路径通过软件流程保证时序，不做同步，其余路径均做同步处理。

16.14 ADC 相关寄存器描述

基地址：0x40017000

表 16-2 ADC 寄存器列表

寄存器	偏移量	描述
ADC_CR	0x00	控制寄存器
ADC_CFGR	0x04	配置寄存器
ADC_SEQR0	0x08	通道采样序列控制寄存器 0
ADC_SEQR1	0x0C	通道采样序列控制寄存器 1
ADC_DIFFSEL	0x10	采样通道差分/单端选择寄存器
ADC_ISR	0x14	中断和状态寄存器
ADC_IER	0x18	中断使能寄存器
ADC_DR	0x1C	数据寄存器

16.14.1 ADC_CR

偏移量：0x00

复位值：0x00000000

31-4	3	2	1	0
RESERVED	STOP	START	DIS	EN
r-0h	rw-0h	rw-0h	w-0h	rw-0h

位 31-4 RESERVED：保留，不可更改。

位 3 STOP：ADC 转换停止控制。

- 写 0：无效
- 写 1：控制 ADC 转换强制停止，读该位为 1 表示 STOP 的关闭动作正在执行

注意：

1. *软件对该位写 1 来强制停止当前转换，当前转换的数据会被丢弃，扫描序列也会回到初始状态；该位硬件自动清零。*
2. *软件查询到 START 关闭后等待 3 拍 ADCCLK 后才能再次配置 START；或等待 1 拍 ADC_CFGR{CLK_DIV} 后才能配置 DIS 关闭 ADC 功能。*
3. *仅在 START=1 且 STOP=0 时对该位的写 1 动作才有效。*
4. *配置 STOP 关闭 ADC 转换前，建议先将触发源关闭，或触发电平处于无效状态。*

位 2 START：ADC 转换开启控制。

- 写 0：无效
- 写 1：开启 ADC 转换，读该位为 1 表示 ADC 正在转换

该位软件置 1 开启 ADC 转换功能，并根据 ADC_CFGR{TRIG_SEL} 的配置决定 ADC 转换马上开始（软件触发模式），或者等待硬件触发事件才开始；仅在 EN=1 且 DIS=0 时才能够配置 ADC 转换开启。

该位由硬件自动清零，分为以下几种情况：

1. 单次转换模式下，且选择软件触发模式时（ADC_CFGR{TRIG_SEL}=00），当 ADC_ISR{EOS} 标志置高时清除 START 位。
2. 非连续模式下，且选择软件触发模式时，当 ADC_ISR{EOC} 标志置高时清除 START。
3. 在任何情况下，执行 STOP 命令清除 START（START 和 STOP 同时清除）。

位 1 DIS：ADC 功能除能控制。

- 写 0：无效
- 写 1：控制 ADC 除能

该位仅在 EN=1 且 START=0（没有转换正在执行）时配置写 1 才有效。

位 0 EN：ADC 功能使能控制。

- 写 0：无效
- 写 1：控制 ADC 使能，读该位为 1 表示 ADC 使能

软件配置该使能后，表示 ADC 可以开始触发转换；该位仅在 ADC_CR 寄存器全为 0 时才能写 1；读该位可以反映 ADC 功能使能状态。软件应当在初始化 ADC 模拟电路后延时至少 100us 以等待电路稳定，然后再使能 ADC 功能。

16.14.2 ADC_CFGR

偏移量：0x04

复位值：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

位 31-23 RESERVED：保留，不可更改。

位 22 WAIT_MODE：等待转换模式控制。

- 0：等待转换不使能
- 1：等待转换使能

等待转换模式，即 ADC_DR 数据被读走或者 ADC_ISR{EOC} 标志被清除后才能接收新的触发请求，可以防止 overrun，但可能 bypass 触发请求。

仅在 ADC_CR{START} 为 0 时该位才能够配置。

位 21-20 CONV_MODE：ADC 转换模式选择。

- 00：单次转换模式
- 01：连续转换模式
- 1x：非连续转换模式

仅在 ADC_CR{START} 为 0 时该位才能够配置。

说明：

1. *单次转换模式，每次触发完成 ADC_SEQR0/1 整个采样序列后即停止，等待下一次触发。*
2. *连续转换模式，触发开始转换后，一直按照 ADC_SEQR0/1 的采样序列循环采样，直到配置 ADC_CR{STOP}。*
3. *非连续转换模式，每次触发完成一次 ADC 采样（按照 ADC_SEQR0/1 的采样序列）即停止，等待下一次触发。*

位 19 OVERRUN_MODE：控制 overrun 时的数据操作。

- 0：当 overrun 发生时，原 ADC_DR 中的数据被保留
- 1：当 overrun 发生时，ADC_DR 被新的转换数据覆盖

仅在 ADC_CR{START} 为 0 时该位才能够配置。

位 18-17 TRIG_SEL：触发源模式和触发极性选择。

- 00：软件触发，ADC_CR{START}上升沿决定转换立即开始
- 01：硬件触发，上升沿触发
- 10：硬件触发，下降沿触发
- 11：硬件触发，上升沿和下降沿均可触发

仅在 ADC_CR{START}为 0 时该位才能够配置。

使用硬件触发时，配置 ADC_CR{START} 后，需要等待 3 拍 ADCCLK 后才可以接收触发信号。

位 16-13 EXT_TRIG_SEL：ADC 转换开始外部触发源选择。

- 0000~0100：Reverse
- 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：Reverse

说明：

1. *仅在 ADC_CR{START} 为 0 时该位才能够配置。*
2. *若使用 GPTIMx 的 TRGO 信号作为触发，则 GPTIMx_CR2{MMS} 仅可以配置为 0x100(OC0REF), 0x101(OC1REF), 0x110(OC2REF), 0x111(OC3REF)。对于 GPTIM2 和 GPTIM3，仅可以配置为 0x100(OC0REF), 0x101(OC1REF)。*
3. *若要实现定时触发或周期触发，需配置所选通道为输出模式，选择相应的输出模式，并根据所需时间配置相应的 GPTIMx_ARR 和 GPTIMx_CCRx。*

位 12 DMA_EN：DMA 功能使能。

- 0：DMA 不使能
- 1：DMA 使能

位 11-0 CLK_DIV：ADCCLK 的时钟预分频选择。

- 000：不分频
- 001：不分频
- n：ADC_IP_CLK=ADCCLK/n，占空比 50%。

注意：

1. *仅在 ADC_CR 均为 0 时，该位配置才有效；ADCCLK 的时钟源选择在 RCC_CR2 中配置。*
2. *时钟分频和时钟源选择需要考虑数据的读出速度，ADC 每 16 拍完成一次采样，若该 ADC 时钟配置过快，软件或 DMA 不能及时读走，则可能造成溢出。*

16.14.3 ADC_SEQR0

偏移量：0x08

复位值：0x00000000

注意：*仅在 ADC_CR{START} 和 ADC_CR{EN} 为 0 时，才能够配置该寄存器。*

31-28	27-24	23-20	19-16	15-12	11-8	7-4	3-0
SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 31-28 SEL7：ADC 采样序列第 7 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

位 27-24 SEL6：ADC 采样序列第 6 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

位 23-20 SEL5：ADC 采样序列第 5 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

位 19-16 SEL4：ADC 采样序列第 4 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

位 15-12 SEL3：ADC 采样序列第 3 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

位 11-8 SEL2：ADC 采样序列第 2 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

位 7-4 SEL1：ADC 采样序列第 1 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

位 3-0 SEL0：ADC 采样序列第 0 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

16.14.4 ADC_SEQR1

偏移量：0x0C

复位值：0x00000000

注意：*仅在 ADC_CR{START} 和 ADC_CR{EN} 为 0 时，才能够配置该寄存器。*

31-28	27-24	23-20	19-16	15-12	11-8	7-4	3-0
SEL15	SEL14	SEL13	SEL12	SEL11	SEL10	SEL9	SEL8
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 31-28 SEL15：ADC 采样序列第 15 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

位 27-24 SEL14：ADC 采样序列第 14 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

位 23-20 SEL13：ADC 采样序列第 13 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

位 19-16 SEL12：ADC 采样序列第 12 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

位 15-12 SEL11：ADC 采样序列第 11 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

位 11-8 SEL10：ADC 采样序列第 10 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

位 7-4 SEL9：ADC 采样序列第 9 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

位 3-0 SEL8：ADC 采样序列第 8 个通道配置。

配置值为采样通道号 1~15，若发现配置为 0，则当前序列结束；若出现 SELx 的通道号相同，则执行多次重复采样。

差分输入只配置正端的通道号即可，负端通道号硬件根据 ADC_DIFFSEL 自动选择。

16.14.5 ADC_DIFFSEL

偏移量：0x10

复位值：0x00000000

注意：*仅在 ADC_CR{START}和 ADC_CR{EN}为 0 时，该寄存器才能够配置。*

31-16	15-9	8-1	0
RESERVED	SEL1	SEL0	RESERVED
r-0h	r-0h	rw-0h	r-0h

位 31-16 RESERVED：保留，不可更改。

位 15-9 SEL1：ADC 通道 9~15 为内部通道。

只支持单端模式，不支持差分模式，这些位只读。

位 8-1 SEL0：ADC 通道 1~8 差分/单端模式选择。

每一位控制一个通道：

- 0：通道 x 为单端模式
- 1：通道 x 为差分模式

差分通道仅支持两个外部相邻通道之间，如通道 2 和通道 3，则该寄存器相应的两个控制位要配置为 1。

位 0 RESERVED：保留，不可更改。

16.14.6 ADC_ISR

偏移量：0x14

复位值：0x00000000

说明：*软件使能 ADC_CR{START} 前建议先清除该寄存器。*

31-3	2	1	0
RESERVED	OVERRUN	EOS	EOC
r-0h	rw1c-0h	rw1c-0h	rw1c-0h

位 31-3 RESERVED：保留，不可更改。

位 2 OVERRUN：ADC 转换 overrun 标志。

- 1：发生 overrun
- 0：没有 overrun 发生

当 EOC 标志为高时（ADC_DR 数据未取走或未配置软件写 1 清零），新的转换完成，该位由硬件置 1。

软件写 1 清零。

位 1 EOS：ADC 通道序列采样完成标志。

- 1：通道序列转换完成
- 0：通道序列转换未完成

当 ADC_SEQR0/1 中的整个通道序列完成一轮转换后，该位由硬件置 1。

软件写 1 清零。

位 0 EOC：ADC 转换完成标志。

- 1：通道转换完成;
- 0：通道转换未完成

某个通道 ADC 转换结束，新的转换数据写入到 ADC_DR 中后该标志由硬件置高。

软件写 1 清零或者读 ADC_DR 后清零。

16.14.7 ADC_IER

偏移量：0x18

复位值：0x00000000

31-3	2	1	0
RESERVED	OVERRUN_INT_EN	EOS_INT_EN	EOC_INT_EN
r-0h	rw-0h	rw-0h	rw-0h

位 31-3 RESERVED：保留，不可更改。

位 2 OVERRUN_INT_EN：ADC 转换 overrun 中断使能。

- 1：使能 overrun 中断
- 0：不使能 overrun 中断

位 1 EOS_INT_EN：ADC 通道序列采样完成中断使能。

- 1：使能通道序列采样完成中断
- 0：不使能通道序列采样完成中断

位 0 EOC_INT_EN：ADC 转换完成中断使能。

- 1：使能转换完成中断
- 0：不使能转换完成中断

16.14.8 ADC_DR

偏移量：0x1C

复位值：0x00000000

31-12	11-0
RESERVED	DATA
r-0h	r-0h

位 31-12 RESERVED：保留，不可更改。

位 11-0 DATA：ADC 转换数据。**在**差分模式下，第 11 位为符号位。

17. BSTIM

17.1 简介

BSTIMER（Base Timer）包含 16bits 计数器，支持自动重装载功能，且支持最多 16bits 可编程的分频计数器。有两个 BSTIMER，分别为 BSTIMER0 和 BSTIMER1。

17.2 功能

BSTIMER 包括如下功能：

1. 16bits 计数器，加法计数，支持自动重加载
2. 分频计数器
3. DMA 控制
4. 支持单脉冲
5. 支持主模式选择
6. 更新事件管理
7. Debug 模式控制
8. 中断信号产生

BSTIMER 的框图如下：

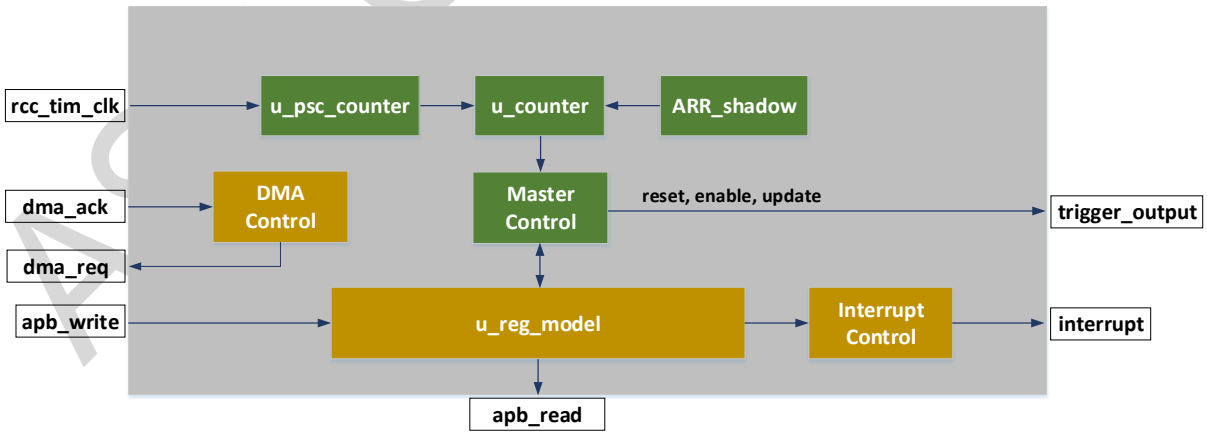

图 17-1 BSTIMER 框图

- **rcc_tim_clk**：BSTIMER 的接口时钟
- **dma_ack**：DMA 回复的 ACK
- **dma_req**：BSTIMER 向 DMA 的请求
- **apb_write**：APB 总线的写
- **apb_read**：APB 总线的读
- **trigger_output**：BSTIMER 的 TRGO 输出
- **interrupt**：BSTIMER 的中断

