HK32F030M 用户手册

版本：1.5

发布日期：2022-03-29

深圳市航顺芯片技术研发有限公司

http://www.hsxp-hk.com

前言

编写目的

本文档介绍了 HK32F030M 系列芯片的功能框图、存储器映射、Flash、中断和事件等功能以及各功能模块的寄存器描述，旨在帮助用户快速开发 HK32F030M 的应用及产品。

读者对象

本文适用于以下读者：

- 开发工程师
- 芯片测试工程师

版本说明

本文档对应的产品系列为 HK32F030M 系列芯片。

修订记录

版本	日期	修订内容
1.0	2020/02/21	首次发布。
1.2	2021/04/16	更新本文档的组织结构。
1.3	2021/12/20	添加了各章节的功能描述内容。
1.4	2022/02/24	更新了“8.2 GPIO 功能描述”、“11.7 内部参考电压”等。
1.5	2022/03/29	删除了 ADC 章节里面没有的增益补偿功能，以及不该有的寄存器（ADC_Gain 和 ADC_DRx），另外还更正了全文中的一些小错误。

目录

1 简介

本文档为 HK32F030M 系列芯片的用户手册。HK32F030M 系列芯片是由深圳市航顺芯片技术研发有限公司研发的经济型 MCU 芯片，包括以下型号：

- HK32F030MF4U6
- HK32F030MF4P6
- HK32F030MD4P6
- HK32F030MJ4M6

用户可以查看《HK32F030M 数据手册》，进一步了解 HK32F030M MCU 的功能特性，如外设接口、电器特性、管脚封装等。

2 系统及存储器概述

本章介绍了 HK32F030M MCU 的系统架构和内部存储器。

2.1 系统架构

HK32F030M MCU 主要包括以下几个模块：

- 主模块：
  - Cortex®-M0 内核和 AHB-Lite 总线
- 从模块：
  - 内部 SRAM
  - 内部 Flash 存储器
  - AHB-Lite 到 APB 的桥，所有的外设都挂在 APB 总线上
  - 连接于 AHB-Lite 总线的 GPIO 口

以 HK32F030MF4P6 为例，HK32F030M MCU 系统架构如下图所示：

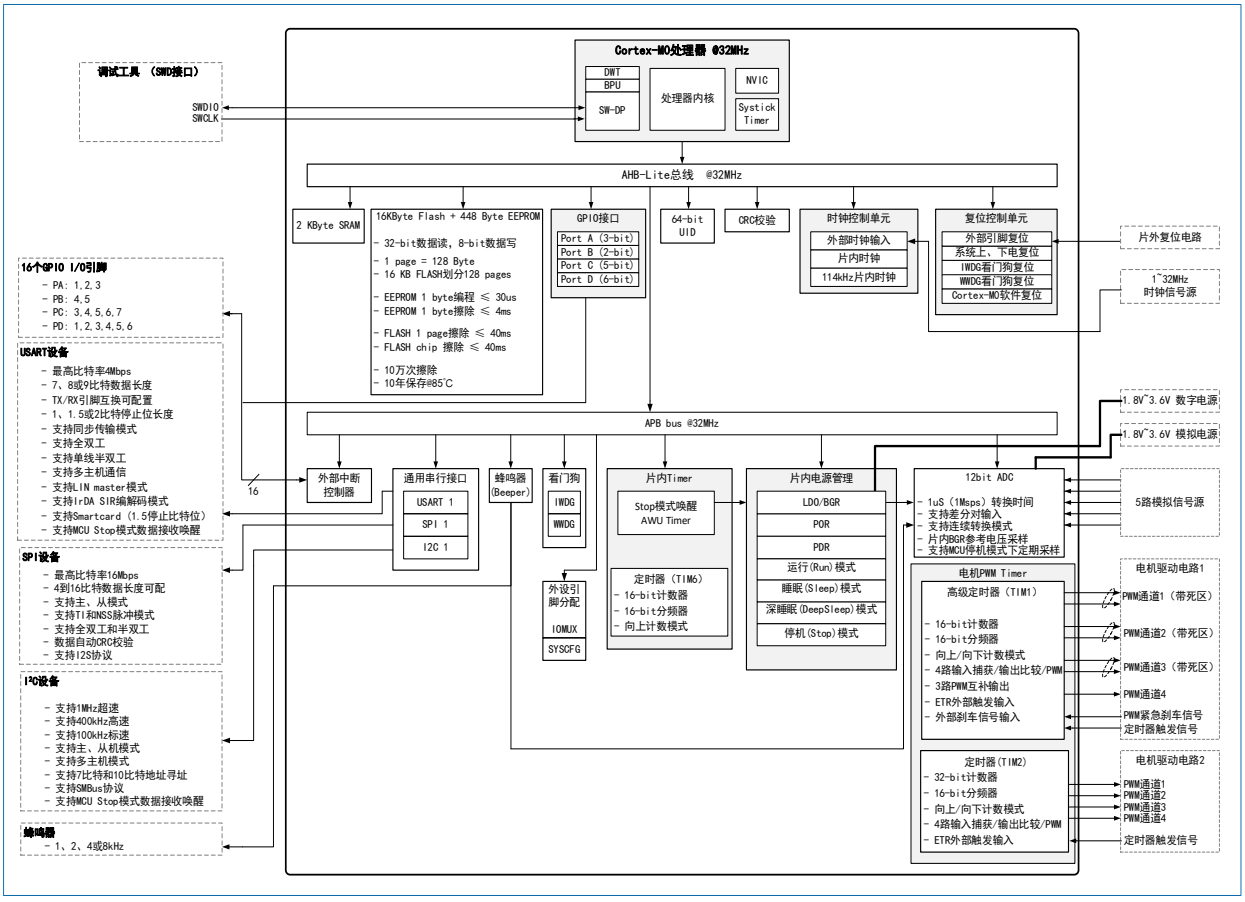


图 2-1 HK32F030MF4P6 系统架构图

2.1.1 总线架构

- ICode 总线

该总线将 Cortex - M0 内核的指令总线与闪存指令接口相连接。指令预取在此总线上完成。

- DCode 总线

该总线将 Cortex - M0 内核的 DCode 总线与闪存存储器的数据接口相连接（常量加载和调试访问）。

- AHB-lite 到 APB 桥

AHB-lite 到 APB 桥在 AHB-lite 与 APB 总线间提供同步连接。

有关连接到桥的不同外设的地址映射请参见图 2-2。

在每次复位之后，所有的外设时钟都关闭（除了 SRAM 及 FLIFT 外）。在用一个外设前，你必须打开相应的 RCC_AHBENR、RCC_APBxENR 寄存器中时钟使能位。

说明：当对 APB 寄存器进行 8 位或者 16 位访问时，该访问会被自动转换成 32 位的访问：桥会自动将 16 位或者 8 位的数据扩展以配合 32 位的宽度。

2.2 存储器映射及寄存器编址

程序存储器、数据存储器、寄存器及 I/O 口统一编址，其线性地址空间达到 4G Byte。HK32F030M MCU 支持小端模式的数据存储，即数据的低字节存放于低地址，数据的高字节存放于高地址。

存储器的寻址空间可分成8 块，每块512M Byte。存储器内的保留区是指暂时未分配给片上存储器和外设的地址空间。

以 HK32F030MF4P6 为例，HK32F030M MCU 存储器映射如下图所示：

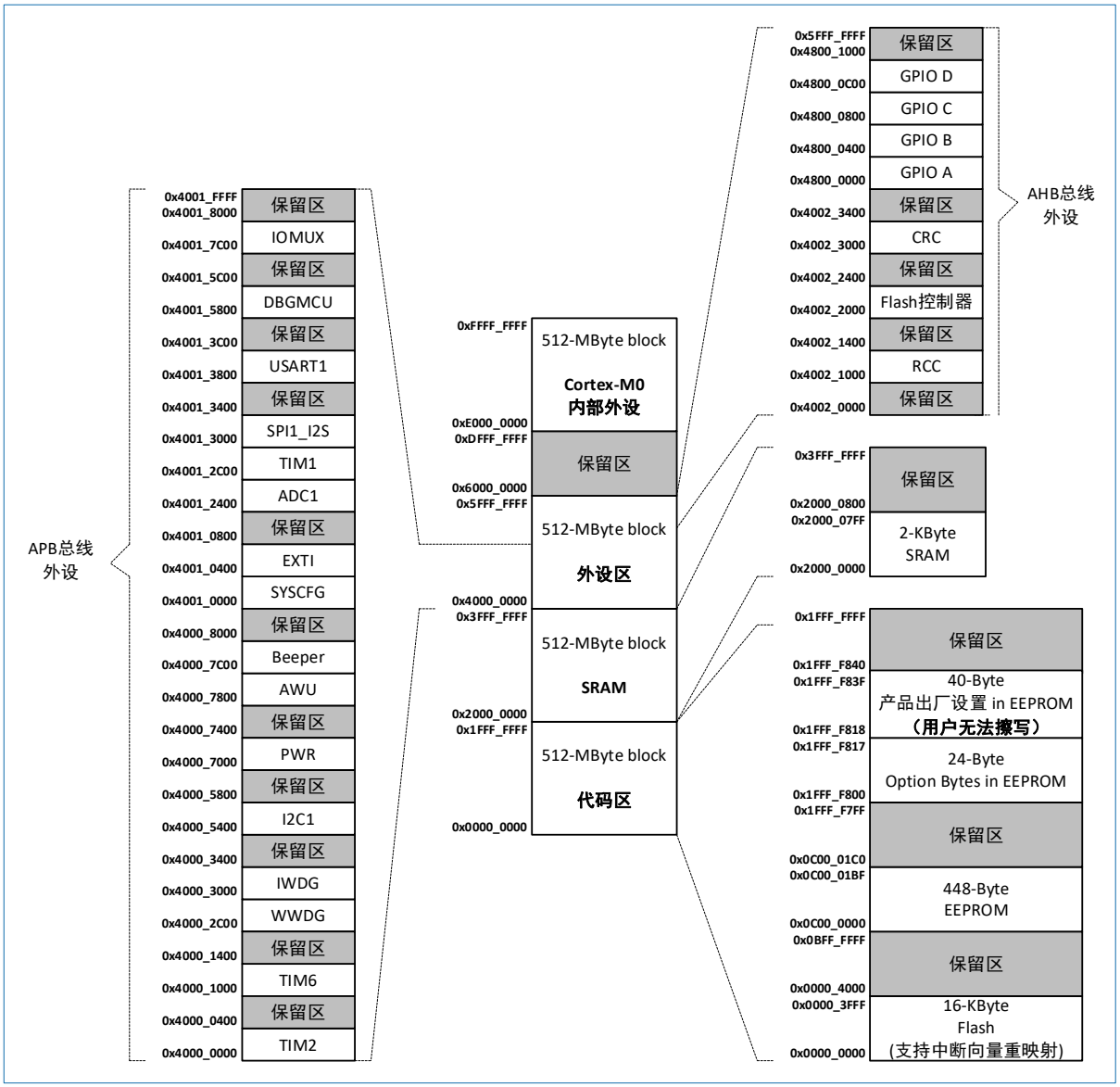


图 2-2 HK32F030MF4P6 存储器映射

2.3 SRAM

HK32F030M MCU 内置 2 Kbyte 的 SRAM，起始地址是 0x2000 0000。SRAM 可以字节（8 位）、半字（16 位）或字（32 位）方式进行访问。CPU 可使用最快的系统时钟且不插入等待周期访问 SRAM。

SRAM 不支持硬件奇偶校验。

2.4 启动配置

系统复位等待至启动延迟时间结束之后，CPU 获取 Flash 地址 0x0000 0000 中存储的堆栈顶地址，然后从 Flash 地址 0x0000 0004 处开始执行代码。

系统启动后，应用程序可通过修改 SYSCFG_CFGR1 寄存器中的 MEM_MODE 位来重新映射存储器地址。与 Cortex-M3®和 Cortex®-M4 内核不同，Cortex®-M0 内核不支持中断向量表（Interrupt Vector Table，IVT）重映射，但 HK32F030M MCU 支持通过配置 INT_VEC_OFFSET 寄存器实现 IVT 重映射功能。

例如，实现在线升级（In Application Programming，IAP）功能，一种方法是用户将中断向量表重映射到 SRAM：

1. 将中断向量表从 Flash 拷贝到 SRAM 的起始地址 0x2000 0000。
2. 配置 SYSCFG_CFGR1 寄存器的 MEM_MODE[1:0]，以实现 SRAM 映射到 Flash 的地址 0x0000 0000。
3. 当有中断发生时，CPU 从重映射到 SRAM 中的 IVT 取中断服务程序地址，然后跳转到 Flash 中执行中断服务程序。

另外一种方法是通过配置 INT_VEC_OFFSET 寄存器将 IVT 重映射到 Flash 地址。配置完成之后，若有中断发生，CPU 从重映射到 Flash 中的 IVT 取中断服务程序地址，然后跳到 Flash 中执行中断服务程序。

3 Flash

Flash 接口遵循 AHB 协议执行指令和数据存取。

3.1 Flash 特性

- Flash 结构
  - 最高 16K 字节主 Flash 块
    - 应用程序区
    - 用户数据区
  - 信息块，其包括：
    - 选项字节（Option byte）：内含硬件及存储保护用户配置选项。
    - 448 字节 EEPROM
- Flash 编程/擦除操作
- 访问/写保护
- Flash 访问位宽：支持半字（16 位）和字节（8 位）写；32 位读
- 低功耗模式

3.2 Flash 功能

3.2.1 Flash 结构

Flash 空间由 32 位宽的存储单元组成，可存储代码和数据。

主 Flash 块可分为 128 页，每页 128 字节。

Flash 选项字节域共有 24 个字节。

表 3-1 Flash 结构

地址	大小（字节）	页号	描述
0x0800 0000-0x0800 007F	128 Byte	Page0	主 Flash 块
0x0800 0080-0x0800 00FF	128 Byte	Page1	
0x0800 0100-0x0800 017F	128 Byte	Page2	
0x0800 0180-0x0800 01FF	128 Byte	Page3	
……	……	……	
0x0800 1E00-0x0800 1E7F	128 Byte	Page60	
0x0800 1E80-0x0800 1EFF	128 Byte	Page61	
0x0800 1F00-0x0800 1F7F	128 Byte	Page62	
0x0800 1F80-0x0800 1FFF	128 Byte	Page63	
……	……	……	
0x0800 3E00-0x0800 3E7F	128 Byte	Page124	

地址	大小（字节）	页号	描述
0x0800 3E80-0x0800 3EFF	128 Byte	Page125	
0x0800 3F00-0x0800 3F7F	128 Byte	Page126	
0x0800 3F80-0x0800 3FFF	128 Byte	Page127	
0x1FFF F800-0x1FFF F817	24 Byte	-	选项字
0x1FFF F818-0x1FFF F83F	40 Byte	-	系统配置

3.2.2 读操作

嵌入式 Flash 模块可以直接寻址访问。任何对该 Flash 内容的读操作都须经过专门的判断过程。

指令和数据的访问都是通过 AHB 总线完成，并按照 Flash 访问控制寄存器（Flash_ACR）所指定的等待周期进行访问。

3.2.3 读保护

将选项字节中的RDP 字节置位，然后重新复位，读保护功能则被激活。系统存储区不受读保护字节的影响，但该区域不允许编程和擦除操作。Flash 存储器的读保护级别和 RDP 选项字节及其按位取反之后的内容的对应关系，如下表：

表 3-2 读保护级别/ RDP 字节及其按位取反值的对应关系

RDP 字节值	RDP 按位取反之后的值	读保护级别
0xAA	0x55	Level 0
任意值，除 0xAA 和 0xCC 外	任意值（不要求按位取反），除 0x55 和 0x33 外	Level 1（默认）
0xCC	0x33	Level 2

读保护状态包括三个级别：

- Level 0：无保护

  允许对主 Flash 区域和选项字节进行读写和擦除操作。

- Level 1：读保护

  这是 RDP 选项字节被擦除之后的默认保护级别。对应的 RDP 值为除 0xAA 和 0xCC 以外的任意值或者其按位取反之后的值。

  - 用户模式：在用户模式下执行的代码允许对主 Flash 和选项字节做全部操作。
  - 调试模式：包括 boot RAM。在调试模式下或运行在 boot RAM 状态下，不允许访问主 Flash 区和备份寄存器。在调试模式下，任何简单的读访问都会引起总线错误并引发硬件错误中断。主 Flash 区也禁止写和擦除操作，以防范恶意程序修改代码。当 RDP 字节的内容被重新改为 Level 0（0xAA）的级别时，CPU 硬件先执行整片擦除操作。

- Level 2：不支持调试

  Level 2 包含了 Level 1 的保护功能，且 Cortex M0 的调试接口被禁止了，也不支持从 RAM 启动、系统区启动等功能。

  在用户模式下，允许对主 Flash 区进行读写和擦除操作；但选项字节区仅支持读取和写入操作，不支持擦除操作。

  在 Level 2 级别时，不能改写 RDP 字节，因此 Level 2 保护级别不能被清除。设置为 Level2 是不

可恢复的操作。当试图改写 RDP 字节时，FLASH_SR 寄存器中的保护错误标志 WRPRTERR 会被置位并引发一个中断。

表 3-3 不同工作模式下保护级别和保护状态的对应关系

区域	保护级别	用户代码执行			调试/从 RAM 或从系统区域启动		
		读	写	擦除	读	写	擦除
主 Flash 区域	1	允许	允许	允许	禁止	禁止	禁止 (3)
	2	允许	允许	允许	禁止 (1)	禁止 (1)	禁止 (1)
系统区域	1	允许	禁止	禁止	允许	禁止	禁止
	2	允许	禁止	禁止	禁止 (1)	禁止 (1)	禁止 (1)
选项字节	1	允许	允许 (2)	允许	允许	允许 (3)	允许
	2	允许	允许 (3)	禁止	禁止 (1)	禁止 (1)	禁止 (1)
备份寄存器	1	允许	允许	允许	禁止	禁止	允许
	2	允许	允许	允许	禁止 (1)	禁止 (1)	禁止 (1)

(1). 当使能 Level2 保护级别，调试口被禁止从 RAM/系统启动。

(2). 当 RDP 被改成不保护时，主 Flash 会被擦除。

(3). 除 RDP 以外的其他选项字节均能被再次编程。

3.2.3.1 改变读保护级别

修改 RDP 的值（除 0xCC 以外的值）就能将读保护级别从 Level 0 级迁移到 Level 1 级别。将 RDP 写入 0xCC，就可以直接进入 Level 2 级别。从 level 1 进入到 Level 0，一定会经过整片擦除阶段。因为在修改 RDP 成功进入到 Level 0 之前，MCU 已经启动了整片擦除。

3.2.4 写保护

写保护是通过配置选项字节中的 WRP 位，然后重新复位系统以使能写保护功能。一共 32 位写保护控制位，每个 WRP 比特置位对应保护 1 Kbyte 主 Flash。该系列的 Flash 是 16 Kbyte，WRP0\WRP1 一共 16 位控制有效。

如果写入或擦除一个受写保护的扇区，会引起 FLASH_SR 中的 WRPRTERR 标志位被置位。

写保护的解除

以下是解除写保护操作的实例：

1. 置位 FLASH_CR 中的 OPTER 位擦除整个选项字节区域。
2. 向 RDP 写入 0xAA 从而解除所有保护，这会引起整片擦除。

选项字节的写保护

选项字节默认被写保护且随时可读。必须先向 FLASH_OPTKEYR 寄存器顺序写入关键字，才能对选项字节进行写/擦除操作。填入正确的关键字会引起 FLASH_CR 中的 OPTWRE 置位，表明解锁成功；通过对 OPTWRE 位清零，能够禁止对选项字节的写操作。

3.2.5 主 Flash 写和擦除操作

电路编程（In Circuit Programing，ICP）使用 SWD 或 Bootloader 的方法在线改变 Flash 的内容，将用户代码烧录到 MCU 中。ICP 提供了一种简单高效的方法，免除了烧写芯片时的芯片装夹等问题。

与 ICP 方法不同的是，IAP 能够使用 MCU 支持的任何通信接口下载程序或者数据。IAP 允许用户在运行程序的过程中重写应用程序，前提是 IAP 的烧录引导程序已经写入 MCU。

写和擦除操作在整个产品工作电压范围内都可以完成。该操作涉及以下寄存器的配置：

- Flash 关键字寄存器（FLASH_KEYR）
- Flash 状态寄存器（FLASH_SR）
- Flash 控制寄存器（FLASH_CR）
- Flash 地址寄存器（FLASH_AR）
- 写保护寄存器（FLASH_WRPR）

在写/擦除 Flash 时，不能对 Flash 进行取指或数据访问，并且任何中断请求亦不会被响应。此时，中断会被 CPU 内核挂起，直到写/擦除完成之后才会响应中断。只要 CPU 不访问 Flash 空间，进行中的 Flash 写操作不会影响 CPU 的运行。即在对 Flash 进行写/擦除操作时，任何对 Flash 的访问都会令总线停顿，直到写/擦操作完成后才会继续执行 Flash 的访问。

在对 Flash 进行写/擦除操作时，内部 RC 振荡器（HSI）必须处于开启状态。

3.2.5.1 主 Flash 空间的解锁

复位后，Flash 存储器默认处于受保护状态，以避免意外擦除。FLASH_CR 寄存器的值通常不允许改写。只有对 FLASH_KEYR 寄存器进行解锁操作后，才具有对 FLASH_CR 寄存器的访问权限。FLASH_KEYR 寄存器的解锁操作包括以下步骤：

1. 向 FLASH_KEYR 寄存器写入关键字 KEY1=0x45670123；
2. 向 FLASH_KEYR 寄存器写入关键字 KEY2=0xCDEF89AB。

错误的操作顺序将会锁死 FLASH_CR 直至下次复位。当写入关键字错误时，会由总线错误触发一次硬件错误中断。

- 如果 KEY1 出错，就会立即触发中断。
- 如果 KEY1 正确且 KEY2 错误时，就会在 KEY2 错的时刻触发中断。

3.2.5.2 主 Flash 擦除

主 Flash 存储器可以按页为单位擦除，也可以整片擦除。

注意：

HK32F030M 系列芯片擦除后为随机值。

- **擦除页**

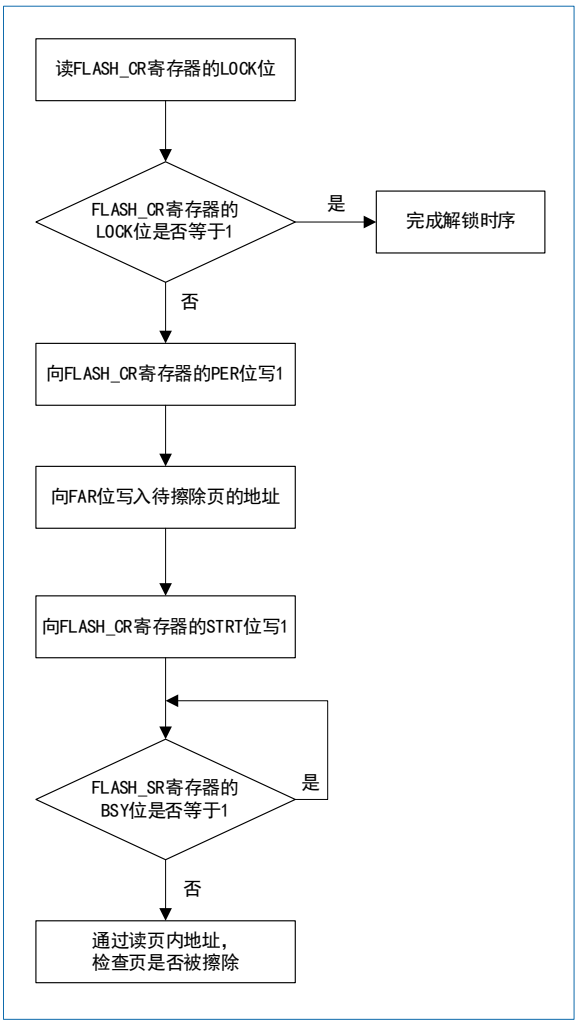


图 3-1 Flash 擦除页的流程

擦除页的操作步骤：

1. 检查 FLASH_SR 寄存器中的 BSY 位，以确认上次操作已经结束。
2. 将 FLASH_CR 寄存器中的 PER 位置为 1，以选择按页擦除。
3. 写 FLASH_AR 寄存器的 FAR 位，写入待擦除页的地址。
4. 将 FLASH_CR 寄存器中的 STRT 位置为 1，以启动擦除操作。
5. 将 FLASH_CR 寄存器中的 PER 位置 0，以恢复默认值。
6. 等待 FLASH_SR 中的 BSY 变为 0，表明擦除操作完成。
7. 检查 FLASH_SR 寄存器的 EOP 标志（若 Flash 擦除成功会置位 EOP），然后软件清除该标志位。

- **整片擦除**

整片擦除命令可以一次擦除整个 Flash 扇区。整片擦除的具体步骤如下：

1. 检查 FLASH_SR 寄存器的 BSY 位，以确认上次操作已经结束。
2. 将 FLASH_CR 寄存器中的 MER 位置 1，以选择整片擦除。
3. 将 FLASH_CR 寄存器中的 STRT 位置为 1，以启动擦除操作。
4. 将 FLASH_CR 寄存器中的 MER 位置 0，以选择整片擦除类型。

5. 等待 FLASH_SR 中的 BSY 位置 0，整片擦除操作结束。

6. 检查 FLASH_SR 寄存器的 EOP 标志位（如果 Flash 擦除成功会置位 EOP），然后软件清除该标志位。

说明：整片擦除命令对信息块不起作用。

3.2.5.3 主 Flash 编程

主 Flash 一次可以编程 16 位（半字）。解锁 Flash 后，当 FLASH_CR 寄存器中的 PG 位为 1 时，则向指定的地址写入半字的数据，即完成一次 Flash 编程操作。

注意：

若 Flash 处于解锁状态，FLASH_CR 寄存器中的 PG 位为 1 时，写入的数据长度不是半字，会引起硬件错误中断。

主 Flash 编程的流程如下图所示：

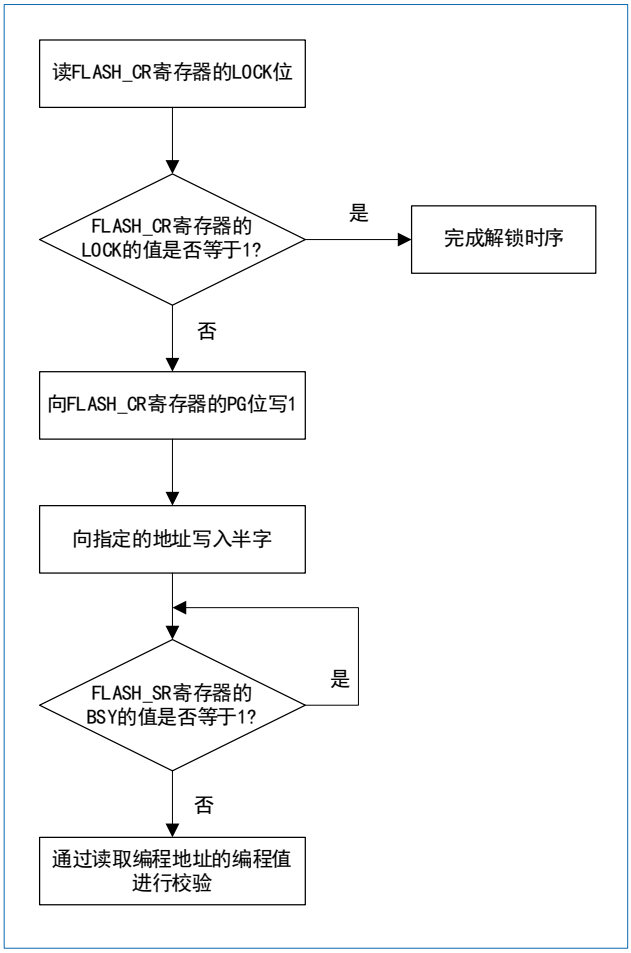


图 3-2 对 Flash 的编程

当待编程地址所对应的 FLASH_WRPR 寄存器的写保护位生效时，不允许编程操作，否则会产生编程错误告警。编程操作结束后，FLASH_SR 寄存器的 EOP 位会提示该操作是否成功。

主 Flash 存储器的标准编程流程如下：

1. 检查 FLASH_SR 中的 BSY 位，以确认上次操作已经结束。

2. 将 FLASH_CR 寄存器中的 PG 位置为 1，以写入 Flash。

3. 根据配置，以半字为单位向目标地址写入数据。

4. 在完成 Flash 的数据写入后，将 FLASH_CR 寄存器中的 PG 位置为 0。
5. 等待 FLASH_SR 寄存器中的 BSY 变为 0。
6. 检查 FLASH_SR 寄存器的 EOP 标志位（如果 Flash 编程成功会置位 EOP），然后软件清除该标志位。

3.2.6 Flash 中断

表 3-4 Flash 中断事件和事件标志

中断事件	事件标志	使能控制位
操作结束	EOP	EOPIE
写保护错误	WRPRTERR	ERRIE

3.3 Flash 选项字节

Flash 选项字节域共有 24 个字节，根据用户的应用需求进行配置。

一个 32 位选项字可划分成如下表所示的选项字节：

表 3-5 选项字划分成选项字节

[31:24]	[23:16]	[15:8]	[7:0]
选项字节 1 按位取反	选项字节 1	选项字节 0 按位取反	选项字节 0

选项字若有内容更新，需复位系统才能生效。前面 4 个选项字是以按位取反之后和选项字节组合的形式存储，如下表所示。

表 3-6 选项字节结构

地址	[31:24]	[23:16]	[15:8]	[7:0]
0X1FFF F800	nUSER	USER	nRDP	RDP
0X1FFF F804	nDATA1	DATA1	nDATA0	DATA0
0X1FFF F808	nWRP1	WRP1	nWRP0	WRP0
0X1FFF F80C	nWRP3	WRP3	nWRP2	WRP2
0X1FFF F810	IWDG_INI_KEY[15:0]		保留	IWDG_RL_IV[11:0]
0X1FFF F814	DBG_CLK_CTL[15:0]		LSI_LP_CTL[15:0]	

表 3-7 选项字节描述

地址	位域	选项字节描述
0x1FFF F800	31:24	nUSER：USER 按位取反
	23:16	USER：用户选项字节 USER 内容存储于 FLASH_OBR[15:8]，用于配置如下特性： ● 选择硬件或软件看门狗事件。 ● 当进入停机模式时，复位事件。 ○ 位 23：保留 ○ 位 22：保留

地址	位域	选项字节描述
		○ 位 21：保留 ○ 位 20：保留 ○ 位 19：保留 ○ 位 18：保留 ○ 位 17：nRST_STOP - 0：当进入停机模式产生复位 - 1：不产生复位 ○ 位 16：WDG_SW - 0：硬件看门狗 - 1：软件看门狗
	15:8	nRDP：RDP 按位取反
	7:0	RDP：Flash 读保护选项字节 该字节的值定义了 Flash 读保护级别。 ● 0xAA：级别 0 ● 0xXX（除 0xAA 和 0xCC 取值外）：级别 1 ● 0xCC：级别 2
0x1FFF F804	31:0	DATAx：用户数据 ● 位[31:24]：nDATA1 ● 位[23:16]：DATA1（存于FLASH_OBR[31:24]） ● 位[15:8]：nDATA0 ● 位[7:0]：DATA0（存于FLASH_OBR[23:16]）
0x1FFF F808	31:0	WRPx：Flash 写保护选项字节 ● 位[31:24]：nWRP1 ● 位[23:16]：WRP1（存于 FLASH_WRPR[15:8]） ● 位[15:8]：nWRP0 ● 位[7:0]：WRP0（存于 FLASH_WRPR[7:0]） ○ 0：写保护使能 ○ 1：写保护禁能 Flash的写保护范围是按照每一位（Bit）对应 4 页（Page）进行控制。WRP0 作用于0～31 页；WRP1作用于32~63 页。
0x1FFF F80C	31:0	WRPx：Flash 写保护选项字节 ● 位[31:24]：nWRP3 ● 位[23:16]：WRP3（存于 FLASH_WRPR[31:24]） ● 位[15:8]：nWRP2 ● 位[7:0]：WRP2（存于 FLASH_WRPR[23:16]） ○ 0：写保护使能 ○ 1：写保护禁能 Flash的写保护范围是按照每一位（Bit）对应 4 页（Page）进行控制。WRP2 作用于64～95 页。WRP3 作用于96～127 页。
0x1FFF F810	31:16	IWDG_INI_KEY：决定IWDG_RL_IV是否生效。 当IWDG_INI_KEY[31:16]为0x5b1e时，IWDG_RL_IV配置有效，否则无效。
	15:12	保留值
	11:0	IWDG_RL_IV[11:0]：IWDG 重载值。

地址	位域	选项字节描述
		当IWDG_INI_KEY[15:0]为0x5b1e时，IWDG_RL_IV配置有效，否则无效。 IWDG 重载值计算公式参见“15 独立看门狗（IWDG）章节”。 *需要注意的是：* *1、 使用该功能，必须程序中要对 IWDG 喂狗；* *2、 如果没有对 IWDG 喂狗，且该值设置太小，IWDG 复位间隔时间短，可能会导致芯片无法正常烧录。*
0x1FFF F814	31:16	DBG_CLK_CTL：关闭或打开 CPU 内部 Debug 时钟。 当存储的值为0x12de时，关闭CPU内部Debug时钟，SWD 将不能访问 MCU；否则保持调试时钟为打开状态。
	15:0	LSI_LP_CTL：决定 MCU 在使能IWDG后再进入停机模式时，是否需要被IWDG周期唤醒。 ● 该位域配置为 0x369c时： MCU 进入停机（Stop）模式后，可根据 LSION的设置关闭 LSI。 MCU 被唤醒后，LSI 恢复为进入停机模式之前的状态。 ● 若未配置该位域： 在使能IWDG 后再进入停机模式，MCU 会被 IWDG 周期唤醒。

每次系统复位后，信息块数据被读取和存储到相应的选项字节寄存器（FLASH_OBR）和闪存保护寄存器（FLASH_WRPR）中。

每个选项字节都有其值按位取反之后的值存放在信息块中，其目的用于校验选项字节的正确性。当选项字节装载后，CPU 会检查选项字节的正确性。若选项字节与其按位取反之后的值比较不一致时，会产生选项字节校验错（OPTERR）信息。当产生 OPTERR 后，CPU 会强制将相应的选项字节值变为 0xFF。当选项字节与其按位取反之后的值都为 0xFF 时，CPU 不会比较其与按位取反之后的值的差异。

注意：

选项字节的擦除和编程操作在整个产品工作电压范围内都可以完成。

3.3.1 选项字节擦除

选项字节擦除操作涉及以下 5 个寄存器的配置：

- Flash 关键字寄存器（FLASH_OPTKEYR）
- Flash 状态寄存器（FLASH_SR）
- Flash 控制寄存器（FLASH_CR）
- Flash 地址寄存器（FLASH_AR）

擦除选项字节前的准备：

需要先对 FLASH_KEYR 和 FLASH_OPTKEYR 写入关键字 KEY 以解锁 Flash，然后进行选项字节擦除操作。

选项字节是按字节擦除的，将目标地址擦除之后的值为随机值，步骤如下：

1. 检查 FLASH_SR 寄存器中的 BSY 位，以确保上次操作结束。
2. 将 FLASH_CR 寄存器中的 OPTWRE 位置 1，以允许改写选项字节。
3. 置 FLASH_CR 寄存器中的 OPTER 位为 1，以选择按字节擦除。

注意：将待擦除的地址写入 FLASH_AR 寄存器，待擦除的地址必须在里已有的地址，否则可能会出现不可预计的错误。

4. 将 FLASH_CR 寄存器中的 STRT 位置为 1，以启动擦除操作。
5. 等待 FLASH_SR 中的 BSY 位变成 0，表明擦除操作结束。
6. 将 FLASH_CR 寄存器中的 OPTER 位置为 0，以恢复默认值。

3.3.2 选项字节编程

选项字节区按照半字为单位进行编程。该区大小总共 12 个半字，包括：

- 1 个读保护
- 1 个用户数据
- 2 个硬件配置
- 4 个写保护
- 2 个 IWDG 配置
- 2 个调试时钟控制

选项字节编程前的准备：

需要先对 FLASH_KEYR 和 FLASH_OPTKEYR 写入关键字 KEY 以解锁 Flash，然后进行选项字节编程操作。LSB 值会自动转化为 MSB，以适应选项字节的位定义。

选项字节编程的步骤如下：

1. 检查 FLASH_SR 寄存器中的 BSY 位，以确保上次操作结束。
2. 将 FLASH_CR 寄存器中的 OPTWRE 位置 1，以使能选项字节编程。
3. 将 FLASH_CR 寄存器中的 OPTPG 位置为 1，以选择半字方式写入。
4. 写数据（半字）到目标地址。
5. 等待 FLASH_SR 中的 BSY 位为 0，表示编程操作结束。
6. 将 FLASH_CR 寄存器中的 OPTPG 位置为 0，以恢复默认值。

当读保护选项字节由保护状态变成非保护状态时，会执行一次整片擦除，然后才允许改写读保护位数据。若用户仅想改写选项字节区以外的数据，则不会引起整片擦除，该机制用于保护 Flash 的内容。

3.4 EEPROM

EEPROM 共有 448 个字节可用，其地址范围是 0x0C00 0000~0x0C00 01BF。

EEPROM 的擦除和编程操作在整个产品工作电压范围内都可以完成。

3.4.1 EEPROM 的擦除

该操作涉及以下 5 个寄存器的配置：

- Flash 关键字寄存器（FLASH_KEYR）
- Flash 状态寄存器（FLASH_SR）
- Flash 控制寄存器（FLASH_CR）
- Flash 地址寄存器（FLASH_AR）
- Flash EEPROM 控制寄存器（FLASH_ECR）

EEPROM 擦除前的准备：

需要先对 FLASH_KEYR 写入关键字 KEY 以解锁 Flash，然后进行 EEPROM 擦除操作。

EEPROM 是按字节擦除的，将目标地址擦除之后的值为随机值，步骤如下：

1. 检查 FLASH_SR 寄存器中的 BSY 位，以确保上次操作结束。
2. 将 FLASH_ECR 寄存器中的 EEPROM_ER 位置 1，准备按字节擦除 EEPROM 空间。
3. 将待擦除的 EEPROM 的地址写入 FLASH_AR 寄存器。
4. 将 FLASH_CR 寄存器中的 STRT 位置为 1，以启动擦除操作。
5. 等待 FLASH_SR 中的 BSY 位变成 0，表明擦除操作结束。
6. 将 FLASH_ECR 寄存器中的 EEPROM_ER 位置为 0，以恢复默认值。

3.4.2 EEPROM 的编程

EEPROM 编程前的准备：

需要先对 FLASH_KEYR 写入关键字 KEY 以解锁 Flash，然后进行 EEPROM 编程操作。

EEPROM 编程的步骤如下：

1. 检查 FLASH_SR 寄存器中的 BSY 位，以确保上次操作结束。
2. 将 FLASH_ECR 寄存器中的 EEPROM_BPG 位置 1，准备按字节编程 EEPROM 空间。
3. 写数据（字节）到目标地址。
4. 等待 FLASH_SR 中的 BSY 位为 0，表示编程操作结束。
5. 将 FLASH_ECR 寄存器中的 EEPROM_BPG 位置为 0，以恢复默认值。

3.5 Flash 寄存器

基地址：0x4002 2000

空间大小：0x400

3.5.1 Flash 访问控制寄存器（FLASH_ACR）

地址偏移：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res													LATENCY[2:0]		
													rw		

位 31:3	Res：保留 必须保持复位值。
位 2:0	LATENCY[2:0]：等待周期 本位预设 HCLK 周期和 Flash 访问时间的比率关系。 • 当 1.8V ≤V_{DD} ≤ 2.4V 时：