17.3 接口时钟

BSTIMER 接口时钟源为 PCLK，不能设置为其它时钟源。时钟使能和复位配置可以参考 RCC 章节。

17.4 计数器

计数器仅支持向上计数，计数到 ARR，然后继续计数，这样计数器的值会从 ARR 变为 0，同时状态标记位 UIF 置位，如果更新事件中断请求使能即 UIE 置位，则也会产生中断，此时表示一个计数周期完成。下个计数周期计数器继续从 0 开始计数，如此循环往复。

17.5 自动重加载

ARR 可软件配置寄存器 BSTIM_CR1 的 ARPE 是否使用启用影子寄存器，如果 ARPE=0，则禁用影子寄存器，软件写入的值直接同步更新到 ARR 供计数器使用，如果 ARPE=1，则软件写入的值不会立即生效，直到更新事件到来，才会将该值更新到 ARR 供计数器使用。

17.6 分频计数器

BSTIMER 支持 16-bit（1~65535）可编程分频，此功能通过分频计数器 BSTIM_PSC 实现。接口时钟作为分频计数器的时钟，寄存器 BSTIM_CR1 的 CEN 作为分频计数器的计数使能，当分频计数器计数到预先加载的分频值后，输出一个脉冲，作为下一级计数器的计数使能，然后分频计数器归零重新计数，依次类推。

分频计数器的分频值默认启用影子寄存器，即软件的写操作不会立即生效，而是直到更新事件（UG 事件置位、计数溢出）到来，才会将新的分频值写入影子寄存器，此时该分频值才正式生效。软件读操作读取的是写入的寄存器值，而不是影子寄存器，如果在更新事件到来前有多次写操作，则会覆盖之前写入的值。计数和分频波形如下:

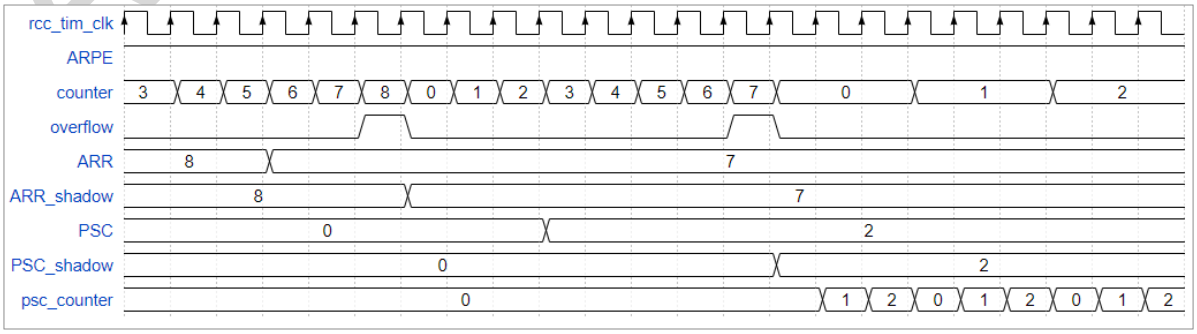

图 17-2 计数和分频波形

17.7 DMA 控制

BSTIMER 支持 DMA 功能，使能 DMA 功能后，其所有寄存器除 BSTIM_SR、BSTIM_EGR 外与 memory 之间可以相互传递数据，BSTIM_SR 只能被读取数据，BSTIM_EGR 只能被写入数据。通过寄存器 BSTIM_DIER 的 UDE 位使能 DMA，当有更新事件时则会产生 DMA 请求，DMA 返回的 ACK 信号会清除模块的 DMA 请求信号。

17.8 支持单脉冲

BSTIMER 支持单脉冲计数模式，通过置位寄存器 BSTIM_CR1 的 OPM 位使能该模式，在该模式下，当计数器计数到 ARR 值后会归零并停止计数（CEN 硬件自动清零），除非再次初始化才会重新计数，如下图所示：

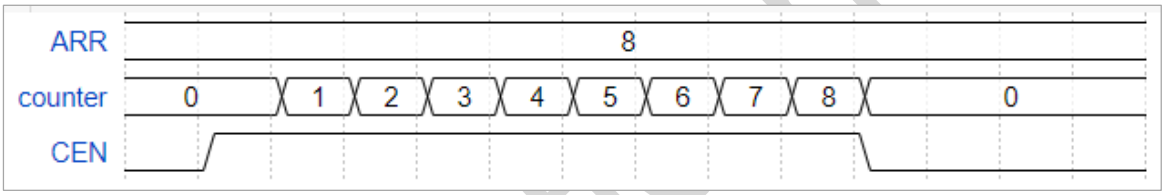

图 17-3 单脉冲波形

17.9 支持主模式选择

BSTIMER 可以与其他内部模块级联，并作为主机使用，通过产生触发输出信号（TRGO）来控制 DAC。TRGO 信号的来源可以由软件配置寄存器 BSTIM_CR2 的 MMS，具体如下：

- MMS=3’b000：复位模式，此时 UG 标志位将作为 TRGO 信号输出给外部从机。
- MMS=3’b001：使能模式，此时计数器的计数使能 CEN 将作为 TRGO 信号输出给外部从机。
- MMS=3’b010：更新模式，此时将更新事件将作为 TRGO 信号输出。
- MMS 的其它值为保留值。

17.10 更新事件管理

更新事件主要有以下事件源：

1. 计数器的溢出事件（overflow），即计数器的值从 ARR 变为 0。
2. UG 置位（软件置位），即配置寄存器 BSTIM_EGR 的 UG 位。

与更新事件管理相关的控制信号主要是寄存器 BSTIM_CR1 的 URS 和 UDIS，具体控制如下：

- 若 UDIS=0，URS=0，则 overflow、UG 置位会初始化计数器和分频计数器，如果启用影子寄存器，更新事件则会把写入的值更新到影子寄存器中（ARR 取决于 ARPE），UIF 会置位，如果使能了中断或 DMA，则会产生中断或 DMA 请求。
- 若 UDIS=0，URS=1，则 overflow、UG 置位会初始化计数器和分频计数器，如果启用影子寄存器，更新事件将会把写入的值更新到影子寄存器中（ARR 取决于 ARPE），UIF 只会在 overflow 情况下置位，如果使能了中断或 DMA，则会产生中断或 DMA 请求。
- 若 UDIS=1（忽略 URS），则只有 UG 置位仍会初始化计数器和分频计数器，但是影子寄存器不会被更新，且 UIF 不会置位，因此也不会产生相应中断或 DMA 请求。

17.11 Debug 模式控制

BSTIMER 可由软件配置 debug 下是否停止计数，通过 SYSCFG 的 CR2 寄存器来实现 BSTIMER0 和 BSTIMER1 的 DEBUG 模式计数控制，如果使能该功能，则进入系统 debug 模式时，BSTIMER 停止计数（计数器不会被初始化）。

17.12 中断信号

BSTIMER 的中断信号如下：

表 17-1 BSTIMER 中断信号

中断名称	描述
更新事件中断	计数器溢出、UG 置位均可以产生更新事件中断

上述中断的使能通过配置寄存器 BSTIM_DIER 的 UIE 位实现，中断状态可以通过寄存器 BSTIM_SR 获得。

17.13 BSTIMER 相关寄存器描述

BSTIMER0 基地址：0x4000C000

BSTIMER1 基地址：0x4001C000

表 17-2 BSTIMER 寄存器列表

寄存器	偏移量	描述
BSTIM_CR1	0x00	状态寄存器 1
BSTIM_CR2	0x04	状态寄存器 2
BSTIM_DIER	0x0c	DMA/中断使能寄存器
BSTIM_SR	0x10	状态寄存器
BSTIM_EGR	0x14	事件寄存器
BSTIM_CNT	0x24	计数器寄存器
BSTIM_PSC	0x28	计数器分频值
BSTIM_ARR	0x2c	计数器重装载值

17.13.1 BSTIM_CR1

偏移量：0x00

复位值：0x00000000

31-8	7	6-4	3	2	1	0
RESERVED	ARPE	RESERVED	OPM	URS	UDIS	CEN
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 31-8 RESERVED：保留。

位 7 ARPE：重装载影子寄存器使能。

- 0：BSTIM_ARR 影子寄存器除能
- 1：BSTIM_ARR 影子寄存器使能

位 6-4 RESERVED：保留。

位 3 OPM：单脉冲模式使能。

- 0：单脉冲模式除能
- 1：单脉冲模式使能，计数器在下一次更新事件停止计数

位 2 URS：更新事件源选择，该位仅影响中断和 DMA 标志位（UIF），不影响内部逻辑。

- 0：计数器溢出、UG 位置位，均可以置位 UIF
- 1：只有计数器溢出事件可以置位 UIF

位 1 UDIS：更新事件除能。

- 0：更新事件使能，可以产生更新事件。
- 1：更新事件除能，影子寄存器和 UIF 均不会被更新，但是此时计数器和分频计数器仍可以被 UG 置位事件初始化。

位 0 CEN：计数器使能，单脉冲模式下 CEN 由硬件清零。

- 0：计数器除能
- 1：计数器使能

17.13.2 BSTIM_CR2

偏移量：0x00

复位值：0x00000000

31-7	6-4	3-0
RESERVED	MMS	RESERVED
rw-0h	rw-0h	rw-0h

位 31-7 RESERVED：保留。

位 6-4 MMS：主模式选择，可以配置 TRGO 输出。

- 000：复位模式，UG 将作为 TRGO 信号输出
- 001：使能模式，CEN 将作为 TRGO 信号输出
- 010：更新模式，更新事件（内部信号）将作为 TRGO 信号输出
- 其它值：保留

位 3-0 RESERVED：保留。

17.13.3 BSTIM_DIER

偏移量：0x0c

复位值：0x00000000

31-9	8	7-1	0
RESERVED	UDE	RESERVED	UIE
rw-0h	rw-0h	rw-0h	rw-0h

位 31-9 RESERVED：保留。

位 8 UDE：更新事件 DMA 请求使能。

- 0：禁用更新事件 DMA 请求
- 1：使能更新事件 DMA 请求

位 7-1 RESERVED：保留。

位 0 UIE：更新事件中断请求使能。

- 0：禁用更新事件中断请求
- 1：使能更新事件中断请求

17.13.4 BSTIM_SR

偏移量：0x10

复位值：0x00000000

31-1	0
RESERVED	UIF
r-0h	r-0h

位 31-1 RESERVED：保留。

位 0 UIF：更新事件标志。

- 0：无事件
- 1：更新事件发生

17.13.5 BSTIM_EGR

偏移量：0x14

复位值：0x00000000

31-1	0
RESERVED	UG
w-0h	w-0h

位 31-1 RESERVED：保留。

位 0 UG：更新事件产生。

- 0：无动作
- 1：产生一次更新事件

17.13.6 BSTIM_CNT

偏移量：0x24

复位值：0x00000000

31-16	15-0
RESERVED	CNT
rw-0h	rw-0h

位 31-16 RESERVED：保留。

位 15-0 CNT：计数器计数值。

17.13.7 BSTIM_PSC

偏移量：0x28

复位值：0x00000000

31-16	15-0
RESERVED	PSC
rw-0h	rw-0h

位 31-16 RESERVED：保留。

位 15-0 PSC：时钟分频值为 PSC+1。

17.13.8 BSTIM_ARR

偏移量：0x2c

复位值：0x0000ffff

31-16	15-0
RESERVED	ARR
rw-0h	rw-ffffh

位 31-16 RESERVED：保留。

位 15-0 ARR：计数器重装载值。

18. RTC

18.1 简介

RTC（Real Time Clock）是一个独立的 BCD 计时器/计数器，有两个 32 位寄存器包含秒、分、小时（12 或 24 小时格式）、星期、日期（月的日期）、月和年，用二进制编码的十进制格式表示（BCD），还有一个 32 位寄存器表示亚秒值。RTC 支持在低功耗模式运行。

18.2 功能

RTC 包括如下功能：

1. 日历计数功能，采用 BCD 格式，支持秒、分、小时、日、星期、月、年、
2. 支持 ppm 调整，调整步长 0.5ppm，支持+/-1024 ppm 调整
3. 支持低功耗唤醒
4. tamper/wakeup IO 检测功能，支持有效电平选择，滤波拍数可配置
5. 周期计数功能，32 位计数器
6. 闹钟功能，支持两个闹钟，支持 Mask 选择与日历匹配
7. tamper/wakeup 报警清除 retention sram 功能
8. 内部信号 IO 输出，包括 alarm0 匹配脉冲，alarm1 匹配脉冲，周期计数配置脉冲，秒信号输出
9. 支持日历计数值读取
10. 支持 sub-second 计数值读取
11. 支持周期计数的计数值读取
12. 支持中断信号产生

18.3 接口时钟

RTC 接口时钟源有 XO32K 和 RCO32K，XO32K 的时钟精度一般比 RCO32K 要高。

时钟配置和选择可以参考 *RCC* 章节。

18.4 日历

RTC 日历时间和日期可以通过两种类型的寄存器访问获取，一种是异步的，一种是同步的，两种寄存器分别如下：

- **异步寄存器** RTC_SYNDATA 和 RTC_SYNDATA_H，RTC_SYNDATA 表示时分秒，RTC_SYNDATA_H 表示年月日星期。
- **同步寄存器** RTC_CALENDAR_R 和 RTC_CALENDAR_R_H，RTC_CALENDAR_R 表示时分秒，RTC_CALENDAR_R_H 表示年月日星期。

18.4.1 读取日历

只介绍通过同步寄存器读取日历，由于采用打拍同步，需要多次读到同一值，才可以使用，不能仅读取一次。读取顺序如下：

(1) 先读取 RTC_SUB_SECOND 寄存器的值，获取 SUBSECOND_COUNT 拍数。

(2) 连续两次读取 RTC_CALENDAR_R 的值，如果两次的值不等，则继续读取，直到连续两次的值相等。

(3) 连续两次读取 RTC_CALENDAR_R_H 的值，如果两次的值不等，则继续读取，直到连续两次的值相等。

(4) 最后再读一次 RTC_SUB_SECOND 寄存器的值，如果与步骤(1) 的值不一致，则重新从步骤(1) 开始读取寄存器的值。

(5) 由于 SUBSECOND_COUNT 从最大值变为 0 时，寄存器 RTC_CALENDAR_R 或 RTC_CALENDAR_R_H 的值可能没有产生变化，因此如果 SUBSECOND_COUNT 为 0 时，则继续重新从步骤(1) 开始读取寄存器的值，如果 SUBSECOND_COUNT 不为 0，则完整的日历时间读取完成。

SUBSECOND_COUNT 拍数转换为微秒 sub-second，先要通过 RCC 获知 RTC 的接口时钟 fRTCCLK，则计算 sub-second 的公式如下：

sub-second=(1000000*SUBSECOND_COUNT)/fRTCCLK

18.4.2 设置日历

设置日历需要配置两个 RTC 的寄存器为 RTC_CALENDAR_H 和 RTC_CALENDAR，RTC_CALENDAR_H 为年月日星期信息，RTC_CALENDAR 为时分秒信息，RTC_SUB_SECOND 寄存器不能配置，所以 sub-second 不能配置。日历配置顺序如下：

(1) 读 RTC_SR1 寄存器，等待寄存器中全部的 WRITE_XXX_DONE 位为 1 后，即 RTC_SR1 寄存器的 bit1 至 bit11 全为 1，才允许对寄存器进行写操作。

(2) 配置 RTC_CALENDAR_H 寄存器，即配置年月日星期。

(3) 读 RTC_SR1 寄存器，等待全部的 WRITE_XXX_DONE 为 1 后，即 RTC_SR1 寄存器的 bit1 至 bit11 全为 1，才允许对寄存器进行写操作。

(4) 配置 RTC_CALENDAR 寄存器，即配置时分秒。

18.5 RTC PPM 调整

RTC 频率的校准分辨率约为 0.5ppm 范围从 -1024ppm 到 +1024ppm，为 0 时表示不调整，通过配置寄存器 RTC_PPMADJUST 的值来设置 PPM 的调整值，0 ppm 对应 RTC_PPMADJUST 寄存器的值应该为 0x7FFF。设置 PPM 调整的流程如下:

(1) 读 RTC_SR1 寄存器，等待寄存器中全部的 WRITE_XXX_DONE 位为 1 后，即 RTC_SR1 寄存器的 bit1 至 bit11 全为 1，才允许对寄存器进行写操作。

(2) 配置 RTC_PPMADJUST 寄存器的值。

18.6 低功耗唤醒

RTC 可以通过中断或唤醒信号从 sleep、stop、standby 中唤醒。

表 18-1 RTC 唤醒源

休眠模式	唤醒描述
Sleep	RTC 所有中断可以使设备从 sleep 模式中唤醒
Stop0 、Stop1 、Stop2 、Stop3	Wakeup/tamper IO、alarm、cyc 的唤醒信号可以使设备从 stop 模式中唤醒
Standby	Wakeup/tamper IO、alarm、cyc 的唤醒信号可以使设备从 standby 模式中唤醒

设置 wakeup/tamper IO、alarm、cyc 的唤醒信号是通过寄存器 RTC_CR，对应寄存器中的 bit 名称分别如下表:

表 18-2 配置唤醒信号使能的 bit 信息

功能	RTC_CR 寄存器 bit 信息
WAKEUP_IO0	WAKEUP0_WKEN1
WAKEUP_IO1	WAKEUP1_WKEN1
WAKEUP_IO2	WAKEUP2_WKEN1
TAMPER	TAMPER_WKEN1
ALARM0	RTC_ALARM0_WKEN
ALARM1	RTC_ALARM1_WKEN
CYC	CYC_WKEN

18.7 tamper/wakeup IO 检测

tamper/wakeup IO 的输入事件可以配置为边沿检测，也可以配置为电平检测，电平检测时可以配置滤波。边沿检测表示检测 GPIO 的上升沿或下降沿，电平检测表示检测 GPIO 的高电平或低电平，如果高电平有效，GPIO 输入高电平则会检测到输入事件，如果低电平有效，GPIO 输入低电平则会检测到输入事件。当检测到输入事件可以配置产生如下操作:

- 清除保留 SRAM 的内容
- 产生一个中断，并且能够从 sleep 模式中唤醒
- 产生唤醒信号，能够从 stop、standby 模式中唤醒

18.7.1 初始化和配置流程

在 tamper/wakeup 初始化之前需要配置对应的 GPIO 为 tamper/wakeup 功能。如果是电平检测，需要配置 GPIO 的上下拉，高电平有效则配置下拉，低电平有效则配置上拉。以 tamper 为例，其初始化和配置流程如下:

(1) 如果是电平检测，则通过 RTC_CR 寄存器的 TAMPER_FILTER_CFG 位配置滤波拍数，并且配置有效电平即低电平有效还是高电平有效，此 bit 位为 RTC_CR 寄存器的 TAMPER_LEVEL_SEL，最后使能电平唤醒，即配置 RTC_CR 寄存器的 TAMPER_WKEN0。***如果是边沿检测，则忽略此步骤。***

(2) 如果需要从 stop、standby 模式中唤醒，则配置唤醒使能，即配置 RTC_CR 寄存器的 TAMPER_WKEN1，***否则忽略此步骤***。

(3) 置位 RTC_CR 寄存器的 TAMPER_EN 来使能 tamper 功能。

18.7.2 清除保留 SRAM

当检测到 tamper/wakeup IO 的输入事件时可以配置清除保留 SRAM 的内容，置位 RTC_CR 寄存器的 RTC_RET_SRAM_ERASE_EN 的对应的 bit 位可以配置该功能。bit0 对应 wakeupio0，bit1 对应 wakeupio1，bit2 对应 wakeupio2，bit3 对应 tamper。

18.8 周期计数功能

周期计数功能可以定时产生中断或唤醒事件。定时时间通过配置寄存器 RTC_CYC_MAX_VALUE 的值进行设置，计算定时时间 time 先要通过 RCC 获知 RTC 的接口时钟 fRTCCLK，寄存器 RTC_CYC_MAX_VALUE 的值为 CYC_MAX_VALUE，则计算 time 的公式如下，单位为微秒:

$$time = (1000000 * CYC\_MAX\_VALUE) / fRTCCLK$$

周期计数过程中可以读取已经经过的拍数，就可以得到本轮计数开始到当前时刻的时间间隔 interval，已经经过的拍数通过读取 RTC_CYC_CNT_VALUE 的值 CYC_CNT_VALUE 获得，

计算 interval 的公式如下，单位为微秒：

interval= (1000000 * CYC_CNT_VALUE) / fRTCCLK

周期计数功能的配置流程如下：

(1) 已知 time，根据上述公式计算得到 CYC_MAX_VALUE 的值，把此值配置到寄存器 RTC_CYC_MAX_VALUE 中。

(2) 如果需要从 stop、standby 模式中唤醒，则配置唤醒使能，即配置 RTC_CR 寄存器的 CYC_WKEN，否则无需配置。

(3) 使能周期计数功能，即置位 RTC_CR 寄存器的 CYC_START_COUNTER。

18.9 闹钟功能

RTC 提供两个闹钟为闹钟 0 和闹钟 1，两个闹钟都支持 Mask 选择与日历匹配，Mask 配置可以选择匹配 sub-second、秒、分钟、小时、日期、星期，日期和星期匹配只能二选一。

如果寄存器 RTC_ALARMx 中的 Bit 位 ALARMx_WEEK_SEL 为 0，则只能选择是否匹配日期，如果 ALARMx_WEEK_SEL 为 1 则只能选择是否匹配星期，x 表示 0 或 1。当没有配置秒和 sub-second 匹配而配置了分钟匹配，则闹钟定时时间到了后，在当前一分钟内会以一秒为间隔产生 60 次中断或唤醒事件，如果使能了闹钟中断则会产生中断，如果使能了闹钟唤醒功能则会产生唤醒事件，当没有配置秒、sub-second 和分钟的匹配而配置了小时匹配，则闹钟定时时间到了后，在当前一小时内会以一秒为间隔产生 3600 次中断或唤醒事件。

闹钟的秒、分钟、小时、日期、星期 Mask 通过配置寄存器 RTC_ALARMx 中的 ALARMx_MASK 字段实现，sub-second 的 Mask 设置通过配置寄存器 RTC_ALARMx_SUB 的 RTC_ALARMx_SUB_MASK 实现，sub-second 的定时值设置通过配置寄存器 RTC_ALARMx_SUB 的 RTC_ALARMx_SUB_VALUE 实现，RTC_ALARMx_SUB_VALUE 的值表示拍数，拍数转换成时间的公式与周期计数的拍数转换成时间的一样，x 表示 0 或 1，分别对应闹钟 0 和闹钟 1。以闹钟 0 为例描述闹钟配置流程如下：

(1) 设置日历。

(2) 配置闹钟 0 定时值，配置寄存器 RTC_ALARM0 的 ALARM0_VALUE 字段，即配置定时时间的小时、分钟、秒、日期或星期。

(3) 配置闹钟 0 的 sub-second 定时值，即配置寄存器 RTC_ALARM0_SUB 的 RTC_ALARM0_SUB_VALUE。

(4) 配置闹钟 0 的小时、分钟、秒、日期或星期的 Mask。

(5) 配置闹钟 0 的 sub-second 的 Mask。

(6) 如果需要闹钟的中断，则使能中断，如果需要唤醒事件，则使能闹钟 0 的唤醒功能，即配置寄存器 RTC_CR 的 bit 位 RTC_ALARM0_WKEN。

(7) 使能闹钟 0 功能，通过配置寄存器 RTC_ALARM0 的 ALARM0_EN。

(8) 使能日历功能，即置位寄存器 RTC_CR 的 bit 位 RTC_START_RTC。

18.10 内部信号 IO 输出

内部的能通过 IO 输出的信号包括闹钟 0 脉冲，闹钟 1 脉冲，周期计数脉冲，秒信号输出，闹钟、周期计数脉冲是宽度为一拍的脉冲，alarm 脉冲是在定时时间到的时候输出，周期计数脉冲是在每次计数满一个周期后输出，是周期输出的，秒信号是占空比为 50%的方波，周期为 1 秒。IO 输出的电平可以被取反，配置寄存器 RTC_CR2 的 RTC_OUT_POL 位为 0 时表示原电平，为 1 是表示电平取反。配置寄存器 RTC_CR2 的 RTC_OUT_SEL 进行输出信号的选择。

18.11 中断

RTC 的中断信号如下:

表 18-3 RTC 中断信号

中断名称	描述
闹钟 0 中断	闹钟 0 定时时间到产生的中断
闹钟 1 中断	闹钟 1 定时时间到产生的中断
周期计数中断	计数满一个周期时产生的中断
Tamper 中断	Tamper 检测到输入事件时产生的中断
Wakeupio0 中断	Wakeupio0 检测到输入事件时产生的中断
Wakeupio1 中断	Wakeupio1 检测到输入事件时产生的中断
Wakeupio2 中断	Wakeupio2 检测到输入事件时产生的中断
秒中断	秒信号在每一秒产生的中断

上述中断的使能通过配置寄存器 RTC_CR1 实现，除秒中断的中断状态外，其它所有中断的中断状态都可以通过寄存器 RTC_SR 获得，秒中断的中断状态通过寄存器 RTC_SR1 的 SECOND_SR 位获得。

18.12 RTC 相关寄存器描述

基地址：0x4000E000

表 18-4 RTC 寄存器列表

寄存器	偏移量	描述
RTC_CR	0x00	控制寄存器
RTC_ALARM0	0x04	闹钟 0 寄存器
RTC_ALARM1	0x08	闹钟 1 寄存器
RTC_PPMADJUST	0x0c	PPM 调整寄存器
RTC_CALENDAR	0x10	日历配置时分秒寄存器
RTC_CALENDAR_H	0x14	日历配置年月日星期寄存器
RTC_CYC_MAX_VALUE	0x18	周期计数值配置寄存器
RTC_SR	0x1c	中断状态寄存器
RTC_SYNDATA	0x20	日历异步读取时分秒寄存器
RTC_SYNDATA_H	0x24	日历异步读取年月日星期寄存器
RTC_CR1	0x28	中断使能寄存器
RTC_SR1	0x2c	操作状态寄存器
RTC_CR2	0x30	控制寄存器 2
RTC_SUB_SECOND	0x34	读取 sub-second 寄存器
RTC_CYC_CNT_VALUE	0x38	去读周期计数值寄存器
RTC_ALARM0_SUB	0x3c	闹钟 0 的 sub-second 寄存器
RTC_ALARM1_SUB	0x40	闹钟 1 的 sub-second 寄存器
RTC_CALENDAR_R	0x44	日历同步读取时分秒寄存器
RTC_CALENDAR_R_H	0x48	日历同步读取年月日星期寄存器