	○ 000：0 时钟等待周期（适用于 HCLK ≤ 16 MHz） ○ 001：1 个时钟等待周期（适用于 16 MHz < HCLK ≤ 32 MHz） ○ 010：保留 • 当 2.4V < V_{DD} ≤ 3.6V 时： ○ 000：0 等待周期（适用于 HCLK ≤ 24 MHz） ○ 001：1 个时钟等待周期（适用于 24 MHz < HCLK ≤ 48 MHz） ○ 010：保留 • 011：3 个时钟等待周期 • 100：7 个时钟等待周期 • 101：9 个时钟等待周期 • 110：19 个时钟等待周期 • 111：39 个时钟等待周期 通过配置 LATENCY，能使 CPU 在极低的频率下运行应用程序；配置的等待周期越大，芯片的功耗越低。 *说明：rw 表示可读写。*

3.5.2 Flash 关键字寄存器（FLASH_KEYR）

地址偏移：0x04

复位值：0xXXXX XXXX

说明：X 表示不定值。

该寄存器仅支持写。若读该寄存器，返回值为 0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FKEYR[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FKEYR[15:0]															
w															

位 31:0	FKEYR：Flash 关键字 该位域用于存储解锁 Flash 的关键字。 • KEY1：0x4567 0123 • KEY2：0xCDEF 89AB

3.5.3 Flash 选项关键字寄存器（FLASH_OPTKEYR）

地址偏移：0x08

复位值：0xXXXX XXXX

说明：X 表示不定值。

该寄存器仅支持写。若读该寄存器，返回值为 0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPTKEYR[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTKEYR[15:0]															
w															

位 31:0	OPTKEYR：选项字节关键字 该位域用于存储可解锁 OPTWRE 的关键字。 • KEY1：0x4567 0123 • KEY2：0xCDEF 89AB

3.5.4 Flash 状态寄存器（FLASH_SR）

地址偏移：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res										EOP	WRPRTERR	Res			BSY
										rw	rw				r

位 31:6	Res：保留 必须保持复位值。
位 5	EOP：操作结束 当 Flash 操作（写/擦除）完成时，由硬件置位。该位由软件写 1 清零。 *注意：写或擦除操作成功后，硬件才会置位 EOP。*
位 4	WRPRTERR：写保护错误标志 当出现对写保护区域的写操作时，该位被硬件置位。该位由软件写 1 清零。
位 3:1	Res：保留 必须保持复位值。
位 0	BSY：忙标志 该位标明 Flash 操作正在进行。当开始 Flash 操作时，该位被硬件置为“1”。当操作结束时或发生错误时，该位由硬件清零。 *说明：r 表示仅读。*

3.5.5 Flash 控制寄存器（FLASH_CR）

地址偏移：0x10

复位值：0x0000 0080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res			EOPIE	Res	ERRIE	OPTWRE	Res	LOCK	STRT	OPTER	OPTPG	Res	MER	PER	PG
			rw		rw	rw		rw	rw	rw	rw		rw	rw	rw

位 31:13	Res：保留 必须保持复位值。
位 12	EOPIE：操作结束中断使能 该位可使 FLASH_SR 中的 EOP 位变为 1 时，产生中断请求。 • 0：中断禁用 • 1：中断使能
位 11	Res：保留 必须保持复位值。
位 10	ERRIE：操作错误中断使能 该位使 FLASH_SR 中的 WRPRTERR 位变为 1 时，产生中断请求。 • 0：中断禁用 • 1：中断使能
位 9	OPTWRE：选项字节写使能 该位为 1 时，允许改写选项字节。FLASH_OPTKEYR 寄存器写入正确的关键字序列，该位被置 1。 该位可由软件清零。
位 8	Res：保留 必须保持复位值。
位 7	LOCK：锁定 Flash 标志 当该位为 1时，表明 Flash 为锁定状态。通过解锁时序将该位清零。当解锁不成功时，该位就一直为1了，除非下次复位重新操作。 *注意：该位只能写 1。*
位 6	STRT：启动 该位会触发一个擦除操作，仅由软件置 1，仅在 BSY 为 0 时被清零。
位 5	OPTER：选项字节擦除 只能按字节擦除。
位 4	OPTPG：选项字节写入 只能按半字的方式写入。
位 3	Res：保留 必须保持复位值。

位 2	MER：整片擦除 整片擦除时选择。
位 1	PER：页擦除 页擦除时选择。
位 0	PG：半字写入 Flash 写入时可配置该位。

3.5.6 Flash 地址寄存器（FLASH_AR）

地址偏移：0x14

复位值：0x0000 0000

该寄存器通过硬件根据当前和上一次操作进行更新。对于页擦除操作，该寄存器该由软件来更新以便瞄准要擦除的页。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FAR [31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAR[15:0]															
w															

位 31:0	FAR：Flash 地址 当 PG 位使能时，该位域为写入 Flash 的地址；或当 PER 位使能时，选择待擦除的页。 *注意：当 FLASH_SR 中的 BSY 为 1 时，禁止写该寄存器。*

3.5.7 Flash 选项字节寄存器（FLASH_OBR）

地址偏移：0x1C

复位值：0xXXXX XX0X

选项字节的写入值决定了该寄存器的复位值；复位时，选项字节加载环节中比较选项字节及其反码（按位取反）的结果决定了 OPTERR 位的复位值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA1								DATA0							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res						nRST_STOP	WDG_SW	Res					RDPRT[1:0]		OPTERR
r						r	r						r		r

位 31:24	DATA1：用户数据
位 23:16	DATA0：用户数据
位 15:10	Res：用户选项字节的保留位

位 9	nRST_STOP：进入停机模式是否产生复位。 该位属于用户选项字节。 • 0：当进入停机模式产生复位。 • 1：不产生复位。
位 8	WDG_SW：选择软件或硬件看门狗 该位属于用户选项字节。 • 0：硬件看门狗 • 1：软件看门狗
位 7:3	Res：保留 必须保持复位值。
位 2:1	RDPRT[1:0]：读保护状态 • 00：当前处于 Level0 读保护状态。 • 01：当前处于 Level1 读保护状态。 • 11：当前处于 Level2 读保护状态。
位 0	OPTERR：选项字节错误 当该位置 1 时，表明加载选项字节互补关系不成立。

3.5.8 Flash 写保护寄存器（FLASH_WRPR）

地址偏移：0x20

复位值：0xXXXX XXXX

选项字节的写入值决定了该寄存器的复位值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRP[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRP[15:0]															
r															

位 31:0	WRP：写保护 该寄存器包含保持由复位后载入的写保护选项字节。

3.5.9 Flash 控制寄存器 2（FLASH_ECR）

地址偏移：0x70

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res													EEPROM_BPG	EEPROM_ER	BPG

	rw	rw	rw

位 31:3	Res：保留 必须保持复位值。
位 2	EEPROM_BPG：字节编程 EEPROM 空间
位 1	EEPROM_ER：字节擦除 EEPROM 空间
位 0	BPG：字节编程 Flash 空间

3.5.10 中断向量表偏移寄存器（INT_VEC_OFFSET）

偏移地址：0x74

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res		INT_VEC_OFFSET[13:0]													
		rw													

位 31:14	Res：保留 必须保持复位值。
位 13:0	INT_VEC_OFFSET[13:0]：中断向量表重映射偏移地址 *注意：INT_VEC_OFFSET[13:0]的最低两位必须配置为0，可设置的最小偏移量为4。*

若 INT_VEC_OFFSET 设置为 Y_Addr，则：

- 中断向量表重映射后，CPU 访问 0x0800 0000~0x0800 00FF 这一段地址时，实际访问到的地址是:（Y_Addr + 0x0800 0000）~（Y_Addr + 0x0800 00FF）;
- 当 INT_VEC_OFFSET 的值大于或等于 0x100 时，原本物理地址 0x0800 0000 到 0x0800 00FF 中的值将不能被访问到。

注意：若开发一个带 Bootloader 和应用程序（APP）的项目，需要在 APP 工程内调用 Boot 工程中的函数 fun_y，那么函数 fun_y 必须定义在地址 0x0800 00FF 以后。

下图说明了中断向量表地址映射及偏移的关系：

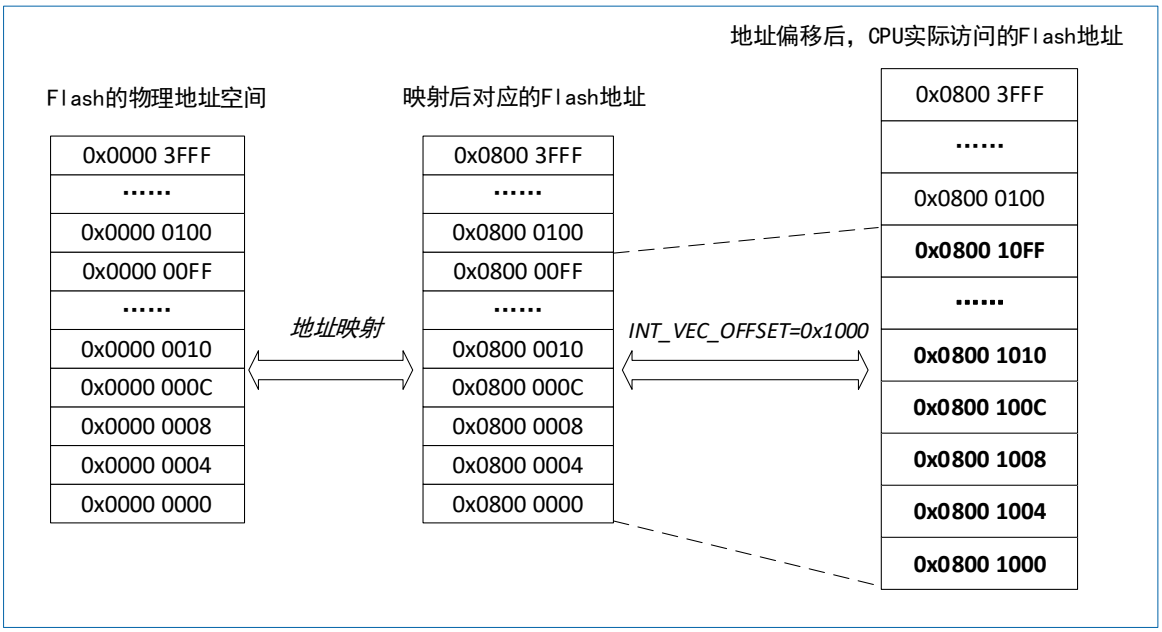


图 3-3 中断向量表地址映射及偏移

Flash 的物理地址空间：Flash 的物理地址空间从 0x0000 0000 开始向上增长，中断向量表也从低地址开始存放。

CPU 访问 Flash 的地址映射规则：规定了 CPU 访问 0x0800 0000 地址，实则为访问 Flash 物理地址 0x0000 0000。

INT_VEC_OFFSET = 0x1000：把 0x00000000 ~ 0x000000FF 偏移 0x1000，则中断向量表也被一起偏移映射了 0x1000。则 CPU 访问 0x0800 1000 地址，也就是访问了 FLASH 的 0x0000 1000 地址。这在 IAP 升级应用中很方便。

4 CRC 计算单元（CRC）

循环冗余校验（CRC）计算单元用于验证数据传输和数据存储的完整性。CRC 计算单元在运行期间计算出软件的签名，并将其和链接时所产生并存储于在指定存储地址的参考签名进行比较。

4.1 CRC 主要功能

- 采用的 CRC-32（与以太网标准相同）多项式 0x4C11DB7

  $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 能处理 8 位、16 位和 32 位数据宽度
- 可编程 CRC 初始值
- 单输入/输出 32 位数据寄存器
- 输入缓冲器可避免计算期间发生总线阻塞
- 8 位通用寄存器（可用于临时存储）
- I/O 数据的可反转性选项 CRC 功能说明
- 对于 32 位数据大小，CRC 计算在 4 个 AHB 时钟周期（HCLK）内完成。

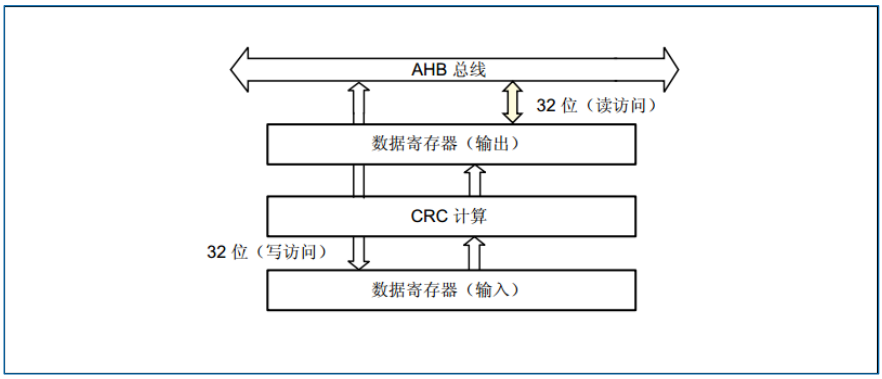


图 4-1 CRC 计算单元框图

4.2 CRC 功能描述

CRC 计算单元具有单个 32 位读/写数据寄存器（CRC_DR）。CRC_DR 用于保存输入的新数据（写访问）和之前 CRC 计算的结果（读访问）。

对 CRC_DR 寄存器的每次写操作都会对之前的 CRC 值（存于 CRC_DR 中）和新值做一次 CRC 计算。CRC 计算支持整个 32 位数据字或逐个字节计算，具体取决于写入数据的位宽。

CRC_DR 寄存器可按字、右对齐半字和右对齐字节进行访问。对于其它 CRC 寄存器，只支持 32 位访问。

计算时间取决于数据宽度：

- 32 位数据需要 4 个 AHB 时钟周期
- 16 位数据需要 2 个 AHB 时钟周期
- 8 位数据需要 1 个 AHB 时钟周期

输入缓冲器中可立即写入第二个数据，无需因之前的 CRC 计算而等待。

CRC 计算单元可动态调整数据大小，从而能最大程度地减少给定字节数的写访问次数。例如，对 5 个字节进行 CRC 计算时，可先写入一个字，然后写入一个字节。

输入数据的顺序可反转，以管理各种数据存放方式（双字/单字/字节、大端/小端等）。可对 8 位、

16 位和 32 位数据执行反转操作，具体取决于 CRC_CR 寄存器中的 REV_IN[1:0]位。

例如，输入数据 0x1A2B3C4D 在 CRC 计算中用作：

- 按字节执行位反转后的 0x58D43CB2
- 按半字执行位反转后的 0xD458B23C
- 按全字执行位反转后的 0xB23CD458

通过将 CRC_CR 寄存器中 REV_OUT 位置 1，也可以将输出数据反转。该操作按位进行，例如：输出数据 0x1122 3344 将转换为 0x22CC 4488。

配置 CRC_CR 寄存器中的 RESET 控制位可将 CRC 计算器初始化为可编程值（默认值为 0xFFFF FFFF）。

可使用 CRC_INIT 寄存器对 CRC 初始值进行编程。对 CRC_INIT 寄存器进行写访问时，会自动初始化 CRC_DR 寄存器。

CRC_IDR 寄存器可用于保存与 CRC 计算相关的临时值。CRC_IDR 不受 CRC_CR 寄存器中的 RESET 位影响。

4.3 CRC 寄存器

基地址：0x4002 3000

空间大小：0x400

4.3.1 数据寄存器（CRC_DR）

地址偏移：0x00

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DR[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
rw															

位 31:0	DR[31:0]：数据寄存器 该寄存器用于存放待计算的新数据，直接将其写入即可。读取该寄存器得到的是上次 CRC 计算的结果。如果读出或写入的数据不足 32 位，则表示该数据仅针对有意义的位。

4.3.2 独立数据寄存器（CRC_IDR）

地址偏移：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								IDR[7:0]							
								rw							

位 31:8	Res：保留 必须保持复位值。

位 7:0	IDR[7:0]：通用目的 8 位数据寄存器 该寄存器可用作 1 个字节的临时存储。CRC_CR 寄存器中的 RESET 位引起的复位操作不会影响该寄存器。

4.3.3 控制寄存器（CRC_CR）

地址偏移：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								REV_OUT	REV_IN[1:0]		Res				RESET
								rw	rw						rs

位 31:8	Res：保留 必须保持复位值。
位 7	REV_OUT：翻转输出数据 该位控制输出数据的翻转。 • 0：不翻转 • 1：翻转
位 6:5	REV_IN[1:0]：翻转输入数据 该位域控制输入数据的翻转。 • 00：不翻转 • 01：按字节为单位翻转 • 10：按半字为单位翻转 • 11：按字为单位翻转
位 4:1	Res：保留 必须保持复位值。
位 0	RESET：复位控制 该位用于复位整个CRC计算单元，并将CRC_INIT寄存器中的值更新到 CRC_DR 寄存器。 该位由软件置位，由硬件清零。 *说明：rs 所表示的含义，参见“24.1 寄存器描述中的缩略语”。*

4.3.4 CRC 初值寄存器（CRC_INIT）

地址偏移：0x10

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CRC_INIT[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC_INIT[15:0]															
rw															

位 31:0	CRC_INIT[31:0]：CRC 预置的初值 该寄存器用于设置CRC 的初值。

5 电源控制（PWR）

5.1 电源

该系列芯片的工作电压（V_{DD}）为 1.8~3.6 V。通过内置的电压调节器提供所需的 1.2 V 内核电源。

该系列芯片片内数字逻辑的电源由片内 LDO 提供。

片内 LDO 的输出电压可通过寄存器设置，以便于软件根据应用场景最大程度地优化芯片的电流功耗。芯片运行（Run）模式和停机（Stop）模式的 LDO 输出电压可以分别独立设置。

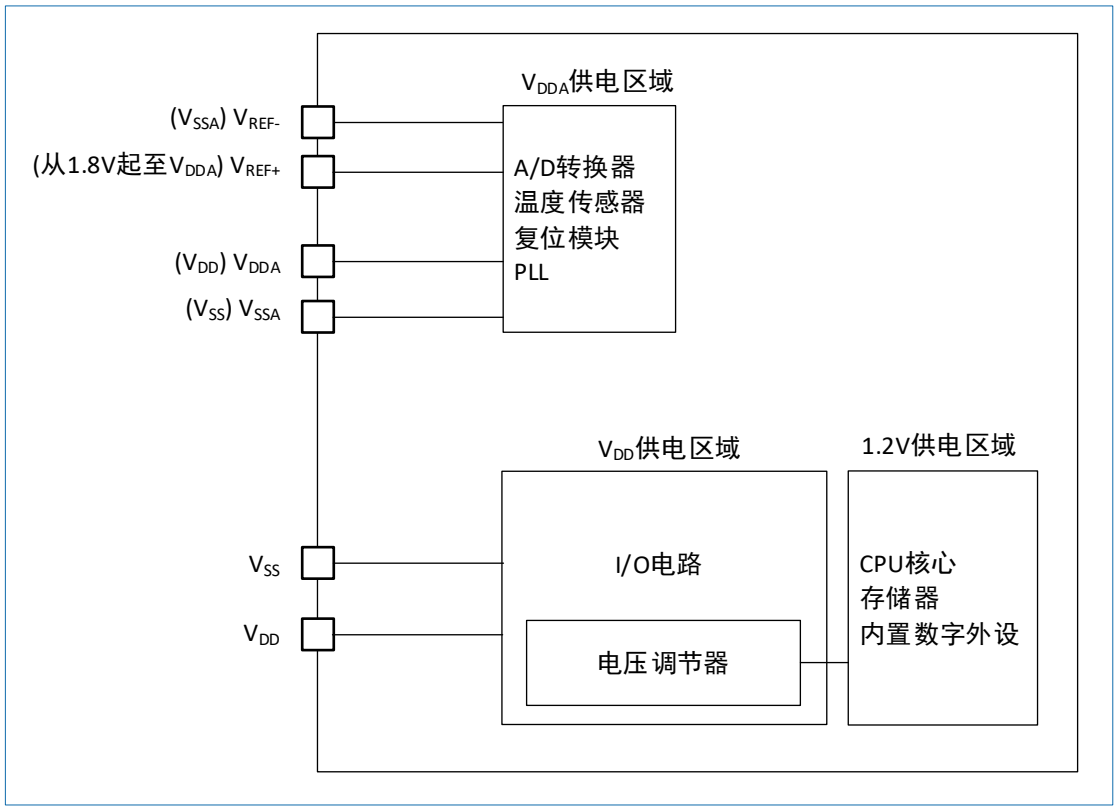


图 5-1 电源框图

注意：V_{DDA} 和 V_{SSA} 必须分别连到 V_{DD} 和 V_{SS}。

5.1.1 独立的 A/D 转换器供电和参考电压

为了提高转换的精确度，ADC 使用一个独立的电源供电，以过滤和屏蔽来自印刷电路板上的毛刺干扰。

- ADC 的电源引脚为 V_{DDA}
- 独立的电源地 V_{SSA}

如果有 V_{REF-}引脚（依封装而定），它必须连接到 V_{SSA}。

没有 V_{REF+}和 V_{REF-}引脚，它们在芯片内部与 ADC 的电源（V_{DDA}）和地（V_{SSA}）相连。

5.1.2 电压调节器

复位后，电压调节器总是使能的。根据应用方式，调节器以 2 种不同的模式工作。

- 运行（Run）模式：调节器以正常功耗模式提供 1.2 V 电源（内核、内存和外设）。
- 停机（Stop）模式：调节器以低功耗模式提供 1.2 V 电源，以保存寄存器和 SRAM 的内容。

5.2 上电/掉电复位（POR/PDR）

该系列芯片内部有一个完整的上电复位（POR）和掉电复位（PDR）电路。当供电电压达到 POR/PDR 阈值时，系统即能正常工作。

当 V_{DD}/V_{DDA} 低于指定的限位电压 V_{POR}/V_{PDR} 时，系统保持为复位状态，而无需外部复位电路。关于上电复位和掉电复位的更多细节，请参考数据手册的电气特性部分。

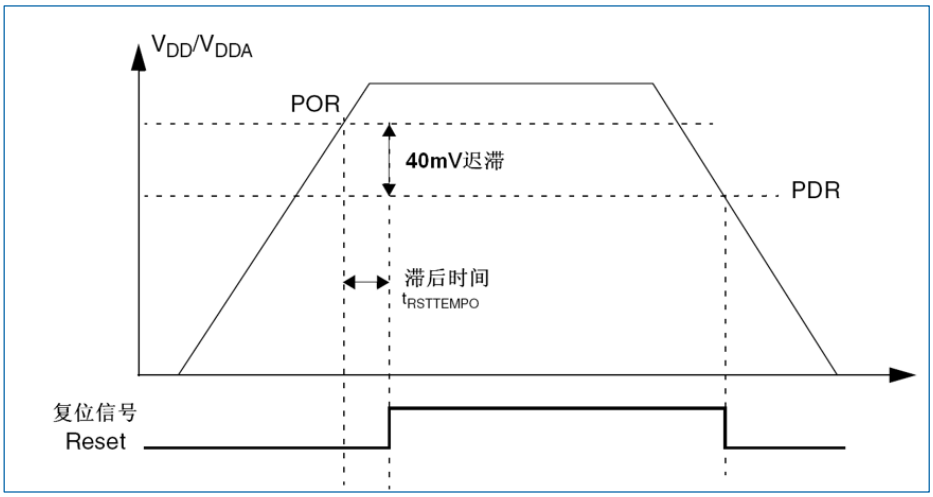


图 5-2 上电复位和掉电复位的波形图

5.3 低功耗模式

在系统或电源复位以后，MCU 处于运行状态。当 CPU 不需继续运行时（例如等待某个外部事件时），可以利用多种低功耗模式来节省功耗。用户需要根据最低电源消耗、最短启动时间和可用的唤醒源等条件，选定一个最佳的低功耗模式。

该系列芯片支持以下低功耗模式：

- 睡眠（Sleep）模式

  Cortex®-M0 内核停止，所有外设包括 Cortex-M0 核心的外设，如 NVIC、系统时钟（SysTick）等仍在运行。

- 深度睡眠（Deep-Sleep）模式

  在深度睡眠模式下，系统时钟降低至 114 kHz 以节省功耗。此模式下，仅有 CPU 停止工作，所有外设处于工作状态并可通过中断/事件唤醒 CPU。深度睡眠模式的功耗高于停机模式。

- 停机（Stop）模式

  在停机模式下，内核的所有时钟被关闭，HSI 振荡器被关闭。可以通过任一配置成 EXTI 的信号把 MCU 从停机模式中唤醒。

在运行模式下，可以通过以下任意方式降低功耗：

- 降低系统时钟频率。
- 关闭 APB 和 AHB 总线上未被使用的外设时钟。

表 5-1 低功耗模式的进入/唤醒条件

工作模式	进入条件	唤醒条件	内部核电源时钟状态	V_{DD} 主区域时钟状态	电压调节器状态
睡眠模式（Sleep）	1. 设置 PWR_CR:LPDS = 0； 2. 软件执行 WFI/WFE	由任何一个普通 irq 中断事件唤醒，包括 System ticker。	CPU 时钟关闭，对其他时钟和 ADC	开启	开启

工作模式	进入条件	唤醒条件	内部核电源时钟状态	V_{DD} 主区域时钟状态	电压调节器状态
	指令进入。		时钟无影响		
深度睡眠模式（DeepSleep）	1. 将时钟切换为 LSI； 2. 设置 PWR_CR:LPDS = 0； 3. 软件执行 WFI/WFE 指令进入。	由任何一个普通 IRQ 中断事件唤醒，包括 System ticker。	CPU 时钟关闭，对其他时钟和 ADC 时钟无影响	开启	开启
停机模式（Stop）	1. 设置 PWR_CR:LPDS = 0； 2. 设置 CM0 系统控制寄存器的 SLEEPDEEP 位； 3. 软件执行 WFI/WFE 指令进入。	• 支持 Beeper 驱动 ADC 采样预唤醒，当满足条件后真正唤醒。 • 支持自动唤醒定时器（AWU）唤醒。	所有时钟停止	HSI 关闭	开启或者处于低功耗状态（在 PWR_CR 中设置）

5.3.1 降低系统时钟

在运行模式下，通过对预分频寄存器进行编程，可以降低任意一个系统时钟（SYSCLK、HCLK、PCLK）的速度。进入睡眠模式前，也可以利用预分频器来降低外设的时钟频率。参见章节：“6.3.2 时钟配置寄存器（RCC_CFGR）”。

5.3.2 外部时钟的控制

在运行模式下，随时可以通过停止为各外设和内存提供时钟（HCLK 和 PCLK）来减少功耗。在睡眠模式下为了进一步减少功耗，可在执行 WFI 或 WFE 指令前关闭所有外设的时钟。

通过设置 AHB 外设时钟使能寄存器（参见“6.3.6 AHB 外部时钟使能寄存器（RCC_AHBENR）”）APB 外设时钟使能寄存器（参见“6.3.7 APB 外设时钟使能寄存器 2（RCC_APB2ENR）和 6.3.8 APB1 外设时钟使能寄存器（RCC_APB1ENR）”）来开关各个外设模块的时钟。

5.3.3 睡眠（Sleep）模式

5.3.3.1 进入睡眠模式

通过执行 WFI 或 WFE 指令进入睡眠状态。根据 Cortex®-M0 系统控制寄存器中的 SLEEPONEXIT 位的值，有两种选项可用于选择睡眠模式进入机制：

- SLEEP-NOW：如果 SLEEPONEXIT 位被清除，当 WFI 或 WFE 被执行时，MCU 立即进入睡眠模式。
- SLEEP-ON-EXIT：如果 SLEEPONEXIT 位被置位，系统从最低优先级的中断处理程序中退出时，MCU 就立即进入睡眠模式。

在睡眠模式下，所有的 I/O 引脚都保持它们在运行模式时的状态。

关于如何进入睡眠模式，更多的细节参考表 5-2 和表 5-3。

5.3.3.2 退出睡眠模式

如果执行 WFI 指令进入睡眠模式，任意一个被嵌套向量中断控制器（NVIC）响应的外设中断都能将系统从睡眠模式唤醒。

如果执行 WFE 指令进入睡眠模式，则一旦发生唤醒事件时，MCU 将从睡眠模式退出。唤醒事件可以通过下述方式产生：

- 在外设控制寄存器中使能一个中断，而不是在 NVIC 中使能，并且在 Cortex-M0 系统控制寄存

器中使能 SEVONPEND 位。当 MCU 从 WFE 中唤醒后，外设的中断挂起位和外设的 NVIC 中断通道挂起位（在 NVIC 中断清除挂起寄存器中）必须被清除。

- 配置一个外部或内部的 EXTI 线为事件模式。当 MCU 从 WFE 中唤醒后，因为与事件线对应的挂起位未被设置，不必清除外设的中断挂起位或 NVIC 中断请求（IRQ）通道挂起位。

该模式唤醒所需的时间最短，因为中断的进入或退出没有消耗时间。关于如何退出睡眠模式，更多的细节参考表 5-2 和表 5-3。

表 5-2 SLEEP-NOW 模式

SLEEP-NOW 模式	说明
进入条件	在以下条件下，执行 WFI（等待中断）或 WFE（等待事件）指令： ● SLEEPDEEP = 0 ● SLEEPONEXIT = 0 参考 Cortex-M0 系统控制寄存器。
退出条件	● 如果执行 WFI 进入睡眠模式： 中断：参考“10.1.3 中断和异常向量”。 ● 如果执行 WFE 进入睡眠模式： 唤醒事件：参考“10.2.4 唤醒事件管理”。
唤醒延时	无

表 5-3 SLEEP-ON-EXIT 模式

SLEEP-ON-EXIT 模式	说明
进入	在以下条件下，执行 WFI 指令： ● SLEEPDEEP = 0 ● SLEEPONEXIT = 1 参考 Cortex-M0 系统控制寄存器。
退出	中断：参考“10.1.3 中断和异常向量”。
唤醒延时	无

5.3.4 深度睡眠（DeepSleep）模式

在深度睡眠模式下，系统时钟降低至 114 kHz 以节省功耗。此模式下，仅 CPU 停止工作，所有外设处于工作状态并可发生中断/事件唤醒 CPU。深度睡眠模式的功耗高于停机模式。

5.3.5 停机（Stop）模式

停机模式是在 Cortex®-M0 的深睡眠模式基础上结合了外设的时钟控制机制。在停机模式下，电压调节器可运行在正常或低功耗模式。此时，在内核供电区域的所有时钟都停止，HSI 的功能禁用，SRAM 和寄存器内容被保存下来。

在停机模式下，部分的 I/O 引脚都保持它们在运行模式时的状态。

5.3.5.1 进入停机模式

关于如何进入停机模式，参见表 5-3。

在停机模式下，通过设置电源控制寄存器（PWR_CR）的 LPDS 位使内部调节器进入低功耗模式，能够降低更多的功耗。

如果正在进行 Flash 编程，直到对 Flash 访问完成，系统才进入停机模式。如果正在进行对 APB 的访

问，直到对 APB 访问完成，系统才进入停机模式。停机模式下，可以通过设置独立的控制位，选择以下功能：

- 独立看门狗（IWDG）：可通过写入看门狗的关键字寄存器或硬件选择来启动 IWDG。独立看门狗一旦启动，除非系统复位，否则它不能被停止。参见 IWDG 功能描述节。
- 自动唤醒定时器（AWU）：参见“15 自动唤醒定时器（AWU）”。
- 内部 RC 振荡器（LSI RC）：通过控制/状态寄存器（RCC_CSR）的 LSION 位来设置。

在停机模式下，如果在进入该模式前 ADC 没有被关闭，那么这些外设仍然耗电。通过设置寄存器 ADC_CR 的 ADON 位为 0，可关闭该外设。

5.3.5.2 退出停机模式

关于如何退出停机模式，可参见表 5-1。当一个中断或唤醒事件导致退出停机模式时，HSI RC 振荡器被选为系统时钟。

当电压调节器处于低功耗模式下，当系统从停机模式退出时，将会有一段额外的启动延时。如果在停机模式期间保持内部调节器开启，则退出启动时间会缩短，但相应的功耗会增加。

5.3.6 调试模式

默认情况下，如果在调试 MCU 时，使 MCU 进入停机，将断开调试连接。这是因为 Cortex®-M0 的内核的时钟被禁用。

然而，通过设置 DBGMCU_CR 寄存器中的某些配置位，可以在低功耗模式下调试软件。更多的细节请参考章节：低功耗模式的调试支持。

5.4 PWR 寄存器

基地址：0x4000 7000

空间大小：0x400

5.4.1 电源控制寄存器（PWR_CR）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res															LPDS
															rw

位 31:1	Res：保留 必须保持复位值。
位 0	LPDS：电压调节器状态 该位用软件设置或清除。 • 0：在停机模式下，电压调节器开启（处于正常耗模式）。 • 1：在停机模式下，电压调节器处于低功耗模式。

5.4.2 内部参考电压输出选择寄存器（PWR_VREF_SEL）

偏移地址：0x68

复位值：0x8000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VTEST_SEL[1:0]		Res													
rw															

位 15:14	VTEST_SET[1:0]：控制内部参考电压输出 • 10：0.8 V 参考电压输出到ADC • 11：LDO电压输出到ADC（默认1.2V） • 其他：关闭参考电压输出
位 13:0	Res：保留 必须保持复位值。

6 复位和时钟控制（RCC）

6.1 复位

该系列芯片有两种复位方式：系统复位、电源复位。

6.1.1 系统复位

系统复位将复位除时钟控制寄存器 CSR 中的复位标志以外的所有寄存器为它们的复位值。

当以下任一事件发生时，将产生系统复位：

- NRST 引脚上的低电平（外部复位）
- 窗口看门狗事件（WWDG 复位）
- 独立看门狗事件（IWDG 复位）
- 软件复位（SW 复位）
- 低功耗管理复位

可通过查看 RCC_CSR 控制状态寄存器中的复位状态标志位识别复位事件来源。

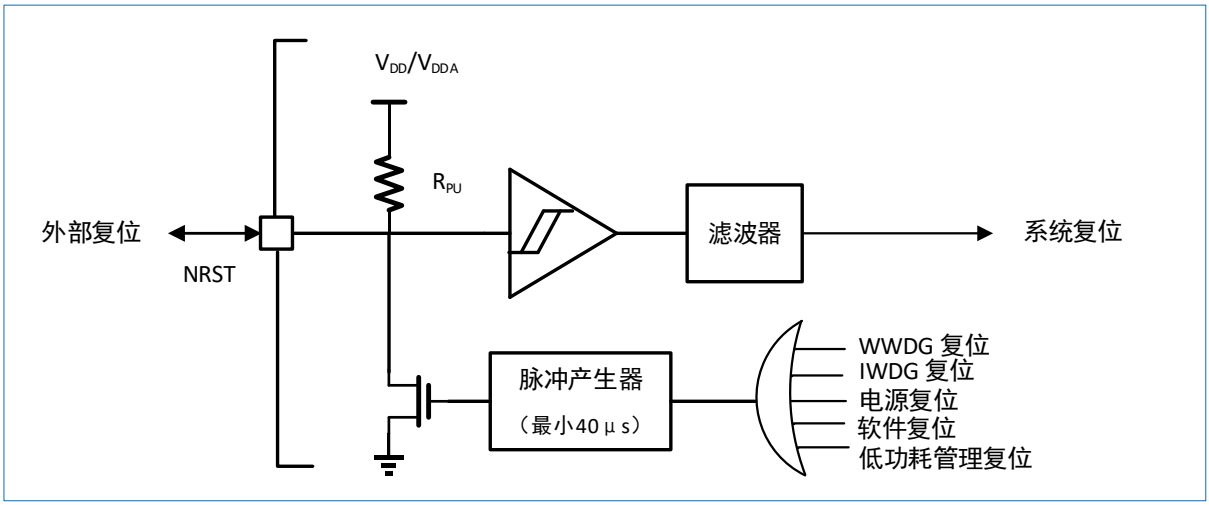


图 6-1 复位电路

复位源（包括电源复位）将最终作用于 NRST 引脚，并在复位过程中保持低电平。复位入口矢量被固定在地址 0x0000 0004。

芯片内部的复位信号会在 NRST 引脚上输出。脉冲发生器保证每一个内部复位源都能有至少 40 μs 的脉冲延时。当 NRST 引脚被拉低产生外部复位时，它将产生复位脉冲。

软件复位

通过将 Cortex-M0 应用中断和复位控制寄存器中的 SYSRESETREQ 位置 1，可实现软件复位。请参考 Cortex-M0 技术参考手册获得进一步信息。

低功耗管理复位

在进入停机模式时，产生低功耗管理复位：

通过将用户选项字节中的 nRST_STOP 位置零将使能该复位。这时，即使正处于进入停机模式的进程，系统将被复位而不是进入停机模式。

6.1.2 电源复位

当以下事件发生时，将产生电源复位：

- 上电/掉电复位（POR/PDR）

HK32F030M MCU 内部集成了上电复位（POR）/掉电复位（PDR）电路。该电路始终处于工作状态，以保证系统在供电超过 POR/PDR 阈值时正常工作。当 V_{DD} 小于 POR/PDR 阈值时，MCU 将被复位，无需使用外部复位电路。

说明：若需了解 POR/PDR 阈值，请参见 HK32F030M 数据手册的电气参数章节。

6.2 时钟

支持多种时钟源驱动系统时钟：

- 内部高速（HSI）32MHz RC 振荡器时钟
- 内部低速（LSI）振荡器时钟
- GPIO 外部输入时钟

每种时钟源都可以独立地打开或关断。当它们不用时，可以将其关断来降低功耗。有多个分频器可用于配置 AHB 和 APB 时钟域，AHB 和 APB 域的最大时钟频率为 32 MHz。Cortex 系统定时器由 AHB 时钟（HCLK）驱动，其可由 AHB/8 时钟频率直接驱动（通过 Cortex Systick 配置位来配置）。

所有的外设时钟由其所在的总线时钟（HCLK 或 PCLK）驱动，以下除外：

- 闪存编程接口时钟（FLITFCLK）由 HSI 32MHz 时钟或者 SYSCLK 驱动（由软件选择）。
- I2S 始终由 SYSCLK 直接驱动。
- I2C 的时钟为下列的时钟源之一（由软件选择）：
  - SYSCLK
  - HSI 32MHz 由 I2C-HSI 分频器分频
  - APB 时钟（PCLK）
- Cortex 的 FCLK 由 AHB 时钟（HCLK）直接提供。
- AHB bus，ARM core，memory 由 AHB 时钟（HCLK）直接提供。
- ADC 时钟由下列之一时钟得到（由软件选择 ）：
  - APB 时钟（PCLK）由 ADC 分频器 2/4 分频
  - HSI 32MHz 由 ADC-HSI 分频器分频
- USART1 的时钟为下列的时钟源之一（由软件选择）：
  - PCLK
  - SYSCLK
  - HSI 32MHz RC 振荡器时钟通过 USART-HSI 分频器来分频时钟
- Timer 时钟由 APB 时钟（PCLK）或者 APB 时钟（PCLK）2 倍频提供（由软件设置，硬件判定执行）。
- RCC 将 AHB 时钟（HCLK）8 分频后作为 Cortex 系统定时器（SysTick）的外部时钟。通过对 SysTick 控制与状态寄存器的设置，可选择 HCLK/8 时钟作为 SysTick 时钟。