18.12.1 RTC_CR

偏移量：0x00

复位值：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

位 31-29 RESERVED：保留。

位 28 RTC_START_RTC：RTC 日历计数使能控制。

- 0：不使能
- 1：使能

位 27 RTC_ALARM0_WKEN：ALARM0_SR 唤醒使能。

- 0：不使能
- 1：使能

位 26 RTC_ALARM1_WKEN：ALARM1_SR 唤醒使能。

- 0：不使能
- 1：使能

位 25 CYC_WKEN：CYC_SR 唤醒使能。

- 0：不使能
- 1：使能

位 24 CYC_START_COUNTER：定时功能使能控制。

- 0：不使能
- 1：使能

位 23 TAMPER_EN：TAMPER 功能使能。

- 0：不使能
- 1：使能

位 22 TAMPER_LEVEL_SEL：TAMPER 有效电平选择。

- 0：低电平有效
- 1：高电平有效

位 21 TAMPER_WKEN0：TAMPER 电平唤醒使能。

- 0：低电平有效
- 1：高电平有效

TAMPER_EN 为 0 时，仍可用于唤醒。

位 20 TAMPER_WKEN1：TAMPER_SR 唤醒使能。

- 0：不使能
- 1：使能

位 19-18 TAMPER_FILTER_CFG：TAMPER 的滤波控制。

- 0：不滤波
- 1：滤波 1 拍 rtc 接口时钟
- 2：滤波 3 拍 rtc 接口时钟
- 3：滤波 7 拍 rtc 接口时钟

位 17 WAKEUP0_EN：WAKEUP0 功能使能。

- 0：不使能
- 1：使能

位 16 WAKEUP0_LEVEL_SEL：WAKEUP0 有效电平选择。

- 0：低电平有效
- 1：高电平有效

位 15 WAKEUP0_WKEN0：WAKEUP0 电平唤醒使能。

- 0：不使能
- 1：使能

WAKEUP0_EN 为 0 时，仍可用于唤醒。

位 14 WAKEUP0_WKEN1：WAKEUP0_SR 唤醒使能。

- 0：不使能
- 1：使能

位 13-12 WAKEUP0_FILTER_CFG：WAKEUP0 的滤波控制。

- 0：不滤波
- 1：滤波 1 拍 rtc 接口时钟
- 2：滤波 3 拍 rtc 接口时钟
- 3：滤波 7 拍 rtc 接口时钟

位 11 WAKEUP1_EN：WAKEUP1 功能使能。

- 0：不使能
- 1：使能

位 10 WAKEUP1_LEVEL_SEL：WAKEUP1 有效电平选择。

- 0：低电平有效
- 1：高电平有效

位 9 WAKEUP1_WKEN0：WAKEUP1 电平唤醒使能。

- 0：不使能
- 1：使能

WAKEUP1_EN 为 0 时，仍可用于唤醒。

位 8 WAKEUP1_WKEN1：WAKEUP1_SR 唤醒使能。

- 0：不使能
- 1：使能

位 7-6 WAKEUP1_FILTER_CFG：WAKEUP1 的滤波控制。

- 0：不滤波
- 1：滤波 1 拍 rtc 接口时钟
- 2：滤波 3 拍 rtc 接口时钟
- 3：滤波 7 拍 rtc 接口时钟

位 5 WAKEUP2_EN：WAKEUP2 功能使能。

- 0：不使能
- 1：使能

位 4 WAKEUP2_LEVEL_SEL：WAKEUP2 有效电平选择。

- 0：低电平有效
- 1：高电平有效

位 3 WAKEUP2_WKEN0：WAKEUP2 电平唤醒使能。

- 0：不使能
- 1：使能

WAKEUP2_EN 为 0 时，仍可用于唤醒。

位 2 WAKEUP2_WKEN1：WAKEUP2_SR 唤醒使能。

- 0：不使能
- 1：使能

位 1-0 WAKEUP2_FILTER_CFG：WAKEUP2 的滤波控制。

- 0：不滤波
- 1：滤波 1 拍 rtc 接口时钟
- 2：滤波 3 拍 rtc 接口时钟
- 3：滤波 7 拍 rtc 接口时钟

18.12.2 RTC_ALARM0

偏移量：0x04

复位值：0x00000000

31	30	29-26	25-0
ALARM0_EN	ALARM0_WEEK_SEL	ALARM0_MASK	ALARM0_VALUE
rw-0h	rw-0h	rw-0h	rw-0h

位 31 ALARM0_EN：闹钟 0 使能控制。

- 0：不使能
- 1：使能

位 30 ALARM0_WEEK_SEL：闹钟 0 星期与日期选择。

- 0：匹配日期
- 1：匹配星期，星期为日期个位的低 3bit

位 29-26 ALARM0_MASK：闹钟 0 的 Mask 配置。

[26] 控制是否匹配秒。

- 0：匹配
- 1：不匹配

[27] 控制是否匹配分钟。

- 0：匹配
- 1：不匹配

[28] 控制是否匹配小时。

- 0：匹配
- 1：不匹配

[29] 控制是否匹配日期或星期。

- 0：匹配
- 1：不匹配

位 25-0 ALARM0_VALUE：闹钟 0 定时值配置。当 rtc 日历与闹钟 0 配置匹配时，可以产生 ALARM0_SR。

[3:0]：秒个位

[6:4]：秒十位

[10:7]：分钟个位

[13:11]：分钟十位

[17:14]：小时个位

[19:18]：小时十位

[23:20]：日期个位，其中 [22:20] 也可代表星期

[25:24]：日期十位

18.12.3 RTC_ALARM1

偏移量：0x08

复位值：0x00000000

31	30	29-26	25-0
ALARM1_EN	ALARM1_WEEK_SEL	ALARM1_MASK	ALARM1_VALUE
rw-0h	rw-0h	rw-0h	rw-0h

位 31 ALARM1_EN：闹钟 1 使能控制。

- 0：不使能
- 1：使能

位 30 ALARM1_WEEK_SEL：闹钟 1 星期与日期选择。

- 0：匹配日期
- 1：匹配星期，星期为日期个位的低 3bit

位 29-26 ALARM1_MASK：闹钟 1 的 Mask 配置。

[26] 控制是否匹配秒。

- 0：匹配
- 1：不匹配

[27] 控制是否匹配分钟。

- 0：匹配
- 1：不匹配

[28] 控制是否匹配小时。

- 0：匹配
- 1：不匹配

[29] 控制是否匹配日期或星期。

- 0：匹配
- 1：不匹配

位 25:0 ALARM1_VALUE：闹钟 1 定时值配置。当 rtc 日历与闹钟 1 配置匹配时，可以产生 ALARM1_SR。

[3:0]：秒个位

[6:4]：秒十位

[10:7]：分钟个位

[13:11]：分钟十位

[17:14]：小时个位

[19:18]：小时十位

[23:20]：日期个位，其中 [22:20] 也可代表星期

[25:24]：日期十位

18.12.4 RTC_PPMADJUST

偏移量：0x0c

复位值：0x00007fff

31-16	15-0
RESERVED	PPMADJUST_VALUE
r-0h	rw-7fffh

位 31-16 RESERVED：保留。

位 15-0 PPMADJUST_VALUE：32k 时钟频率 ppm 调整配置。1 代表 0.5ppm。可配置范围为 77ff 到 87ff。

- 77ff：调整+1024 ppm
- 7800：调整+1023.5 ppm
- ...
- 7ffd：调整+1 ppm
- 7ffe：调整+0.5 ppm
- 7fff：调整 0 ppm，即不调整
- 8000：调整-0.5 ppm
- 8001：调整-1 ppm
- …
- 87fe：调整-1023.5ppm
- 87ff：调整-1024ppm

18.12.5 RTC_CALENDAR

偏移量：0x10

复位值：0x00000000

31-20	19-0
RESERVED	CALENDAR_VALUE
r-0h	w-0h

位 31-20 RESERVED：保留。

位 19-0 CALENDAR_VALUE：rtc 日历数据低位。

[3:0]：秒个位

[6:4]：秒十位

[10:7]：分钟个位

[13:11]：分钟十位

[17:14]：小时个位

[19:18]：小时十位

18.12.6 RTC_CALENDAR_H

偏移量：0x14

复位值：0x00000841

31-22	21-0
RESERVED	CALENDAR_H_VALUE
r-0h	w-841h

位 31-22 RESERVED：保留。

位 21-0 CALENDAR_H_VALUE：rtc 日历数据高位。

[3:0]：日期个位

[5:4]：日期十位

[9:6]：月份个位

[10]：月份十位

[13:11]：星期

[17:14]：年个位

[21:18]：年十位

18.12.7 RTC_CYC_MAX_VALUE

偏移量：0x18

复位值：0x00008000

31-0
CYC_MAX_VALUE
rw-8000h

位 31-0 CYC_MAX_VALUE：周期计数值配置，当周期计数器的计数值等于 CYC_MAX_VALUE 时，触发周期唤醒状态位，周期计数使用 rtc 接口时钟进行计数。

18.12.8 RTC_SR

偏移量：0x1c

复位值：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

位 31-7 RESERVED：保留。

位 6 ALARM0_SR：闹钟 0 触发状态位。硬件置 1，软件写 1 清 0。

- 0：未发生
- 1：发生

位 5 ALARM1_SR：闹钟 1 触发状态位。硬件置 1，软件写 1 清 0。

- 0：未发生
- 1：发生

位 4 CYC _SR：周期唤醒触发状态位。硬件置 1，软件写 1 清 0。

- 0：未发生
- 1：发生

位 3 TAMPER_SR：tamper 触发状态位。硬件置 1，软件写 1 清 0。

- 0：未发生
- 1：发生

位 2 WAKEUP0_SR：wakeup0 触发状态位。硬件置 1，软件写 1 清 0。

- 0：未发生
- 1：发生

位 1 WAKEUP1_SR：wakeup1 触发状态位。硬件置 1，软件写 1 清 0。

- 0：未发生
- 1：发生

位 0 WAKEUP2_SR：wakeup2 触发状态位。硬件置 1，软件写 1 清 0。

- 0：未发生
- 1：发生

18.12.9 RTC_SYNDATA

偏移量：0x20

复位值：0x00000000

31-20	19-0
RESERVED	SYN_DATA
r-0h	r-0h

位 31-20 RESERVED：保留。

位 19-0 SYN_DATA：rtc 日历数据低位。读 rtc 日历时由硬件更新，软件只读。

[3:0]：秒个位

[6:4]：秒十位

[10:7]：分钟个位

[13:11]：分钟十位

[17:14]：小时个位

[19:18]：小时十位

18.12.10 RTC_SYNDATA_H

偏移量：0x24

复位值：0x00000000

31-22	21-0
RESERVED	SYN_DATA_H
r-0h	r-0h

位 31-22 RESERVED：保留。

位 21-0 SYN_DATA_H：rtc 日历数据高位。读 rtc_calendar 时由硬件更新，软件只读。

[3:0]：日期个位

[5:4]：日期十位

[9:6]：月份个位

[10]：月份十位

[13:11]：星期

[17:14]：年个位

[21:18]：年十位

18.12.11 RTC_CR1

偏移量：0x28

复位值：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

位 31-8 RESERVED：保留。

位 7 SECOND_SR_INT_EN：秒中断使能。

- 0：不使能
- 1：使能

位 6 ALARM0_SR_INT_EN：ALARM0_SR 中断使能。

- 0：不使能
- 1：使能

位 5 ALARM1_SR_INT_EN：ALARM1_SR 中断使能。

- 0：不使能
- 1：使能

位 4 CYC_SR_INT_EN：CYC_SR 中断使能。

- 0：不使能
- 1：使能

位 3 TAMPER_SR_INT_EN：TAMPER_SR 中断使能。

- 0：不使能
- 1：使能

位 2 WAKEUP0_SR_INT_EN：WAKEUP0_SR 中断使能。

- 0：不使能
- 1：使能

位 1 WAKEUP1_SR_INT_EN：WAKEUP1_SR 中断使能。

- 0：不使能
- 1：使能

位 0 WAKEUP2_SR_INT_EN：WAKEUP2_SR 中断使能。

- 0：不使能
- 1：使能

18.12.12 RTC_SR1

偏移量：0x2c

复位值：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

位 31-12 RESERVED：保留。

位 11 WRITE_ALARM0_SUB_DONE：WRITE_ALARM0_SUB 操作完成状态。硬件控制置 1 与清 0。

- 0：操作正在进行中
- 1：操作已完成

位 10 WRITE_ALARM1_SUB_DONE：WRITE_ALARM1_SUB 操作完成状态。硬件控制置 1 与清 0。

- 0：操作正在进行中
- 1：操作已完成

位 9 SECOND_SR：秒中断状态。硬件置 1，软件写 1 清 0。

- 0：无中断发生
- 1：有中断发生

位 8 WRITE_RTCCR2_DONE：WRITE_RTCCR2 操作完成状态。硬件控制置 1 与清 0。

- 0：操作正在进行中
- 1：操作已完成

位 7 WRITE_RTCCR_DONE：WRITE_RTCCR 操作完成状态。硬件控制置 1 与清 0。

- 0：操作正在进行中
- 1：操作已完成

位 6 WRITE_ALARM0_DONE：WRITE_ALARM0 操作完成状态。硬件控制置 1 与清 0。

- 0：操作正在进行中
- 1：操作已完成

位 5 WRITE_ALARM1_DONE：WRITE_ALARM1 操作完成状态。硬件控制置 1 与清 0。

- 0：操作正在进行中
- 1：操作已完成

位 4 WRITE_PPMADJUST_DONE：WRITE_PPMADJUST 操作完成状态。硬件控制置 1 与清 0。

- 0：操作正在进行中
- 1：操作已完成

位 3 WRITE_CALENDAR_DONE：WRITE_CALENDAR 操作完成状态。硬件控制置 1 与清 0。

- 0：操作正在进行中
- 1：操作已完成

位 2 WRITE_CYC_MAX_VALUE_DONE：WRITE_CYC_MAX_VALUE 操作完成状态。硬件控制置 1 与清 0。

- 0：操作正在进行中
- 1：操作已完成

位 1 WRITE_RTCSR_DONE：WRITE_RTCSR 操作完成状态。硬件控制置 1 与清 0。

- 0：操作正在进行中
- 1：操作已完成

位 0 READ_CALENDAR_DONE：READ_CALENDAR 操作完成状态。硬件控制置 1 与清 0。

- 0：操作正在进行中
- 1：操作已完成

18.12.13 RTC_CR2

偏移量：0x30

复位值：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

位 31-8 RESERVED：保留。

位 7 RTC_OUT_POL：rtc io 输出电平选择。

- 0：原电平
- 1：取反电平

位 6-4 RTC_OUT_SEL：rtc io 输出选择。

- 0-3：输出 0
- 4：alarm0 脉冲
- 5：alarm1 脉冲
- 6：cyc 脉冲
- 7：秒信号，占空比 50%

位 3-0 RTC_RET_SRAM_ERASE_EN：tamper 与 wakeup 报警触发 ret sram 清除的功能使能。[0] 对应 wakeup0，[1] 对应 wakeup1，[2] 对应 wakeup2，[3] 对应 tamper。

- 0：不使能
- 1：使能

18.12.14 RTC_SUB_SECOND

偏移量：0x34

复位值：0x00000000

31-15	14-0
RESERVED	RTC_SUB_SECOND_VALUE
r-0h	r-0h

位 31-15 RESERVED：保留。

位 14-0 RTC_SUB_SECOND_VALUE：rtc 日历计数中的秒以下部分的计数器值，由于采用打拍同步，需要多次读到同一值，才可以使用，不能仅读取一次（可能由于同步而读错）。

18.12.15 RTC_CYC_CNT_VALUE

偏移量：0x38

复位值：0x00000000

31-0
CYC_CNT_VALUE
r-0h

位 31-0 CYC_CNT_VALUE：周期计数器值，由于采用打拍同步，需要多次读到同一值，才可以使用，不能仅读取一次（可能由于同步而读错）。

18.12.16 RTC_ALARM0_SUB

偏移量：0x3c

复位值：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

位 31-20 RESERVED：保留。

位 19-16 RTC_ALARM0_SUB_MASK：闹钟 0 的 sub-second Mask 配置，使用 alarm0 的 sub-second 功能时，ppm 调整功能建议不使用。

- 0：sub-second 定时值配置不生效，仅在每秒考虑 rtc_alarm0 是否匹配。
- 1：匹配 sub-second 计数的[0]，其它 sub-second 计数位不生效。
- 2：匹配 sub-second 计数的[1:0]，其它 sub-second 计数位不生效。
- N：匹配 sub-second 计数的[N-1:0]，其它 sub-second 计数位不生效。

位 15 RESERVED：保留。

位 14-0 RTC_ALARM0_SUB_VALUE：闹钟 0 sub-second 定时值配置。当 rtc 日历计数值与闹钟 0 配置匹配时，产生 ALARM0_SR。

18.12.17 RTC_ALARM1_SUB

偏移量：0x40

复位值：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

位 31-20 RESERVED：保留。

位 19-16 RTC_ALARM1_SUB_MASK：闹钟 1 的 sub-seconde Mask 配置，使用 alarm1 的 sub-second 功能时，ppm 调整功能建议不使用。

- 0：sub-second 定时值配置不生效，仅在每秒考虑 rtc_alarm1 是否匹配。
- 1：匹配 sub-second 计数的[0]，其它 sub-second 计数位不生效。
- 2：匹配 sub-second 计数的[1:0]，其它 sub-second 计数位不生效。
- N：匹配 sub-second 计数的[N-1:0]，其它 sub-second 计数位不生效。

位 15 RESERVED：保留。

位 14-0 RTC_ALARM1_SUB_VALUE：闹钟 1 sub-second 定时值配置。当 rtc 日历计数值与闹钟 1 配置匹配时，产生 ALARM1_SR。

18.12.18 RTC_CALENDAR_R

偏移量：0x44

复位值：0x00000000

31-20	19-0
RESERVED	CALENDAR_SYNC
r-0h	r-0h

位 31-20 RESERVED：保留。

位 19-0 CALENDAR_SYNC：rtc 日历计数中的时分秒的计数器值，由于采用打拍同步，需要多次读到同一值，才可以使用，不能仅读取一次（可能由于同步而读错）。

18.12.19 RTC_CALENDAR_R_H

偏移量：0x48

复位值：0x00000841

31-22	21-0
RESERVED	CALENDAR_H_SYNC
r-0h	r-841h

位 31-22 RESERVED：保留。

位 21-0 CALENDAR_H_SYNC：rtc 日历计数中的年月日星期的计数器值，由于采用打拍同步，需要多次读到同一值，才可以使用，不能仅读取一次（可能由于同步而读错）。

19. LPUART

19.1 简介

LPUART（Low Power Universal Asynchronous Receiver/Transmitter）是一种低功耗的串口外设，32K 时钟下波特率最高支持 9600。在极低功耗模式下，LPUART 也可以被接收到的数据唤醒。

LPUART 支持 CTS/RTS 流量控制。

LPUART 支持 DMA 请求。

19.2 主要特性

- 波特率可配置
- 数据格式可配置（5-8 位数据，1-2 位停止位，0-1 位奇偶校验位）
- 支持 DMA 请求
- TX/RX FIFO 深度为 1
- 支持 CTS/RTS 流控
- 支持中断请求
- 支持低功耗唤醒

19.3 功能描述

19.3.1 数据格式

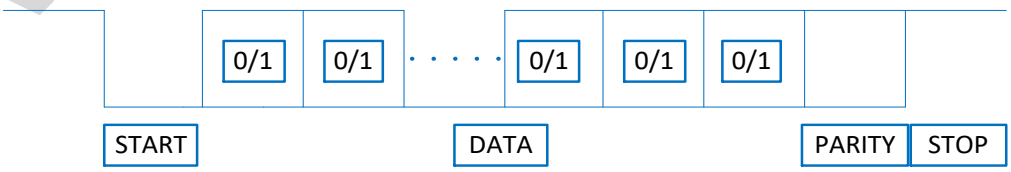

图 19-1 LPUART 的数据传输格式

在空闲时，LPUART 的数据线应该保持在高电平。

在数据传输时，依次传输起始位（START），数据位（DATA），奇偶校验位（PARITY）和停止位（STOP）。各个位的含义如下：

(1) **起始位：**先发送一个 0 信号，表示数据传输开始。

(2) **数据位：**根据配置，依次传输 5-8 个 bit。

(3) **奇偶校验位：**数据位后，传输一个 bit 的奇偶校验位，也可以配置为无奇偶校验位。

(4) **停止位：**数据传输结束的标志，可以是 1 或者 2 个 bit。

19.3.2 波特率产生

LPUART 波特率的配置支持小数分频，其主要通过 LPUART_BAUD_RATE_INT 和 LPUART_BAUD_RATE_FRA 两个寄存器来配置。

以 UART 接口时钟频率为 32.768KHz，波特率为 9600 为例，分频系数为 32768/9600=3.413，则寄存器 LPUART_BAUD_RATE_INT 配置为 3，寄存器 LPUART_BAUD_RATE_FRA 配置为 7（0.413*16=6.608，四舍五入为 7）。

19.3.3 CTS/RTS 流控

两个 LPUART 之间的连接如下图:

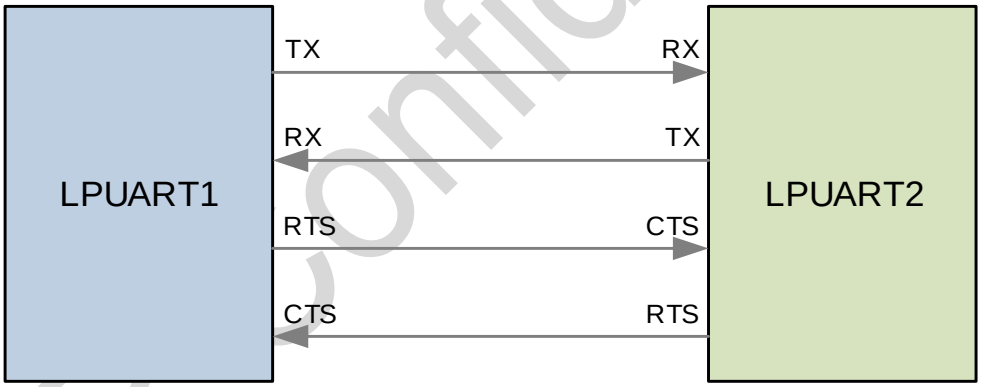

图 19-2 两个 LPUART 设备之间的连接

RTS（Require To Send，发送请求）为输出信号，用于指示本设备准备好可接收数据，低电平有效，低电平说明本设备可以接收数据。

CTS（Clear To Send，发送允许）为输入信号，用于判断是否可以向对方发送数据，低电平有效，低电平说明本设备可以向对方发送数据。

19.3.4 DMA 请求

LPUART DMA 发送过程：

(1) 将寄存器 LPUART_CR1 中的 DMA_TX_EN 位配置为使能；

(2) 将寄存器 LPUART_DATA 地址配置为 DMA 的目的地址；

(3) 将发送数据的内存地址配置为 DMA 的源地址；

(4) 配置 DMA 的 DATA_WIDTH 为 0（数据位宽为 8bit）；

(5) 配置 DMA 的 SRC_MSIZE 和 DEST_MSIZE 为 0（burst length 为 1）；

(6) 配置 DMA 的数据传输总长度；

(7) 配置 DMA 的 handshake 类型为 DMA_HANDSHAKE_LPUART_TX；

(8) 激活 DMA 通道。

当 DMA 传输完成后，会将 DMA CHENREG 寄存器的 CH_EN 位清 0。

LPUART DMA 接收过程：

(1) 将寄存器 LPUART_CR1 中的 DMA_RX_EN 位配置为使能；

(2) 将寄存器 LPUART_DATA 地址配置为 DMA 的源地址；

(3) 将数据接收的内存地址配置为 DMA 的目的地址；

(4) 配置 DMA 的 DATA_WIDTH 为 0（数据位宽为 8bit）；

(5) 配置 DMA 的 SRC_MSIZE 和 DEST_MSIZE 为 0（burst length 为 1）；

(6) 配置 DMA 的数据传输总长度；

(7) 配置 DMA 的 handshake 类型为 DMA_HANDSHAKE_LPUART_RX；

(8) 激活 DMA 通道。

当 DMA 传输完成后，会将 DMA CHENREG 寄存器的 CH_EN 位清 0。

19.3.5 中断信号

LPUART 的中断信号主要有：

- TX_DONE 中断
- TXFIFO_EMPTY 中断
- RXFIFO_NOT_EMPTY 中断
- RX_OVERFLOW 中断
- STOP_ERR 中断
- PARITY_ERR 中断
- START_INVALID 中断
- RX_DONE 中断
- START_VALID 中断

19.3.6 低功耗唤醒

LPUART 的低功耗唤醒包括 RX 低电平唤醒，有效 START 唤醒，RX_DONE 唤醒。

通过配置 LPUART_CR0 寄存器的 LPUART_WAKEUP_EN 位来使能唤醒方式。

19.4 LPUART 相关寄存器描述

寄存器基地址：0x40005000

表 19-1 LPUART 寄存器列表

寄存器	偏移量	描述
LPUART_CR0	0x00	控制寄存器 0
LPUART_CR1	0x04	控制寄存器 1
LPUART_SR0	0x08	状态寄存器 0
LPUART_SR1	0x0C	状态寄存器 1
LPUART_DATA	0x10	数据寄存器

19.4.1 LPUART_CR0

偏移量：0x00

复位值：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

位 31-27 RESERVED：保留，不能修改。

位 26 LPUART_RTS_EN：lpuart rts 流控使能。

- 0：不使能
- 1：使能

位 25 LPUART_RX_EN：lpuart 接收使能。

- 0：不使能
- 1：使能

位 24-22 LPUART_WAKEUP_EN：lpuart 唤醒使能。

[22] 为 rx 低电平唤醒使能。

- 0：不使能
- 1：使能

[23] 为有效 start 唤醒使能。

- 0：不使能
- 1：使能

[24] 为接收数据完成唤醒使能。

- 0：不使能
- 1：使能

位 21-10 LPUART_BAUD_RATE_INT：波特率分频系数的整数部分。

分频系数等于 UART 接口时钟频率/波特率；

以 UART 接口时钟频率为 32.768KHz，波特率为 9600 为例，分频系数为 32768/9600=3.413，lpuart_baud_rate_int 配置为 3，lpuart_baud_rate_fra 配置为 0.413*16=6 或 7。