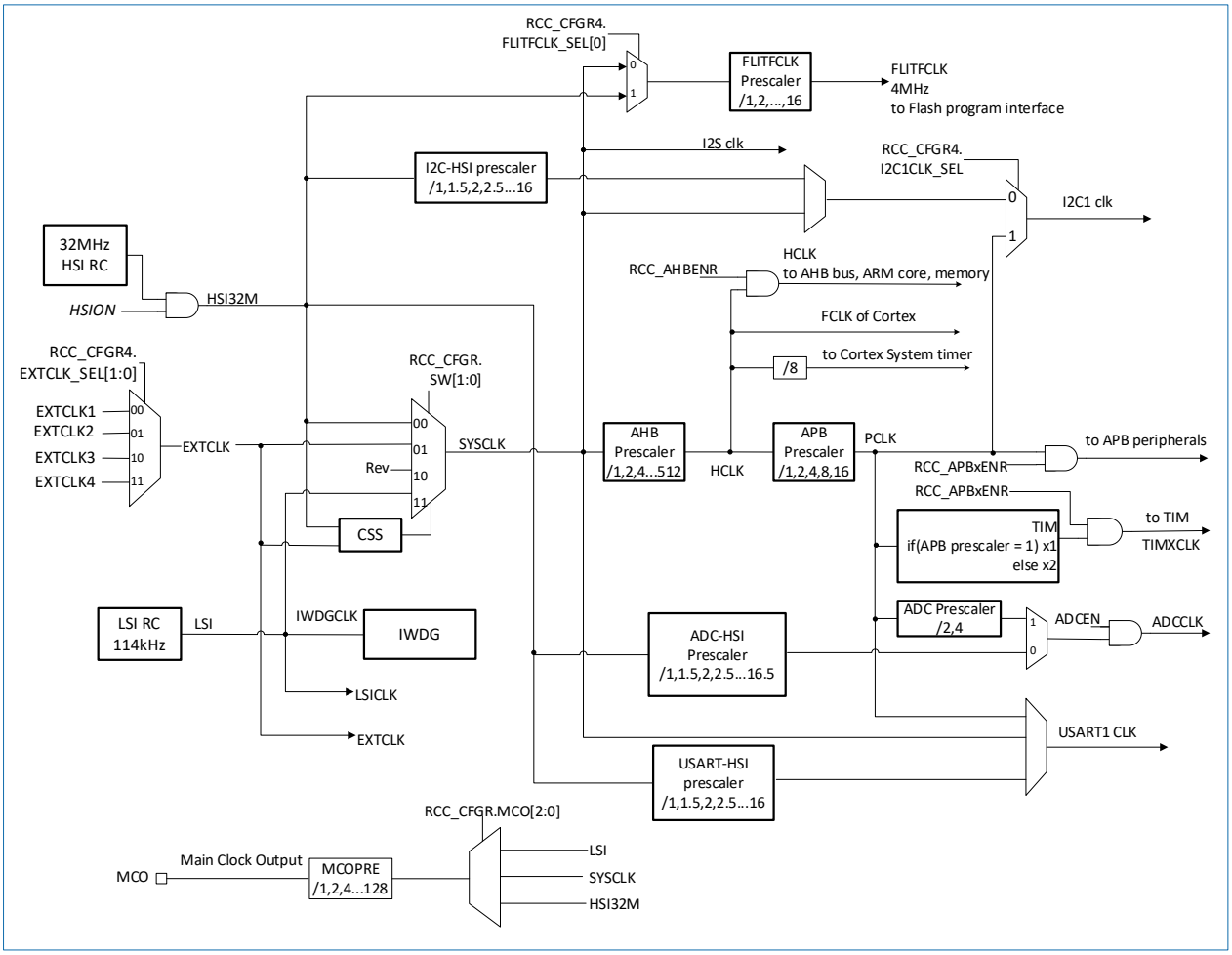


图 6-2 MCU 时钟树

6.2.1 HSI 时钟

HSI 时钟信号由 32 MHz 内部 RC 振荡器生成的，可直接作为系统时钟在从复位重启唤醒后，HSI 时钟始终用作系统时钟。

HSI RC 振荡器能够在不需要任何外部器件的条件下提供系统时钟。

6.2.2 GPIO 外部时钟输入

在这个模式里，必须提供外部时钟。用户可以通过外部 GPIO 输入最高 32MHz 时钟作为 SYSCLK。通过 RCC_CFGR4.EXTCLK_SEL[1:0]选择作为外部时钟输入的 IO。用户可通过设置在时钟控制寄存器（RCC_CR）中的 EXTCLKON 位来选择这一模式。占空比为 40%～60%的外部时钟信号（方波、正弦波或三角波）必须连到对应的外部 GPIO 引脚上。

用户还可配置时钟控制寄存器（RCC_CR）中的 CSSON 位来打开/关闭时钟检测功能。

6.2.3 LSI 时钟

LSI RC 可作为低功耗时钟源在停机模式下保持运行，供独立看门狗（IWDG）使用。时钟频率在 114 kHz 左右。由于 LSI 的特性，会有±10%的误差。

LSI RC 振荡器可通过控制/状态寄存器（RCC_CSR）中的 LSION 位打开或关闭。

控制/状态寄存器（RCC_CSR）中的 LSIRDY 标志指示低速内部振荡器是否稳定。在启动时，需等待硬件将该位置 1 后，才能使用此时钟。如果在 RCC_CIR 中使能中断，则可产生中断。

从 IWDG 激活后，LSI 振荡器不能通过 LSION=0 停止。LSI 振荡器通过系统复位停止（通过硬件使用

FLASH_OBR 寄存器中的 WDG_SW 位使能 IWDG 的情况除外）。如果 IWDG 已通过软件使能，则必须在系统复位后再次使能 LSI 振荡器，以确保 IWDG 正常工作。

可通过测量 LSI 振荡器的频散来获得精度水平可接受的 IWDG 超时（当 LSI 用作这些外设的时钟源时）。

6.2.4 系统时钟（SYSCLK）选择

可以使用 3 种不同的时钟源来驱动系统时钟（SYSCLK）：

- HSI32M 振荡器
- LSI 振荡器
- EXTCLK 外部时钟

系统复位后，HSI 振荡器被选为系统时钟。当时钟源被直接作为系统时钟时，它将不能被停止。

时钟的切换只有在目标时钟源可用的情况下才能进行。假如系统选择了未准备好的时钟源作为当前系统时钟，那么只有在目标时钟源准备好之后才真正执行切换时钟源的操作。时钟配置寄存器（RCC_CFGR）指示当前系统时钟采用哪个时钟源作为系统时钟。

6.2.5 看门狗时钟

如果独立看门狗（Independent watchdog，IWDG）已通过硬件选项或软件访问的方式启动，则 LSI 振荡器将被强制打开，且不能禁止。在 LSI 振荡器稳定后，时钟将提供给 IWDG。

如果已通过软件使能 IWDG，则系统复位后会禁止 LSI。之后，必须再次使能 LSI 振荡器，以确保 IWDG 正常工作。

6.2.6 时钟输出功能（MCO）

MCU 时钟输出（Microcontroller clock output，MCO）功能允许使用可配置的预分频值（1、2、4、8、16、32、64、128）将时钟输出到外部 MCO 引脚上。必须在复用功能模式下对相应 GPIO 端口的配置寄存器进行编程。可选择以下 3 种时钟信号之一作为 MCO 时钟：

- LSI
- SYSCLK
- HSI

时钟信号选择由时钟配置寄存器（RCC_CFGR）中的 MCO[3:0]位控制。

6.3 RCC 寄存器

基地址：0x4002 1000

空间大小：0x400

6.3.1 时钟控制寄存器（RCC_CR）

偏移地址：0x00

复位值：0x0000 XX83

访问：无等待状态，字、半字和字节访问。

说明：X 表示不定值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res												CSSON	Res	EXTCLKRDY	EXTCLKON
												rw		r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res		HSICAL[5:0]						HSITRIM[4:0]					Res	HSIRDY	HSION
		rw						rw						r	rw

位 31:20	Res：保留 必须保持复位值。
位 19	CSSON：打开/关闭时钟检测 当EXTCLKON 被置位后，等待 EXTCLKRDY 变为 1，CSS硬件就开始检测GPIO 输入时钟频率。如果检测到GPIO输入时钟频率低于CSS_THRESHOLD设定阈值，则置位CSSF 并产生NMI，然后停止检测GPIO输入时钟。该位由软件置位和清零。 • 0：关闭时钟检测。 • 1：开启时钟检测（如果GPIO输入时钟没有稳定，则自动关闭时钟检测）。
位 18	Res：保留 必须保持复位值。
位 17	EXTCLKRDY：指示 GPIO 输入时钟是否稳定 • 0：GPIO输入时钟未稳定 • 1：GPIO输入时钟稳定
位 16	EXTCLKON：GPIO输入时钟使能 • 0：关闭GPIO输入时钟 • 1：使能GPIO输入时钟
位 15:14	Res：保留 必须保持复位值。
位 13:8	HSICAL[5:0]：HSI 粗调 该位的复位值由出厂设置而定。粗调步进为2%，值越大，设置的频率越高。
位 7:3	HSITRIM[4:0]：HSI精调 精调步进为0.2%，值越大，设置的频率越高。
位 2	Res：保留 必须保持复位值。
位 1	HSIRDY：HSI 就绪标志 • 0：HSI未稳定 • 1：HSI 已稳定
位 0	HSION：HSI 使能 该位可由软件置位或清除。 当从STOP模式唤醒时，硬件自动将该位置 1；当GPIO被用作 SYSCLK 且CSS检测到GPIO输入时钟异常，硬件也自动将该位置 1。

	选择 HSI 作为SYSCLK时，软件不能清除该位。 • 0：关闭HSI • 1：打开HSI

6.3.2 时钟配置寄存器（RCC_CFGR）

偏移地址：0x04

复位值：0x0000 0010

访问：无等待周期，支持字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	MCOPRE[2:0]			MCO[3:0]				Res							
	rw			rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res					PPRE[2:0]			HPRE[3:0]				SWS[1:0]		SW[1:0]	
					rw			rw						rw	

位 31	Res：保留 必须保持复位值。
位 30:28	MCOPRE[2:0]：MCU 时钟输出（MCO）分频系数 • 000：MCO/1 • 001：MCO/2 • 010：MCO/4 … • 111：MCO/128
位 27:24	MCO[3:0]：MCU 时钟输出 • 0000：MCO 无时钟输出 • 0010：MCO 输出 LSI • 0100：MCO 输出 SYSCLK • 0101：MCO 输出 HSI • 其他：保留
位 23:11	Res：保留 必须保持复位值。
位 10:8	PPRE[2:0]：PCLK 的分频系数 • 0xx：HCLK/1 • 100：HCLK/2 • 101：HCLK/4 • 110：HCLK/8 • 111：HCLK/16
位 7:4	HPRE[3:0]：HCLK 的分频系数

	• 0001：SYSCLK/6 • 1000：SYSCLK/2 • 1001：SYSCLK/4 • 1010：SYSCLK/8 • 1011：SYSCLK/16 • 1100：SYSCLK/64 • 1101：SYSCLK/128 • 1110：SYSCLK/256 • 1111：SYSCLK/512 • 其他：SYSCLK/1
位 3:2	SWS[1:0]：系统时钟切换状态 该位域由硬件自动改写值。 • 00：HSI 用作 SYSCLK。 • 01：GPIO 输入时钟用作 SYSCLK。 • 10：保留 • 11：LSI 用作 SYSCLK。
位 1:0	SW[1:0]：系统时钟切换 • 00：选择 HSI 作为 SYSCLK。 • 01：选择 GPIO 输入时钟作为 SYSCLK。 • 10：保留 • 11：选择 LSI 作为 SYSCLK。

6.3.3 时钟中断寄存器（RCC_CIR）

偏移地址：0x08

复位值：0x0000 0000

访问：无等待状态，字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res								CSSC	Res			EXTRDYC	HSIRDYC	Res	LSIRDYC
								w				w	w		w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				EXTRDYIE	HSIRDYIE	Res	LSIRDYIE	CSSF	Res			EXTRDYF	HSIRDYF	Res	LSIRDYF
				rw	rw		rw	r				r	r		r

位 31:24	Res：保留 必须保持复位值。
位 23	CSSC：清除安全时钟标志 • 0：无作用 • 1：清除 CSSF 标志

位 22:20	Res：保留 必须保持复位值。
位 19	EXTRDYC：清除 EXTRDYF 的标志 该位由软件设置。 • 0：无作用 • 1：清除 EXTRDYF 标志
位 18	HSIRDYC：清除 HSIRDYF 的标志 该位由软件设置。 • 0：无作用 • 1：清除 HSIRDYF 标志
位 17	Res：保留 必须保持复位值。
位 16	LSIRDYC：清除 LSIRDYF 的标志 该位由软件设置。 • 0：无作用 • 1：清除 LSIRDYF 标志
位 15:12	Res：保留 必须保持复位值。
位 11	EXTRDYIE：关闭/使能 GPIO 时钟稳定中断 • 0：关闭 GPIO 时钟稳定中断 • 1：使能 GPIO 时钟稳定中断
位 10	HSIRDYIE：关闭/使能 HSI 时钟稳定中断 • 0：关闭 HSI 时钟稳定中断 • 1：使能 HSI 时钟稳定中断
位 9	Res：保留 必须保持复位值。
位 8	LSIRDYIE：关闭/使能 LSI 时钟稳定中断 • 0：关闭 LSI 时钟稳定中断 • 1：使能 LSI 时钟稳定中断
位 7	CSSF：当检测到 GPIO 输入时钟频率异常，硬件置位 CSSF；软件对 CSSC 写 1 时，清除 CSSF。 • 0：没有触发 GPIO 输入频率异常 • 1：检测到 GPIO 输入频率异常

位 6:4	Res：保留 必须保持复位值。
位 3	EXTRDYF：当 GPIO 输入时钟变为稳定态时，由硬件置位；软件写 EXTRDYC，则将该位清零。 • 0：没有触发 GPIO 输入时钟稳定中断 • 1：触发 GPIO 输入时钟稳定中断
位 2	HSIRDYF：当打开 HSI 时钟后，HSI 时钟变为稳定态时由硬件置位。只有通过软件配置 HSION 使 HSI 变为稳定态，硬件才会置 HSIRDYF 为 1。软件对 HSIRDYC 写 1 时，清零 HSIRDYF。 • 0：没有触发 HSI 时钟稳定中断 • 1：触发 HSI 时钟稳定中断
位 1	Res：保留 必须保持复位值。
位 0	LSIRDYF：当 LSI 变为稳定态时，由硬件置位；软件对 LSIRDYC 写 1 时，清零 LSIRDYF。 • 0：没有触发 LSI 时钟稳定中断 • 1：触发 LSI 时钟稳定中断

6.3.4 APB2 外设复位寄存器（RCC_APB2RSTR）

偏移地址：0x0C

复位值：0x0000 0000

访问：无等待周期，支持字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res									DBGMCURST	Res					
									rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	USART1RST	Res	SPI1RST	TIM1RST	Res	ADCRST	Res								SYSCFGRST
	rw		rw	rw		rw									rw

位 31:23	Res：保留 必须保持复位值。
位 22	DBGMCURST：调试 MCU 复位 该位由软件置 1 或清 0。 • 0：无作用 • 1：复位调试 MCU
位 21:15	Res：保留 必须保持复位值。

位 14	USART1RST：USART1 复位 该位由软件置 1 或清 0。 • 0：无作用 • 1：复位 USART1
位 13	Res：保留 必须保持复位值。
位 12	SPI1RST：SPI1 复位 该位由软件置1 或清0。 • 0：无作用 • 1：复位SPI1
位 11	TIM1RST：TIM1 定时器复位 该位由软件置 1 或清 0。 • 0：无作用 • 1：复位 TIM1 定时器
位 10	Res：保留 必须保持复位值。
位 9	ADCRST：ADC 接口复位 该位由软件置 1 或清 0。 • 0：无作用 • 1：复位 ADC 接口
位 8:1	Res：保留 必须保持复位值。
位 0	SYSCFGRST：SYSCFG 复位 该位由软件置 1 或清 0。 • 0：无作用 • 1：复位 SYSCFG 和 COMP

6.3.5 APB1 外设复位寄存器（RCC_APB1RSTR）

偏移地址：0x10

复位值：0x0000 0000

访问：无等待周期，支持字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	IOMUXRST	BEEPERRST	PWRRST	Res						I2C1RST	Res				AWURST
	rw	rw	rw							rw					rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				WWDGRST	Res						TIM6RST	Res			TIM2RST
				rw							rw	rw			rw

位 31	Res：保留 必须保持复位值。
位 30	IOMUXRST：引脚功能多重映射复位 该位由软件置1 或清0。 • 0：无作用 • 1：复位引脚功能多重映射
位 29	BEEPERRST：蜂鸣器复位 该位由软件置1 或清0。 • 0：无作用 • 1：复位蜂鸣器
位 28	PWRRST：电源接口复位 该位由软件置1 或清0。 • 0：无作用 • 1：复位电源接口
位 27:22	Res：保留 必须保持复位值。
位 21	I2C1RST：I2C1 复位 该位由软件置1 或清0。 • 0：无作用 • 1：复位I2C1
位 20:17	Res：保留 必须保持复位值。
位 16	AWURST：停机模式下的自动唤醒复位 • 0：无作用 • 1：复位自动唤醒
位 15:12	Res：保留 必须保持复位值。
位 11	WWDGRST：窗口看门狗复位 该位由软件置1 或清0。 • 0：无作用 • 1：复位窗口看门狗
位 10:5	Res：保留 必须保持复位值。

位 4	TIM6RST：TIM6 定时器复位 该位由软件置1 或清0。 • 0：无作用 • 1：复位TIM6
位 3:1	Res：保留 必须保持复位值。
位 0	TIM2RST：TIM2 定时器复位 该位由软件置1 或清0。 • 0：无作用 • 1：复位TIM2

6.3.6 AHB 外部时钟使能寄存器（RCC_AHBENR）

偏移地址：0x14

复位值：0x0000 0014

访问：无等待周期，支持字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res											IOPDEN	IOPCEN	IOPBEN	IOPAEN	Res
											rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res									CRCEN	Res	FLITFEN	Res	SRAMEN	Res	
									rw		rw		rw		

位 31:21	Res：保留 必须保持复位值。
位 20	IOPDEN：GPIOD 时钟使能 该位由软件置 1 或清 0。 • 0：GPIOD 时钟关闭 • 1：GPIOD 时钟开启
位 19	IOPCEN：GPIOC 时钟使能 该位由软件置 1 或清 0。 • 0：GPIOC 时钟关闭 • 1：GPIOC 时钟开启
位 18	IOPBEN：GPIOB 时钟使能 该位由软件置 1 或清 0。 • 0：GPIOB 时钟关闭 • 1：GPIOB 时钟开启
位 17	IOPAEN：GPIOA 时钟使能

	该位由软件置 1 或清 0。 • 0：GPIOA 时钟关闭 • 1：GPIOA 时钟开启
位 16:7	Res：保留 必须保持复位值。
位 6	CRCEN：CRC 时钟使能 该位由软件置 1 或清 0。 • 0：CRC 时钟关闭 • 1：CRC 时钟开启
位 5	Res：保留 必须保持复位值。
位 4	FLITFEN：FLITF 时钟使能 该位由软件置 1 或清 0 来开启/关闭在睡眠模式下的FLITF 时钟。 • 0：在睡眠模式下FLITF 时钟关闭 • 1：在睡眠模式下FLITF 时钟开启
位 3	Res：保留 必须保持复位值。
位 2	SRAMEN：SRAM 接口时钟使能 该位由软件置 1 或清 0 来开启/关闭在睡眠模式下的SRAM 时钟。 • 0：在睡眠模式下SRAM 接口时钟关闭 • 1：在睡眠模式下SRAM 接口时钟开启
位 1:0	Res：保留 必须保持复位值。

6.3.7 APB 外设时钟使能寄存器 2（RCC_APB2ENR）

偏移地址：0x18

复位值：0x0000 0000

访问：支持字、半字和字节访问；无等待周期，除了上一次的 APB 访问未完成的情况下，必须插入等待周期直到该次访问完成。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res									DBGMCUEN	Res					
									rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	USART1EN	Res	SPI1EN	TIM1EN	Res	ADCEN	Res								SYSCFGEN
	rw		rw	rw		rw									rw

位 31:23	Res：保留 必须保持复位值。
位 22	DBGMCUEN：MCU 调试模块时钟使能 该位由软件置1 或清0。 • 0：MCU 调试模块时钟关闭 • 1：MCU 调试模块时钟开启
位 21:15	Res：保留 必须保持复位值。
位 14	USART1EN：USART1 时钟使能 该位由软件置1 或清0。 • 0：USART1 时钟关闭 • 1：USART1 时钟开启
位 13	Res：保留 必须保持复位值。
位 12	SPI1EN：SPI1 时钟使能 该位由软件置1 或清0。 • 0：SPI1 时钟关闭 • 1：SPI1 时钟开启
位 11	TIM1EN：TIM1 定时器时钟使能 该位由软件置1 或清0。 • 0：TIM1 定时器时钟关闭 • 1：TIM1 定时器时钟开启
位 10	Res：保留 必须保持复位值。
位 9	ADCEN：ADC 接口时钟使能 该位由软件置1 或清0。 • 0：ADC 接口时钟关闭 • 1：ADC 接口时钟开启
位 8:1	Res：保留 必须保持复位值。
位 0	SYSCFGEN：SYSCFG 时钟使能 该位由软件置1 或清0。 • 0：SYSCFG 时钟关闭

	• 1：SYSCFG 时钟开启

6.3.8 APB1 外设时钟使能寄存器（RCC_APB1ENR）

偏移地址：0x1C

复位值：0x0000 0000

访问：支持字、半字和字节访问；无等待周期，除了上一次的 APB 访问未完成的情况下，必须插入等待周期直到该次访问完成。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	IOMUXEN	BEEPEREN	PWREN	Res						I2C1EN	Res				AWUEN
	rw	rw	rw												rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				WWDGEN	Res						TIM6EN	Res			TIM2EN
				rw							rw				rw

位 31	Res：保留 必须保持复位值。
位 30	IOMUXEN：引脚功能多重映射使能 该位由软件置 1 或清 0。 • 0：引脚功能多重映射功能关闭 • 1：引脚功能多重映射功能开启
位 29	BEEPEREN：蜂鸣器使能 该位由软件置 1 或清 0。 • 0：蜂鸣器关闭 • 1：蜂鸣器开启
位 28	PWREN：Power 接口时钟使能 该位由软件置 1 或清 0。 • 0：Power 接口时钟关闭 • 1：Power 接口时钟开启
位 27:22	Res：保留 必须保持复位值。
位 21	I2C1EN：I2C1 时钟使能 该位由软件置 1 或清 0。 • 0：I2C1 时钟关闭 • 1：I2C1 时钟开启
位 20:17	Res：保留 必须保持复位值。
位 16	AWUEN：Stop 模式的自动唤醒使能

	该位由软件置 1 或清 0。 • 0：自动唤醒关闭 • 1：自动唤醒开启
位 15:12	Res：保留 必须保持复位值。
位 11	WWDGEN：看门狗时钟使能 该位由软件置 1 或清 0。 • 0：看门狗时钟关闭 • 1：看门狗时钟开启
位 10:5	Res：保留 必须保持复位值。
位 4	TIM6EN：TIM6 定时器时钟使能 该位由软件置 1 或清 0。 • 0：TIM6 时钟关闭 • 1：TIM6 时钟开启
位 3:1	Res：保留 必须保持复位值。
位 0	TIM2EN：TIM2 定时器时钟使能 该位由软件置 1 或清 0。 • 0：TIM2 时钟关闭 • 1：TIM2 时钟开启

6.3.9 控制/状态寄存器（RCC_CSR）

偏移地址：0x24

复位值：0x0000 0000

除了复位标志（RMVF）外，RCC_CSR 寄存器的其他位域由系统复位进行复位，RMVF 位由电源复位清除。

访问：0 到3 等待周期，支持字、半字和字节访问；当连续对该寄存器进行访问时，将插入等待状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWRRSTF	WWDGRSTF	IWDGRSTF	SFTRSTF	PORRSTF	PINRSTF	Res	RMVF	Res							
r	r	r	r	r	r	r	r								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res														LSIRDY	LSION
														r	rw

位 31	LPWRRSTF：低功耗复位标志

	在低功耗管理复位发生时，该位由硬件置 1。 该位由软件通过写 RMVF 位清除。 • 0：无低功耗管理复位发生 • 1：发生低功耗管理复位
位 30	WWDGRSTF：窗口看门狗复位标志 在窗口看门狗复位发生时，该位由硬件置 1。 该位由软件通过写 RMVF 位清除。 • 0：无窗口看门狗复位发生 • 1：发生窗口看门狗复位
位 29	IWDGRSTF：独立看门狗复位标志 在独立看门狗复位发生时，该位由硬件置 1。 该位由软件通过写 RMVF 位清除。 • 0：无看门狗复位发生 • 1：发生看门狗复位
位 28	SFTRSTF：软件复位标志 在软件复位发生时，该位由硬件置 1。 该位由软件写 RMVF 位清除。 • 0：无软件复位发生 • 1：发生软件复位
位 27	PORRSTF：上电/掉电复位标志 在 NRST 引脚复位发生时，该位由硬件置 1。 该位由软件通过写 RMVF 位清除。 • 0：无 NRST 引脚复位发生 • 1：发生 NRST 引脚复位
位 26	PINRSTF：NRST 引脚复位标志 在 NRST 引脚复位发生时，该位由硬件置 1。 该位由软件通过写 RMVF 位清除。 • 0：无 NRST 引脚复位发生 • 1：发生 NRST 引脚复位
位 25	Res：保留 必须保持复位值。
位 24	RMVF：清除复位标志 该位由软件置 1 来清除复位标志。 • 0：无作用 • 1：清除复位标志

位 23:2	Res：保留 必须保持复位值。
位 1	LSIRDY：LSI 振荡器就绪 通过硬件置 1 或清 0 来指示内部 LSI 振荡器是否就绪。在 LSION 清零后，等待 3 个 LSI 振荡器的周期后 LSIRDY 被清零。 • 0：LSI 振荡器未就绪 • 1：LSI 振荡器就绪
位 0	LSION：LSI 振荡器使能 该位由软件置 1 或清 0。 • 0：LSI 振荡器关闭 • 1：LSI 振荡器开启

6.3.10 AHB 外设复位寄存器（RCC_AHBRSTR）

偏移地址：0x28

复位值：0x0000 0000

访问：无等待周期，支持字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res											IOPDRST	IOPCRST	IOPBRST	IOPARST	Res
											rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res									CRCRST	Res					
									rw						

位 31:21	Res：保留 必须保持复位值。
位 20	IOPDRST：GPIOD 口复位 该位由软件置1 或清0。 • 0：无作用 • 1：复位GPIOD 口
位 19	IOPCRST：GPIOC 口复位 该位由软件置1 或清0。 • 0：无作用 • 1：复位GPIOC 口
位 18	IOPBRST：GPIOB 口复位 该位由软件置1 或清0。 • 0：无作用 • 1：复位GPIOB 口

位 17	IOPARST：GPIOA 口复位 该位由软件置1 或清0。 • 0：无作用 • 1：复位GPIOA 口
位 16:7	Res：保留 必须保持复位值。
位 6	CRCRST：CRC 复位 该位由软件置1 或清0。 • 0：无作用 • 1：复位CRC
位 5:0	Res：保留 必须保持复位值。

6.3.11 时钟配置寄存器 3（RCC_CFGR3）

偏移地址：0x30

复位值：0x0000 0000

访问：无等待周期，支持字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res											I2C1SW	Res		USART1SW	
											rw	rw		rw	

位 31:5	Res：保留 必须保持复位值。
位 4	I2C1SW：选择 I2C 时钟源 该位和 RCC_CFGR4.I2C1CLK_SEL 共同决定 I2C1 时钟源，由软件置 1 和清0。 当 RCC_CFGR4.I2C1CLK_SEL 为 0 时，I2C1 的时钟源为： • 0：HSI 时钟被选为I2C1 时钟源（缺省）。 • 1：系统时钟（SYSCLK）被选为I2C1 的时钟源。
位 3:2	Res：保留 必须保持复位值。
位 1:0	USART1SW[1:0]：USART1 时钟源选择 由软件置 1 和清0 来选择 USART1 的时钟源。 • 00：PCLK 被选为 USART1 的时钟源（缺省）。 • 01：系统时钟（SYSCLK）被选为 USART1 的时钟源。

	• 10：保留 • 11：HSI 时钟被选为 USART1 的时钟源。

6.3.12 控制寄存器（RCC_CSS）

偏移地址：0xe0

复位值：0x1e00 0000

访问：无等待周期，支持字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSS_THRESHOLD[6:0]							Res								
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res															

位 31:25	CSS_THRESHOLD：控制 CSS 计数器的阈值。 当 CSS 功能开启后： • RCC 使用 HSI32MHz 时钟来采样 GPIO 外部输入时钟分频后的波形。 • 如果 GPIO 输入频率非常低，则即使 GPIO 输入时钟正常也可能触发 CSS 中断，而通过配置 CSS_THRESHOLD 的值可避免这种情况。产生 CSS 中断的最低 GPIO 输入频率大约为 16M/ CSS_THRESHOLD。因此若 CSS_THRESHOLD 等于复位值，当 GPIO 输入小于 1 MHz 时就会产生 NMI 中断。
位 24:0	Res：保留 必须保持复位值。

6.3.13 时钟配置寄存器 4（RCC_CFGR4）

偏移地址：0xe8

复位值：0x280a 280a

访问：无等待周期，支持字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	ADCHSIPRE[4:0]					EXTCLK_SEL[1:0]		Res			I2CHSIPRE[4:0]				
	rw					rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C1CLK_SEL	FLITFCLK_PRE[3:0]				FLITFCLK_SEL[1:0]		Res				USARTHSIPRE[4:0]				
rw	rw				rw						rw				

位 31	Res：保留 必须保持复位值。
位 30:26	ADCHSIPRE[4:0]：ADC 时钟相对于 HSI32M 的分频系数 • 00：HSI32M/1 • 01：HSI32M/1.5 • 02：HSI32M/2 • 03：HSI32M/2.5

	• … • 1f：HSI32M/16.5
位 25:24	EXTCLK_SEL[1:0]：外部时钟输入管脚选择 • 00：外部时钟管脚 1（EXTCLK1）选择作为输入源 • 01：外部时钟管脚 2（EXTCLK2）选择作为输入源 • 10：外部时钟管脚 3（EXTCLK3）选择作为输入源 • 11：外部时钟管脚 4（EXTCLK4）选择作为输入源
位 23:21	Res：保留 必须保持复位值。
位 20:16	I2CHSIPRE[4:0]：I2C 时钟相对于 HSI32M 的分频系数 分频设置和 ADCHSIPRE 相同。
位 15	I2C1CLK_SEL：选择 I2C1 的时钟 • 0：I2C1 时钟由 RCC_CFGR3.I2C1SW 决定 • 1：PCLK 作为 I2C1 时钟
位 14:11	FLITFCLK_PRE[3:0]：Flash 擦写操作时钟分频因子 分频后的 FLITFCLK 必须满足：2 MHz ≤ FLITFCLK ≤ 6 MHz，其值越大，Flash 擦除所需时间越短。 分频系数等于 FLITFCLK_PRE + 1。 *注意：不能在执行 Flash 编程和擦除的时候，更改 FLITFCLK_PRE 寄存器的值。*
位 10:9	FLITFCLK_SEL[1:0]：FLITFCLK 分频时钟源选择 • 00：选择 HSI32MHz 分频时钟作为 FLITFCLK 分频时钟源。 • 01：选择 SYSCLK 作为 FLITFCLK 分频时钟源。 • 其他：保留 *注意：不能在执行 Flash 编程和擦除的时候更改 FLITFCLK_SEL 寄存器的值。*
位 8:5	Res：保留 必须保持复位值。
位 4:0	USARTHSIPRE[4:0]：USART 相对于 HSI32M 的分频系数 分频设置和 ADCHSIPRE 相同。

7 系统配置控制器（SYSCFG）

HK32F030M 有一组配置寄存器，系统配置寄存器的主要目的如下：

- 代码区开始区存储的重映射。
- 管理外部中断与 GPIO 的连接。
- 管理系统的可靠性特性。

7.1 SYSCFG 寄存器

基地址：0x4001 0000

空间大小：0x400

7.1.1 SYSCFG 配置寄存器 1（SYSCFG_CFGR1）

偏移地址：0x00

复位值：0x0000 0000

复位后，该寄存器的值对应芯片的实际启动模式。可通过软件选择物理地址重新映射，从而绕过硬件引导程序选择。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCKUP_LOCK	Res														
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res														MEM_MODE	
														rw	

位 31	LOCKUP_LOCK：Cortex-M0 LOCKUP 位使能 该位通过软件置 1，系统复位。它用于断开和连通 Cortex-M0 LOCKUP（Hardfault）到 TIM1 Break 输入之间的连接。 • 0：Cortex-M0 LOCKUP 输出到 TIM1 Break 输入断开。 • 1：Cortex-M0 LOCKUP 输出到 TIM1 Break 输入连接。
位 30:2	Res：保留 必须保持复位值。
位 1:0	MEM_MODE：存储映射选择位 该位域由软件置位和清 0。它控制存储器映射到片内地址 0x0000 0000。 • x0：主闪存存储器映射到0x0000 0000 • 11：嵌入式RAM 映射到0x0000 0000

7.1.2 SYSCFG 外部中断配置寄存器 1（SYSCFG_EXTICR1）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
rw				rw				rw				rw			

位 31:16	Res：保留 必须保持复位值。
位 4x+3:4x （x = 0..3）	EXTIx[3:0]：选择 EXTIx 的外部中断源（x = 0..3） 该位域由软件设置。 • x000：PA[x]引脚 • x001：PB[x]引脚 • x010：PC[x]引脚 • x011：PD[x]引脚 • 其它配置：保留

7.1.3 SYSCFG 外部中断配置寄存器 2（SYSCFG_EXTICR2）

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
rw				rw				rw				rw			

位 31:16	Res：保留 必须保持复位值。
位 4(x-4)+3:4(x-4) （x = 4..7）	EXTIx[3:0]：选择 EXTIx 的外部中断源（x = 4..7） 该位域由软件设置。 • x000：PA[x]引脚 • x001：PB[x]引脚 • x010：PC[x]引脚 • x011：PD[x]引脚 • 其它配置：保留

8 通用 I/O（GPIO）

本章介绍该系列芯片的 GPIO 功能和寄存器描述。

8.1 GPIO 的主要特性

- 输出状态：推挽或开漏（均带上拉/下拉功能）
- 输入状态：浮空、上拉/下拉
- 模拟功能
- 复用功能选择寄存器
- 从输出数据寄存器（GPIOx_ODR）或外设（复用功能输出）输出数据。
- 可以为每个 I/O 口选择不同的速度。
- 将数据输入到输入寄存器（GPIOx_IDR）或外设（复用功能输入）。
- 置位和复位寄存器（GPIOx_BSRR），对 GPIOx_ODR 具有按位写权限。
- 锁定机制（GPIOx_LCKR），可冻结 I/O 端口配置。
- 快速翻转，每次翻转最快只需要两个时钟周期。
- 引脚复用灵活，允许将 I/O 引脚用作 GPIO 或者多种外设功能中的一种。
- 所有 I/O 都可以选择打开或关闭施密特特性。

8.2 GPIO 功能描述

根据数据手册中列出的每个 I/O 端口的特性，可通过软件将通用 I/O（GPIO）端口的各个端口位分别配置为以下任意模式：

- 输入浮空
- 输入上拉
- 输入下拉
- 模拟输入
- 具有上拉或下拉的开漏输出
- 具有上拉或下拉的推挽输出
- 具有上拉或下拉的复用功能推挽
- 具有上拉或下拉的复用功能开漏
- 所有 I/O 可以打开或关闭施密特特性

每个 I/O 端口均可自由编程，但 I/O 端口寄存器必须按字（32 位）、半字（16 位）、或者是字节进行访问。GPIOx_BSRR 寄存器和 GPIOx_BRR 寄存器旨在实现对 GPIOx_ODR 寄存器的原子读写操作，以确保在读取和修改访问之间发生中断请求也不会有问题。

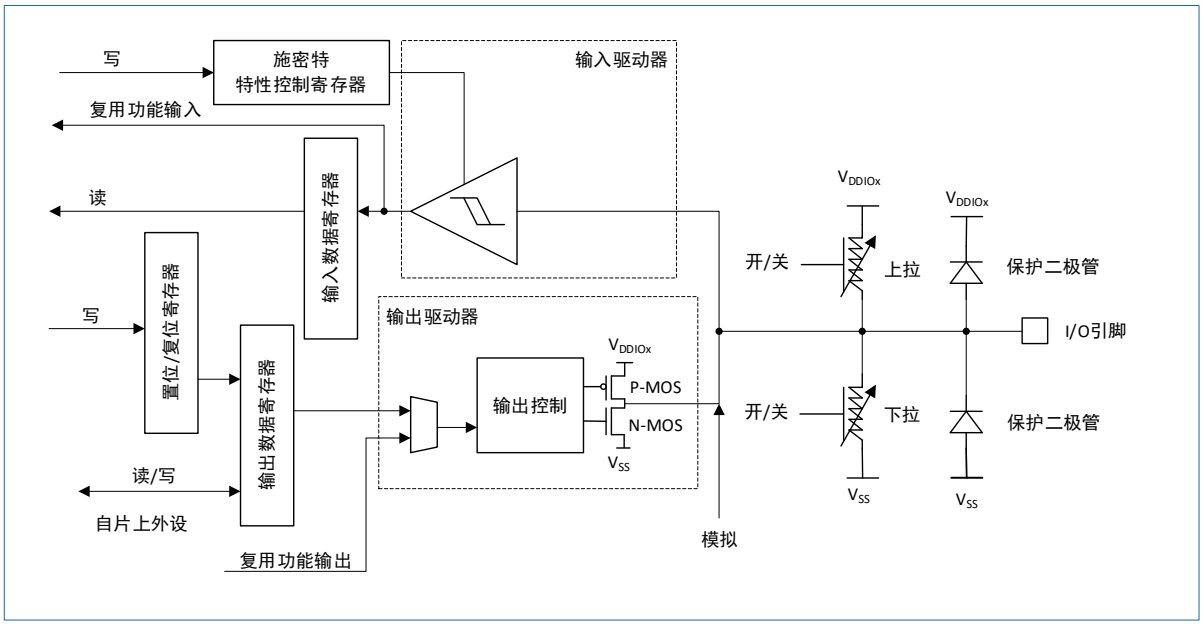


图 8-1 I/O 端口位的基本结构

表 8-1 端口位配置表

MODER（i）[1:0]	OTYPER（i）	OSPEEDR（i）[1:0]	PUPDR（i）[1:0]		I/O 配置	
01	0	OSPEEDER[1:0]	0	0	通用输出	推挽
	0		0	1	通用输出	推挽+上拉
	0		1	0	通用输出	推挽+下拉
	0		1	1	保留	
	1		0	0	通用输出	开漏
	1		0	1	通用输出	开漏+上拉
	1		1	0	通用输出	开漏+下拉
	1		1	1	保留（通用输出开漏）	
10	0	OSPEEDER[1:0]	0	0	复用功能	推挽
	0		0	1	复用功能	推挽+上拉
	0		1	0	复用功能	推挽+下拉
	0		1	1	保留	
	1		0	0	复用功能	开漏
	1		0	1	复用功能	开漏+上拉
	1		1	0	复用功能	开漏+下拉
	1		1	1	保留	
00	X（1）	X	0	0	输入	浮空
	X	X	0	1	输入	上拉

MODER（i）[1:0]	OTYPER（i）	OSPEEDR（i）[1:0]	PUPDR（i）[1:0]		I/O 配置	
	X	X	1	0	输入	下拉
	X	X	1	1	保留（输入浮空）	
11	X	X	0	0	输入/输出	模拟
	X	X	0	1	保留	
	X	X	1	0		
	X	X	1	1		

（1）“X”表示无影响，不起作用。

8.2.1 通用 I/O（GPIO）

在复位期间及复位刚完成时，复用功能尚未激活，大多数 I/O 端口都会默认为浮空输入模式，但调试引脚例外：

- PB5：SWCLK 处于下拉状态
- PD5：SWDIO 处于上拉状态

当引脚配置为输出后，写入到输出寄存器（GPIOx_ODR）的值将在 I/O 引脚上输出。可以在推挽模式下或开漏模式下使用输出驱动器（仅驱动低电平，高电平为高阻态）。

输入数据寄存器（GPIOx_IDR）每隔 1 个 AHB 时钟周期捕获一次 I/O 引脚的数据。

所有 GPIO 引脚都具有内部弱上拉及下拉电阻，可根据 GPIOx_PUPDR 寄存器中的值来打开/关闭。

8.2.2 I/O 引脚复用功能复用器和映射

器件 I/O 引脚通过一个复用器连接到外设/模块，该复用器一次仅允许一个外设的复用功能（AF）连接到同一个 I/O 引脚。这可以确保共用同一个 I/O 引脚的外设之间不会发生冲突。

每个 I/O 引脚都有一个复用器，该复用器采用多达 8 路复用功能输入（AF0 到 AF7），可通过 GPIOx_AFRL（针对引脚 0 至 7）寄存器对这些输入进行配置。

复位后，复用器选择为复用功能 0（AF0）。在复用模式下通过 GPIOx_MODER 寄存器配置 IO。

器件数据手册中详细说明了每个引脚的特定复用功能分配。

除了这种灵活的 I/O 复用架构外，各外设还可以将复用功能映射到不同 I/O 引脚，以优化小型封装中可用外设的数量。

要在指定配置下使用 I/O，用户必须按照以下步骤操作：

- 调试功能：每个器件复位后，立即将这些引脚分配为可由调试主机使用的复用功能引脚。
- GPIO：在 GPIOx_MODER 寄存器中将所需 I/O 配置为输出、输入或模拟。
- 外设复用功能：
  A. 在 GPIOx_AFRL 寄存器中，将 I/O 连接到所需的 AFx。
  B. 通过 GPIOx_OTYPER、GPIOx_PUPDR 和 GPIOx_OSPEEDR 寄存器，分别选择类型、上拉/下拉以及输出速度。
  C. 在 GPIOx_MODER 寄存器中将所需的 I/O 配置为复用功能。
- 其它功能：

  对于 ADC，在 GPIOx_MODER 寄存器中将所需 I/O 配置为模拟模式，并在 ADC 寄存器中配置所

需功能。

对于 WKUP 和振荡器的其它功能，在相关的 PWR 和 RCC 寄存器中配置所需功能。这些功能优先于标准 GPIO 寄存器中的配置。

8.2.3 I/O 端口控制寄存器

每个通用 I/O 端口包括：

- 7 个 32 位的控制寄存器，可配置多达 8 个 IO。
  - 端口模式寄存器 GPIOx_MODER，用于选择 I/O 模式（输入、输出、AF 或模拟）。
  - 端口输出类型寄存器 GPIOx_OTYPER，用于选择输出类型（推挽或开漏）。
  - 端口输出速度寄存器 GPIOx_OSPEEDR，用于选择速度。
  - 端口上下拉寄存器 GPIOx_PUPDR，用于选择上拉/下拉。
  - 端口置位/复位寄存器 GPIOx_BSRR，用于端口置位/复位。
  - 端口复位寄存器 GPIOx_BRR，用于端口复位。
  - 端口施密特寄存器 GPIOx_IOSR，用于配置 I/O 的施密特特性。
    器件复位后 I/O 的施密特特性默认开启（包括 SWDIO 和 SWCLK 两个引脚）
- 1 个 32 位锁定寄存器 GPIOx_LCKR
- 1 个 32 位复用功能选择寄存器 GPIOx_AFRL

8.2.4 I/O 端口数据寄存器

2 个 32 位的数据寄存器：

- 端口输入数据寄存器 GPIOx_IDR，通过 I/O 输入的数据存储到该寄存器。
- 端口输出数据寄存器 GPIOx_ODR，用于存储待输出数据，可以其进行读/写访问。

8.2.5 I/O 数据位操作

置位/复位寄存器 GPIOx_BSRR 是一个 32 位寄存器，它允许应用程序在输出数据寄存器 GPIOx_ODR 中对各个单独的数据位执行置位和复位操作。GPIOx_ODR 中的每个数据位对应于 GPIOx_BSRR 中的两个控制位：BSy 和 BRy（y=15..0）。当将 BSy 置位时，会置位对应的 ODRy；当将 BRy 置位时会复位对应的 ODRy。向 GPIOx_BSRR 中任何位写 0 都不会对 GPIOx_ODR 中的对应位产生任何的影响。如果在 GPIOx_BSRR 中同时尝试对 BSy 和 BRy 写'1'时，对 BSy 写'1'的操作优先，对 BRy 写'1'的操作被忽略掉。

使用 GPIOx_BSRR 寄存器更改 GPIOx_ODR 中各位的值是一个"单次"操作，不会锁定 GPIOx_ODR 位。随时都可以直接访问 GPIOx_ODR 位。GPIOx_BSRR 寄存器只是提供了一种对 GPIOx_ODR 寄存器执行原子按位处理的方法。

在对 GPIOx_ODR 进行位操作时，软件无需禁止中断：在一次原子 AHB 写访问中，可以修改一个或多个位。

8.2.6 GPIO 锁定机制

通过将特定的写序列应用到 GPIOx_LCKR 寄存器，可以冻结 GPIO 控制寄存器。冻结的寄存器包括 GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_IOSR。

对 GPIOx_LCKR 寄存器执行写操作必须写入特定的写序列。当正确的 LOCK 序列写到此寄存器的第 16 位后，会使用 LCK[7:0]的值来锁定对应 I/O 的配置（在写序列期间，LCK[15:0]的值必须保持不变）。将 LOCK 序列写到某个端口位后，在执行下一次 MCU 复位或外设复位之前，将无法对该端口位的值进行更改。每个 GPIOx_LCKR 位都会冻结 GPIO 控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、

GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_IOSR）中的对应位。

8.2.7 I/O 复用功能输入输出

每个通用 I/O 端口包括 1 个 32 位复用功能选择寄存器 GPIOx_AFRL。这两个寄存器用于从每个 I/O 可用的复用功能输入/输出中进行选择。根据应用程序的要求，用户可将某个复用功能连接到指定的 I/O 引脚上。由于 AF 选择信号由复位功能输入和复用功能输出共用，所以只需要为指定的 I/O 的复用功能输入/输出选择一个通道即可。

8.2.8 外部中断线/唤醒线

所有 I/O 端口都具有外部中断功能。如果使用外部中断线，必须将端口配置为输入模式。

8.2.9 输入配置

对 I/O 端口进行编程作为输入时：

- 输出驱动器被禁用。
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开上拉或和下拉电阻。
- 每隔 1 个 AHB 时钟周期，输入数据寄存器对 I/O 引脚上的数据进行一次采样。
- 对输入数据寄存器的读访问可获取 I/O 状态。

8.2.10 输出配置

对 I/O 端口进行编程作为输出时：

- 输出缓冲器被打开
  - 开漏模式：输出寄存器中的’0’可激活 N-MOS，而输出寄存器中的’1’会使端口保持高阻态（Hi-Z），P-MOS 始终不激活。
  - 推挽模式：输出寄存器中的’0’可激活 N-MOS，输出寄存器中的’1’可激活 P-MOS。
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开上拉和下拉电阻。
- 输入数据寄存器每隔 1 个 AHB 时钟周期对 I/O 引脚上的数据采样一次。
- 在开漏模式时，对输入数据寄存器的读访问可获取 I/O 的状态。
- 在推挽模式时，对输出数据寄存器的读访问可获取最后的写入值。

8.2.11 复用功能配置

对 I/O 端口进行编程作为复用功能时：

- 可将输出缓冲器配置为开漏或推挽模式。
- 输出缓冲器由来自外设的信号驱动（发送使能和数据）。
- 根据 GPIOx_IOSR 寄存器的配置使能或禁用施密特触发器。
- 根据 GPIOx_PUPDR 寄存器的配置决定是否打开上拉电阻和下拉电阻。
- 输入数据寄存器每隔 1 个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样。
- 对输入数据寄存器的读访问可获取 I/O 状态。

8.2.12 模拟配置

对 I/O 端口进行编程作为模拟配置时：

- 输出缓冲器被禁用。
- 施密特触发器被强制停用，I/O 引脚每个模拟输入的功耗变为零。施密特触发器的输出被强制

为恒定值’0’。

- 弱上拉和下拉电阻被硬件强制关闭。
- 对输入数据寄存器的读访问值为’0’。

8.2.13 施密特功能配置

当 I/O 工作在模拟模式下时，施密特触发器被强制关闭。除此之外的其他模式下，I/O 的施密特触发器状态由 GPIOx_IOSR 寄存器中的位决定。在数字信号采样时，建议开启施密特触发器，这样可以增强数据采样的搞干扰能力。

8.3 GPIO 寄存器

基地址：（GPIOA; GPIOB, GPIOC, GPIOD）=（0x4800 0000; 0x4800 0400, 0x4800 0800, 0x4800 0C00）

空间大小：（GPIOA; GPIOB, GPIOC, GPIOD）=（0x400; 0x400, 0x400, 0x400）

8.3.1 GPIO 端口模式寄存器（GPIOx_MODER）（x = A..D）

偏移地址：0x00

复位值：0x2800 0000（端口 A）/ 0x0000 0000（其他端口）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7		MODER6		MODER5		MODER4		MODER3		MODER2		MODER1		MODER0	
rw		rw		rw		rw		rw		rw		rw		rw	

位 31:16	Res：保留 必须保持复位值。
位 2y+1:2y（y=0..7）	MODERy[1:0]：端口 x 的工作模式（y=0..7） 该位域可由软件配置 I/O 口模式。 • 00：输入模式（复位状态） • 01：通用输出模式 • 10：复用功能模式 • 11：模拟模式

8.3.2 GPIO 端口输出类型寄存器（GPIOx_OTYPER）（x = A..D）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8	Res：保留

	必须保持复位值。
位 y （y=0..7）	OTy：端口x 的输出类型（y=0..7） 该位由软件配置I/O 口的输出类型。 • 0：推挽输出（复位状态） • 1：开漏输出

8.3.3 GPIO 口输出速度寄存器（GPIOx_OSPEEDR）（x = A..D）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEEDR7		OSPEEDR6		OSPEEDR5		OSPEEDR4		OSPEEDR3		OSPEEDR2		OSPEEDR1		OSPEEDR0	
rw		rw		rw		rw		rw		rw		rw		rw	

位 31:16	Res：保留 必须保持复位值。
位 2y+1:2y （y=0..7）	OSPEEDRy[1:0]：端口x 的输出速度（y=0..7） 该位域由软件配置I/O 口的速度。 • 00：低速 • 01：中速

8.3.4 GPIO 口上拉/下拉寄存器（GPIOx_PUPDR）（x = A..D）

偏移地址：0x0C

复位值：0x2400 0000（端口 A）/ 0x0000 0000（其它端口）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR7		PUPDR6		PUPDR5		PUPDR4		PUPDR3		PUPDR2		PUPDR1		PUPDR0	
rw		rw		rw		rw		rw		rw		rw		rw	

位 31:16	Res：保留 必须保持复位值。
位 2y+1:2y （y=0..7）	PUPDRy[1:0]：端口x 配置为上拉或下拉。（y=0..7） 该位域由软件配置I/O 口为上拉或下拉。 • 00：无上拉和下拉 • 01：上拉 • 10：下拉 • 11：保留

8.3.5 GPIO 端口输入数据寄存器（GPIOx_IDR）（x = A..D）

偏移地址：0x10

复位值：0x0000 XXXX（X 表示不定值）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
								r	r	r	r	r	r	r	r

位 31:8	Res：保留 必须保持复位值。
位 y （y=0..7）	IDRy：端口 x 的输入数据（y=0..7） 该位包含相应 I/O 口的输入值，只能读。

8.3.6 GPIO 端口输出数据寄存器（GPIOx_ODR）（x = A..D）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8	Res：保留 必须保持复位值。
位 y （y=0..7）	ODRy：端口输出数据（y=0..7） 该位由软件读写。

8.3.7 GPIO 端口置位/复位寄存器（GPIOx_BSRR）（x = A..D）

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res								BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
								w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
								w	w	w	w	w	w	w	w

位 31:24	Res：保留 必须保持复位值。

位 y （y=16..23）	BRy：端口x 复位位y 该位只能写。 读该位时返回值为0。若 BSx 和 BRx 同时设置，BSx 有优先权。 • 0：对相应的 ODRy 位无影响 • 1：复位相应的 ODRy 位
位 15:8	Res：保留 必须保持复位值。
位 y （y=0..7）	BSy：端口x 设置位y（y=0..7） 该位只能写。 读该位时返回值为0。 • 0：对相应的ODRy 位无影响 • 1：置位相应的ODRy 位

8.3.8 GPIO 端口配置锁定寄存器（GPIOx_LCKR）（x = A..B）

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															LCKK
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:17	Res：保留 必须保持复位值。
位 16	LCKK：锁定键 该位可随时读取，仅能由锁定键写序列来改写。 • 0：端口配置锁定键不激活 • 1：端口配置锁定键激活。GPIOx_LCKR 寄存器保持锁定，直到 MCU 复位产生才解除锁定。 锁定键写序列： 1. 写 LCKR[16] = ‘1’ + LCKR[15:0] 2. 写 LCKR[16] = ‘0’ + LCKR[15:0] 3. 写 LCKR[16] = ‘1’ + LCKR[15:0] 4. 读 LCKR 5. 可选操作：读LCKR[16] = ‘1’（确认锁定是否激活） *注意：在锁定键写序列时，不能更改 LCK[15:0]的值。*

位 15:8	Res：保留 必须保持复位值。
位 y （y=0..7）	LCKy：端口x 锁定（y=0..7） 该位可读/写，但仅LCKK 为 0 时写。 • 0：端口配置未锁定 • 1：端口配置锁定

8.3.9 GPIO 复用功能低位寄存器（GPIOx_AFRL）（x = A..D）

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFRL7[3:0]				AFRL6[3:0]				AFRL5[3:0]				AFRL4[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFRL3[3:0]				AFRL2[3:0]				AFRL1[3:0]				AFRL0[3:0]			
rw				rw				rw				rw			

位 4y+3:4y （y = 0..7）	AFRLy[3:0]：端口 x 引脚y 的复用功能选择（y = 0..7） 可由软件配置复用功能I/O 口。 AFRLy 的选择： • 0000：AF0 • 0001：AF1 • 0010：AF2 • 0011：AF3 • 0100：AF4 • 0101：AF5 • 0110：AF6 • 0111：AF7 • 其他值：保留

8.3.10 GPIO 端口位复位寄存器（GPIOx_BRR）（x=A..D）

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
								w	w	w	w	w	w	w	w

位 31:8	Res：保留 必须保持复位值。

位 y （y=0..7）	BRy：端口x 复位位y（y= 0..7） 该位只能写，读该位的返回值为 0。 • 0：对相应的ODRx 位无影响 • 1：复位相应的ODRx 位

8.3.11 GPIO 端口输入输出施密特寄存器（GPIOx_IOSR）（x=A..D）

地址偏移：0x30

复位值：0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IOSEN15	IOSEN14	IOSEN13	IOSEN12	IOSEN11	IOSEN10	IOSEN9	IOSEN8	IOSEN7	IOSEN6	IOSEN5	IOSEN4	IOSEN3	IOSEN2	IOSEN1	IOSEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16	Res：保留 必须保持复位值。
位 y （y=0..7）	IOSENy：端口 x 配置位 y（y = 0..7）施密特特性开关 • 0：使能 IO 施密特功能 • 1：禁能 IO 施密特功能

9 引脚选择功能（IOMUX）

9.1 功能介绍

HK32F030M 的小型封装（SON8/TSSOP16）产品，可通过 IOMUX 实现单根引脚对应多个 GPIO 或外设 IO 的映射控制。以 SON8 封装产品为例进一步说明。

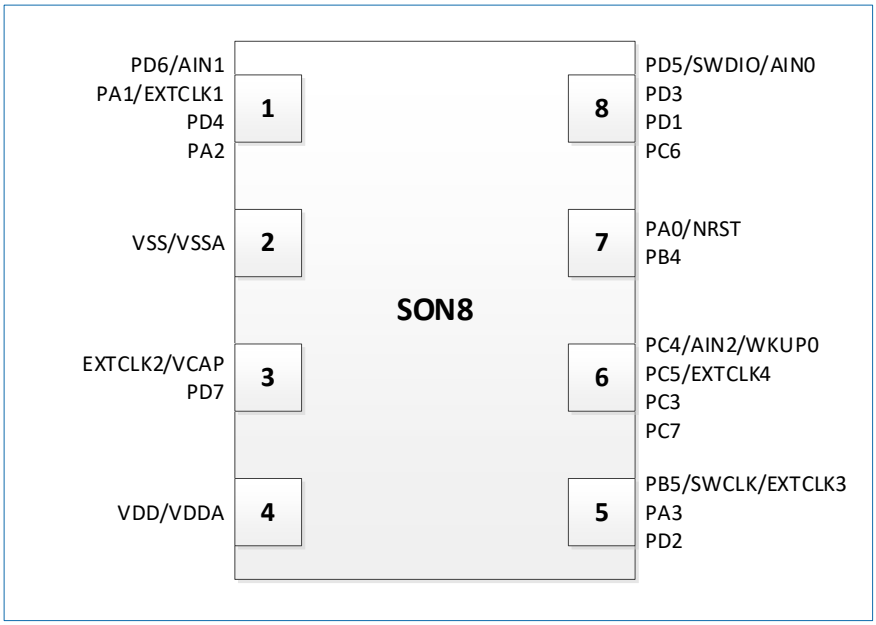


图 9-1 SON8 封装的管脚排列

如图 9-1 所示，芯片初始复位后，第 8 引脚的功能为“PD5（及 SYSCFG 配置中对应的外设 IO）”；通过配置 IOMUX 寄存器，可以将第 8 脚功能重映射到 PD3（及 SYSCFG 配置中对应的外设 IO）、PD1（及 SYSCFG 配置中对应的外设 IO）或 PC6（及 SYSCFG 配置中对应的外设 IO）。

通过 IOMUX 配置，SON8/TSSOP16 封装产品可以使用 18 个 GPIO 以及片内所有外设 IO。

9.2 IOMUX 寄存器

基地址：0x4001 7C00

空间大小：0x400

9.2.1 IOMUX 引脚功能选择寄存器（PIN_FUNC_SEL）

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res													PB5_I2C1_SEL	PC4_TIM1_SEL	PC3_TIM1_SEL
													rw	rw	rw

位 31:3	Res：保留 必须保持复位值。

位 2	PB5_I2C1_SEL：PB5 引脚的功能选择 当 PB5_AF 设置为 AF0 时，PB5_I2C1_SEL 为： • 0：PB5 作为 SWCLK 输入引脚。 • 1：PB5 作为 I2C1 的 SDA 引脚。
位 1	PC4_TIM1_SEL：PC4 引脚的功能选择 当 PC4_AF 设置为 AF3 时，PC4_TIM1_SEL 为： • 0：PC4 作为 TIM1 的 CH4 引脚。 • 1：PC4 作为 TIM1 的 CH2N 引脚。
位 0	PC3_TIM1_SEL：PC3 引脚的功能选择 当 PC3_AF 设置为 AF3 时，PC3_TIM1_SEL 为： • 0：PC3 作为 TIM1 的 CH3 引脚。 • 1：PC3 作为 TIM1 的 CH1N 引脚。

9.2.2 IOMUX 引脚选择寄存器（PKG_PIN_SEL）

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res							PD6_PIN_SEL		PD5_PIN_SEL		PC4_PIN_SEL		PB5_PIN_SEL		NRSTPA0_PIN_SEL
							rw		rw		rw		rw		rw

位 31:9	Res：保留 必须保持复位值。
位 8:7	PD6_PIN_SEL[1:0]：引脚功能选择 • 00：PD6 功能 • 01：PA1 功能（16 和 20 引脚产品禁止此设置） • 10：PD4 功能 • 11：PA2 功能（16 和 20 引脚产品禁止此设置）
位 6:5	PD5_PIN_SEL[1:0]：引脚功能选择 • 00：PD5 功能 • 01：PD3 功能（16 和 20 引脚产品禁止此设置） • 10：PD1 功能 • 11：PC6 功能（16 和 20 引脚产品禁止此设置）
位 4:3	PC4_PIN_SEL[1:0]：引脚功能选择

	• 00：PC4 功能 • 01：PC5 功能（16 和 20 引脚产品禁止此设置） • 10：PC3 功能 • 11：PC7 功能（16 和 20 引脚产品禁止此设置）
位 2:1	PB5_PIN_SEL[1:0]：引脚功能选择 • 00/11：PB5 功能 • 01：PA3 功能（16 和 20 引脚产品禁止此设置） • 10：PD2 功能
位 0	NRSTPA0_PIN_SEL：引脚功能选择 • 0：NRST/PA0 功能 • 1：PB4 功能（16 和 20 引脚产品禁止此设置） *注意：* *NRSTPA0_PIN_SEL 只能通过上电复位。* *只有当 NRST_PIN_KEY[15:0]=0x5AE1 时，才能写入 NRSTPA0_PIN_SEL。*

9.2.3 IOMUX 功能控制寄存器（NRST_PIN_KEY）

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRST_PIN_KEY[15:0]															
rw															

位 31:16	Res：保留 必须保持复位值。
位 15:0	NRST_PIN_KEY[15:0]：NRST_PA0_SEL 的功能从 NRST 改变到 PA0 之前，必须先将该寄存器设置为 0x5AE1。 *注意：* *当 CPU 改变 PKG_PIN_SEL.NRSTPA0_PIN_SEL 或 NRST_PA0_SEL 的值后，NRST_PIN_KEY[15:0]会被系统硬件复位。*

9.2.4 IOMUX 引脚功能控制寄存器（NRST_PA0_SEL）

偏移地址：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res															NRST_PA0_SEL

	rw

位 31:1	Res：保留 必须保持复位值。
位 0	NRST_PA0_SEL：选择 NRST/PA0 引脚的功能 • 0：NRST 功能 • 1：PA0 功能 *注意：* *NRST_PA0_SEL 只能上电复位。* *只有在 NRST_PIN_KEY[15:0]=0x5AE1 时，才能写 NRST_PA0_SEL。*

9.2.5 IOMUX 引脚功能控制寄存器（TIM2_CH0_IN_SEL）

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res														TIM2_CH0_IN_SEL[1:0]	
														rw	

位 31:2	Res：保留 必须保持复位值。
位 1:0	TIM2_CH0_IN_SEL[1:0]：TIM2 通道 1 的输入来源选择 • 00/11：TIM2 通道 1 的输入来源于外部引脚 • 01：TIM2 通道 1 的输入来源于分频后的 HSI 时钟 • 10：TIM2 通道 1 的输入来源于 114 kHz LSI 时钟

10 中断和事件（NVIC 和 EXTI）

本章主要介绍了 NVIC 和 EXTI 的功能特性。

10.1 嵌套向量中断控制器（NVIC）

10.1.1 NVIC 主要特性

嵌套向量中断控制器（NVIC）和处理器核的接口紧密相连，可以实现低延迟的中断处理和高效地处理晚到的中断。嵌套向量中断控制器管理着包括中断在内的所有异常。

- 21 个可屏蔽中断通道（不包括 16 个Cortex-M0 的中断线）
- 4 个可编程的中断优先级（2 位中断优先级）
- 低延迟的异常和中断处理
- 电源管理控制
- 系统控制寄存器的实现

10.1.2 系统嘀嗒校准值寄存器

系统嘀嗒（SysTick）校准值设置为 4000，当 SysTick 时钟设置为 4 MHz（f_{HCLK}/8 的最大值）时，会产生 1 ms 时间基准。

10.1.3 中断和异常向量

表 10-1 NVIC 表

位置	优先级		名称	描述	地址
-	-	-	-	保留	0x0000 0000
-	-3	固定	Reset	复位（Reset）	0x0000 0004
-	-2	固定	NMI	非屏蔽中断 （Non-maskable interrupt）	0x0000 0008
-	-1	固定	HardFault	所有类型错误（All class of fault）	0x0000 000C
-	3	可配置	SVCall	通过 SWI 指令进行的系统服务调度 （System service call via SWI instruction）	0x0000 002C
-	5	可配置	PendSV	可挂起的系统服务请求 （Pendable request for system service）	0x0000 0038
-	6	可配置	SysTick	系统嘀嗒定时器 （System tick timer）	0x0000 003C
0	7	可配置	WWDG	窗口看门狗中断 （Window Watchdog interrupt）	0x0000 0040
1	8	-	-	保留	0x0000 0044
2	9	可配置	EXTI11	外部线 11 的自动唤醒中断 （AWU_WKP）	0x0000 0048

位置	优先级		名称	描述	地址
3	10	可配置	Flash	Flash 全局中断 （Flash global interrupt）	0x0000 004C
4	11	可配置	RCC	RCC 全局中断 （RCC global interrupt）	0x0000 0050
5	12	可配置	EXTI0	EXTI 线 0 中断 （EXTI Line0 interrupt）	0x0000 0054
6	13	可配置	EXTI1	EXTI 线 1 中断 （EXTI Line1 interrupt）	0x0000 0058
7	14	可配置	EXTI2	EXTI 线 2 中断 （EXTI Line2 interrupt）	0x0000 005C
8	15	可配置	EXTI3	EXTI 线 3 中断 （EXTI Line3 interrupt）	0x0000 0060
9	16	可配置	EXTI4	EXTI 线 4 中断 （EXTI Line4 interrupt）	0x0000 0064
10	17	可配置	EXTI5	EXTI 线 5 中断 （EXTI Line5 interrupt）	0x0000 0068
11	18	可配置	TIM1_BRK	TIM1 刹车中断（TIM1 Break interrupt）	0x0000 006C
12	19	可配置	ADC1	ADC 中断（和 EXTI 线 8 共用） （ADC interrupt, combined with EXTI Line8 ）	0x0000 0070
13	20	可配置	TIM1_UP_TRG_COM	TIM1 更新/触发中断 （TIM1 Update/Trigger interrupt）	0x0000 0074
14	21	可配置	TIM1_CC	TIM1 捕获/比较中断 （TIM1 Capture /Compare interrupt）	0x0000 0078
15	22	可配置	TIM2	TIM2 全局中断 （TIM2 global interrupt）	0x0000 007C
16	23	-	-	保留	0x0000 0080
17	24	可配置	TIM6	TIM6 全局中断 （TIM6 global interrupt）	0x0000 0084
18	25	-	-	保留	0x0000 0088
19	26	-	-	保留	0x0000 008C
20	27	-	-	保留	0x0000 0090
21	28	可配置	EXTI6	EXTI 线 6 中断 （EXTI Line6 interrupt）	0x0000 0094
22	29	可配置	EXTI7	EXTI 线 7 中断 （EXTI Line7 interrupt）	0x0000 0098

位置	优先级		名称	描述	地址
23	30	可配置	I2C	I2C 全局中断（和 EXTI 线 10 共用） （I2C global interrupt, combined with EXTI Line 10）	0x0000 009C
24	31	-	-	保留	0x0000 00A0
25	32	可配置	SPI	SPI 全局中断 （SPI global interrupt）	0x0000 00A4
26	33	-	-	保留	0x0000 00A8
27	34	可配置	USART1	USART1 全局中断（和 EXTI 线 9 共用） （USART1 global interrupt, combined with EXTI Line9）	0x0000 00AC
28	35	-	-	保留	0x0000 00B0
29	36	-	-	保留	0x0000 00B4
30	37	-	-	保留	0x0000 00B8
31	38	-	-	保留	0x0000 00BC

10.2 扩展中断和事件控制器（EXTI）

扩展中断及事件控制器（EXTI）负责管理内、外异步中断和事件：

- 向 CPU 输出事件请求。
- 向中断控制器输出中断请求。
- 向电源管理模块输出唤醒请求。

外部中断，可选择上升沿/下降沿触发；内部中断，只能由上升沿触发。外部中断可以被挂起，通过查询挂起状态寄存器可获取挂起状态及对应的中断源；而内部中断，挂起状态应由相应的内置外设提供。

EXTI 管理多达 12 根内外部中断/事件线（8 根中断线、1 根外部事件线和 3 根内部事件线）。其中，事件用于内核唤醒。

每条输入线的中断或事件均可独立屏蔽。内部输入线仅在停机模式下采样。

此外，控制器提供了一种通过写软件中断事件寄存器（EXTI_SWIER）来模拟生成中断/事件的方法。

10.2.1 主要特性

EXTI 控制器的主要特性如下：

- 支持多达 12 个事件/中断请求线
  - 8 条可配置线（EXTI0~7），用于连接 IO
  - 4 条无需配置的线
- 每根中断/事件线都可单独触发和屏蔽
- 每根中断线都有专用的状态位
- 能够检测到比 APB 时钟宽度还小的脉冲

10.2.2 框图

EXTI 结构框图如下所示：

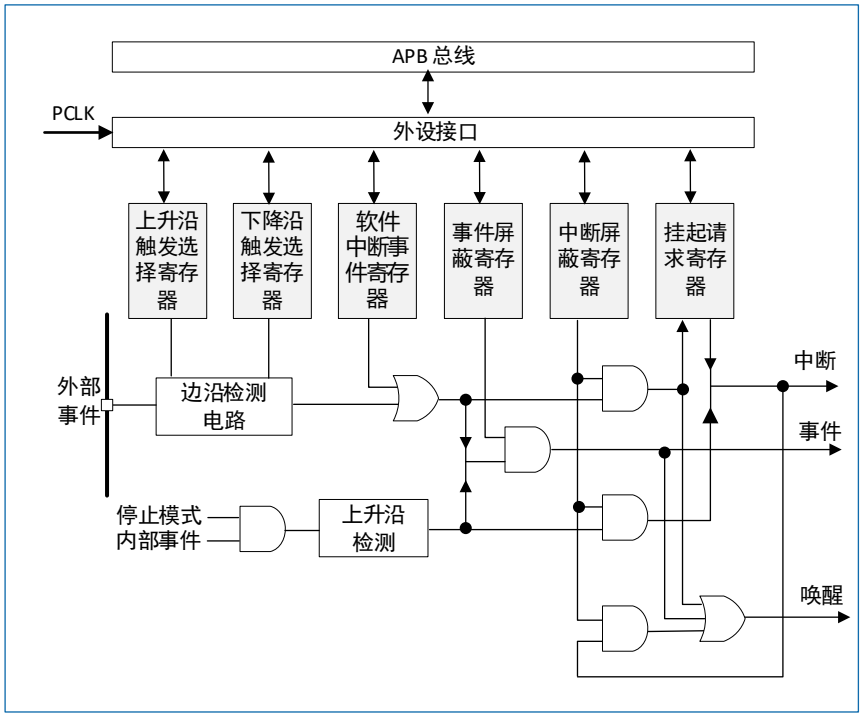


图 10-1 EXTI 框图

10.2.3 EXTI 与周边模块关系

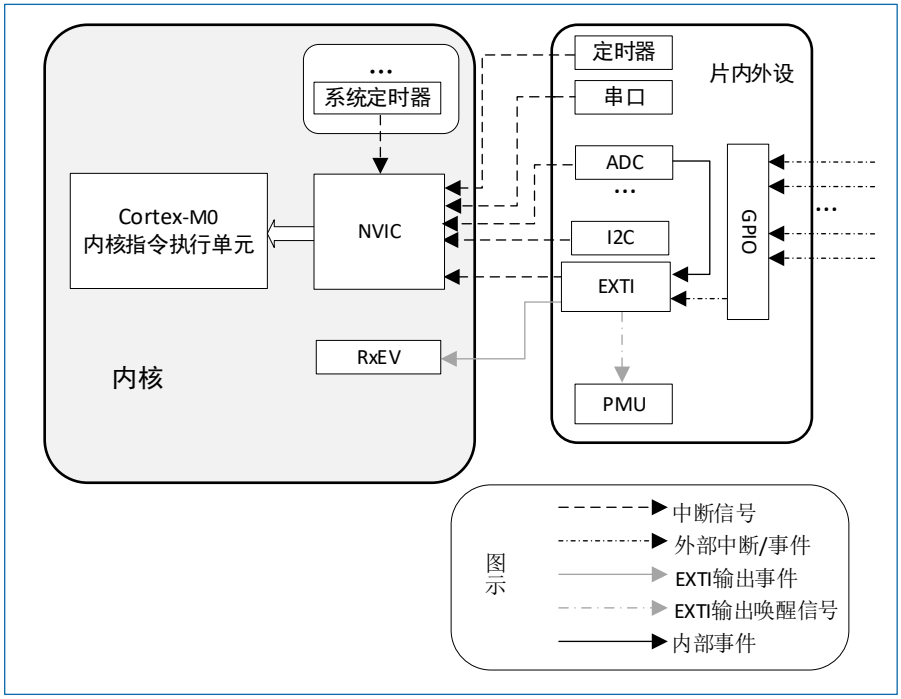


图 10-2 EXTI 与周边模块关系框图

如图 10-2 所示，EXTI 的中断输出与芯片内其他中断源一起汇总于 NVIC；EXTI 输出的事件，输入到 RxEV 模块，用于管理内核唤醒相关操作；EXTI 输出的唤醒信号，输出给 PMU 单元。

10.2.4 唤醒事件管理

HK32F030M 可以处理外部或内部事件来唤醒内核（WFE）。唤醒事件可以通过下述方式产生：

- 在外设的控制寄存器中使能一个中断，但不在 NVIC 中使能，同时在 Cortex®-M0 的系统控制寄存器中使能 SEVONPEND 位。当 CPU 从 WFE 恢复后，需要清除相应外设的中断挂起位和外设 NVIC 中断通道挂起位（在 NVIC 中断清除挂起寄存器中）。

- 配置一个外部或内部 EXTI 线为事件模式，当 CPU 从 WFE 恢复后，因为对应事件线的挂起位没有被置位，不必清除相应外设的中断挂起位或 NVIC 中断通道挂起位。

使用外部 I/O 端口作为唤醒事件，请参考“10.2.6 外部中断/事件线映射”。

10.2.5 功能说明

要产生中断，必须先配置并使能中断线。根据所需的边沿检测条件来配置 2 个触发寄存器，并且通过向中断屏蔽寄存器的相应位写’1’来使能中断请求。当外部中断线上发生了所设置的边沿时，将产生一个中断请求，对应的挂起位也随之被置’1’。在挂起寄存器的对应位写’1’，将清除该中断请求。

如果需要产生事件，必须先配置并使能事件线。根据所需的边沿检测条件来配置 2 个触发寄存器，并且通过向事件屏蔽寄存器的相应位写’1’来使能事件请求。当事件线上发生了需要的边沿时，将产生一个事件请求脉冲，对应的挂起位不被置’1’。

通过软件向软件中断/事件寄存器写’1’，以产生中断/事件请求。

10.2.5.1 硬件中断选择

配置 EXTI 线作为中断源的步骤如下：

1. 配置其中断屏蔽位（EXTI_IMR）。
2. 配置其中断触发选择位（EXTI_RTSR/EXTI_FTSR）。
3. 配置对应到外部中断控制器（EXTI）的 NVIC 中断通道的使能和屏蔽位，以使得 EXTI 线上来的中断能正确地被响应。

10.2.5.2 硬件事件选择

配置 EXTI 线作为事件源的步骤如下：

1. 配置其事件屏蔽位（EXTI_EMR）。
2. 配置其事件触发选择位（EXTI_RTSR/EXTI_FTSR）。

10.2.5.3 软件中断/事件的选择

任意 EXTI 线可以被配置成软件中断/事件线。下面是产生软件中断的步骤：

1. 配置其中断/事件屏蔽位（EXTI_IMR/EXTI_EMR）。
2. 设置软件中断寄存器的请求位（EXTI_SWIER）。

10.2.6 外部中断/事件线映射

16 个 GPIO 口以下图的方式连接到 8 个外部中断/事件线上：

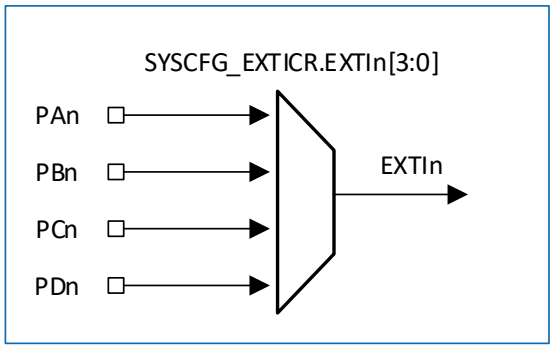


图 10-3 外部中断 GPIO 映射

关于图 10-3 参数说明如下表所示。如需 GPIO 详细信息，请查阅数据手册中管脚定义章节。

表 10-2 参数 n 取值范围

端口	n 取值
PA	1~3
PB	4~5
PC	3~7
PD	1~6

说明：该系列芯片仅 SON8 和 TSSOP16 封装提供 PA0 和 PD7 引脚。

如图 10-3 所示，通过 SYSCFG_EXTICR 配置 GPIO 线上的外部中断/事件，除使能 GPIO 时钟外，也需配置 RCC_APB2ENR.SYSCFGEN 来使能 AFIO 时钟。参见：“6.3.7 APB 外设时钟使能寄存器 2（RCC_APB2ENR）”。

另外 4 根 EXTI 线的连接方式如下：

- EXTI 8 连接 ADC 的 AWD 事件
- EXTI 9 连接 USART 的唤醒事件
- EXTI 10 连接 I2C 的唤醒事件
- EXTI 11 连接 AWU 的唤醒事件

其中 EXTI8~10 作为内部事件，不带 RTSR、FTSR、SWIER 和 PR 寄存器。这 3 个 EXTI 口仅能在停机模式下采集事件的上升沿以产生 ERQ 和 IRQ 唤醒系统。

10.3 EXTI 寄存器

基地址：0x4001 0400

空间大小：0x400

10.3.1 中断屏蔽寄存器（EXTI_IMR）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				IM11	IM10	IM9	IM8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12	Res：保留 必须保持复位值。
位 x （x = 11..0）	IMx：外部/内部线 x 的中断屏蔽位（Interrupt mask on line x） • 0：屏蔽来自线 x 上的中断请求 • 1：允许来自线 x 上的中断请求

10.3.2 事件屏蔽寄存器（EXTI_EMR）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				EM11	EM10	EM9	EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12	Res：保留 必须保持复位值。
位 x （x = 11..0）	EMx：外部/内部线 x 的事件屏蔽位（Event mask on line x） • 0：屏蔽来自线 x 上的事件请求 • 1：允许来自线 x 上的事件请求

10.3.3 上升沿触发选择寄存器（EXTI_RTSR）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				RT11	Res			RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
				rw				rw	rw	rw	rw	rw	rw	rw	rw

位 31:12	Res：保留 必须保持复位值。
位 11	RT11：线 11 的上升沿触发事件配置位（Rising trigger event configuration bit of line 11） • 0：禁止输入线 11 上的上升沿触发（中断和事件） • 1：启用输入线 11 上的上升沿触发（中断和事件）
位 10:8	Res：保留 必须保持复位值。
位 x （x = 7..0）	RTx：线 x 的上升沿触发事件配置位（Rising trigger event configuration bit of line x） • 0：禁止输入线 x 上的上升沿触发（中断和事件） • 1：启用输入线 x 上的上升沿触发（中断和事件）

10.3.4 下降沿触发选择寄存器（EXTI_FTSR）

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Res

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				FT11	Res			FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0
				rw				rw	rw	rw	rw	rw	rw	rw	rw

位 31:12	Res：保留 必须保持复位值。
位 11	FT11：线11 的下降沿触发事件配置位（Falling trigger event configuration bit of line 11） • 0：禁止输入线11 上的下降沿触发（中断和事件） • 1：启用输入线11 上的下降沿触发（中断和事件）
位 10:8	Res：保留 必须保持复位值。
位 x （x = 7..0）	FTx：线x 的下降沿触发事件配置位（Falling trigger event configuration bit of line x） • 0：禁止输入线x 上的下降沿触发（中断和事件） • 1：启用输入线x 上的下降沿触发（中断和事件）

10.3.5 软件中断事件寄存器（EXTI_SWIER）

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				SWI11	Res			SWI7	SWI6	SWI5	SWI4	SWI3	SWI2	SWI1	SWI0
				rw				rw	rw	rw	rw	rw	rw	rw	rw

位 31:12	Res：保留 必须保持复位值。
位 11	SWI11：线 11 的软件中断（Software interrupt on line 11） 当该位为 0 时，将其写 1 会设置 EXTI_PR 中相应的挂起位。如果在 EXTI_IMR 和 EXTI_EMR 中允许产生该中断，则此时将产生一个中断。 通过清除 EXTI_PR 的对应位（写入 1），可以将该位清零。
位 10:8	Res：保留 必须保持复位值。
位 x （x = 7..0）	SWIx：线 x 的软件中断（Software interrupt on line x） 当该位为 0 时，将其写 1 会设置 EXTI_PR 中相应的挂起位。如果在 EXTI_IMR 和 EXTI_EMR 中允许产生该中断，则此时将产生一个中断。 通过清除 EXTI_PR 的对应位（写入 1），可以将该位清零。

10.3.6 请求挂起寄存器（EXTI_PR）

偏移地址：0x14

复位值：0xXXXX XXXX

说明：X 表示不定值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				PIF11	Res			PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0
				rw				rw	rw	rw	rw	rw	rw	rw	rw