位 9-6 LPUART_BAUD_RATE_FRA：波特率分频系数的小数部分，支持 4 位小数分频。

位 5 LPUART_STOP_LEN：uart stop 位长度。

- 0：1 位 stop
- 1：2 位 stop

位 4-2 LPUART_PARITY_CFG：uart 奇偶校验配置。

- 0：偶校验
- 1：奇校验
- 2：校验位固定为 0
- 3：校验位固定为 1
- >3：无奇偶校验位

位 1-0 LPUART_DATA_LEN：uart 数据宽度。

数据宽度等于 LPUART_DATA_LEN+5。

19.4.2 LPUART_CR1

偏移量：0x04

复位值：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

位 31:13 RESERVED：保留，不能修改。

位 12 LPUART_CTS_EN：lpuart cts 流控使能。

- 0：不使能
- 1：使能

位 11 DMA_TX_EN：dma tx 请求使能。

- 0：不使能
- 1：使能

位 10 DMA_RX_EN：dma rx 请求使能。

- 0：不使能
- 1：使能

位 9 LPUART_TX_EN：lpuart 发送使能。

- 0：不使能
- 1：使能

位 8 TX_DONE_INT_EN：tx_done 中断使能。

- 0：不使能
- 1：使能

位 7 TXFIFO_EMPTY_INT_EN：txfifo_empty 中断使能。

- 0：不使能
- 1：使能

位 6 RXFIFO_NOT_EMPTY_INT_EN：rxfifo_not_empty 中断使能。

- 0：不使能
- 1：使能

位 5 RX_OVERFLOW_INT_EN：rx_overflow 中断使能。

- 0：不使能
- 1：使能

位 4 STOP_ERR_INT_EN：stop_err 中断使能。

- 0：不使能
- 1：使能

位 3 PARITY_ERR_INT_EN：parity_err 中断使能。

- 0：不使能
- 1：使能

位 2 START_INVALID_INT_EN：start_invalid 中断使能。

- 0：不使能
- 1：使能

位 1 RX_DONE_INT_EN：rx_done 中断使能。

- 0：不使能
- 1：使能

位 0 START_VALID_INT_EN：start_valid 中断使能。

- 0：不使能
- 1：使能

19.4.3 LPUART_SR0

偏移量：0x08

复位值：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

位 31-6 RESERVED：保留，不能修改。

位 5 RX_OVERFLOW_SR：rx_overflow 状态位，用于指示是否发生接收数据 Buffer 溢出。硬件置 1，软件写 1 清 0。

- 0：未发生

- 1：发生

位 4 STOP_ERR_SR：stop_err 状态位，用于指示是否发生 stop 电平错误。硬件置 1，软件写 1 清 0。

- 0：未发生
- 1：发生

位 3 PARITY_ERR_SR：parity_err 状态位，用于指示是否发生 parity 校验错误。硬件置 1，软件写 1 清 0。

- 0：未发生
- 1：发生

位 2 START_INVALID_SR：start_invalid 状态位，用于指示是否收到错误的 start 位。硬件置 1，软件写 1 清 0。

- 0：未发生
- 1：发生

位 1 RX_DONE_SR：rx_done 状态位，用于指示是否接收完数据。硬件置 1，软件写 1 清 0。

- 0：未发生
- 1：发生

位 0 START_VALID_SR：start_valid 状态位，用于指示是否收到有效的 start 位。硬件置 1，软件写 1 清 0。

- 0：未发生
- 1：发生

19.4.4 LPUART_SR1

偏移量：0x0C

复位值：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

位 31-6 RESERVED：保留，不能修改。

位 5 TX_DONE：tx_done 状态位。硬件置 1，软件写 1 清 0。

- 0：发送未完成
- 1：发送完成

位 4 TXFIFO_EMPTY：txfifo_empty 状态位。硬件置 1，软件写 LPUART_DATA 来清 0 此位。

- 0：非空
- 1：空

位 3 RXFIFO_NOT_EMPTY：rxfifo_not_empty 状态位。硬件置 1，软件读 LPUART_DATA 来清 0 此位。

- 0：空
- 1：非空

位 2 WRITE_CR0_DONE：write cr0 操作完成状态。硬件控制置 1 与清 0。

- 0：操作正在进行中
- 1：操作已完成

位 1 WRITE_SR0_DONE：write sr0 操作完成状态。硬件控制置 1 与清 0。

- 0：操作正在进行中
- 1：操作已完成

位 0 RESERVED：保留，不能修改。

19.4.5 LPUART_DATA

偏移量：0x10

复位值：0x00000000

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

位 31-8 RESERVED：保留，不能修改。

位 7-0 LPUART_DATA：fifo 接口，读操作返回 rx data，写操作写入 tx data。

注意：

1. *如果数据位宽小于 8 时，LPUART_DATA 的低位有效。*
2. *读之前要查询 RXFIFO_NOT_EMPTY 状态位，保证 rxfifo 中有数据；写之前要查询 txfifo_ empty 状态位，保证 txfifo 中可写入。*

20. LPTIM

20.1 简介

LPTIMER（Low Power Timer）是一个 16 位的计时器，由于其有多种时钟源，LPTIMER 能够在除 standby 模式外的所有工作模式下运行，并且支持从所有低功耗工作模式中唤醒。有两个 LPTIMER，分别为 LPTIMER0 和 LPTIMER1。

20.2 功能

LPTIMER 包括如下功能：

1. 支持选择内部时钟与外部时钟作为计数时钟
2. 16bits 计数器，加法计数，支持自动加载
3. 支持两种计数模式，单次计数和连续计数
4. 支持软件触发和外部触发源触发计数
5. 分频计数器
6. 支持产生 PWM
7. 支持单脉冲、Set-once、Timeout 模式输出
8. 支持 DEBUG 模式控制
9. 支持产生通道输出事件、匹配事件、溢出事件、触发事件、DOWN 事件、UP 事件作为唤醒信号输出
10. 正交解码
11. 中断信号产生

LPTIMER 的框图如下：

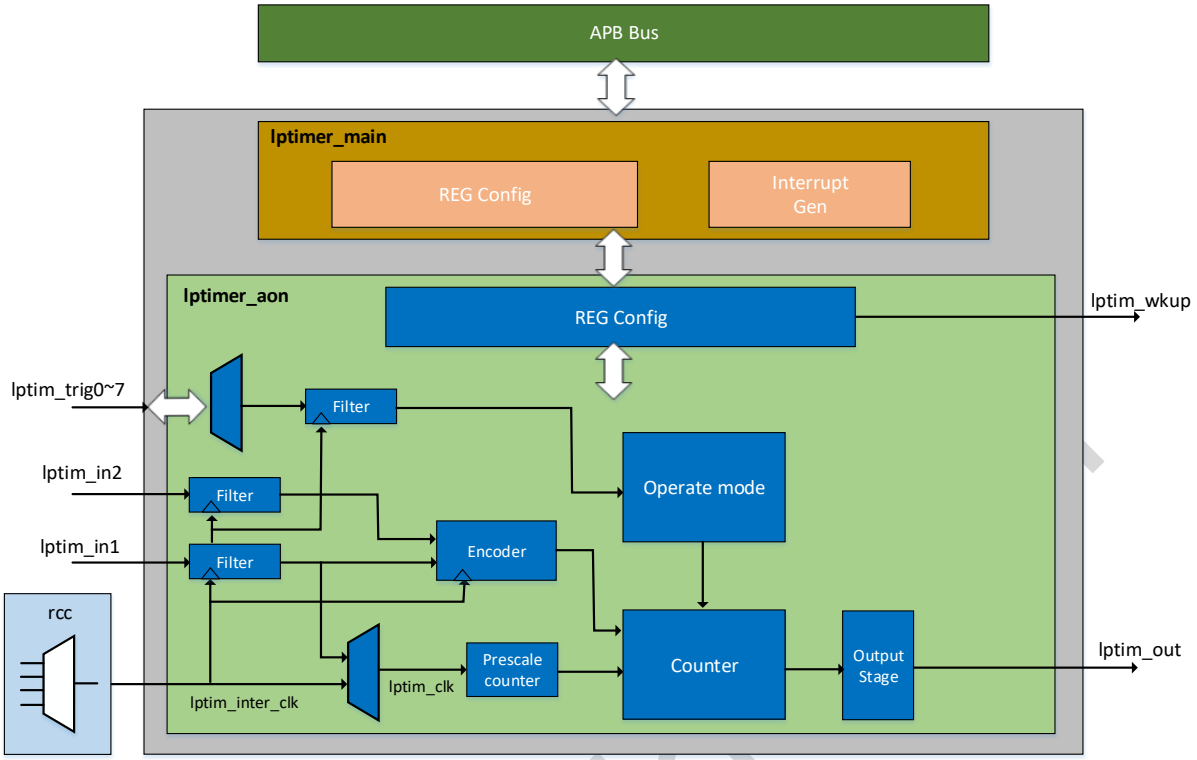

图 20-1 LPTIMER 框图

- **lptim_trig0~7：**LPTIMER 的外部触发源
- **lptim_in2：**LPTIMER 的 IN2 pin 脚
- **lptim_in1：**LPTIMER 的 IN1 pin 脚
- **lptim_wkup：**LPTIMER 的唤醒信号
- **lptim_out：**LPTIMER 的 OUT pin 脚

20.3 接口时钟

LPTIMER 接口时钟源分内部时钟和外部时钟，内部时钟源有 PCLK0、RCO4M、XO32K、RCO32K，外部时钟源通过 IN1 的 GPIO 输入。时钟配置和选择可以参考 RCC 章节。

20.4 计数时钟选择

LPTIMER 除了接口时钟有内部和外部之分外，计数时钟也有内部和外部之分，其内部与外部的时钟源与接口时钟的一致。控制计数时钟选择的寄存器 bit 位为寄存器 LPTIM_CFGR 的 COUNTMODE，值为 0 表示计数器由内部时钟控制，值为 1 表示计数器由外部时钟控制，如果 RCC 模块的寄存器 RCC_CR1 的 LPTIM1_EXT_CLK_SEL 位或 LPTIM_EXT_CLK_SEL 位为 0，即表示 LPTIMER0 或 LPTIMER1 的接口时钟为内部时钟，则 COUNTMODE 的值可以为 0 或 1，即计数时钟既可以为内部时钟也可以为外部时钟，如果 LPTIM1_EXT_CLK_SEL 位或

LPTIM_EXT_CLK_SEL 位为 0，则 LPTIMER0 或 LPTIMER1 的寄存器 LPTIM_CFGR 的只能设置为 0，这时 0 不是表示计数时钟为内部时钟，是表示 COUNTMODE 值需要清零，计数时钟只能为外部时钟。

20.5 计数器

除编码模式外，计数器仅支持向上计数，计数到 ARR 时产生 ARRM 中断，计数器回到 0 重新计数。若使能 timeout 模式，则除了计数器值增加到 ARR 时清零外，触发信号也可以清零计数器重新计数。若使能编码模式，则计数器的计数方向由硬件控制，向上计数到 ARR 时产生 ARRM 事件并清零计数器，向下计数到 0 时则重新加载 ARR 到计数器。

20.6 计数模式

LPTIMER 支持两种计数模式，单次计数和连续计数。单次计数模式下，计数器停止阶段第一个到来的触发信号（硬件或软件）会触发计数器开始计数，计数过程中的触发信号将会被忽略，计数到 ARR 时计数器会停止计数，直到下一次触发信号到来才会再次开始计数，依次类推。连续计数模式下，一旦触发（硬件或软件），计数器会一直计数下去，从 0 到 ARR，然后回到 0 再次计数，如此循环往复。

两种计数模式可以在任意时刻切换（前提是 enable 置位），例如，配置 LPTIMER 为单次计数模式，若置位寄存器 LPTIM_CR 的 CNTSTRT，则计数器计数到 ARR 值将不会停止计数；配置 LPTIMER 为连续计数模式，若置位寄存器 LPTIM_CR 的 SNGSTRT，则计数器计数到 ARR 时将会停止计数，直到下一次触发信号到来。因此状态图如下:

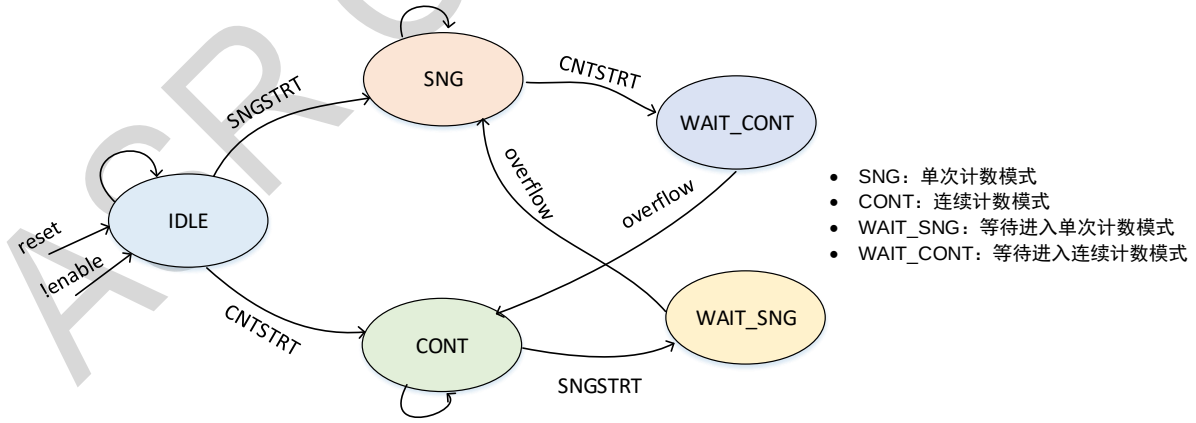

图 20-2 计数模式转换图

20.7 软件触发和外部触发

触发 LPTIMER 计数有两种方式，一种是软件触发，另一种是外部触发源触发。通过寄存器 LPTIM_CFGR 的 TRIGEN 位段进行控制，当值为 0 时为软件触发，非零时为外部触发，当为外部触发时，可以设置外部触发信号上升沿有效、下降沿有效或双沿有效。LPTIMER 有 8 种触发输入源可以选择使用，**LPTIMER0** 的外部触发源如下表：