位 31:12	Res：保留 必须保持复位值。
位 11	PIF11：线 11 挂起位（Pending interrupt flag on line 11） • 0：没有发生触发请求 • 1：发生了选择事件的触发请求。当外部中断线上发生了所选择的边沿事件，该位被置 1。 向该位写入 1 或改变边沿检测的极性进行清除。
位 10:8	Res：保留 必须保持复位值。
位 x （x = 7..0）	PIFx：线 x 挂起位（Pending interrupt flag on line x） • 0：没有发生触发请求 • 1：发生了选择事件的触发请求。当外部中断线上发生了所选择的边沿事件，该位被置 1。 向该位写入 1 或改变边沿检测的极性进行清除。

11 ADC 寄存器模数转换器（ADC）

该系列芯片内置的 12 位 ADC 是逐次趋近型模数转换器。它具有多达 6 个复用通道，可测量来自 5 个外部和 1 个内部的信号。其中 AIN0 ~ AIN4 为外部通道接 IO，AIN5 为内部通道，连接内部参考电压，支持差分输入模式，AIN0 和 AIN1，AIN2 和 AIN3 组成两组差分输入（当 ADC 配置为差分输入模式时，AIN4 和采样内部参考电压（BGR）的 ADC 通道不可用）。各个通道的 A/D 转换支持单次、连续、扫描或不连续采样模式。ADC 的结果可以左对齐或右对齐的存储在一个 16 位数据寄存器中。

ADC 具有模拟看门狗特性，允许应用检测输入电压是否超过了用户自定义的上限或下限阈值。内置硬件过采样器，可提高模拟性能，同时还能减轻 CPU 进行相关计算的负担。

11.1 ADC 主要特性

- 高性能
  - 可配置 12 位、10 位、8 位或 6 位分辨率。
  - ADC 转换时间：12 位分辨率对应的转换时间为 1 μs（1 MHz）；10 位分辨率对应的转换时间为 0.93 μs；若降低分辨率，可进一步缩短转换时间。
  - 自校准
  - 可编程采样时间
  - 内置数据一致性，可确保数据对齐。
- 低功耗
  - 应用可降低 PCLK 频率从而以低功耗运行，同时仍可保持最优的 ADC 性能。例如，无论 PCLK 的频率如何，都保持 1.0 μs 的转换时间。
  - 等待模式：防止 ADC 在低频 PCLK 应用中溢出。
  - 自动关闭模式：ADC 会自动断电，但正在进行转换时除外。这可大幅降低 ADC 的功耗。
- 模拟输入通道
  - 5 路外部模拟输入
  - 1 条用于内部参考电压（VREFINT）的通道
- 可通过以下方式启动 A/D 转换：
  - 通过软件
  - 通过极性可配置的硬件触发器（来自 TIM1、TIM2、TIM6 的内部定时器事件 GPIO 输入事件）
- 转换模式

  可转换单条通道，也可扫描一系列通道。
  - 单次模式：会在每次触发时对选定的输入通道执行一次转换
  - 连续模式：可对选定的输入通道进行连续转换
  - 不连续转换模式
- 在采样结束、转换结束、序列转换结束、发生模拟看门狗事件或溢出事件时产生中断。
- 带模拟看门狗
- ADC 电源要求：1.8V～3.6 V
- ADC 输入范围：$V_{SSA} \le V_{IN} \le V_{DDA}$

11.2 ADC 功能描述

ADC 功能框图如下:

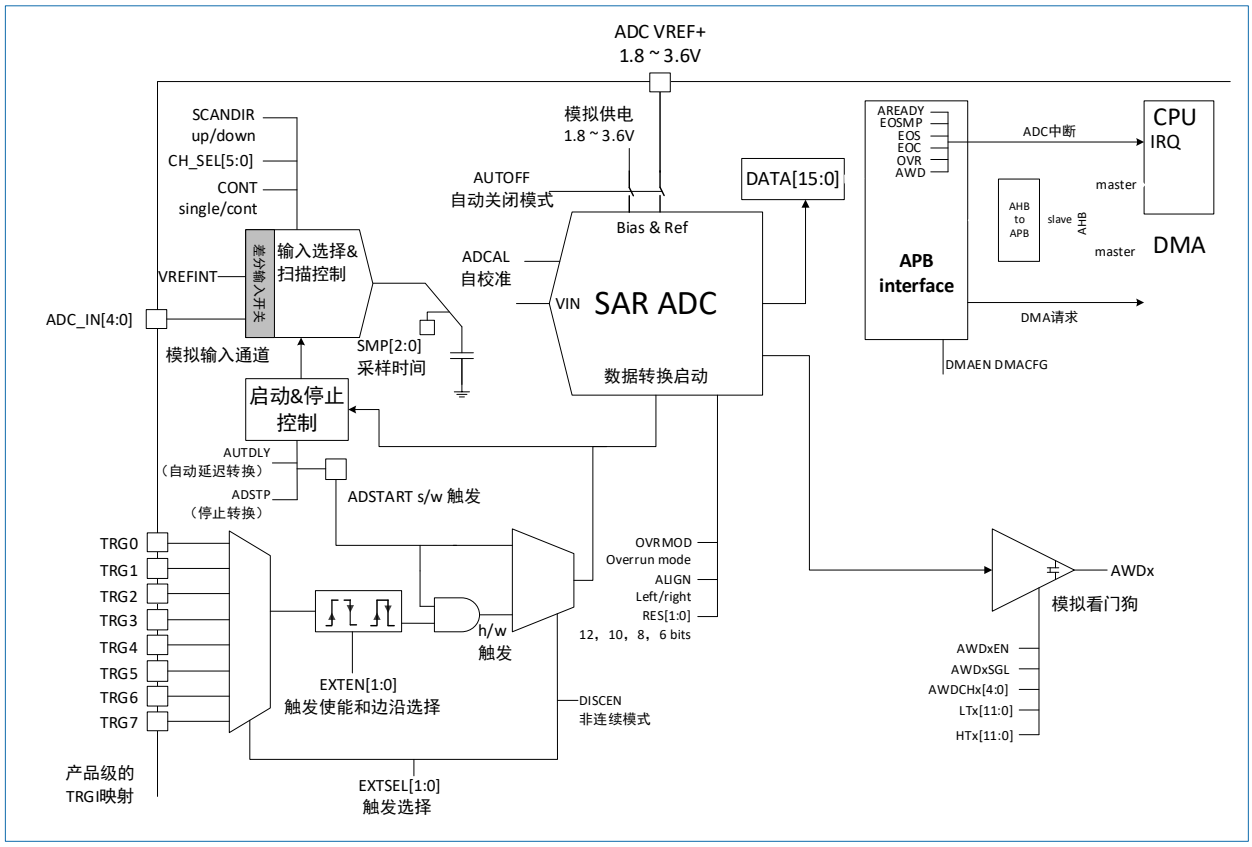


图 11-1 ADC 功能框图

11.2.1 ADC 引脚和内部信号

表 11-1 ADC 引脚

名称	信号类型	备注
V_{DDA}	模拟电源输入	ADC 的模拟电源和正参考电压（$V_{DDA} \geq V_{DD}$）
V_{SSA}	模拟电源地输入	模拟电源接地引脚。电压必须等于 V_{SS}。
ADC_IN[5:0]	模拟输入信号	6 个模拟输入通道

表 11-2 ADC 内部信号

内部信号名称	信号类型	说明
TRGx	输入	ADC 转换触发信号
VREFINT	输入	内部参考输出电压

11.2.2 校准（ADCAL）

ADC 有一个内置自校准模式。校准可显著减小因内部电容器组的变化而造成的精度误差由于制造工艺的不同，各芯片的偏移误差也有所不同。

校准过程中，ADC 会计算校准系数。ADC 下一次掉电之前，会在内部应用此校准系数。校准过程中，不得使用 ADC，必须等待校准完成才行。

校准应在启动 A/D 转换之前进行。仅当 ADC 禁用（ADEN=0）后，才能发起校准。通过软件将 ADCAL 位置 1，以启动校准。ADCAL 位在所有校准序列过程中保持为 1。校准完成后，此位会立即由硬件清零。随后，可从 ADC_DR 寄存器（位 6 到位 0）中读取校准系数。

若禁止了 ADC（ADEN=0），则会保留内部模拟校准。如果 ADC 工作条件发生变化（V_{DDA} 变化是造成 ADC 偏移变化的主要原因，并会导致温度发生小幅变化），建议重新运行校准。

在 ADC 外设复位的情况下，校准系数会丢失。

ADC 在低功耗运行、低功耗睡眠和停机这三种模式下，会保留校准系数。仍可通过软件保存并恢复校准系数，以节省 ADC 重启时间（前提是 ADC 掉电期间的温度和电压保持稳定）。

如果 ADC 已使能但未进行转换（ADEN=1 且 ADSTART=0），可写入校准系数。随后，下次转换启动时，校准系数会自动添加到模拟 ADC 中。这一载入过程是透明的，不会对转换的启动造成延迟。

校准软件程序

1. 确保 ADEN=0。
2. 将 ADCAL 置 1。
3. 等待直到 ADCAL=0。
4. 校准系数可从 ADC_DR 寄存器的位[6:0]读取。

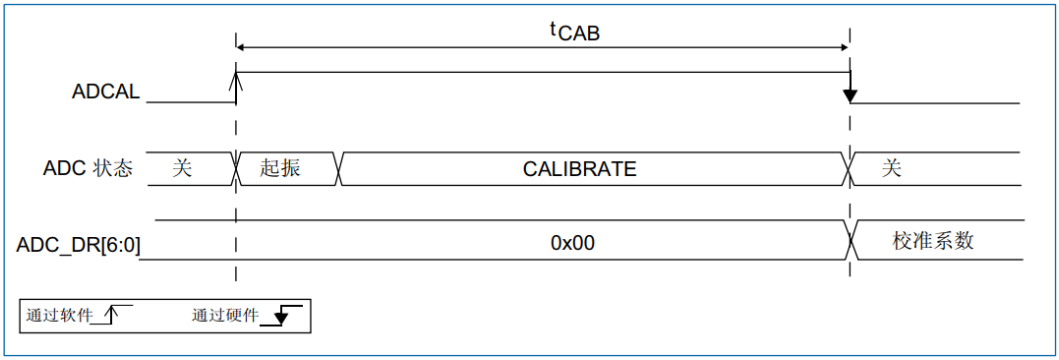


图 11-2 ADC 校准

11.2.3 ADC 开关控制（ADEN, ADDIS, ADRDY）

MCU 上电时，ADC 被禁用并进入掉电模式（ADEN=0）。

如图 11-3 所示，ADC 在开始精确转换之前需要一段稳定时间 t_{STAB}。

以下两个控制位可用于使能或禁止 ADC：

- 将 ADEN 置 1 可使能 ADC。ADC 准备就绪后，ADRDY 标志会立即置 1。
- 将 ADDIS 置 1 可禁用 ADC 并使 ADC 进入掉电模式。随后，ADC 被完全禁止后，ADEN 位和 ADDIS 位会自动由硬件清零。

可通过将 ADSTART 置 1（请参见“11.3 外部触发转换和触发极性（EXTSEL, EXTEN）”）开始进行转换；如果触发器已使能，也可在发生外部触发事件时开始进行转换。

使能 ADC 的流程如下：

1. 对 ADC_ISR 寄存器中的 ADRDY 位写 1。
2. 将 ADC_CR 寄存器中的 ADEN 位置 1。
3. 等待直至 ADC_ISR 寄存器中的 ADRDY=1（ADRDY 会在 ADC 启动时间后置 1）。

如果已通过将 ADC_IER 寄存器中的 ADRDYIE 位置 1 来使能中断，可通过中断进行处理。

禁用 ADC 的流程如下：

1. 检查 ADC_CR 寄存器中的 ADSTART 为 0，以确保当前未执行任何转换。如有需要，可向 ADC_CR 寄存器中的 ADSTP 位写入 1 并等待此位读取值为 0，以此停止正在进行的转换。
2. 将 ADC_CR 寄存器中的 ADDIS 位置 1。
3. 如果应用要求，可等待 ADC_CR 寄存器中的 ADEN=0，这表明 ADC 已完全禁止（ADEN=0 后，ADDIS 会自动复位）。
4. 将 ADC_ISR 寄存器中的 ADRDY 位编程为 1，将此位清零（可选步骤）。

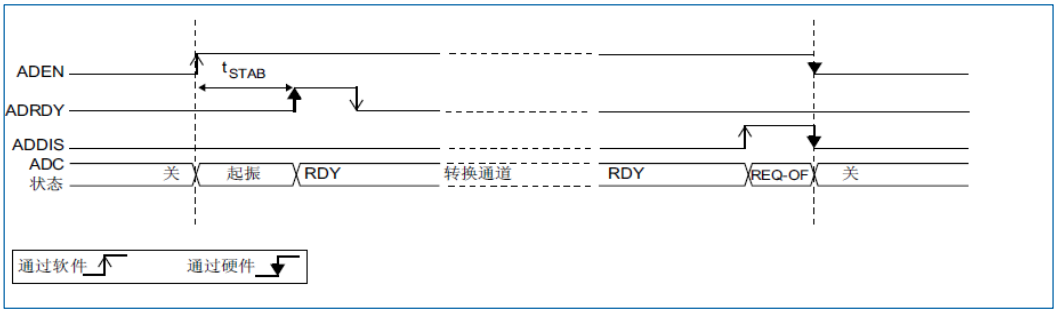


图 11-3 使能/禁用 ADC

说明：在自动关闭模式下（AUTOFF=1），上电/掉电阶段是由硬件自动执行的，不会将ADRDY 置 1。

11.2.4 ADC 时钟（CKMODE）

ADC 采用双时钟域架构，因此，ADC 可由独立于 APB 时钟（PCLK）的时钟提供时钟（ADC 异步时钟）。

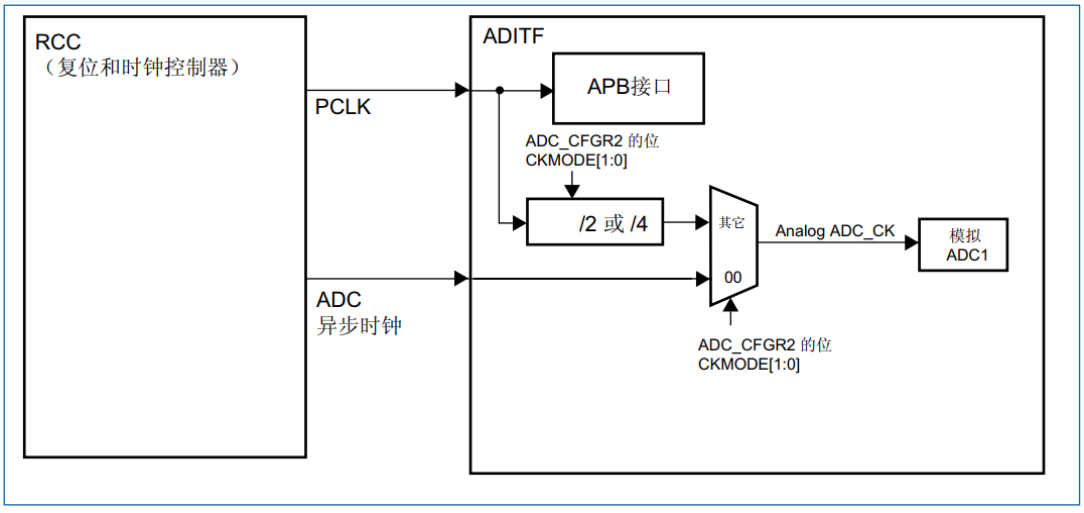


图 11-4 ADC 时钟图

说明：了解 PCLK 和 ADC 异步时钟的使能方式参见“6 复位和时钟控制（RCC）”。

如图 11-4 所示，ADC 的输入时钟可在两个不同的时钟源之间进行选择：

- ADC 时钟可选择名为“ADC 异步时钟”的特定时钟源，该时钟源独立于 APB 时钟，并与 APB 时钟异步。

  更多有关生成该时钟源的信息，请参见“6 复位和时钟控制（RCC）”。

  要选择该时钟，必须将 ADC_CFGR2 寄存器的 CKMODE[1:0]位复位。

  选择该时钟的优势在于：无论选择哪种 APB 时钟，都可以达到最大 ADC 时钟频率。

- ADC 时钟可由 ADC 总线接口的 APB 时钟除以一个可编程因子（2 或 4，由 CKMODE[1:0]位而定）来提供。

  要选择该时钟，ADC_CFGR2 寄存器的 CKMODE[1:0]位不得为“00”。

  选择该时钟的优势在于：不用重新同步时钟域。如果 ADC 由定时器触发，并且应用要求 ADC 精确触发（不存在任何不确定性），可使用此选项。否则，重新同步两个时钟域会为触发时刻带来不确定性。

表 11-3 触发与转换开始之间的延迟

ADC 时钟源	CKMODE[1:0]	触发事件与转换开始之间的延迟
HSI 时钟	00	延迟是不确定的（存在抖动）
PCLK 2 分频	01	延迟是确定的（无抖动），等于 4.25 个 ADC 时钟周期
PCLK 4 分频	10	延迟是确定的（无抖动），等于 4.125 个 ADC 时钟周期

11.2.5 配置 ADC

如果 ADC 已禁用（ADEN=0），必须通过软件写 ADC_CR 寄存器中的 ADCAL 位和 ADEN 位来配置 ADC。

仅当 ADC 已使能且没有待处理的禁用 ADC 的请求时（ADEN=1 且 ADDIS=0），软件才能写 ADC_CR 寄存器中的 ADSTART 位和 ADDIS 位。

对于 ADC_IER、ADC_CFGRx、ADC_SMPR、ADC_TR、ADC_CHSELR 和 ADC_CCR 寄存器中的其他控制位，只有在 ADC 已使能（ADEN=1）且没有转换正在进行（ADSTART=0），软件才可以执行写操作。

如果 ADC 已使能（可能正在进行转换）、并且没有待处理的禁用 ADC 的请求时（ADSTART=1 且 ADDIS=0），软件必须只写 ADC_CR 寄存器中的 ADSTP 位才能停止 ADC。

说明：未采取硬件保护机制来防止软件执行上述规则禁止的写操作。如果发生此类禁止的写访问，ADC 可能会进入未定义状态。要在这种情况下恢复正确的操作，必须禁止 ADC（将 ADEN 以及 ADC_CR 寄存器中的所有位都清零）。

11.2.6 通道选择

复用通道多达 6 条：

- 5 路来自 GPIO 引脚（ADC_IN0~ADC_IN4）的模拟输入。
- 1 路内部模拟输入（内部参考电压）。

可转换单条通道，也可以自动扫描一系列通道。

待转换通道的顺序必须在 ADC_CHSELR 通道选择寄存器中进行编程，每条模拟输入通道都有专用的选择位（CHSEL0~CHSEL5）。

要配置通道的扫描方式，可对 ADC_CFGR1 寄存器中的 SCANDIR 位进行编程：

- SCANDIR=0：正向扫描通道 0 到通道 6
- SCANDIR=1：反向扫描通道 6 到通道 0

V_{REFINT} 内部通道

内部参考电压 V_{REFINT} 连接至通道 ADC_IN5。

11.2.7 可编程采样时间（SMP）

开始转换之前，ADC 需要在待测量电压源与 ADC 内置采样电容之间建立直接连接。该采样时间必须足以使输入电压源为采样电容充电并将电容保持在输入电压水平。

使用可编程采样时间后，可根据输入电压源的输入电阻调整转换速度。

ADC 会在数个 ADC 时钟周期内对输入电压进行采样，时钟周期数可使用 ADC_SMPR 寄存器中的 SMP[2:0]位进行配置。

此可编程采样时间是所有通道共用的。如果应用要求，可通过软件在两次转换之间更改和调整此采样时间。

总转换时间的计算公式如下：

$$t_{CONV} =（采样时间 + 12.5）x\ ADC\ 时钟周期$$

示例：如果 ADC_CLK=16 MHz，采样时间为 1.5 个 ADC 时钟周期，则总转换时间为：

t_{CONV}=（1.5 + 12.5）*x ADC 时钟周期* = 14 个 ADC 时钟周期 = 0.875 µs

ADC 通过将 EOSMP 标志置 1 来指示采样阶段结束。

11.2.8 单次转换模式（CONT=0）

在单次转换模式下，ADC 会执行单次转换序列，对所有通道进行一次转换。当 ADC_CFGR1 寄存器中的 CONT=0 时，会选择此模式。可通过以下方式开始转换：

- 将 ADC_CR 寄存器中的 ADSTART 位置 1。
- 硬件触发事件

在序列中，每次转换完成后：

- 转换后的数据会存储在 16 位 ADC_DR 寄存器中
- EOC（转换结束）标志置 1
- EOCIE 位置 1 时将产生中断

转换序列完成后：

- EOS（序列结束）标志置 1
- EOSIE 位置 1 时将产生中断

随后，ADC 会停止工作，直至发生新的外部触发事件或 ADSTART 位再次置 1。

说明：要转换单个通道，可将序列长度编程为 1。

11.2.9 连续转换模式（CONT=1）

在连续转换模式下，如果发生软件或硬件触发事件，ADC 会执行转换序列，对所有通道进行一次转换，随后会自动重启并持续执行相同的转换序列。当 ADC_CFGR1 寄存器中的 CONT=1 时，会选择此模式。可通过以下方式开始转换：

- 将 ADC_CR 寄存器中的 ADSTART 位置 1
- 硬件触发事件

在序列中，每次转换完成后：

- 转换后的数据会存储在 16 位 ADC_DR 寄存器中
- EOC（转换结束）标志置 1
- EOCIE 位置 1 时将产生中断

转换序列完成后：

- EOS（序列结束）标志置 1
- EOSIE 位置 1 时将产生中断

随后，会立即重启新序列，ADC 会继续重复执行转换序列。

说明：要转换单个通道，可将序列长度编程为 1。

不能同时使能不连续模式和连续模式：禁止同时将 DISCEN 和 CONT 位置 1。

11.2.10 开始转换（ADSTART）

软件通过将 ADSTART 置 1 的方式开始进行 ADC 转换。

ADSTART 置 1 后：

- 在 EXTEN=00 时，会立即开始转换（软件触发）。
- 在 EXTEN≠ 00 时，会在所选硬件触发器的下一有效边沿开始转换。

ADSTART 位还用于指示当前是否正在进行 ADC 操作。可以在 ADSTART=0 时重新配置 ADC，表明 ADC 处于空闲状态。

ADSTART 位由硬件清零：

- 在使用软件触发的单次模式下（CONT=0 且 EXTEN=00）
  - 只要转换序列结束（EOS=1）就清零
- 在使用软件触发的不连续模式下（CONT=0、DISCEN=1 且 EXTEN=00）
  - 转换结束时（EOC=1）清零
- 在所有情况下（CONT=X、EXTEN=XX）
  - 执行由软件调用的 ADSTP 程序之后（参见“11.2.10 开始转换（ADSTART）”）清零。

说明：在连续模式下（CONT=1），由于序列会自动重新启动，因此，当 EOS 标志置 1 时，ADSTART 位不会清零。

如果在单次模式下选择了硬件触发（CONT=0，且 EXTEN≠ 00），当 EOS 置 1 时，ADSTART 不会由硬件清零。这样，便无需通过软件将 ADSTART 再次置 1，并可确保不会错过下一触发事件。

11.2.11 时序

从转换开始到转换结束所经过的时间是配置的采样时间与逐次趋近时间（具体视数据分辨率而定）的总和：

$t_{ADC}=t_{SMPL}+t_{SAR}=[1.5_{|min} + 12.5_{|12bit}] \times t_{ADC\_CLK}$

$t_{ADC} = t_{SMPL} + t_{SAR} = 93.8ns_{|min} + 781.3ns_{|12bit} = 0.875\mu s_{|min}$（对于 f_{ADC_CLK} = 16 MHz）

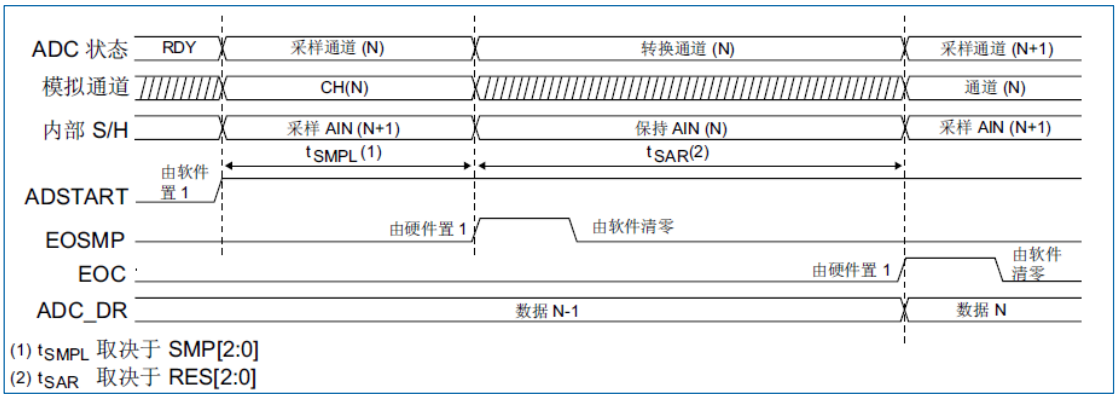


图 11-5 模数转换时间

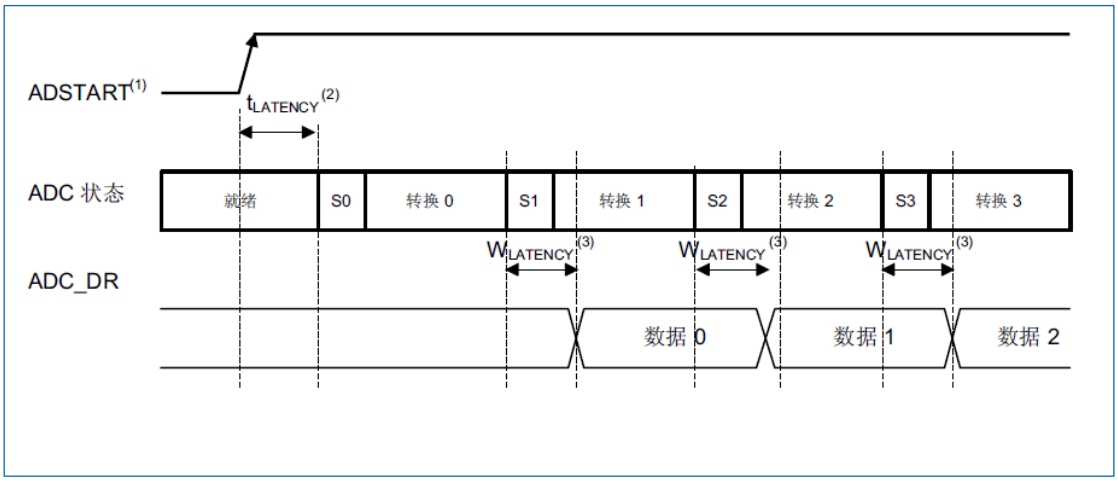


图 11-6 模数转换时序

(1). EXTEN = 00 或 EXTEN ≠ 00

(2). 触发延迟（更多详细信息，请参见数据手册）

(3). ADC_DR 寄存器写入延迟（更多详细信息，请参见数据手册）

11.2.12 停止正在进行的转换（ADSTP）

可通过软件将 ADC_CR 寄存器中的 ADSTP 置 1，停止任何正在进行的转换。这会复位 ADC 操作，ADC 将处于空闲状态，准备好进行新操作。

如果 ADSTP 位由软件置 1，则会中止任何正在进行的转换，并会丢弃转换结果（ADC_DR 寄存器不会更新为当前转换结果）。

扫描序列也会中止并复位（这意味着重启 ADC 将重新开始新的序列）。一旦转换操作完成后，ADSTP 位和 ADSTART 位均由硬件清零，软件必须等待 ADSTART=0，然后才能开始进行新的转换。

图 11-7 停止正在进行的转换

11.3 外部触发转换和触发极性（EXTSEL, EXTEN）

可通过软件或外部事件（定时器捕获事件）触发转换或转换序列。如果 EXTEN[1:0]控制位不等于“0b00”，那么外部事件能够触发所选极性的转换。软件将 ADSTART 位置 1 后，触发选择将立即生效。

在转换进行时发生的硬件触发会被忽略。

如果位 ADSTART=0，则会忽略发生的任何硬件触发。

表 11-4 提供 EXTEN[1:0]值与触发极性之间的对应关系。

表 11-4 配置触发极性

源	EXTEN[1:0]
禁止触发检测	00

源	EXTEN[1:0]
在上升沿检测	01
在下降沿检测	10
在上升沿和下降沿均检测	11

说明：仅当 ADC 未进行转换（ADSTART= 0）时，才可以更改外部触发的极性。

EXTSEL[2:0]控制位用于选择 8 个可能的事件中哪一事件可触发转换。

表 11-5 给出了可用于常规转换的外部触发。

可将 ADC_CR 寄存器中的 ADSTART 位置 1，从而生成软件源触发事件。

表 11-5 ADC 外部触发源

名称	源	EXTSEL[2:0]
TRG0	TIM1_TRGO	000
TRG1	TIM1_CC4	001
TRG2	TIM2_TRGO	010
TRG3	TIM6_TRGO	011
TRG4	TIM1_CC1	100
TRG5	TIM1_CC2	101
TRG6	TIM1_CC3	110
TRG7	IO_TRIG	111

说明：仅当 ADC 未进行转换（ADSTART= 0）时，才可以更改触发选择。

11.3.1 不连续模式（DISCEN）

可将 ADC_CFGR1 寄存器中的 DISCEN 位置 1，以使能此模式。

在该模式下（DISCEN=1），需要通过硬件或软件触发事件启动序列中定义的各个转换。相反，如果 DISCEN=0，单个硬件或软件触发事件会连续启动序列中定义的所有转换。

示例：

- DISCEN=1，待转换通道=0、1、3、5
  - 第一次触发：转换通道 0 并生成 EOC 事件
  - 第二次触发：转换通道 1 并生成 EOC 事件
  - 第三次触发：转换通道 3 并生成 EOC 事件
  - 第四次触发：转换通道 5 并同时生成 EOC 和 EOS 事件
  - 第五次触发：转换通道 0 并生成 EOC 事件
  - 第六次触发：转换通道 1 并生成 EOC 事件
- DISCEN=0，待转换通道=0、1、3、5
  - 第一次触发：转换整个序列：通道 0，然后是通道 1、3 和 5。每次转换都会生成 EOC 事件，最后一次转换还会生成 EOS 事件。

- 任何后续触发事件都将重启整个序列。

说明：不能同时使能不连续模式和连续模式：禁止同时将 DISCEN 和 CONT 位置 1。

11.3.2 可编程分辨率（RES）快速转换模式

可通过降低 ADC 分辨率缩短转换时间（t_{SAR}）。

通过对 ADC_CFGR1 寄存器中的 RES[1:0]位进行编程，可将分辨率配置为 12 位、10 位、8 位或 6 位。对于不要求使用高数据精度的应用，分辨率越低，转换时间越短。

说明：RES[1:0]位必须在 ADEN 位复位后才能进行更改。

转换结果的宽度始终为 12 位，任何未使用的 LSB 位都会读为零。

降低分辨率可缩短逐次趋近步骤所需的转换时间，如表 11-6 所示。

表 11-6 t_{SAR} 与转换分辨率有关的转换时间

RES[1:0]位	t_{SAR}（ADC 时钟周期）	t_{SAR}（ns），fADC=16MHz	t_{SMPL}（min）（ADC 时钟周期）	t_{CONV}（ADC 时钟周期）（使用最短 t_{SMPL}）	t_{CONV} f_{ADC}=16 MHz
12	12.5	781 ns	1.5	14	875 ns
10	11.5	719 ns	1.5	13	812 ns
8	9.5	594 ns	1.5	11	688 ns
6	7.5	469 ns	1.5	9	562 ns

11.3.3 转换结束、采样阶段结束（EOC, EOSMP 标志）

ADC 指示每个转换结束（EOC）事件。

新的转换数据结果出现在 ADC_DR 寄存器中后，ADC 会立即将 ADC_ISR 寄存器中的 EOC 标志置 1。如果 ADC_IER 寄存器中的 EOCIE 位置 1，可产生中断。EOC 标志可通过软件向其写入 1 或读取 ADC_DR 寄存器的方式来清零。

ADC 还通过将 ADC_ISR 寄存器中的 EOSMP 标志置 1 来指示采样阶段结束。EOSMP 标志可通过软件向其写入 1 来清零。如果 ADC_IER 寄存器中的 EOSMPIE 位置 1，可产生中断。此中断用于处理与转换进行同步。通常情况下，可在转换阶段的隐藏时间内访问模拟复用器，这样在下次采样开始时复用器已放置好。

说明：由于采样结束与转换结束之间只有非常短的时间，因此建议使用轮询或 WFE 指令，而不建议使用中断和 WFI 指令。

11.3.4 转换序列结束（EOS 标志）

每次序列（EOS）事件结束时，ADC 都会通知应用。

转换序列的上一数据结果出现在 ADC_DR 寄存器中时，ADC 会立即将 ADC_ISR 寄存器中的 EOS 标志置 1。如果 ADC_IER 寄存器中的 EOSIE 位置 1，可产生中断。EOS 标志可通过软件向其写入 1 的方式来清零。

11.3.5 时序图示例（单次/连续模式硬件/软件触发）

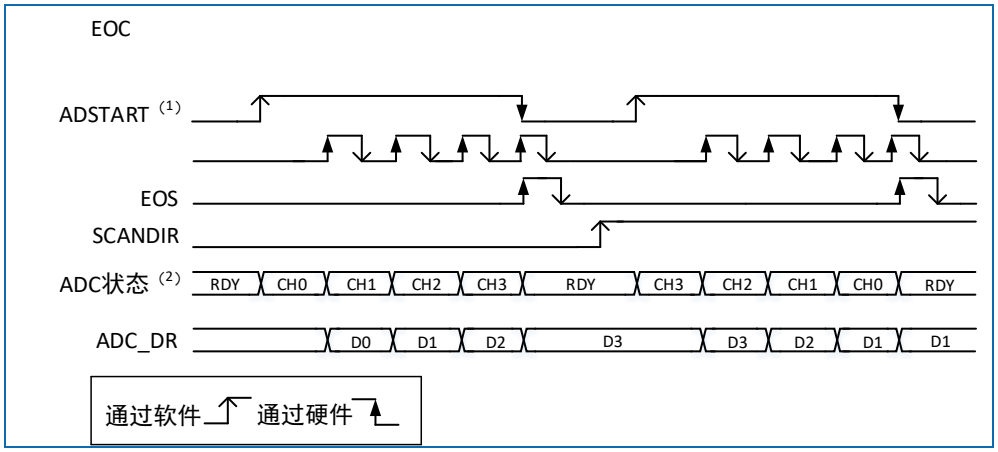


图 11-8 单次序列转换，软件触发

上图说明：EXTEN=00，CONT=0，CHSEL=0x1F，WAIT=0，AUTOFF=0。

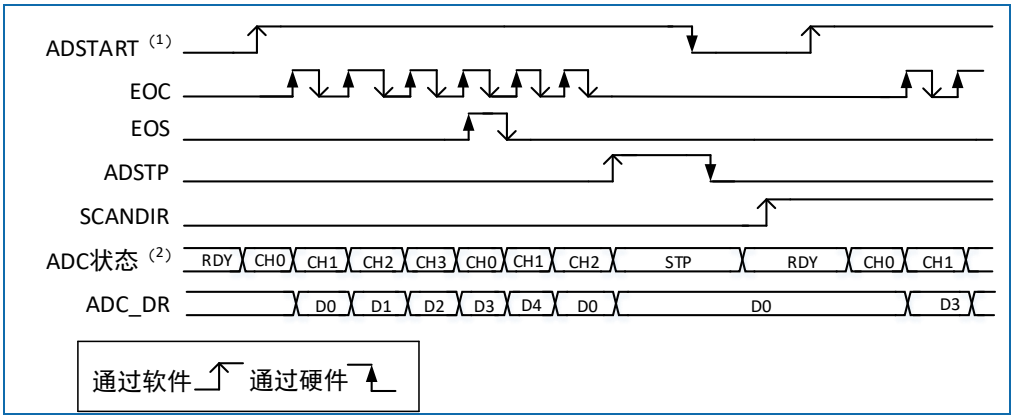


图 11-9 连续序列转换，软件触发

上图说明：EXTEN=00，CONT=1，CHSEL=0x1F，WAIT=0，AUTOFF=0。

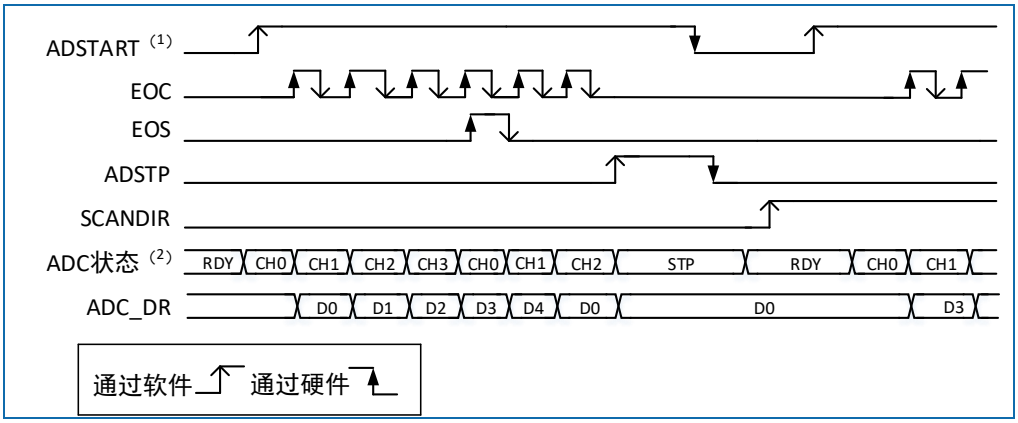


图 11-10 单次序列转换，硬件触发

上图说明：EXTSEL=TRGx（过频），EXTEN=01（上升沿），CONT=0，CHSEL=0x1F，SCANDIR=0，WAIT=0，AUTOFF=0。

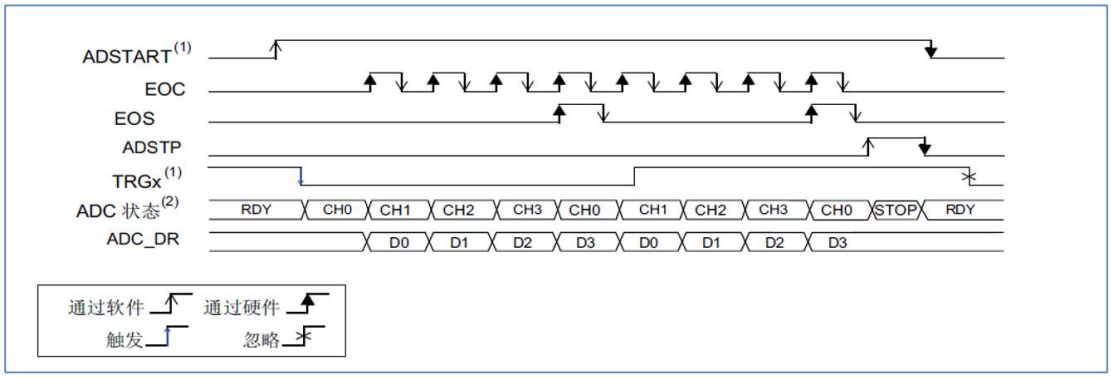


图 11-11 连续序列转换，硬件触发

上图说明：

（1）EXTSEL=TRGx，EXTEN=10（下降沿），CONT=1

（2）CHSEL=0xF，SCANDIR=0，WAIT=0，AUTOFF=0

11.4 数据管理

11.4.1 数据管理和数据对齐（ADC_DR, ALIGN）

每次转换结束时（发生 EOC 事件时），转换后数据的结果都会存储在宽度为 16 位的 ADC_DR 数据寄存器中。

ADC_DR 的格式取决于配置的数据对齐方式和分辨率。

ADC_CFGR1 寄存器中的 ALIGN 位用于选择转换后存储的数据的对齐方式。数据可右对齐（ALIGN=0）或左对齐（ALIGN=1），如图 11-12 所示。

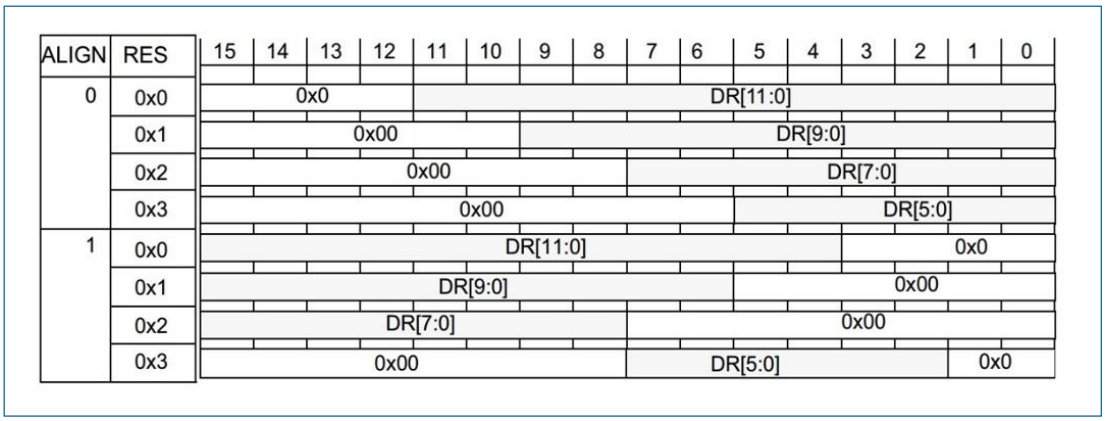


图 11-12 数据对齐方式和分辨率

11.4.2 ADC 溢出（OVR, OVRMOD）

如果转换后的数据未由 CPU 及时读取，在新转换生成数据之前，会由溢出标志（OVR）指示数据溢出事件。

如果新转换完成时 EOC 标志仍为“1”，则 ADC_ISR 寄存器中的 OVR 标志会置 1。

如果 ADC_IER 寄存器中的 OVRIE 位置 1，可产生中断。

如果发生溢出情况，ADC 会保持工作状态并可继续进行转换，除非通过软件将 ADC_CR 寄存器中的 ADSTP 位置 1，从而停止并复位序列。

OVR 标志可通过软件向其写入 1 的方式来清零。

可对 ADC_CFGR1 寄存器中的 OVRMOD 位进行编程，从而配置发生溢出事件时是保留数据还是覆盖

数据：

- OVRMOD=0

  溢出事件会保留数据寄存器的数据，防止其被覆盖：会保留原数据，并会丢弃新的转换结果。如果 OVR 保持为 1，可继续进行转换，但会丢弃所得的数据。

- OVRMOD=1

  数据寄存器会将上一次转换结果覆盖，之前未读取的数据会丢失。如果 OVR 保持为 1，可继续进行转换，ADC_DR 寄存器始终包含最新转换得出的数据。

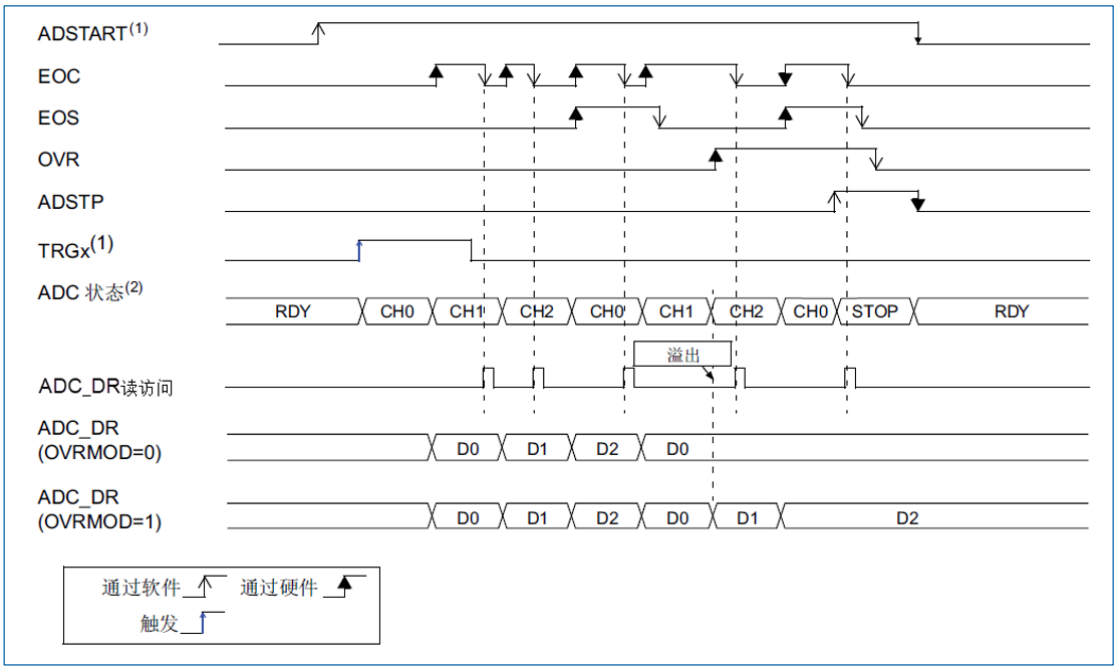


图 11-13 溢出示例（0VR）

11.5 功耗特性

11.5.1 等待模式转换

等待模式转换可用于简化软件，并可优化采用低频时钟的应用（此类应用可能存在 ADC 溢出的风险）的性能。

如果 ADC_CFGR1 寄存器中的 WAIT 位置 1，则仅当之前的数据已完成处理、ADC_DR 寄存器已被读取或者 EOC 位已清零，才会开始新的转换。

通过这种方式，可自动调整 ADC 的速度，使其适应系统读取数据的速度。

说明：转换进行时发生的或在读访问之前的等待时间内发生的硬件触发会被忽略。

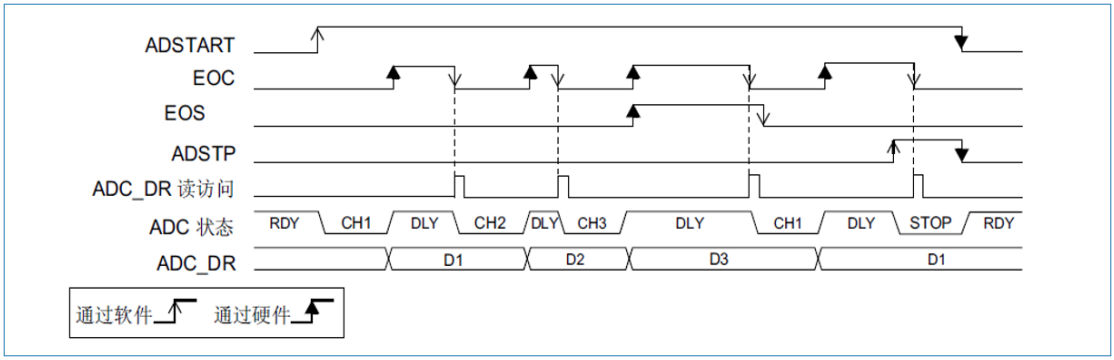


图 11-14 等待模式转换（连续模式，软件触发）

(1) EXTEN=00，CONT=1

(2) CHSEL=0x3，SCANDIR=0，WAIT=1，AUTOFF=0

11.5.2 自动关闭模式（AUTOFF）

ADC 具有自动电源管理功能，也称为自动关闭模式，将 ADC_CFGR1 寄存器中的 AUTOFF 位置 1 可使能此模式。

如果 AUTOFF=1，ADC 始终会在未进行转换时关闭，并会在转换开始后自动唤醒（通过软件或硬件触发）。在启动转换的触发事件和 ADC 的采样时间之间，会自动插入启动时间。随后，转换序列完成后，ADC 会自动禁止。

如果应用需要进行的转换次数相对较少，或者为了合理调整开关 ADC 使用的额外功率和时间而将转换请求的间隔时间设定得足够长（例如采用低频硬件触发），使用自动关闭模式可显著降低应用的功耗。

对于采用低频时钟的应用，可将自动关闭模式与等待模式转换（WAIT=1）结合使用，如果 ADC 在等待过程中自动掉电，将会在应用读取 ADC_DR 寄存器后立即重启。这种组合可显著降低功耗（请参见图 11-16）。

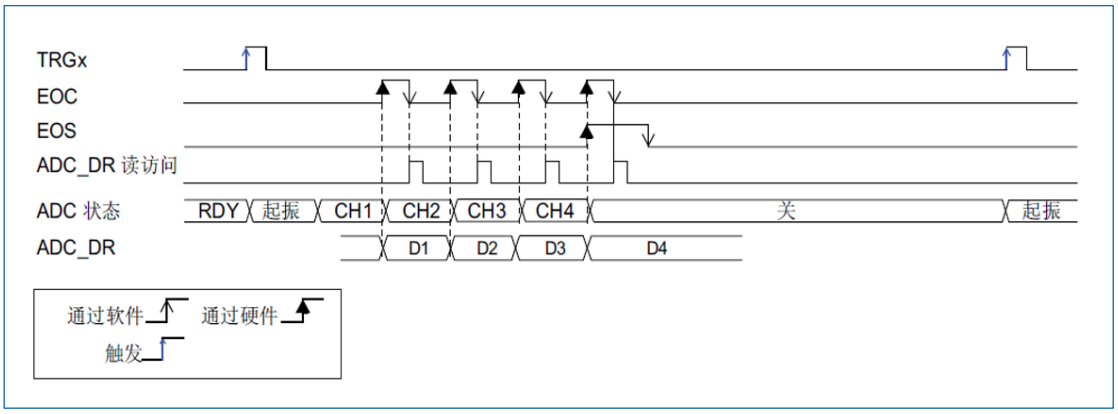


图 11-15 WAIT=0、AUTOFF=1 时的行为

上图的说明：EXTSEL=TRGx，EXTEN=01（上升沿），CONT=x，ADSTART=1，CHSEL=0xF，SCANDIR=0，WAIT=1，AUTOFF=1。

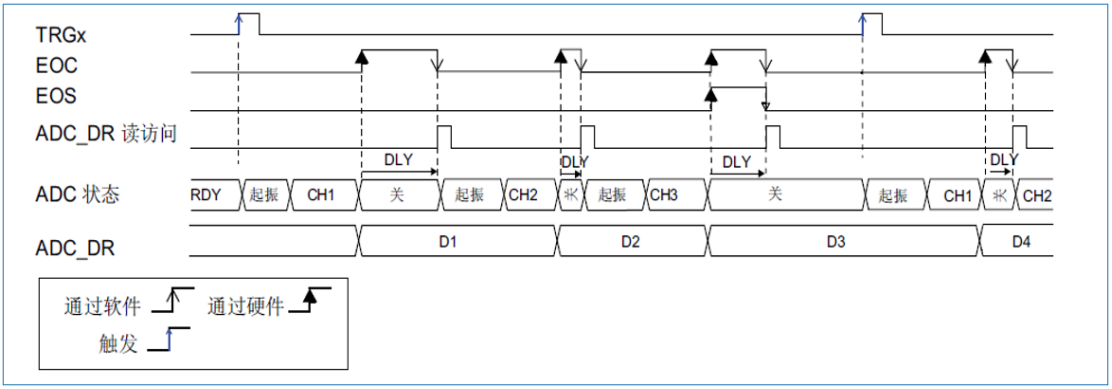


图 11-16 WAIT=1、AUTOFF=1 时的行为

上图的说明：EXTSEL=TRGx，EXTEN=01（上升沿），CONT=x，ADSTART=1，CHSEL=0xF，SCANDIR=0，WAIT=1，AUTOFF=1。

11.6 模拟窗口看门狗（AWDEN, AWDSGL, AWDCH, AWD_HTR/LTR, AWD）

将 ADC_CFGR1 寄存器中的 AWDEN 位置 1，可使能 AWD 模拟看门狗功能。此功能用于监控一条选定的通道或所有已使能的通道（参见表 11-8 模拟看门狗通道选择）是否仍处于所配置的电压范围（窗口）内，如图 11-17 模拟看门狗的保护区域所示。

如果 ADC 转换的模拟电压低于阈值下限或高于阈值上限，则会将 AWD 模拟看门狗状态位置 1。这

些阈值在 ADC_TR 寄存器中的 HT 位和 LT 位进行编程。可以通过将 ADC_IER 寄存器中的 AWDIE 位置 1 来使能中断。

AWD 标志可通过软件向其写入 1 的方式来清零。

如果转换的数据分辨率小于 12 位（取决于 RES[4:3]位），由于内部比较通常会基于 12 位原始转换数据执行（左对齐），因此被编程阈值的 LSB 必须保持清零状态。

表 11-7 介绍了如何对所有可能的分辨率进行比较。

表 11-7 模拟看门狗比较

分辨率位 RES[4:3]	模拟看门狗比较对象		备注
	原始转换数据，左对齐 (1)	阈值	
00：12 位	DATA[11:0]	LT[11:0]和 HT[11:0]	-
01：10 位	DATA[11:2]，00	LT[11:0]和 HT[11:0]	用户必须将 LT[1:0]和 HT[1:0]配置为“00”
10：8 位	DATA[11:4]，0000	LT[11:0]和 HT[11:0]	用户必须将 LT[3:0]和 HT[3:0]配置为“0000”
11：6 位	DATA[11:6]，000000	LT[11:0]和 HT[11:0]	用户必须将 LT[5:0]和 HT[5:0]配置为“000000”

(1). 进行任何对齐计算之前，会对原始转换数据进行看门狗比较。

表 11-8 介绍了如何配置 ADC_CFGR1 寄存器中的 AWDSGL 位和 AWDEN 位，以使能一条或多条通道上的模拟看门狗。

表 11-8 模拟看门狗通道选择

模拟看门狗保护的通道	AWDSGL 位	AWDEN 位
无	X	0
所有通道	0	1
单通道 (1)	1	1

(1). 通过 AWDCH[2:0]位选择。

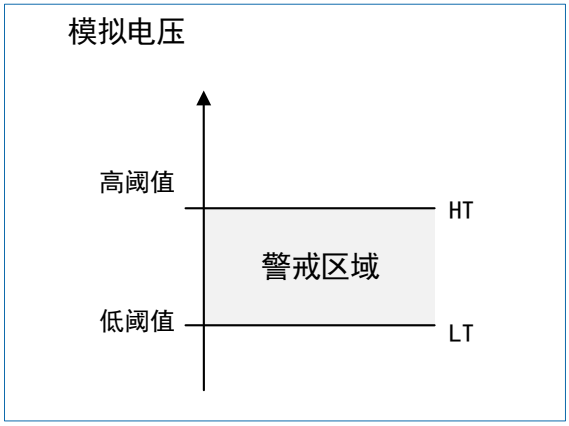


图 11-17 模拟看门狗的保护区域

11.7 内部参考电压

内部参考电压（VREFINT）为 ADC 提供了一个稳定的（带隙基准）电压输出。VREFINT 内部连接到 ADC_IN5 输入通道。

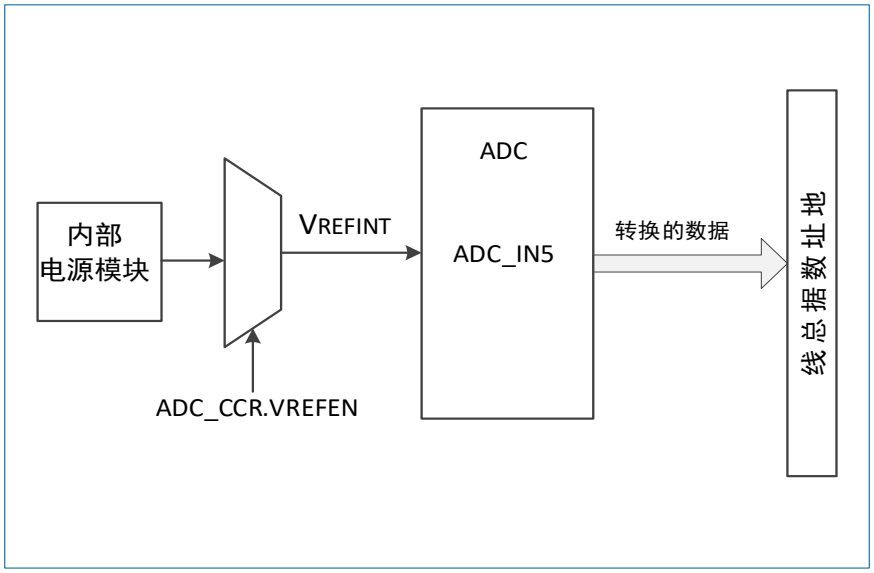


图 11-18 温度传感器和 V_{REFINT} 通道框图

使用内部参考电压计算实际的 V_{DDA} 电压

施加给 MCU 的 VDDA 电源电压可能会有变化，或无法获得准确值。在制造过程中由 ADC 在 VDDA=3.3V 的条件下获得的内置内部参考电压（V_{REFINT}）及其校准数据可用于评估实际的 V_{DDA} 电压水平。

以下公式可求得为器件供电的实际的 VDDA 电压：

$$\frac{V_{REFINT\_CAL}}{4096} * 3.3V = \frac{V_{REFINT\_S}}{4096} * V_{DDA}$$

由上述公式可得：　　$V_{DDA}=3.3 * V_{REFINT\_CAL} / V_{REFINT\_S}$

其中：

- V_{REFINT_CAL} 是内部参考电压 V_{REFINT} 在温度 25°C 时的校准值，存储地址：0x1FFF F834--0x1FFF F837。
- V_{REFINT_S} 表示内部参考电压的实际采样值。

将电源相关的 ADC 测量值转换为绝对电压值

ADC 用于提供对应于模拟电源与施加给转换通道的电压之比的数字值。对于大部分应用用例，需要将该比值转换成与 VDDA 无关的电压。对于 VDDA 已知、ADC 转换值进行了右对齐的应用，可使用以下公式得到该绝对值：

$$V_{CHANNELx}=\frac{V_{DDA}}{FULL\_SCALE}\times ADC\_DATA_x$$

对于VDDA值未知的应用，必须使用内部参考电压，VDDA可替换为使用内部参考电压计算实际的V_{DDA}电压部分提供的表达式，从而得出以下公式：

$$V_{CHANNELx}=\frac{3.3\times V_{REFINT\_CAL}\times ADC\_DATA_x}{V_{REFINT\_DATA}\times FULL\_SCALE}$$

其中：

- V_{REFINT_CAL} 是 VREFINT 校准值。
- ADC_DATAx 是由 ADC 在通道 x 上测得的值（右对齐）。

- V_{REFINT_DATA} 是由 ADC 转换得到的实际 VREFINT 输出值。
- FULL_SCALE 是 ADC 输出的最大数字值。例如，如果分辨率为 12 位，该值为2^{12} - 1 = 4095，如果分辨率为 8 位，该值为2^{8} - 1 = 255。

说明：如果执行 ADC 测量时使用的是输出格式而非 12 位右对齐格式，那么必须先将所有参数转换为兼容格式，然后再进行计算。

11.8 ADC 中断

发生下列任一事件均可生成中断：

- 校准结束（EOCAL 标志）
- ADC 就绪后，ADC 上电（ADRDY 标志）
- 任何转换结束（EOC 标志）
- 转换序列结束（EOS 标志）
- 进行模拟看门狗检测时（AWD 标志）
- 采样阶段结束时（EOSMP 标志）
- 发生数据溢出时（OVR 标志）

可以使用单独的中断使能位以提高灵活性。

表 11-9 ADC 中断

中断事件	事件标志	使能控制位
ADC 就绪	ADRDY	ADRDYIE
转换结束	EOC	EOCIE
转换序列结束	EOS	EOSIE
模拟看门狗状态位置 1	AWD	AWDIE
采样阶段结束	EOSMP	EOSMPIE
上溢	OVR	OVRIE

11.9 ADC 寄存器

基地址：0x4001 2400

空间大小：0x400

11.9.1 ADC 中断和状态寄存器（ADC_ISR）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								AWD	Res		OVR	EOS	EOC	EOSMP	ADRDY
								rw			rw	rw	rw	rw	rw

位 31:8	Res：保留 必须保持复位值。
位 7	AWD：模拟看门狗标志 当转换后的电压超过 ADC_TR 寄存器中设定的电压时，该位由硬件置位。 该位由软件对写 1 清除。 • 0：无模拟看门狗事件产生（或该事件标志由软件获取并清零） • 1：产生了模拟看门狗事件 *说明：r_w1 表示可读；可通过对该位域写 1 清除。对该位域写 0，该位域值无变化。*
位 6:5	Res：保留 必须保持复位值。
位 4	OVR：ADC 溢出 当 ADC 溢出产生时，该位由硬件置位。置位表示新的转换结束但 EOC 位还是为 1。 该位可由软件写 1 请零。 • 0：无溢出事件产生（或该事件标志由软件获取并清零） • 1：产生了溢出事件
位 3	EOS：序列转换结束标志 当CHSEL 位所选的通道序列转换结束后，该位由硬件置位。 该位由软件写入 1 清零。 • 0：序列转换未完成（或该事件标志由软件获取并清零） • 1：序列转换完成
位 2	EOC：转换结束标志 当每个通道新转换结果有效时（存放在ADC_DR 中），该位由硬件置位。 该位由软件写 1 清零或通过读取ADC_DR 寄存器清零。 • 0：通道转换未结束 • 1：通道转换结束
位 1	EOSMP：采样结束标志 在转换期间的采样阶段结束时，该位由硬件置 1。 • 0：不在采样结束阶段（或该事件标志由软件获取并清零） • 1：达到采样阶段结束条件
位 0	ADRDY：ADC 准备好 该位用软件写 1 清零。 • 0：ADC 未准备好（或该事件标志由软件获取并清零） • 1：ADC 已准备好开始转换

11.9.2 ADC 中断使能寄存器（ADC_IER）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								AWDIE	Res		OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE
								rw			rw	rw	rw	rw	rw

位 31:8	Res：保留 必须保持复位值。
位 7	AWDIE：模拟看门狗中断使能 该位由软件置 1 和清 0。 • 0：模拟看门狗中断禁用 • 1：模拟看门狗中断使能
位 6:5	Res：保留 必须保持复位值。
位 4	OVRIE：溢出中断使能 该位由软件置 1 和清 0。 • 0：溢出中断关闭 • 1：溢出中断开启（当 OVR 置位时产生中断）
位 3	EOSIE：序列转换结束中断使能 该位由软件置 1 和清 0。 • 0：EOS 中断关闭 • 1：EOS 中断开启（当 EOS 置位时产生中断）
位 2	EOCIE：转换结束中断使能 该位由软件置 1 和清 0。 • 0：EOC 中断关闭 • 1：EOC 中断开启（当 EOC 置位时产生中断）
位 1	EOSMPIE：采样结束中断使能 该位由软件置 1 和清 0。 • 0：EOSMP 中断关闭 • 1：EOSMP 中断开启（当 EOSMP 置位时产生中断）
位 0	ADRDYIE：ADC 准备好中断使能 该位由软件置 1 和清 0。 • 0：ADRDY 中断关闭

	• 1：ADRDY 中断开启（当ADRDY 置位时产生中断）

11.9.3 ADC 控制寄存器（ADC_CR）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAL	Res														
rs															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res											ADSTP	Res	ADSTART	ADDIS	ADEN
											rs		rs	rs	rs

位 31	ADCAL：ADC 校准 该位由软件设置来启动 ADC 校准。当校准完成后，由硬件清零。 • 0：校准完成 • 1：写 1 时，校准 ADC；读为 1 时，表明校准正在进行。
位 30:5	Res：保留 必须保持复位值。
位 4	ADSTP：ADC 停止转换命令 该位由软件设置来停止和丢弃正在进行中的转换。 当转换停止结束时，该位由硬件清零且 ADC 已准备好接受新的转换命令。 • 0：不发 ADC 停止转换命令 • 1：写 1 用于停止 ADC；读为 1 时，表明 ADSTP 命令正在执行中。
位 3	Res：保留 必须保持复位值。

位 2	ADSTART：ADC 开始转换命令
	该位由软件设置来启动 ADC 转换。
	• 一次转换启动的方式包括：立即启动（由软件配置）和硬件触发产生（硬件触发配置）。启动方式由 EXTEN 决定。该位由硬件清零。
	• 在单次转换模式且选择为软件触发（EXTSEL=0x0）时，序列转换结束（EOS 置位）后该位被清零。
	• 当执行完 ADSTP 命令后，ADSTP 位由硬件清零。
	  ○ 0：无进行中的 ADC 转换。
	  ○ 1：写 1 开始 ADC 转换。读为 1 表明 ADC 正在进行转换。

位 1	ADDIS：ADC 禁止命令 该位由软件设置来禁止 ADC（ADDIS 命令）并使 ADC 处于掉电状态（关断状态）。 一旦 ADC 有效关闭（ADEN 同时被硬件清零）后，由硬件清除该位。 • 0：无 ADDIS 命令正在执行。

	• 1：写 1 关闭 ADC；读为 1 时，表明 ADDIS 命令正在执行。
位 0	ADEN：ADC 使能命令 由软件设置该位来使能 ADC。一旦 ADRDY 标志置为 1 时，表明 ADC 可供使用。执行 ADDIS 命令后，ADC 关断且该位被硬件清零。 • 0：ADC 禁用（关断状态） • 1：写 1 以使能 ADC

11.9.4 ADC 配置寄存器 1（ADC_CFGR1）

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res			AWDCH[2:0]			Res		AWDEN	AWDSGL	Res					DISCEN
			rw					rw	rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AUTOFF	WAIT	CONT	OVEMOD	EXTEN[1:0]		Res	EXTSEL[2:0]			ALIGN	RES[1:0]		SCANDIR	Res	
rw	rw	rw	rw	rw			rw			rw			rw		

位 31:29	Res：保留 必须保持复位值。
位 28:26	AWDCH[2:0]：模拟看门狗监视的通道选择 该位域由软件设置或清 0。 • 000:由模拟看门狗监视的 ADC 模拟输入通道 0 • 001:由模拟看门狗监视的 ADC 模拟输入通道 1 • 010:由模拟看门狗监视的 ADC 模拟输入通道 2 • 011:由模拟看门狗监视的 ADC 模拟输入通道 3 • 100:由模拟看门狗监视的 ADC 模拟输入通道 4 • 101:由模拟看门狗监视的 ADC 模拟输入通道 5 • 其它值：保留
位 25:24	Res：保留 必须保持复位值。
位 23	AWDEN：模拟看门狗使能位 该位由软件置 1 或清 0。 • 0：关闭模拟看门狗 • 1：开启模拟看门狗
位 22	AWDSGL：在单一通道或所有通道使能看门狗 该位由软件置 1 或清 0，以使能 AWDCH 指定的通道或所有通道。 • 0：在所有通道上使能模拟看门狗 • 1：在单一通道上使能模拟看门狗

位 21:17	Res：保留 必须保持复位值。
位 16	DISCEN：断续模式 该位由软件置 1 或清 0，以开启或禁用断续模式。 • 0：禁用断续模式 • 1：开启断续模式
位 15	AUTOFF：自动关断模式 该位由软件置 1 或清 0，以开启或禁用自动关断模式。 • 0：禁用自动关断模式 • 1：开启自动关断模式
位 14	WAIT：等待转换模式 该位由软件置 1 或清 0，以开启或禁用等待转换模式。 • 0：关闭等待转换模式 • 1：开启等待转换模式 *注意：只有当 ADSTART = 0 时，才允许软件写入此位（为了确保上一次转换已完成）。*
位 13	CONT：单次/连续转换模式 该位由软件置 1 和清 0。若该位置 1，则转换为连续模式直到该位清 0。 • 0：单次转换模式 • 1：连续转换模式
位 12	OVRMOD：溢出管理模式 该位由软件置 1 和清 0，以配置数据溢出管理。 • 0：当检测到溢出事件时，ADC_DR 寄存器保持上一次数据。 • 1：当检测到溢出事件时，ADC_DR 寄存器用最后一次的转换数据覆盖。
位 11:10	EXTEN[1:0]：外部触发使能和极性选择 该位域可由软件置位和清 0，以选择外部触发的极性并使能触发器。 • 00：硬件触发检测关闭（可由软件启动转换）。 • 01：在上升沿进行硬件触发检测。 • 10：在下降沿进行硬件触发检测。 • 11：在上升和下降沿进行硬件触发检测。
位 9	Res：保留 必须保持复位值。
位 8:6	EXTSEL[2:0]：外部触发选择 该位域用于选择触发 ADC 转换的外部事件。 • 000：事件 0（TIM1_TRGO）

	• 001：事件 1（TIM1_CC4） • 010：事件 2（TIM2_TRGO） • 011：事件 3（TIM6_TRGO） • 100：事件 4（TIM1_CC1） • 101：事件 5（TIM1_CC2） • 110：事件 6（TIM1_CC3） • 111：事件 7（IO_TRIG）
位 5	ALIGN：数据对齐 该位由软件置 1 或清 0，以选择数据的左或右对齐。 • 0：右对齐 • 1：左对齐
位 4:3	RES[1:0]：数据分辨率 通过软件写入这些位可选择转换的分辨率。 • 00：12 位 • 01：10 位 • 10：8 位 • 11：6 位 *注：仅当 ADEN=0 时，才允许通过软件对这些位执行写操作。*
位 2	SCANDIR：扫描序列方向 该位由软件置 1 或清 0，以选择通道序列中的通道扫描方向。 • 0：向前扫描（从CHSEL0 到CHSEL5） • 1：向后扫描（从CHSEL5 到 CHSEL0）
位 1:0	Res：保留 必须保持复位值。

11.9.5 ADC 配置寄存器 2（ADC_CFGR2）

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CKMODE[1:0]		Res													
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res															

位 31:30	CKMODE[1:0]：ADC 时钟模式 该位域由软件置位和清 0。 • 00：ADCCLK 异步时钟工作模式 • 01：PCLK/2 同步时钟工作模式

	• 10：PCLK/4 同步时钟工作模式 • 11：保留
位 29:0	Res：保留 必须保持复位值。

11.9.6 ADC 采样时间寄存器（ADC_SMPR）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res													SMP[2:0]		
													rw		

位 31:3	Res：保留 必须保持复位值。
位 2:0	SMP[2:0]：采样时间选择 该位域由软件设置，用于选择所选通道的采样时间。 • 000：1.5 个 ADC 时钟周期 • 001：7.5 个 ADC 时钟周期 • 010：13.5 个 ADC 时钟周期 • 011：28.5 个 ADC 时钟周期 • 100：41.5 个 ADC 时钟周期 • 101：55.5 个 ADC 时钟周期 • 110：71.5 个 ADC 时钟周期 • 111：239.5 个 ADC 时钟周期

11.9.7 ADC 看门狗阈值寄存器（ADC_TR）

偏移地址：0x20

复位值：0x0000 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res				HT[11:0]											
				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				LT[11:0]											
				rw											

位 31:28	Res：保留 必须保持复位值。
位 27:16	HT[11:0]：模拟看门狗的阈值上限

	该位域由软件设置，用于定义模拟看门狗的阈值上限。
位 15:12	Res：保留 必须保持复位值。
位 11:0	LT[11:0]：模拟看门狗阈值下限 该位域由软件改写，用于定义模拟看门狗的阈值下限。

11.9.8 ADC 通道选择寄存器（ADC_CHSELR）

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res										CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0
										rw	rw	rw	rw	rw	rw

位 31:6	Res：保留 必须保持复位值。
位 x （x = 0..5）	CHSELx：通道选择（x = 0..5） 该位域由软件设置，用于定义所要转换序列的通道。 • 0：不选择输入通道x 作为转换通道 • 1：选择输入通道x 作为转换通道

11.9.9 ADC 数据寄存器（ADC_DR）

偏移地址：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
r															

位 31:6	Res：保留 必须保持复位值。
位 15:0	DATA[15:0]：转换数据 该位域保存最后转换通道的转换结果，只支持读。 仅在校准完成时，DATA[6:0]值为校准因子。

11.9.10 ADC 通用配置寄存器（ADC_CCR）

偏移地址：0x308

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res									VREEFEN						
									rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res															

位 31:23	Res：保留 必须保持复位值。
位 22	VREFEN：VREFINT 通道使能 该位由软件置 1 或清 0，以打开或关闭 VREFINT 通道。 • 0：VREFINT 通道关闭 • 1：VREFINT 通道开启
位 21:0	Res：保留 必须保持复位值。

11.9.11 ADC 控制寄存器 2（ADC_CR2）

偏移地址：0x3f0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WAKE_EN	Res														
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res														SDIF	Res
														rw	

位 31	WAKE_EN：使能 AWD 唤醒功能 使能 AWD 唤醒功能后，可以在 Stop 模式下检测到 beeper 时钟信号，以便预唤醒。 • 0：AWD 唤醒关闭（默认） • 1：AWD 唤醒使能
位 30:2	Res：保留 必须保持复位值。
位 1	SDIF：差分输入使能 差分输入使能后，通道 0 和 1、2 和 3、4 和 5 组成差分输入（因为通道 5 接内部 VREF，所以差分输入模式下通道 4 - 5 不可用）。 • 0：ADC 通道单端输入模式（默认） • 1：ADC 通道差分输入模式
位 0	Res：保留 必须保持复位值。

ADC 的 AWD 唤醒功能注意事项：

- *Beeper 输出的触发信号不需要设置触发控制寄存器 EXTEN 和 EXTSEL。*
- *ADC 的配置都在这种模式下生效，所以必须使能 AWDEN，不能使能循环和间断模式。在转换的过程中，除 ADC_ISR.AWD 以外的状态寄存器不会被置位。*
- *触发 ADC 单个通道或全部通道的 ADC 转换，和其他触发信号一样的效果。如果在单个通道或全部通道转换完成后，ADC_ISR.AWD 状态位没有置起则重新回到停机模式。*
- *AWD 的唤醒信号根据 AWDSGL 和 AWDCH 设置，可以比较单个通道的结果或比较所有通道的结果。直接将模拟看门狗标志 ADC_ISR.AWD 发送到 EXTI，不需要使能 AWDIE。*

12 高级控制定时器（TIM1）

高级控制定时器（TIM1）由一个 16 位的自动装载计数器组成，它由一个可编程的预分频器驱动。

高级控制定时器适合多种用途，包含测量输入信号的脉冲宽度（输入捕获），或者产生输出波形（输出比较、PWM、死区时间的互补 PWM 等）。

使用定时器预分频器和 RCC 时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

高级控制定时器（TIM1）和其他定时器完全独立，不共享任何资源。

表 12-1 TIM1 特性

符号	参数	条件	最小值	典型值	最大值	单位
$t_{res（TIM）}$	定时器分辨时间	$f_{TIMxCLK}$=32 MHz	-	31.2	-	ns
f_{EXT}	定时器的 CH1 至 CH4，外部输入的时钟频率	-	-	$f_{TIMxCLK}/2$	-	MHz
		$f_{TIMxCLK}$=32 MHz	-	16	-	MHz
t_{MAX_COUNT}	当选择内部时钟时，16 位计数器的时钟周期	-	-	2^{16}	-	$t_{TIMxCLK}$
		$f_{TIMxCLK}$=32 MHz	-	2048	-	μs

12.1 TIM1 主要功能

TIM1 定时器的功能包括：

- 16 位向上、向下、向上/向下自动装载计数器
- 16 位可编程（可实时修改）预分频器，计数器时钟频率的分频系数为 1~65536 之间的任意数值。
- 4 个独立通道：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿或中间对齐模式）
  - 单脉冲模式输出
- 死区时间可编程的互补输出
- 使用外部信号控制定时器和定时器互连的同步电路

- 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- 刹车输入信号可以将定时器输出信号置于复位状态或者一个已知状态
- 以下事件发生时产生中断：
  - 更新：计数器向上溢出/向下溢出，计数器初始化（通过软件或者内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或者由内部/外部触发计数）
  - 输入捕获
  - 输出比较
  - 刹车信号输入
- 支持针对定位的增量（正交）编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理

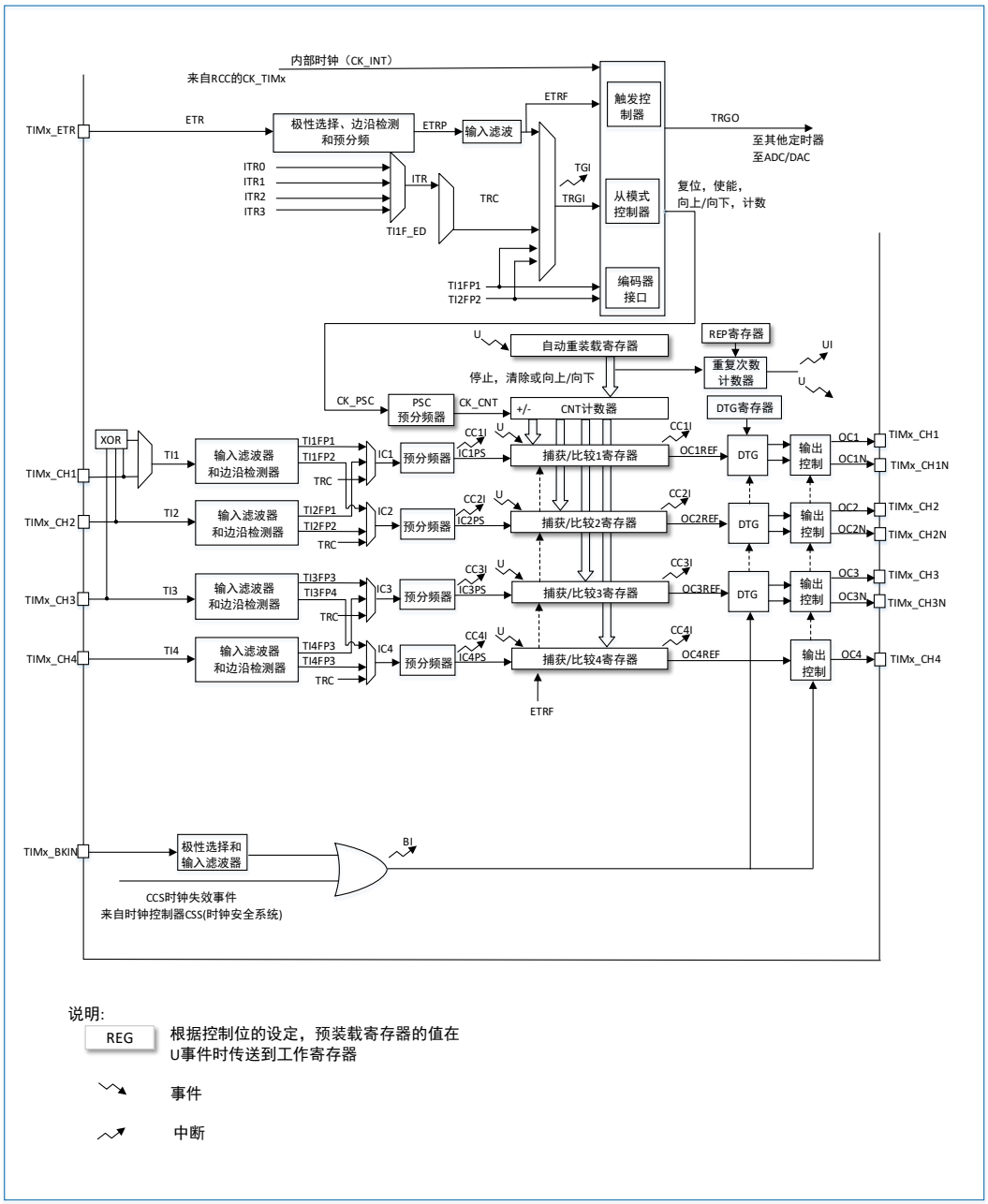


图 12-1 高级控制定时器框图

12.2 TIM1 功能描述

12.2.1 时基单元

可编程高级控制定时器的主要部分是一个 16 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数、向下计数或者向上向下双向计数。此计数器时钟由预分频器分频得到。

计数器、自动装载寄存器和预分频器寄存器可以由软件读写。即使计数器还在运行，读写仍然有效。

时基单元包含：

- 计数器寄存器（TIM1_CNT）
- 预分频器寄存器（TIM1_PSC）
- 自动装载寄存器（TIM1_ARR）
- 重复次数寄存器（TIM1_RCR）

自动装载寄存器是预先装载的，写或读自动重装载寄存器将访问预装载寄存器。根据在 TIM1_CR1 寄存器中的自动装载预装载使能位（ARPE）的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件（向下计数时的下溢条件）并当 TIM1_CR1 寄存器中的 UDIS 位等于 0 时，产生更新事件。更新事件也可以由软件产生。随后会详细描述每一种配置下更新事件的产生。