表 20-1 LPTIMER0 的外部触发源

TRIGSEL	External Trigger	Comment
lptim_ext_trig0	lptim_etr	Lptimer etr pin input
lptim_ext_trig1	comp0	Comp0 output
lptim_ext_trig2	comp1	Comp1 output
lptim_ext_trig3	rtc_cyc_counter	RTC cyc counter output pulse
lptim_ext_trig4	rtc_alarm0	RTC alarm0 output pulse
lptim_ext_trig5	rtc_alarm1	RTC alarm1 output pulse
lptim_ext_trig6	gpio	GPIO58
lptim_ext_trig7	gpio	GPIO59

LPTIMER1 的外部触发源如下表：

表 20-2 LPTIMER1 的外部触发源

TRIGSEL	External Trigger	Comment
lptim_ext_trig0	lptim_etr	Lptimer etr pin input
lptim_ext_trig1	comp0	Comp0 output
lptim_ext_trig2	comp1	Comp1 output
lptim_ext_trig3	rtc_cyc_counter	RTC cyc counter output pulse
lptim_ext_trig4	rtc_alarm0	RTC alarm0 output pulse
lptim_ext_trig5	rtc_alarm1	RTC alarm1 output pulse
lptim_ext_trig6	gpio	GPIO60
lptim_ext_trig7	gpio	GPIO61

20.8 分频计数器

通过计数模式状态机产生的计数使能信号，可以软件配置分频，支持 1、2、4、8、16、32、64、128 分频，通过配置寄存器 LPTIM_CFGR 的 PRESC 位段进行分频配置。该分频通过计数器实现，即上一级电路产生的计数使能信号将作为该分频计数器的计数使能控制计数，当分频计数器计数到预先加载的分频值后，输出一个脉冲，作为下一级计数器的计数使能，然后分频计数器归零重新计数，依次类推。

20.9 PWM

LPTIMER 可以产生 PWM 波形，波形的极性可以通过寄存器 LPTIM_CFGR 的 WAVPOL 比特控制，占空比可以通过寄存器 LPTIM_CMP 和 LPTIM_ARR 的值进行控制。以软件触发和内部时钟计数为例，配置 PWM 的流程如下：

(1) 配置寄存器 LPTIM_CFGR 的 COUNTMODE 为 0，即设置内部时钟计数。

(2) 寄存器 LPTIM_CFGR 的 PRESC 为默认值，即不设置计数器分频。

(3) 配置寄存器 LPTIM_CFGR 的 PRELOAD 的值为 0，即不使能寄存器 LPTIM_CMP 和 LPTIM_ARR 的缓存功能。如果需要也可以使能。

(4) 配置寄存器 LPTIM_CFGR 的 WAVPOL 为 0，即波形输出不反相。

(5) 配置寄存器 LPTIM_CFGR 的 WAVE 为 0。

(6) 寄存器 LPTIM_CFGR 的 TRIGEN 位段的值为 0，即软件触发。

(7) 使能 LPTIMER，就是置位寄存器 LPTIM_CR 的 ENABLE。

(8) 设置寄存器 LPTIM_ARR 和 LPTIM_CMP 的值。

(9) 使能连续计数功能，通过置位寄存器 LPTIM_CR 的 CNTSTRT 实现。

20.10 支持单脉冲、Set-once、Timeout 模式输出

单脉冲模式下，计数器未计数时，检测到第一个触发信号，则计数使能置位，若计数到 ARR 或 enable 清零或模块复位，则计数使能清零，计数过程中的触发信号将会被忽略，如下图：

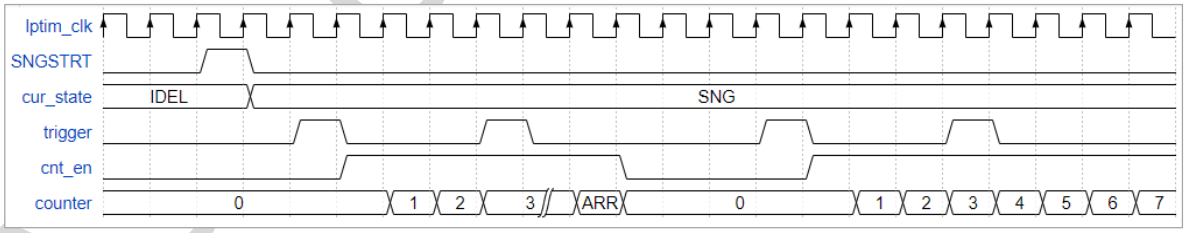

图 20-3 单脉冲计数

单脉冲模式通过配置寄存器 LPTIM_CFGR 的 WAVE 为 0 以及寄存器 LPTIM_CR 的 SNGSTRT 为 1 实现。

Set-once 模式下，检测到第一个触发信号后，计数使能置位，若计数到 ARR，则计数使能清零，计数过程中的触发信号将会被屏蔽，屏蔽信号通过 mask 实现，即检测到第一个触发信号后，mask 有效，屏蔽之后的所有触发信号，如下图：

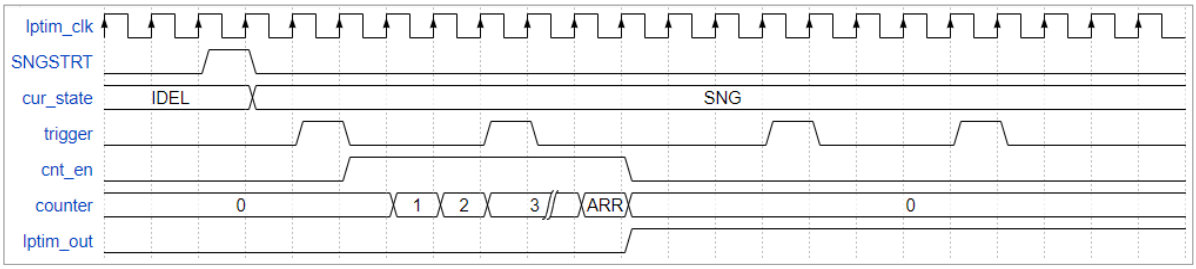

图 20-4 Set-once 计数

Set-once 模式通过配置寄存器 LPTIM_CFGR 的 WAVE 为 1 以及寄存器 LPTIM_CR 的 SNGSTRT 为 1 实现。

Timeout 模式与连续计数模式类似，一旦被触发，计数使能一直有效，区别是，计数过程中的触发信号会让计数器从 0 开始重新计数，且输出波形也会被清除，如下图:

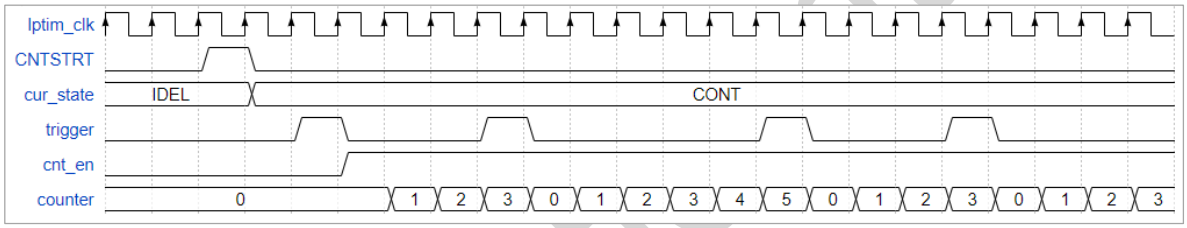

图 20-5 Timeout 计数

Timeout模式通过配置寄存器LPTIM_CFGR的WAVE为0以及寄存器LPTIM_CR的CNTSTRT为 1 实现。

20.11 正交解码

LPTIMER 支持正交编码计数功能，可以通过 IN1 和 IN2 输入正交信号，进行计数和方向检测。编码模式共有三种，仅在上升沿计数、下降沿计数以及双边沿计数，编码模式的使能通过寄存器 LPTIM_CFGR 的 ENC 控制，编码模式的边沿控制通过寄存器 LPTIM_CFGR 的 CKPOL 来实现。在此功能下，两个通道输入可以配置数字滤波功能，滤波使能通过寄存器 LPTIM_CFGR 的 CKFLT_ENABLE 控制，滤波值通过寄存器 LPTIM_CFGR 的 CKFLT 进行配置。通过两个通道信号的组合，可以产生计数使能和方向控制信号，控制计数器加减。具体的组合方式见下表:

表 20-3 正交编码通道信号

编码模式	IN1/IN2 电平	IN1		IN2	
		上升沿	下降沿	上升沿	下降沿
上升沿计数	高电平	向下计数	-	向上计数	-
	低电平	向上计数	-	向下计数	-
下降沿计数	高电平	-	向上计数	-	向下计数
	低电平	-	向下计数	-	向上计数
双沿计数	高电平	向下计数	向上计数	向上计数	向下计数
	低电平	向上计数	向下计数	向下计数	向上计数

IN1 和 IN2 输入信号频率必须小于 LPTIMER 时钟频率的 1/4。

20.12 支持 DEBUG 模式控制

LPTIMER 可由软件配置 debug 下是否停止计数，通过 SYSCFG 的 CR2 寄存器来实现 LPTIMER0 和 LPTIMER1 的 DEBUG 模式计数控制，如果使能该功能，则进入系统 debug 模式时，LPTIMER 停止计数（计数器不会被初始化）。

20.13 唤醒信号

LPTIMER 有 6 种唤醒信号输出，分别是，

- **通道输出信号**，此时通道输出将作为唤醒信号输出。
- **匹配事件（CMPM）**，此时计数器与寄存器 LPTIM_CMP 的匹配事件将作为唤醒信号输出。
- **溢出事件（ARRM）**，此时 overflow 事件将作为唤醒信号输出。
- **触发事件（EXTTRIG）**，此时有效的触发事件将作为唤醒信号输出。
- **DOWN 事件**，若计数方向由向上计数变为向下计数，DOWN 事件会置位，此时 DOWN 事件会作为唤醒信号输出。
- **UP 事件**，若计数方向由向下计数变为向上计数，UP 事件会置位，此时 UP 事件会作为唤醒信号输出。

以上唤醒信号除了通道输出信号，均为 LPTIM_ISR 寄存器的标志位，且有独立的使能位，使能位分别为寄存器 LPTIM_CFGR 的 *OUT_WKUP_EN*、*CMPM_WKUP_EN*、*ARRM_WKUP_EN*、*EXTTRIG_WKUP_EN*、*DOWN_WKUP_EN*、*UP_WKUP_EN* 比特位，唤醒信号与相应使能位是 AND 的关系，各唤醒源之间是 OR 的关系。

20.14 中断信号

LPTIMER 的中断信号如下:

表 20-4 LPTIMER 的中断信号

中断名称	描述
DOWN 中断	编码模式下，表示计数方向由向上变为向下
UP 中断	编码模式下，表示计数方向由向下变为向上
ARROK 中断	表示 ARR 值加载完成
CMPOK 中断	表示 CMP 值加载完成
EXTTRIG 中断	表示检测到有效触发边沿
ARRM 中断	表示计数器值到达 ARR
CMPM 中断	表示计数器值与 CMP 匹配

上述中断的使能通过配置寄存器 LPTIM_IER 实现，所有中断的中断状态都可以通过寄存器 LPTIM_SR1 获得。

20.15 LPTIMER 相关寄存器描述

LPTIMER0 基地址：0x4000D000

LPTIMER1 基地址：0x4000D800

表 20-5 LPTIMER 寄存器列表

寄存器	偏移量	描述
LPTIM_ISR	0x00	状态寄存器
LPTIM_ICR	0x04	状态清除寄存器
LPTIM_IER	0x08	中断使能寄存器
LPTIM_CFGR	0x0c	配置寄存器，该寄存器需在 LPTIM_CR 寄存器的 ENABLE 清零时修改
LPTIM_CR	0x10	控制寄存器
LPTIM_CMP	0x14	比较寄存器
LPTIM_ARR	0x18	重装载寄存器
LPTIM_CNT	0x1c	计数器寄存器
LPTIM_CSR	0x20	清除状态标志寄存器，表示使用寄存器 LPTIM_ICR 清除 LPTIM_ISR 某些状态位时的是否清除完成标志
LPTIM_SR1	0x24	中断标志寄存器，中断标志位会被寄存器 LPTIM_ICR 立即清零

20.15.1 LPTIM_ISR

偏移量：0x00

复位值：0x00000180

31-9	8	7	6	5	4	3	2	1	0
RESERVED	CROK	CFGROK	DOWN	UP	ARROK	CMPOK	EXTTRIG	ARRM	CMPM
r-0h	r-1h	r-1h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h