计数器由预分频器的时钟输出 CK_CNT 驱动，仅当设置了计数器 TIM1_CR1 寄存器中的计数器使能位（CEN）时，CK_CNT 才有效。（更多有关使能计数器的细节，请参见“12.2.19 TIM1 定时器和外部触发的同步”）。

注意：在设置了 TIM1_CR 寄存器的 CEN 位的一个时钟周期后，计数器开始计数。

预分频器描述

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个（在 TIM1_PSC 寄存器中的）16 位寄存器控制的 16 位计数器。因为这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 12-2 和图 12-3 给出了在预分频器运行时，更改预分频器参数的例子。

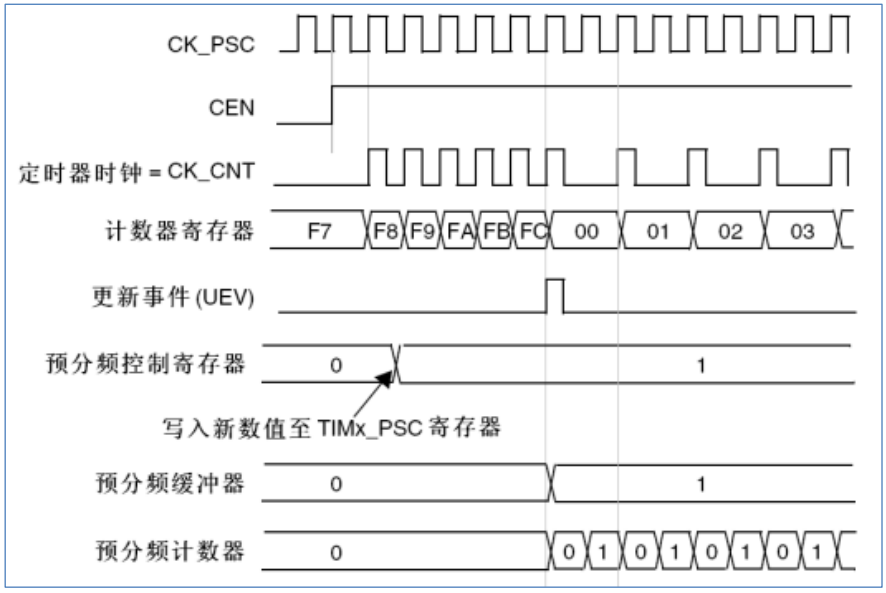


图 12-2 当预分频器的参数从 1 变到 2 时，计数器的时序图

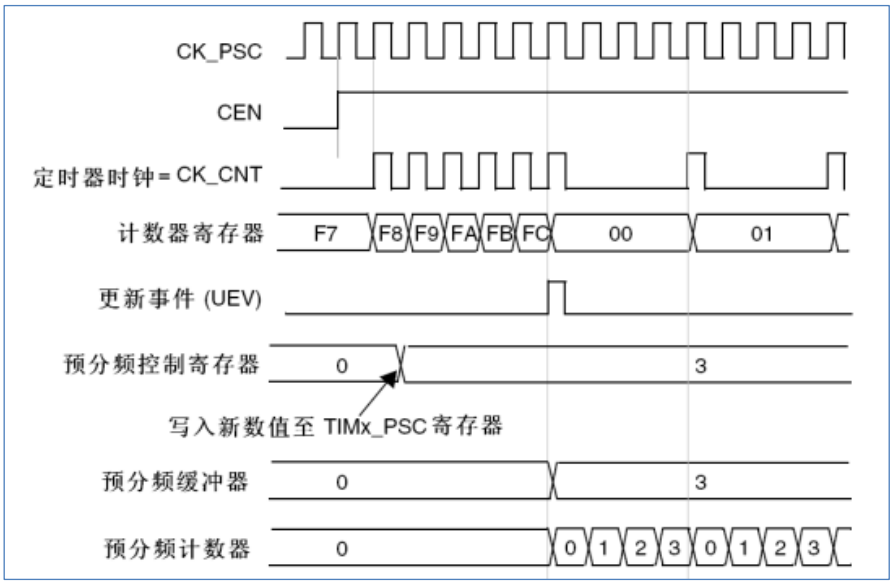


图 12-3 当预分频器的参数从 1 变到 4 时，计数器的时序图

12.2.2 计数器模式

12.2.2.1 向上计数模式

在向上计数模式中，计数器从 0 计数到自动加载值（TIM1_ARR 计数器的内容），然后重新从 0 开始计数并且产生一个计数器溢出事件。

如果使用了重复计数器功能，在向上计数达到设置的重复计数次数（TIM1_RCR）时，产生更新事件（UEV）；否则每次计数器溢出时才产生更新事件。

在 TIM1_EGR 寄存器中（通过软件方式或者使用从模式控制器）设置 UG 位也同样可以产生一个更新事件。

设置 TIM1_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清'0'之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清'0'，同时预分频器的计数也被清零（但预分频器的数值不变）。此外，如果设置了 TIM1_CR1 寄存器中的 URS 位（选择更新请求），设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志（即不产生中断）。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时（依据 URS 位）设置更新标志位（TIM1_SR 寄存器中的 UIF 位）。

- 重复计数器被重新加载为 TIM1_RCR 寄存器的内容。
- 自动装载影子寄存器被重新置入预装载寄存器的值（TIM1_ARR）。
- 预分频器的缓冲区被置入预装载寄存器的值（TIM1_PSC 寄存器的内容）。

下面给出一些例子，当 TIM1_ARR=0x36 时计数器在不同时钟频率下的动作。

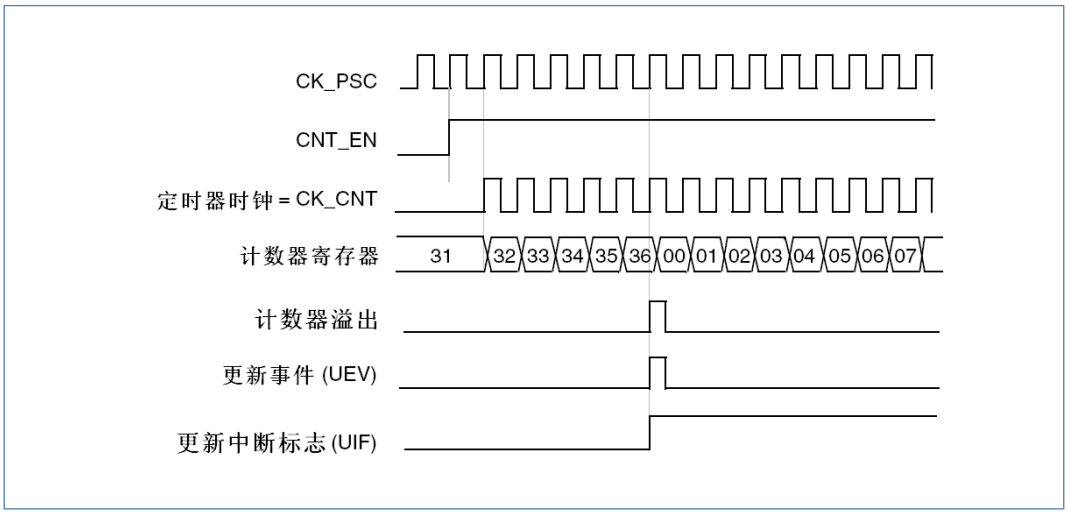


图 12-4 计数器时序图，内部时钟分频因子为 1

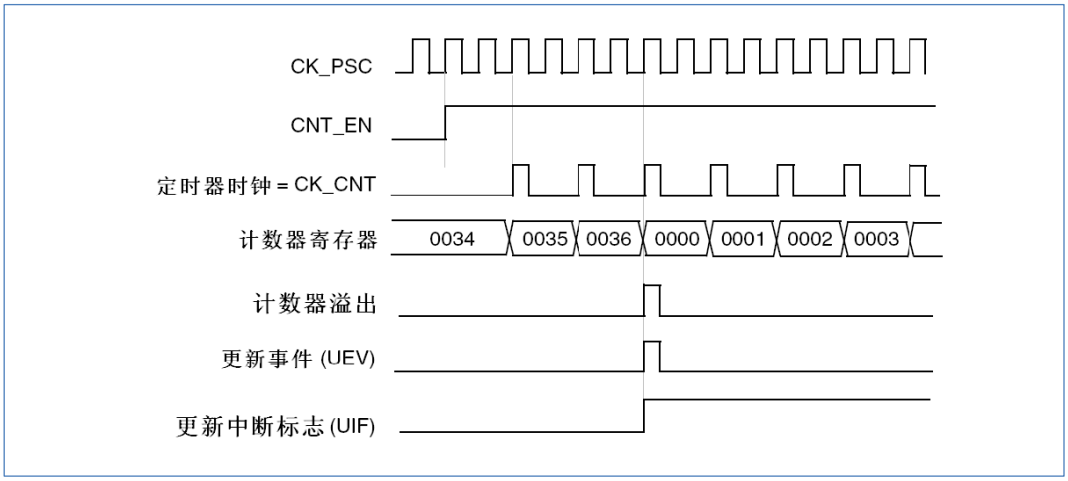


图 12-5 计数器时序图，内部时钟分频因子为 2

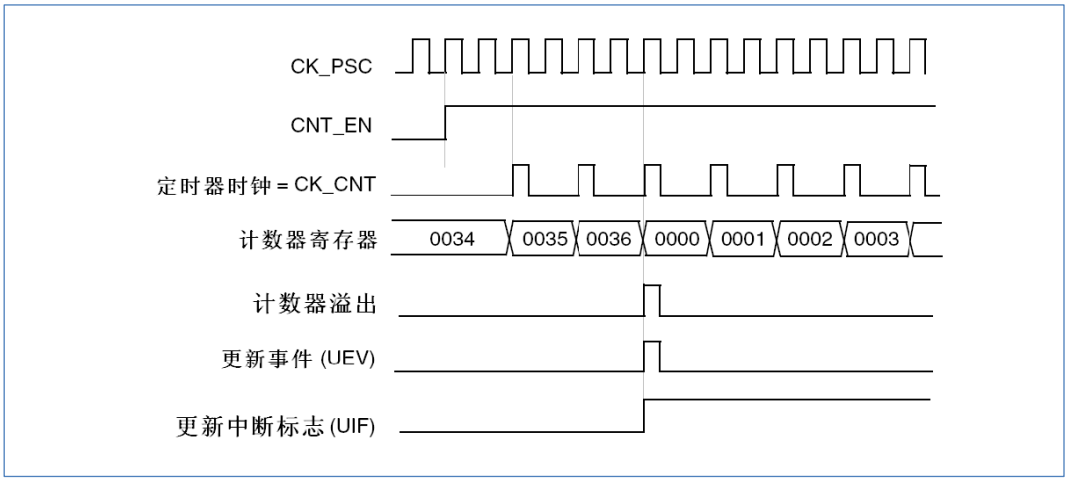


图 12-6 计数器时序图，内部时钟分频因子为 4

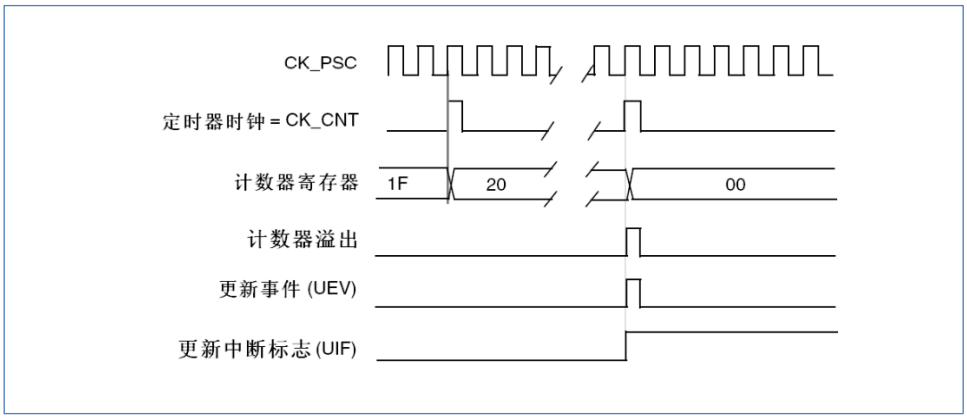


图 12-7 计数器时序图，内部时钟分频因子为 N

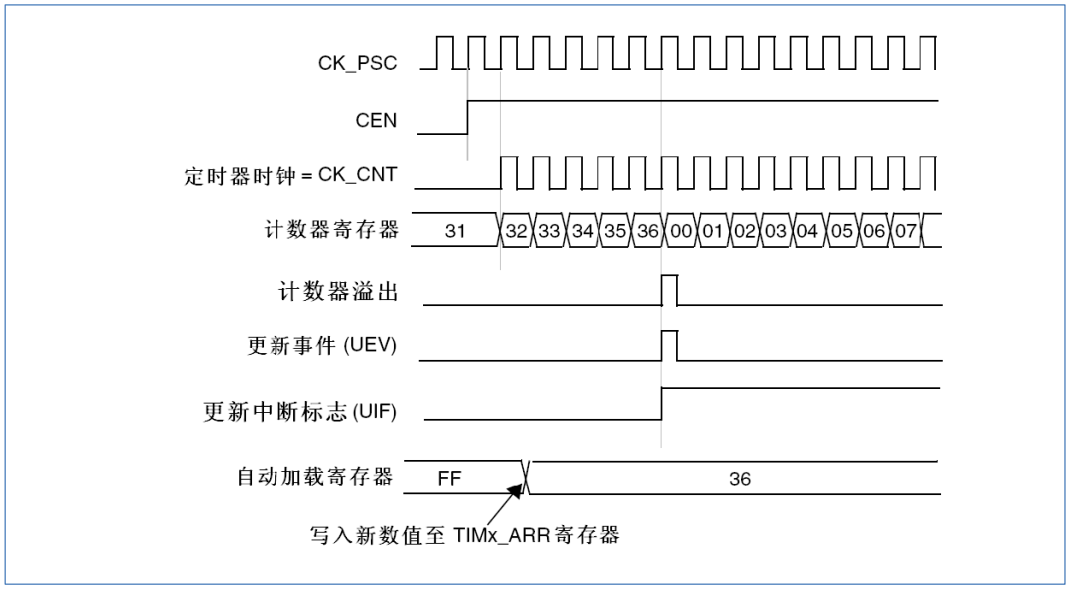


图 12-8 计数器时序图，当 ARPE=0 时的更新事件（TIM1_ARR 没有预装入）

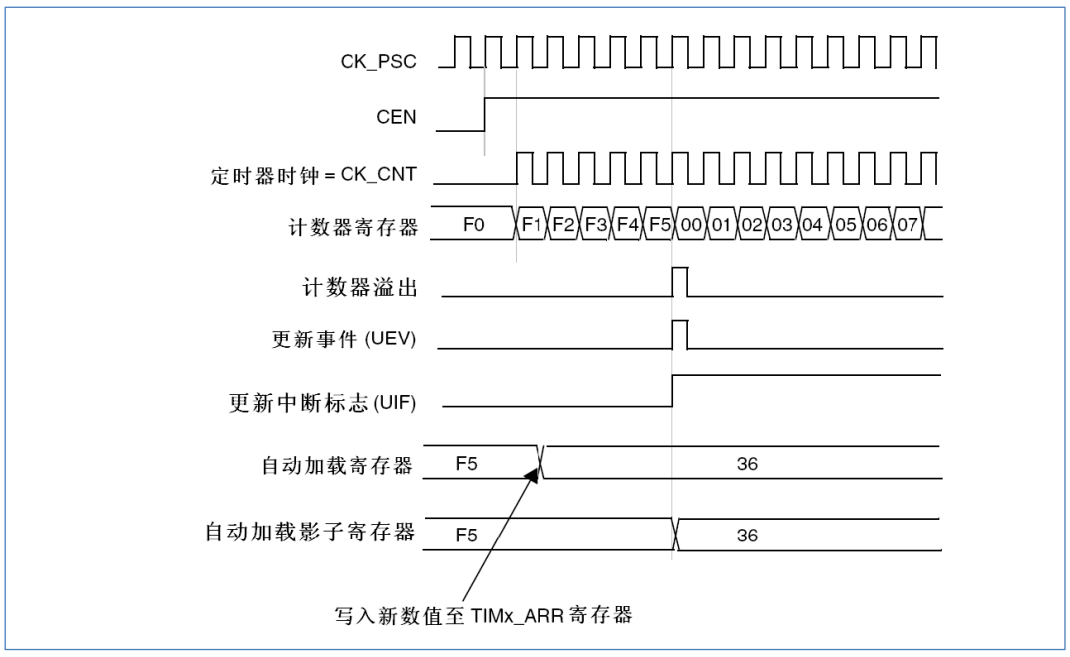


图 12-9 计数器时序图，当 ARPE=1 时的更新事件（预装入了 TIM1_ARR）

12.2.2.2 向下计数模式

在向下模式中，计数器从自动装入的值（TIM1_ARR 计数器的值）开始向下计数到 0，然后从自动装

入的值重新开始并且产生一个计数器向下溢出事件。

如果使用了重复计数器，当向下计数重复了重复计数寄存器（TIM1_RCR）中设定的次数后，将产生更新事件（UEV），否则每次计数器下溢时才产生更新事件。

在 TIM1_EGR 寄存器中（通过软件方式或者使用从模式控制器）设置 UG 位，也同样可以产生一个更新事件。

设置 TIM1_CR1 寄存器的 UDIS 位可以禁止 UEV 事件。这样可以避免向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，并且预分频器的计数器重新从 0 开始（但预分频系数不变）。

此外，如果设置了 TIM1_CR1 寄存器中的 URS 位（选择更新请求），设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志（因此不产生中断），这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且（根据 URS 位的设置）更新标志位（TIM1_SR 寄存器中的 UIF 位）也被设置。

- 重复计数器被重置为 TIM1_RCR 寄存器中的内容。
- 预分频器的缓存器被加载为预装载的值（TIM1_PSC 寄存器的值）。
- 当前的自动加载寄存器被更新为预装载值（TIM1_ARR 寄存器中的内容）。

说明：自动装载在计数器重载入之前被更新，因此下一个周期将是预期的值。

以下是一些当 TIM1_ARR=0x36 时，计数器在不同时钟频率下的操作例子。

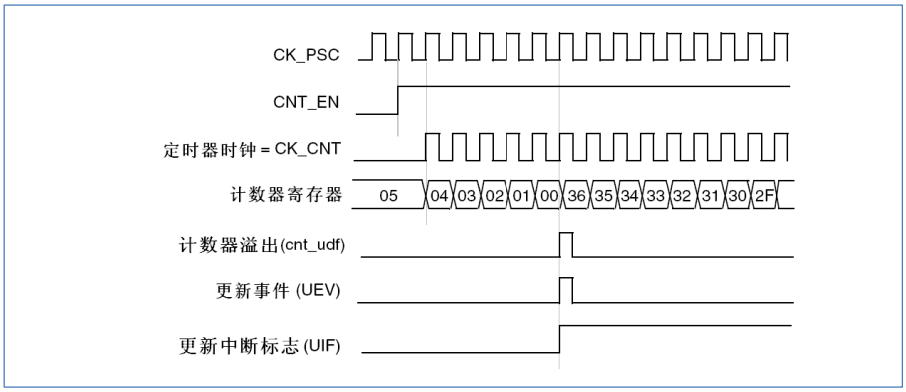


图 12-10 计数器时序图，内部时钟分频因子为 1

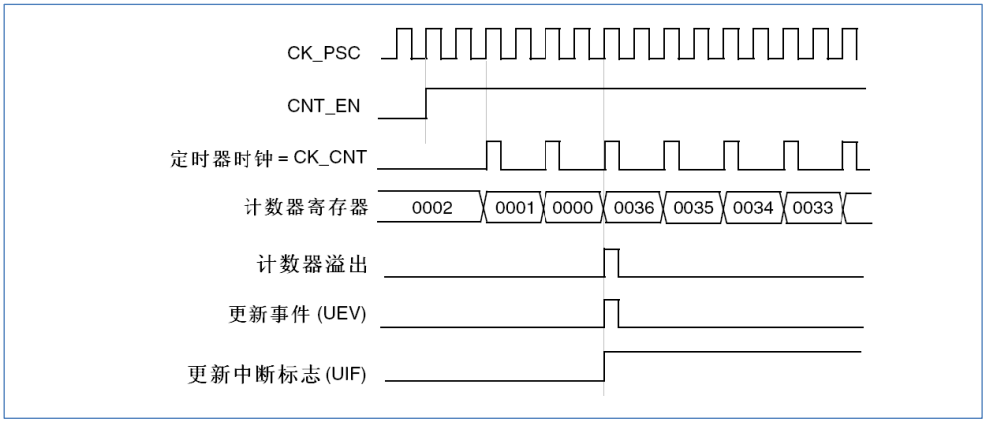


图 12-11 计数器时序图，内部时钟分频因子为 2

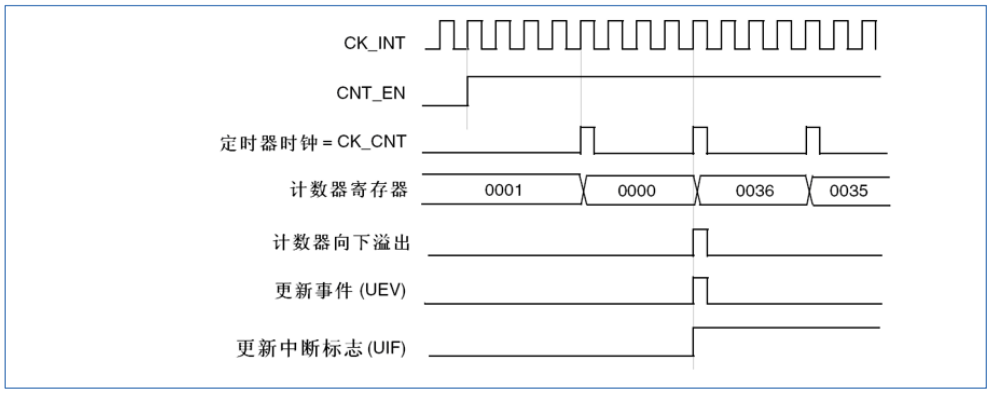


图 12-12 计数器时序图，内部时钟分频因子为 4

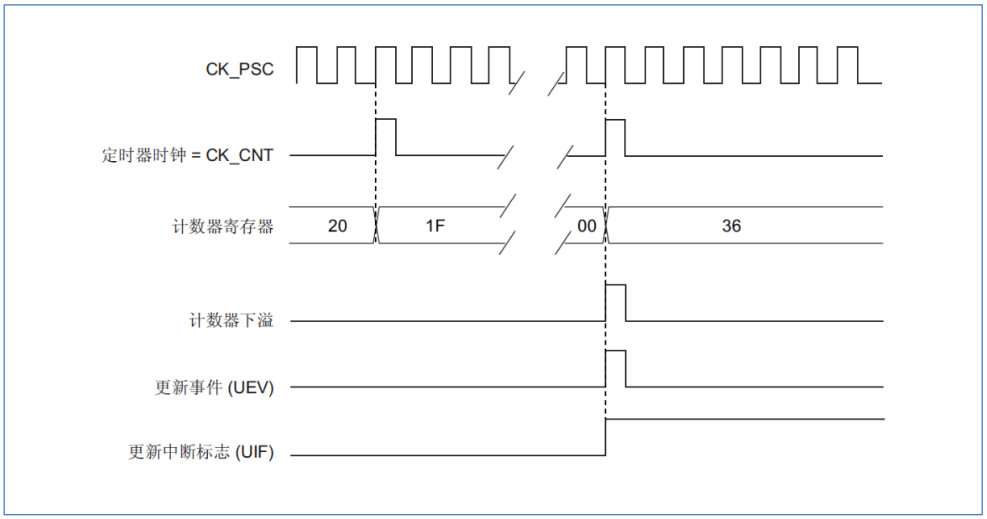


图 12-13 计数器时序图，内部时钟分频因子为 N

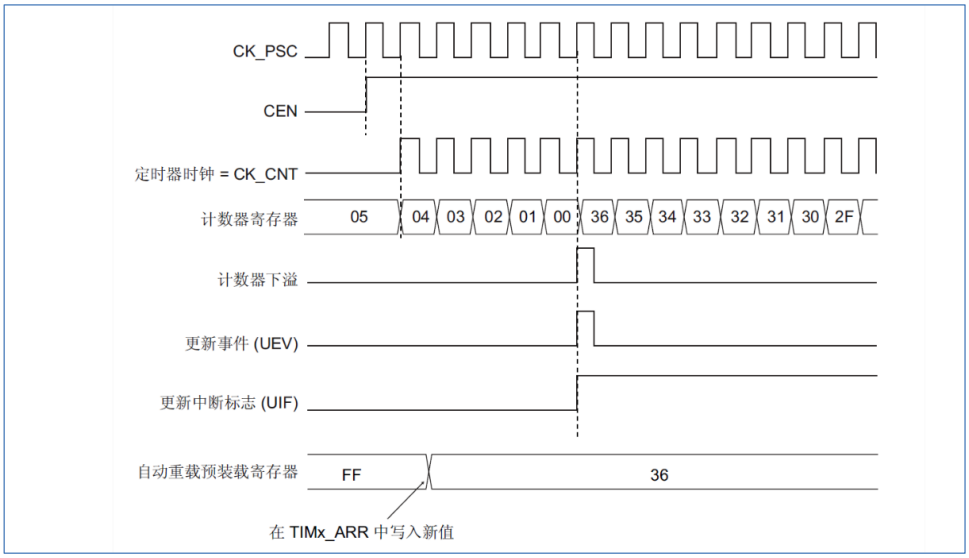


图 12-14 计数器时序图，当没有使用重复计数器时的更新事件

12.2.2.3 中央对齐模式（向上/向下计数）

在中央对齐模式，计数器从 0 开始计数到自动重载值（TIM1_ARR 寄存器的值）减 1，产生一个计数器溢出事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。

在此模式下，不能写入 TIM1_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

可以在每次计数上溢和每次计数下溢时产生更新事件；也可以通过（软件或者使用从模式控制器）设置 TIM1_EGR 寄存器中的 UG 位产生更新事件。然后，计数器重新从 0 开始计数，预分频器也重新从 0

开始计数。

设置 TIM1_CR1 寄存器中的 UDIS 位可以禁止 UEV 事件。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

此外，如果设置了 TIM1_CR1 寄存器中的 URS 位（选择更新请求），设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志（因此不产生中断），这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且（根据 URS 位的设置）更新标志位（TIM1_SR 寄存器中的 UIF 位）也被设置。

- 重复计数器被重置为 TIM1_RCR 寄存器中的内容。
- 预分频器的缓存器被加载为预装载（TIM1_PSC 寄存器）的值。
- 当前的自动加载寄存器被更新为预装载值（TIM1_ARR 寄存器中的内容）。

说明：如果因为计数器溢出而产生更新，自动重装载将在计数器重载入之前被更新，因此下一个周期将是预期的值（计数器被装载为新的值）。

以下是一些计数器在不同时钟频率下的操作的例子：

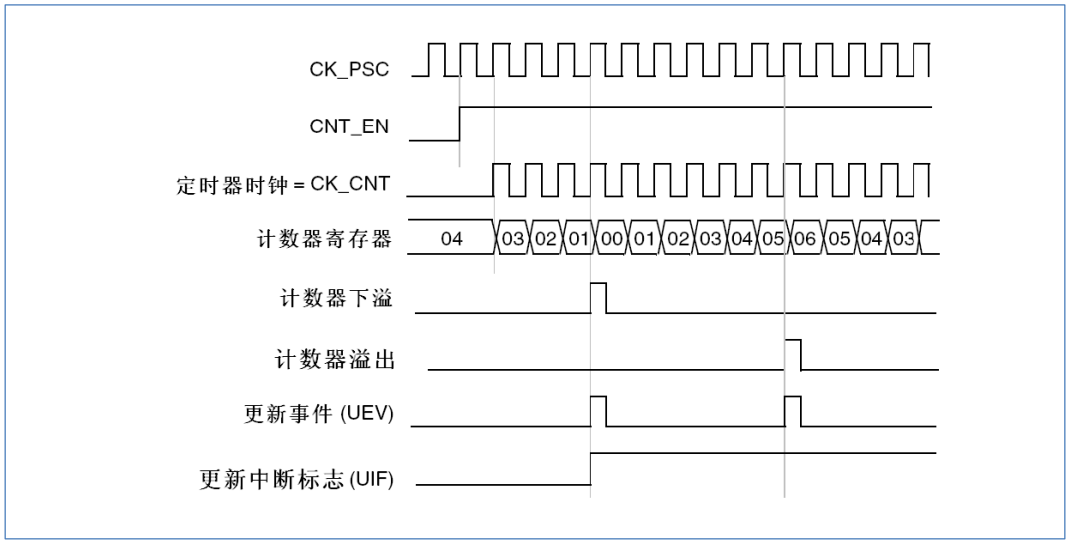


图 12-15 计数器时序图，内部时钟分频因子为 1，TIM1_ARR=0x6（这里使用了中心对齐模式 1）

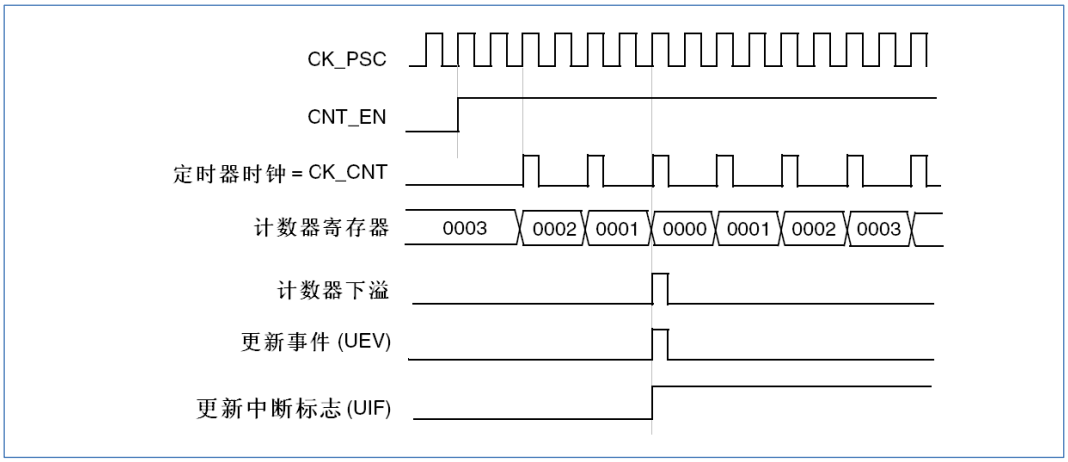


图 12-16 计数器时序图，内部时钟分频因子为 2（这里使用了中心对齐模式 1）

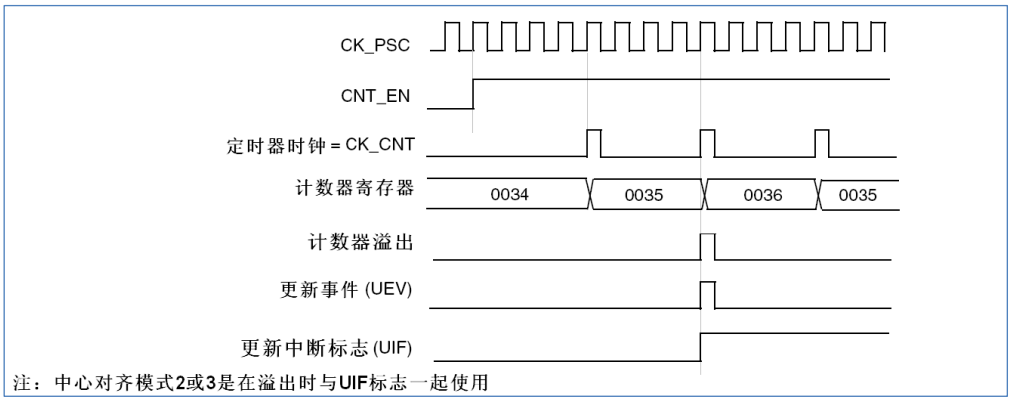


图 12-17 计数器时序图，内部时钟分频因子为 4，TIM1_ARR=0x36（这里使用了中心对齐模式 2 或 3）

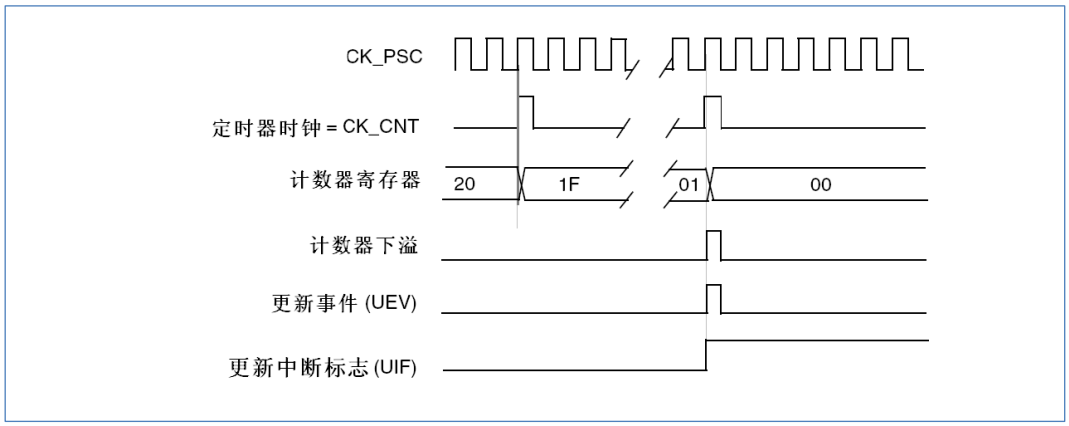


图 12-18 计数器时序图，内部时钟分频因子为 N

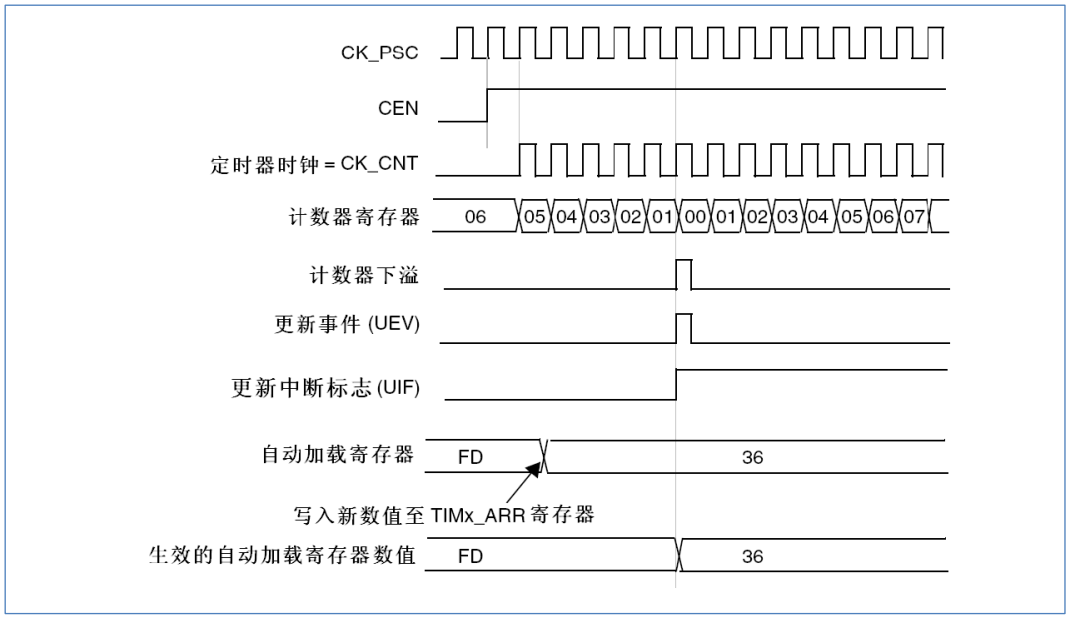


图 12-19 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

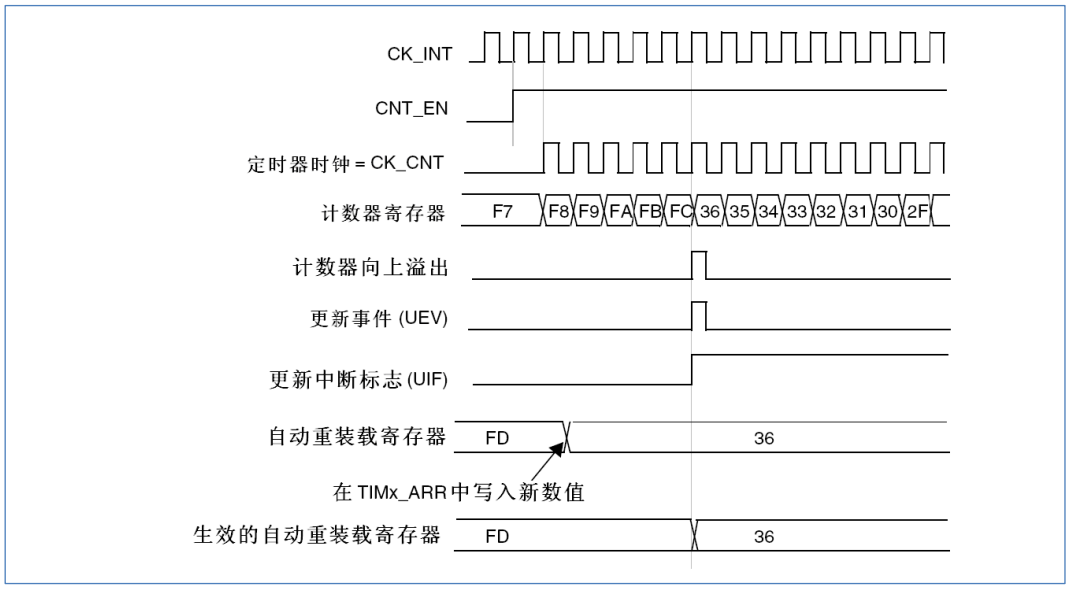


图 12-20 计数器时序图，ARPE=1 时的更新事件（计数器溢出）

12.2.3 重复计数器

“时基单元”解释了计数器上溢/下溢时更新事件（UEV）是如何产生的，然而事实上它只能在重复计数达到 0 的时候产生。这个特性对产生 PWM 信号非常有用。

这意味着在每 N 次计数上溢或下溢时，数据从预装载寄存器传输到影子寄存器（TIM1_ARR 自动重载入寄存器，TIM1_PSC 预装载寄存器，还有在比较模式下的捕获/比较寄存器 TIM1_CCRx），N 是 TIM1_RCR 重复计数寄存器中的值。

重复计数器在下述任一条件成立时递减：

- 向上计数模式下每次计数器溢出时
- 向下计数模式下每次计数器下溢时
- 中央对齐模式下每次上溢和每次下溢时。虽然这样限制了 PWM 的最大循环周期为 128，但它能够在每个 PWM 周期 2 次更新占空比。在中央对齐模式下，因为波形是对称的，如果每个 PWM 周期中仅刷新一次比较寄存器，则最大的分辨率为 2 x Tck。

重复计数器是自动加载的，重复速率是由 TIM1_RCR 寄存器的值定义（参看图 12-21）。当更新事件由软件产生（通过设置 TIM1_EGR 中的 UG 位）或者通过硬件的从模式控制器产生，则无论重复计数器的值是多少，立即发生更新事件，并且 TIM1_RCR 寄存器中的内容被重载入到重复计数器。