位 31-9 RESERVED：保留。

位 8 CROK：上一次对寄存器 LPTIM_CR 的写操作状态。该位由硬件控制，写操作前需要检查上一次写操作是否完成。

- 0：正在进行写操作
- 1：上一次对 LPTIM_CR 的写操作已经完成

位 7 CFGROK：上一次对 LPTIM_CFGR 的写操作状态。该位由硬件控制，写操作前需要检查上一次写操作是否完成。

- 0：正在进行写操作
- 1：上一次对 LPTIM_CFGR 的写操作已经完成

位 6 DOWN：编码模式下计数方向由向上变为向下。

- 0：计数方向未发生由上向下的变化
- 1：计数方向由向上变为向下

可以通过写 LPTIM_ICR 寄存器清零，但是需要时间同步清零脉冲，因此无法立即清除。

位 5 UP：编码模式下计数方向由向下变为向上。

- 0：计数方向未发生由下向上的变化
- 1：计数方向由向下变为向上

可以通过写 LPTIM_ICR 寄存器清零，但是需要时间同步清零脉冲，因此无法立即清除。

位 4 ARROK：ARR 值加载状态。

- 0：未加载完成
- 1：加载完成

可以通过写 LPTIM_ICR 寄存器清零，但是需要时间同步清零脉冲，因此无法立即清除。

位 3 CMPOK：CMP 值加载状态。

- 0：未加载完成
- 1：加载完成

可以通过写 LPTIM_ICR 寄存器清零，但是需要时间同步清零脉冲，因此无法立即清除。

位 2 EXTTRIG：是否检测到有效触发边沿。

- 0：未检测到有效触发边沿
- 1：检测到有效触发边沿

可以通过写 LPTIM_ICR 寄存器清零，但是需要时间同步清零脉冲，因此无法立即清除。

位 1 ARRM：计数器值是否到达 ARR 值。

- 0：计数器值是未到达 ARR
- 1：计数器值到达 ARR

可以通过写 LPTIM_ICR 寄存器清零，但是需要时间同步清零脉冲，因此无法立即清除。

位 0 CMPM：计数器值与 CMP 值匹配状态。

- 0：计数器值与 CMP 值未匹配
- 1：计数器值与 CMP 值匹配

可以通过写 LPTIM_ICR 寄存器清零，但是需要时间同步清零脉冲，因此无法立即清除。

20.15.2 LPTIM_ICR

偏移量：0x04

复位值：0x00000000

31-7	6	5	4	3	2	1	0
RESERVED	DOWNCF	UPCF	ARROKCF	CMPOKCF	EXTTRIGCF	ARRMCF	CMPMCF
w-0h	w-0h	w-0h	w-0h	w-0h	w-0h	w-0h	w-0h

位 31-7 RESERVED：保留。

位 6 DOWNCF：清除 DOWN 标志位。软件写 1 清除标记位，该位由硬件清零。

- 0：无操作
- 1：清除操作

位 5 UPCF：清除 UP 标志位。软件写 1 清除标记位，该位由硬件清零。

- 0：无操作
- 1：清除操作

位 4 ARROKCF：清除 ARROK 标志位。软件写 1 清除标记位，该位由硬件清零。

- 0：无操作
- 1：清除操作

位 3 CMPOKCF：清除 CMPOK 标志位。软件写 1 清除标记位，该位由硬件清零。

- 0：无操作
- 1：清除操作

位 2 EXTTRIGCF：清除 EXTTRIG 标志位。软件写 1 清除标记位，该位由硬件清零。

- 0：无操作
- 1：清除操作

位 1 ARRMCF：清除 ARRM 标志位。软件写 1 清除标记位，该位由硬件清零。

- 0：无操作
- 1：清除操作

位 0 CMPMCF：清除 CMPM 标志位。软件写 1 清除标记位，该位由硬件清零。

- 0：无操作
- 1：清除操作

20.15.3 LPTIM_IER

偏移量：0x08

复位值：0x00000000

31-7	6	5	4	3	2	1	0
RESERVED	DOWNIE	UPIE	ARROKIE	CMPOKIE	EXTTRIGIE	ARRMIE	CMPMIE
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 31-7 RESERVED：保留。

位 6 DOWNIE：DOWN 中断使能。

- 0：禁用中断
- 1：使能中断

位 5 UPIE：UP 中断使能。

- 0：禁用中断
- 1：使能中断

位 4 ARROKIE：ARROK 中断使能。

- 0：禁用中断
- 1：使能中断

位 3 CMPOKIE：CMPOK 中断使能。

- 0：禁用中断
- 1：使能中断

位 2 EXTTRIGIE：EXTTRIG 中断使能。

- 0：禁用中断
- 1：使能中断

位 1 ARRMIE：ARRM 中断使能。

- 0：禁用中断
- 1：使能中断

位 0 CMPMIE：CMPM 中断使能。

- 0：禁用中断
- 1：使能中断

20.15.4 LPTIM_CFGR

偏移量：0x0c

复位值：0x00000000

31	30	29	28	27	26
RESERVED	OUT_WKUP_EN	DOWN_WKUP_EN	UP_WKUP_EN	EXTTRIG_WKUP_EN	ARRM_WKUP_EN
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h
25	**24**	**23**	**22**	**21**	**20**
CMPM_WKUP_EN	ENC	COUNTMODE	PRELOAD	WAVPOL	WAVE
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h
19	**18-17**	**16**	**15-13**	**12**	**11-9**
TIMEOUT	TRIGEN	RESERVED	TRIGSEL	RESERVED	PRESC
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h
8	**7-6**	**5**	**4-3**	**2-1**	**0**
TRGLT_ENABLE	TRGFLT	CKFLT_ENABLE	CKFLT	CKPOL	RESERVED
rw-0h	rw-0h	rw-0h	rw-0h	rw-0h	rw-0h

位 31 RESERVED：保留。

位 30 OUT_WKUP_EN：LPTIM_OUT 唤醒使能。

- 0：LPTIM_OUT 不能触发唤醒信号
- 1：LPTIM_OUT 可以触发唤醒信号

位 29 DOWN_WKUP_EN：DOWN 事件唤醒使能。

- 0：DOWN 事件不能触发唤醒信号
- 1：DOWN 事件可以触发唤醒信号

位 28 UP_WKUP_EN：计数匹配唤醒使能。

- 0：计数匹配不能触发唤醒信号
- 1：计数匹配事件触发唤醒信号

位 27 EXTTRIG_WKUP_EN：外部触发事件唤醒使能。

- 0：外部触发事件不能触发唤醒信号
- 1：外部触发事件可以触发唤醒信号

位 26 ARRM_WKUP_EN：计数溢出事件唤醒使能（ENC 模式除外）。

- 0：计数溢出不能触发唤醒信号
- 1：计数溢出事件触发唤醒信号

位 25 CMPM_WKUP_EN：计数匹配事件唤醒使能。

- 0：计数匹配不能触发唤醒信号
- 1：计数匹配事件触发唤醒信号

位 24 ENC：编码模式使能。

- 0：禁用编码模式

- 1：使能编码模式

位 23 COUNTMODE：计数模式选择。

- 0：计数器由内部时钟控制
- 1：计数器由外部时钟控制

位 22 PRELOAD：寄存器缓存使能。

- 0：ARR 和 CMP 直接由软件操作
- 1：ARR 和 CMP 由更新事件更新

位 21 WAVPOL：输出波形极性。

- 0：输出不反相
- 1：输出反相

位 20 WAVE：波形形状。

- 0：禁用 Set-once，选择 PWM 或单脉冲模式
- 1：使能 Set-once 模式

位 19 TIMEOUT：Timeout 模式使能。

- 0：禁用 Timeout 模式
- 1：使能 Timeout 模式

位 18:17 TRIGEN：外部触发使能及极性选择。

- 00：软件触发
- 01：外部触发上升沿有效
- 10：外部触发下降沿有效
- 11：外部触发双沿有效

位 16 RESERVED：保留。

位 15-13 TRIGEN：外部触发源选择。

- 000：lptim_ext_trig0
- 001：lptim_ext_trig1
- 010：lptim_ext_trig2
- 011：lptim_ext_trig3
- 100：lptim_ext_trig4
- 101：lptim_ext_trig5
- 110：lptim_ext_trig6
- 111：lptim_ext_trig7

位 12 RESERVED：保留。

位 11-9 TRIGEN：时钟分频。

- 000：/1
- 001：/2
- 010：/4
- 011：/8
- 100：/16
- 101：/32
- 110：/64
- 111：/128

位 8 TRGLT_ENABLE：触发输入滤波器使能，必须先配置滤波器长度，再使能。

- 0：禁用触发输入滤波器
- 1：使能触发输入滤波器

位 7-6 TRGFLT：触发输入滤波器配置。

- 00：无操作
- 01：使能滤波器，滤波器长度 N=2
- 10：使能滤波器，滤波器长度 N=4
- 11：使能滤波器，滤波器长度 N=8

位 5 CKFLT_ENABLE：外部时钟滤波器使能，必须先配置滤波器长度，再使能。

- 0：禁用外部时钟滤波器
- 1：使能外部时钟滤波器

位 4-3 CKFLT：外部时钟滤波器配置。

- 00：无操作
- 01：使能滤波器，滤波器长度 N=2
- 10：使能滤波器，滤波器长度 N=4
- 11：使能滤波器，滤波器长度 N=8

位 2-1 CKPOL：Encoder 模式控制。

- 00：选择 Encoder 模式 1，上升沿计数
- 01：选择 Encoder 模式 2，下降沿计数
- 10：选择 Encoder 模式 3，双沿计数
- 11：保留

位 0 RESERVED：保留。

20.15.5 LPTIM_CR

偏移量：0x10

复位值：0x00000000

31-3	2	1	0
RESERVED	CNTSTRT	SNGSTRT	ENABLE
rw-0h	rw-0h	rw-0h	rw-0h

位 31-7 RESERVED：保留。

位 2 CNTSTRT：连续计数模式使能。

- 0：不使能
- 1：使能连续计数模式，写 1 开始连续计数模式，若在连续计数模式过程中置位 SNGSTRT，则在下一次计数到 ARR 时停止计数（切换到单次计数模式）。该比特位需在 ENABLE 置位后修改。

位 1 SNGSTRT：单次计数模式使能。

- 0：不使能
- 1：使能单次计数模式，写 1 开始单次计数模式，若在单次计数模式过程中置位 CNTSTRT，则在下一次计数到 ARR 时继续计数（切换到连续计数模式）。该比特位需在 ENABLE 置位后修改。

位 0 ENABLE：LPTIMER 使能。

- 0：禁用 LPTIMER
- 1：使能 LPTIMER

20.15.6 LPTIM_CMP

偏移量：0x14

复位值：0x00000000

31-16	15-0
RESERVED	CMP
rw-0h	rw-0h

位 31-16 RESERVED：保留。

位 15-0 CMP：比较值，需在寄存器 LPTIM_CR 的 ENABLE 置位后才能修改。

20.15.7 LPTIM_ARR

偏移量：0x18

复位值：0x00000001

31-16	15-0
RESERVED	ARR
rw-0h	rw-0h

位 31-16 RESERVED：保留。

位 15-0 ARR：重装载值，需在寄存器 LPTIM_CR 的 ENABLE 置位后才能修改。

20.15.8 LPTIM_CNT

偏移量：0x1c

复位值：0x00000000

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

位 31-16 RESERVED：保留。

位 15-0 CNT：计数结果，读该值时，连续两次读到的结果一致才算有效。

20.15.9 LPTIM_CSR

偏移量：0x20

复位值：0x0000001f

31-5	4	3	2	1	0
RESERVED	DOWN_CLR_DONE	UP_CLR_DONE	EXTTRIG_CLR_DONE	ARRM_CLR_DONE	CMPM_CLR_DONE
r-0h	r-0h	r-0h	r-0h	r-0h	r-0h

位 31-5 RESERVED：保留。

位 4 DOWN_CLR_DONE：DOWN 清除完成。

- 0：正在清除 DOWN 标志位
- 1：清除成功

位 3 UP_CLR_DONE：UP 清除完成。

- 0：正在清除 UP 标志位
- 1：清除成功

位 2 EXTTRIG_CLR_DONE：EXTTRIG 清除完成。

- 0：正在清除 EXTTRIG 标志位
- 1：清除成功

位 1 ARRM_CLR_DONE：ARRM 清除完成。

- 0：正在清除 ARRM 标志位
- 1：清除成功

位 0 CMPM _CLR_DONE：CMPM 清除完成。

- 0：正在清除 CMPM 标志位
- 1：清除成功

20.15.10 LPTIM_SR1

偏移量：0x24

复位值：0x00000000

31-7	6	5	4	3	2	1	0
RESERVED	DOWN	UP	ARROK	CMPOK	EXTTRIG	ARRM	CMPM
r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h	r-0h

位 31-7 RESERVED：保留。

位 6 DOWN：编码模式下计数方向由向上变为向下。

- 0：计数方向未发生由上向下的变化
- 1：计数方向由向上变为向下

位 5 UP：编码模式下计数方向由向下变为向上。

- 0：计数方向未发生由下向上的变化
- 1：计数方向由向下变为向上

位 4 ARROK：ARR 值加载状态。

- 0：未加载完成
- 1：加载完成

位 3 CMPOK：CMP 值加载状态。

- 0：未加载完成
- 1：加载完成

位 2 EXTTRIG：是否检测到有效触发边沿。

- 0：未检测到有效触发边沿
- 1：检测到有效触发边沿

位 1 ARRM：计数器值是否到达 ARR 值。

- 0：计数器值是未到达 ARR
- 1：计数器值到达 ARR

位 0 CMPM：计数器值与 CMP 值匹配状态。

- 0：计数器值与 CMP 值未匹配
- 1：计数器值与 CMP 值匹配