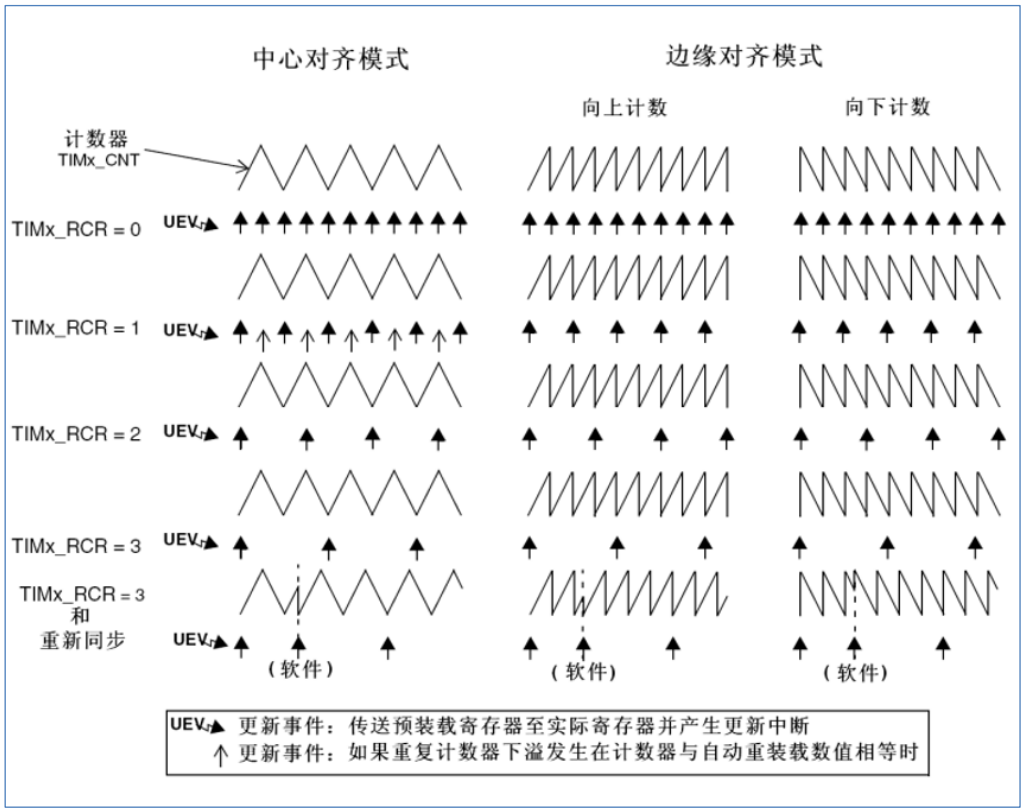


图 12-21 不同模式下更新速率的例子，及 TIM1_RCR 的寄存器设置

12.2.4 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟（CK_INT）
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 内部触发输入（ITRx）：使用一个定时器作为另一个定时器的预分频器。如可以配置一个定时器 Timer1 作为另一个定时器 Timer2 的预分频器。详细信息，参见“13.2.15.1 使用一个定时器作为另一个定时器的预分频器”。

内部时钟源（CK_INT）

如果禁止了从模式控制器（SMS=000），则 CEN、DIR（TIM1_CR1 寄存器）和 UG 位（TIM1_EGR 寄存器）是事实上的控制位，并且只能被软件修改（UG 位仍被自动清除）。只要 CEN 位被写成‘1’，预分频器的时钟就由内部时钟 CK_INT 提供。

下图显示控制电路和向上计数器在一般模式下，不带预分频器时的操作。

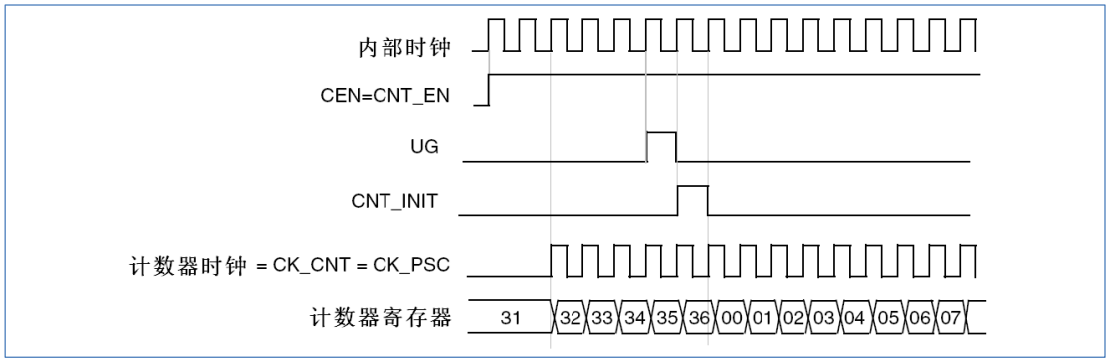


图 12-22 一般模式下的控制电路，内部时钟分频因子为 1

外部时钟源模式 1

当 TIM1_SMCR 寄存器的 SMS=111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

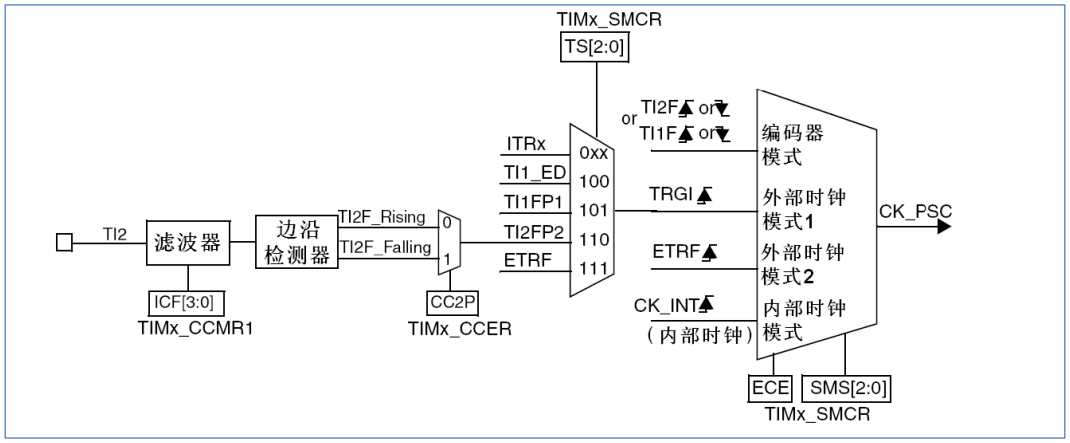


图 12-23 TI2 外部时钟连接例子

例如，配置向上计数器在 T12 输入端的上升沿计数，使用下列步骤：

1. 配置 TIM1_CCMR1 寄存器 CC2S=01，配置通道 2 检测 TI2 输入的上升沿。
2. 配置 TIM1_CCMR1 寄存器的 IC2F[3:0]，选择输入滤波器带宽（如果不需要滤波器，保持 IC2F=0000）。
3. 配置 TIM1_CCER 寄存器的 CC2P=0，选定上升沿极性。
4. 配置 TIM1_SMCR 寄存器的 SMS=111，选择定时器外部时钟模式 1。
5. 配置 TIM1_SMCR 寄存器中的 TS=110，选定 TI2 作为触发输入源。
6. 设置 TIM1_CR1 寄存器的 CEN=1，启动计数器。

说明：捕获预分频器不用作触发，所以不需要对它进行配置。

当上升沿出现在 TI2，计数器计数一次，且 TIF 标志被设置。

在 TI2 的上升沿和计数器实际时钟之间的延时，取决于在 TI2 输入端的重新同步电路。

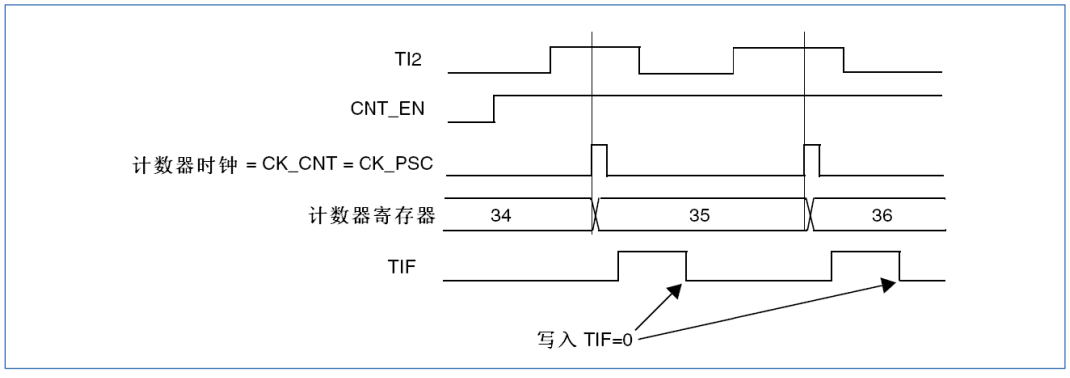


图 12-24 外部时钟模式 1 下的控制电路

外部时钟源模式 2

选定此模式的方法为：令 TIM1_SMCR 寄存器中的 ECE=1。

计数器能够在外部触发 ETR 的每一个上升沿或下降沿计数。

下图是外部触发输入的框图。

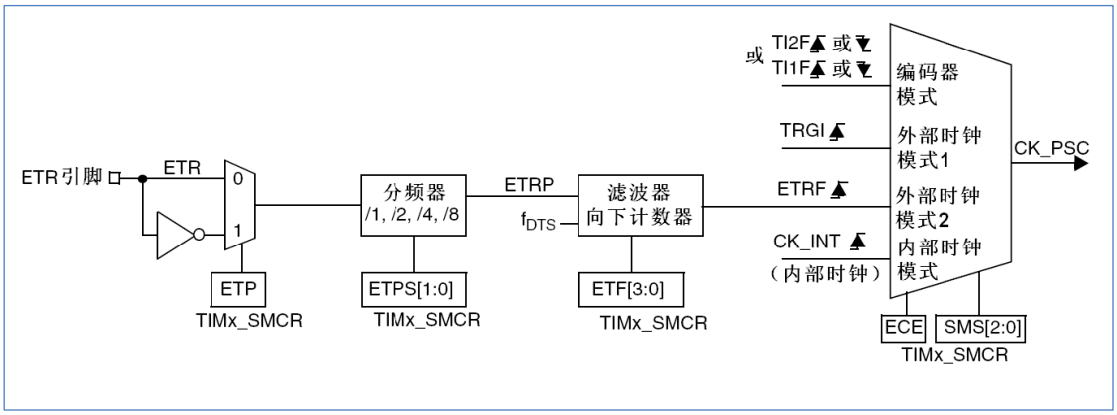


图 12-25 外部触发输入框图

例如，配置在 ETR 上每 2 个上升沿计数一次的向上计数器，使用下列步骤：

1. 置 TIM1_SMCR 寄存器中的 ETF[3:0]=0000。
2. 设置预分频器，置 TIM1_SMCR 寄存器中的 ETPS[1:0]=01。
3. 选择 ETR 的上升沿检测，置 TIM1_SMCR 寄存器中的 ETP=0。
4. 开启外部时钟模式 2，写 TIM1_SMCR 寄存器中的 ECE=1。
5. 启动计数器，写 TIM1_CR1 寄存器中的 CEN=1。

计数器在每 2 个 ETR 上升沿计数一次。在 ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的重新同步电路。

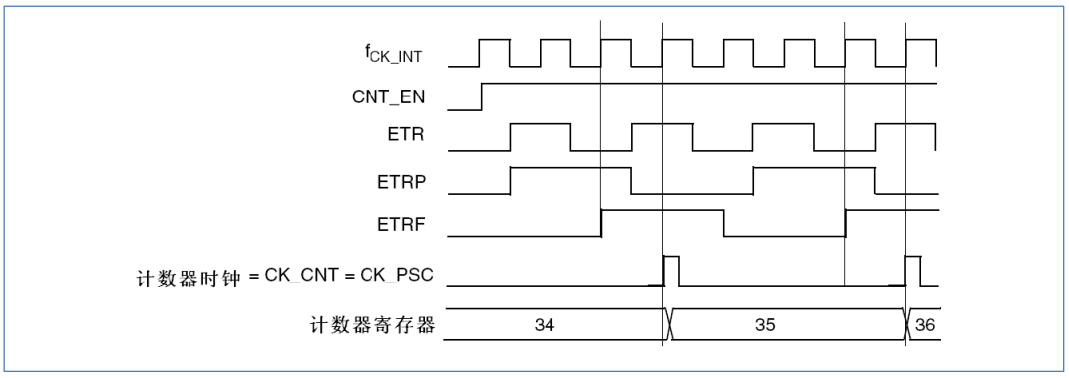


图 12-26 外部时钟模式 2 下的控制电路

12.2.5 捕获/比较通道

每一个捕获/比较通道都是围绕着一个捕获/比较寄存器（包含影子寄存器），包括捕获的输入部分（数字滤波、多路复用和预分频器）和输出部分（比较器和输出控制）。

输入部分对相应的 TIx 输入信号采样，并产生一个滤波后的信号 TIxF。然后，一个带极性选择的边沿检测器产生一个信号（TIxFPx），它可以作为从模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器（ICxPS）。

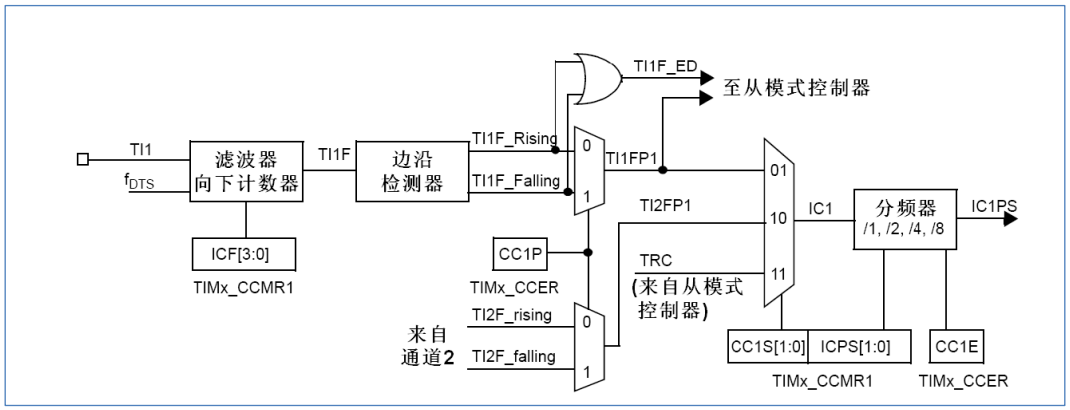


图 12-27 捕获/比较通道（如：通道 1 输入部分）

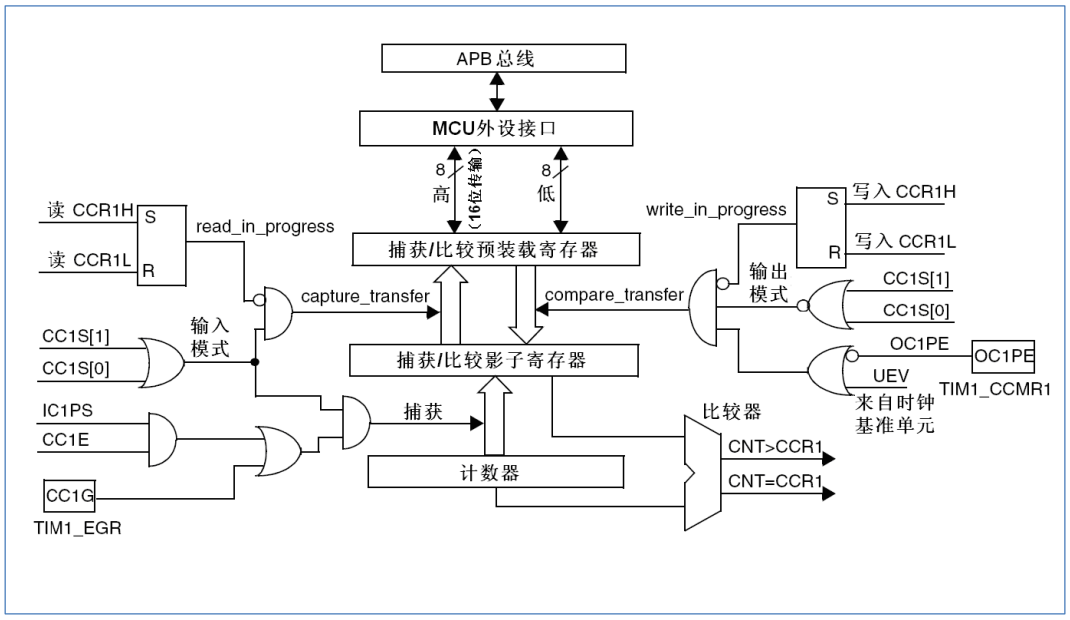


图 12-28 捕获/比较通道 1 的主电路

输出部分产生一个中间波形 OCxRef（高有效）作为基准，链的末端决定最终输出信号的极性。

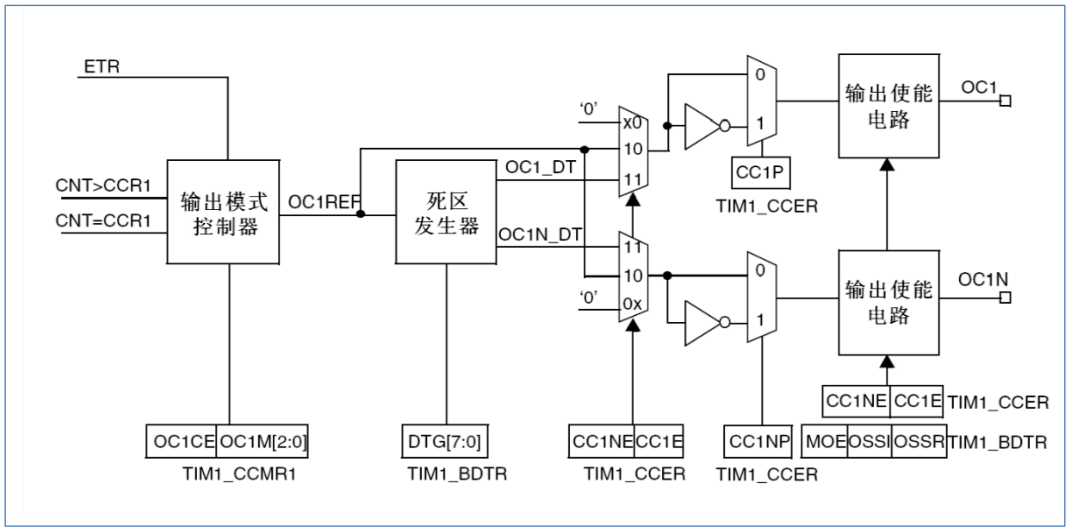


图 12-29 捕获/比较通道的输出部分（通道 1 至 3）

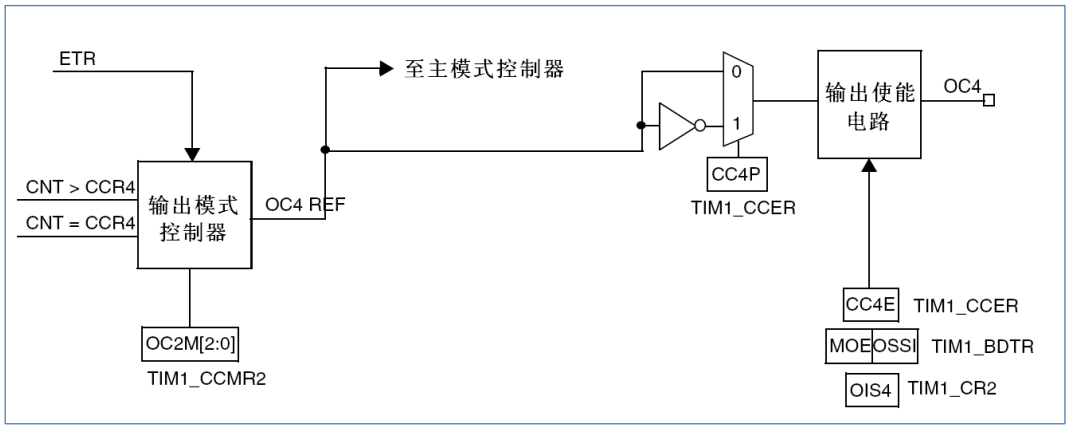


图 12-30 捕获/比较通道的输出部分（通道 4）

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。读写过程仅操作预装载寄存器。

在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

12.2.6 输入捕获模式

在输入捕获模式下，当检测到 ICx 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（TIM1_CCRx）中。当发生捕获事件时，相应的 CCxIF 标志（TIM1_SR 寄存器）被置 1，如果开放了中断，则将产生中断。如果发生捕获事件时 CCxIF 标志已经为高，那么重复捕获标志 CCxOF（TIM1_SR 寄存器）被置 1。写 CCxIF=0 可清除 CCxIF，或读取存储在 TIM1_CCRx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIM1_CCMR1 寄存器中，步骤如下：

1. 选择有效输入端：TIM1_CCR1 必须连接到 TI1 输入，所以写入 TIM1_CCR1 寄存器中的 CC1S=01，只要 CC1S 不为’00’，通道被配置为输入，并且 TIM1_CCR1 寄存器变为只读。
2. 根据输入信号的特点，配置输入滤波器为所需的带宽（即输入为 TIx 时，输入滤波器控制位是 TIM1_CCMRx 寄存器中的 ICxF 位）。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以（以 fDTS 频率）连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIM1_CCMR1 寄存器中写入 IC1F=0011。
3. 选择 TI1 通道的有效转换边沿，在 TIM1_CCER 寄存器中写入 CC1P=0（上升沿）。
4. 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止（写 TIM1_CCMR1 寄存器的 IC1PSC=00）。
5. 设置 TIM1_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
6. 如果需要，通过设置 TIM1_DIER 寄存器中的 CC1IE 位允许相关中断请求。

当发生一个输入捕获时：

- 产生有效的电平转换时，计数器的值被传送到 TIM1_CCR1 寄存器。
- CC1IF 标志被设置（中断标志）。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置 1。
- 如设置了 CC1IE 位，则会产生一个中断。

为了处理捕获溢出，建议在读出捕获溢出标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

说明：设置 TIM1_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断。

12.2.7 PWM 输入模式

该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这 2 个 ICx 信号为边沿有效，但是极性相反。
- 其中一个 TIxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。

例如，你需要测量输入到 TI1 上的 PWM 信号的长度（TIM1_CCR1 寄存器）和占空比（TIM1_CCR2 寄存器），具体步骤如下（取决于 CK_INT 的频率和预分频器的值）：

1. 选择 TIM1_CCR1 的有效输入：置 TIM1_CCMR1 寄存器的 CC1S=01（选中 TI1）。
2. 选择 TI1FP1 的有效极性（用来捕获数据到 TIM1_CCR1 中和清除计数器）：置 CC1P=0（上升沿有效）。
3. 选择 TIM1_CCR2 的有效输入：置 TIM1_CCMR1 寄存器的 CC2S=10（选中 TI1）。
4. 选择 TI1FP2 的有效极性（捕获数据到 TIM1_CCR2）：置 CC2P=1（下降沿有效）。
5. 选择有效的触发输入信号：置 TIM1_SMCR 寄存器中的 TS=101（选择 TI1FP1）。
6. 配置从模式控制器为复位模式：置 TIM1_SMCR 中的 SMS=100。
7. 使能捕获：置 TIM1_CCER 寄存器中 CC1E=1 且 CC2E=1。

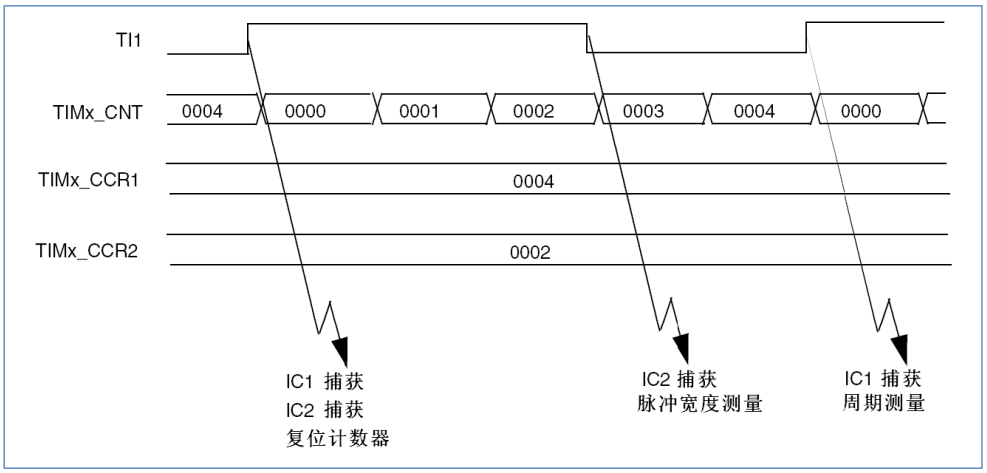


图 12-31 PWM 输入模式时序

因为只有 TI1FP1 和 TI2FP2 连到了从模式控制器，所以 PWM 输入模式只能使用 TIM1_CH1/TIM1_CH2 信号。

12.2.8 强制输出模式

在输出模式（TIM1_CCMRx 寄存器中 CCxS=00）下，输出比较信号（OCxREF 和相应的 OCx/OCxN）能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。置 TIM1_CCMRx 寄存器中相应的 OCxM=101，即可强置输出比较信号（OCxREF/OCx）为有效状态。这样 OCxREF 被强置为高电平（OCxREF 始终为高电平有效），同时 OCx 得到 CCxP 极性相反的信号。

例如：CCxP=0（OCx 高电平有效），则 OCx 被强置为高电平。

置 TIM1_CCMRx 寄存器中的 OCxM=100，可强置 OCxREF 信号为低。

该模式下，在 TIM1_CCRx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因

此仍然会产生相应的中断请求。这将会在下面的输出比较模式一节中介绍。

12.2.9 输出比较模式

此项功能是用来控制一个输出波形，或者指示一段给定的时间已经到时。

当计数器与捕获/比较寄存器的内容相同时，输出比较功能按如下操作：

- 将输出比较模式（TIM1_CCMRx 寄存器中的 OCxM 位）和输出极性（TIM1_CCER 寄存器中的 CCxP 位）定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平（OCxM=000）、被设置成有效电平（OCxM=001）、被设置成无效电平（OCxM=010）或进行翻转（OCxM=011）。
- 设置中断状态寄存器中的标志位（TIM1_SR 寄存器中的 CCxIF 位）。
- 若设置了相应的中断屏蔽（TIM1_DIER 寄存器中的 CCxIE 位），则产生一个中断。

TIM1_CCMRx 中的 OCxPE 位选择 TIM1_CCRx 寄存器是否需要使用预装载寄存器。

在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。

同步的精度可以达到计数器的一个计数周期。输出比较模式（在单脉冲模式下）也能用来输出一个单脉冲。

输出比较模式的配置步骤：

1. 选择计数器时钟（内部、外部、预分频器）。
2. 将相应的数据写入 TIM1_ARR 和 TIM1_CCRx 寄存器中。
3. 如果要产生一个中断请求，设置 CCxIE 位。
4. 选择输出模式，例如：
   A. 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚，设置 OCxM=011。
   B. 置 OCxPE=0 禁用预装载寄存器。
   C. 置 CCxP=0 选择极性为高电平有效。
   D. 置 CCxE=1 使能输出。
5. 设置 TIM1_CR1 寄存器的 CEN 位启动计数器。

TIM1_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器（OCxPE=‘0’，否则 TIM1_CCRx 的影子寄存器只能在发生下一次更新事件时被更新）。下图给出了一个例子。

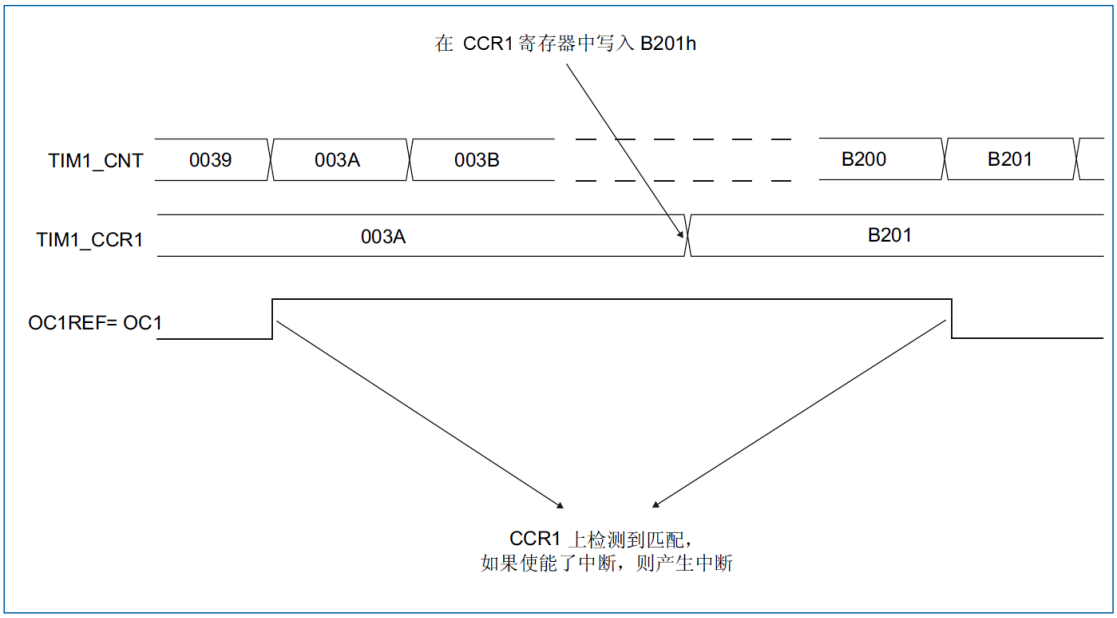


图 12-32 输出比较模式，翻转 OC1

12.2.10 PWM 模式

脉冲宽度调制模式可以产生一个由 TIM1_ARR 寄存器确定频率、由 TIM1_CCRx 寄存器确定占空比的信号。

在 TIM1_CCMRx 寄存器中的 OCxM 位写入’110’（PWM 模式 1）或’111’（PWM 模式 2），能够独立地设置每个 OCx 输出通道产生一路 PWM。必须通过设置 TIM1_CCMRx 寄存器的 OCxPE 位使能相应的预装载寄存器，最后还要设置 TIM1_CR1 寄存器的 ARPE 位，（在向上计数或中心对称模式中）使能自动重装载的预装载寄存器。

仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 TIM1_EGR 寄存器中的 UG 位来初始化所有的寄存器。

OCx 的极性可以通过软件在 TIM1_CCER 寄存器中的 CCxP 位设置，它可以设置为高电平有效或低电平有效。OCx 的输出使能通过 CCxE、CCxNE、MOE、OSSI 和 OSSR 位（TIM1_CCER 和 TIM1_BDTR 寄存器中）的组合控制。参见 TIM1 捕捉/比较使能寄存器（TIM1_CCER）的描述。

在 PWM 模式（模式 1 或模式 2）下，TIM1_CNT 和 TIM1_CCRx 始终在进行比较，（依据计数器的计数方向）以确定是否符合 TIM1_CCRx≤TIM1_CNT 或者 TIM1_CNT≤TIM1_CCRx。

根据 TIM1_CR1 寄存器中 CMS 位的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

12.2.10.1 PWM 边沿对齐模式

- 向上计数配置

当 TIM1_CR1 寄存器中的 DIR 位为低的时候执行向上计数。

下面是一个 PWM 模式 1 的例子。当 TIM1_CNT<TIM1_CCRx 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIM1_CCRx 中的比较值大于自动重装载值（TIM1_ARR），则 OCxREF 保持为’1’。如果比较值为 0，则 OCxREF 保持为’0’。下图为 TIM1_ARR=8 时边沿对齐的 PWM 波形实例。

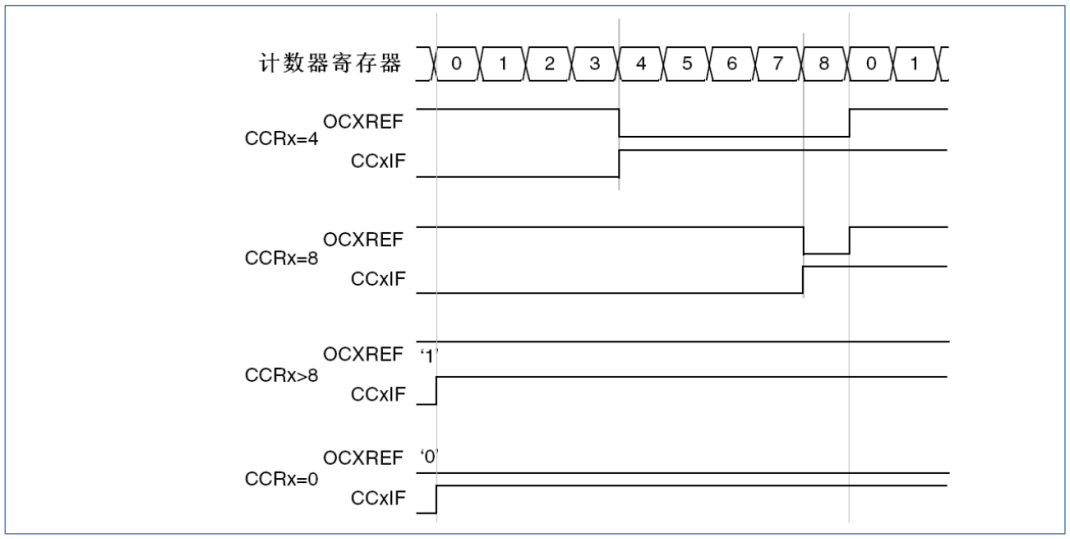


图 12-33 边沿对齐的 PWM 波形（ARR=8）

- 向下计数的配置

当 TIM1_CR1 寄存器的 DIR 位为高时执行向下计数。

在 PWM 模式 1，当 TIM1_CNT>TIM1_CCRx 时参考信号 OCxREF 为低，否则为高。如果 TIM1_CCRx 中的比较值大于 TIM1_ARR 中的自动重装载值，则 OCxREF 保持为'1'。该模式下不能产生 0%的 PWM 波形。

12.2.10.2 PWM 中央对齐模式

当 TIM1_CR1 寄存器中的 CMS 位不为'00'时，为中央对齐模式（所有其他的配置对 OCxREF/OCx 信号都有相同的作用）。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时被置 1、在计数器向下计数时被置 1、或在计数器向上和向下计数时被置 1。TIM1_CR1 寄存器中的计数方向位（DIR）由硬件更新，不要用软件修改它。

下图给出了一些中央对齐的 PWM 波形的例子。

- TIM1_ARR=8
- PWM 模式 1
- TIM1_CR1 寄存器的 CMS=01，在中央对齐模式 1 下，当计数器向下计数时设置比较标志。

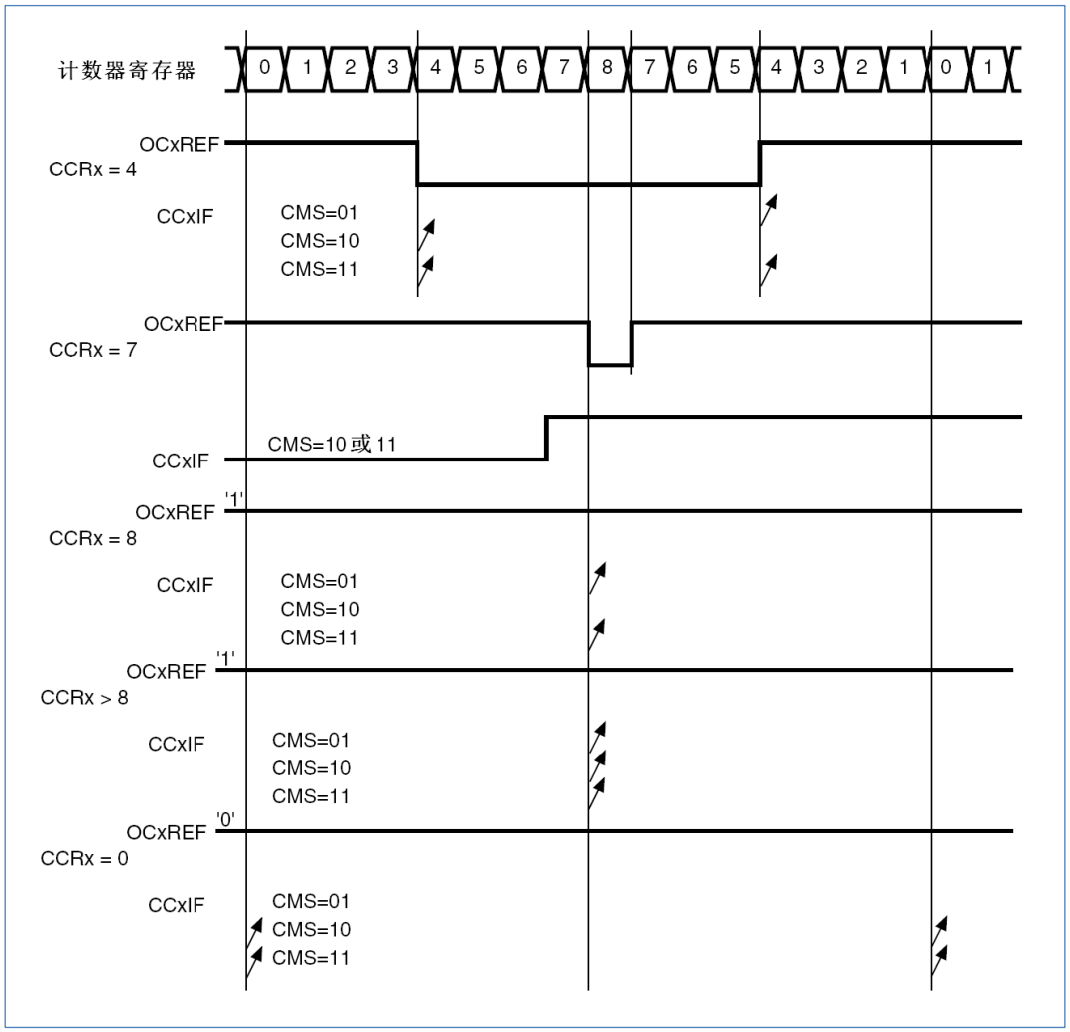


图 12-34 中央对齐的 PWM 波形（APR=8）

使用中央对齐模式的提示：

- 进入中央对齐模式时，使用当前的向上/向下计数配置；这就意味着计数器向上还是向下计数取决于 TIM1_CR1 寄存器中 DIR 位的当前值。此外，软件不能同时修改 DIR 和 CMS 位。
- 不推荐当运行在中央对齐模式时改写计数器，因为这会产生不可预知的结果。特别地：
  - 如果写入计数器的值大于自动重加载的值（TIM1_CNT>TIM1_ARR），则方向不会被更新。例如，如果计数器正在向上计数，它就会继续向上计数。
  - 如果将 0 或者 TIM1_ARR 的值写入计数器，方向被更新，但不产生更新事件 UEV。
- 使用中央对齐模式最保险的方法，就是在启动计数器之前产生一个软件更新（设置 TIM1_EGR 位中的 UG 位），并且不要在计数进行过程中修改计数器的值。

12.2.11 互补输出和死区插入

高级控制定时器（TIM1）能够输出两路互补信号，并且能够管理输出的瞬时关断和接通。

这段时间通常被称为死区，用户应该根据连接的输出器件和它们的特性（电平转换的延时、电源开关的延时等）来调整死区时间。

配置 TIM1_CCER 寄存器中的 CCxP 和 CCxNP 位，可以为每一个输出独立地选择极性（主输出 OCx 或互补输出 OCxN）。

互补信号 OCx 和 OCxN 通过下列控制位的组合进行控制：TIM1_CCER 寄存器的 CCxE 和 CCxNE 位，TIM1_BDTR 和 TIM1_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位，参见表 12-4。特别的是，在转换到 IDLE 状态时（MOE 下降到 0）死区被激活。

同时设置 CCxE 和 CCxNE 位将插入死区，如果存在刹车电路，则还要设置 MOE 位。每一个通道都有一个 10 位的死区发生器。参考信号 OCxREF 可以产生 2 路输出 OCx 和 OCxN。假定 OCx 和 OCxN 输出为高时有效：

- OCx 输出信号与参考信号相同，只是它的上升沿相对于参考信号的上升沿有一个延迟。
- OCxN 输出信号与参考信号相反，只是它的上升沿相对于参考信号的下降沿有一个延迟。

如果延迟大于当前有效的输出宽度（OCx 或者 OCxN），则不会产生相应的脉冲。

下列几张图显示了死区发生器的输出信号和当前参考信号 OCxREF 之间的关系（假设 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1）。

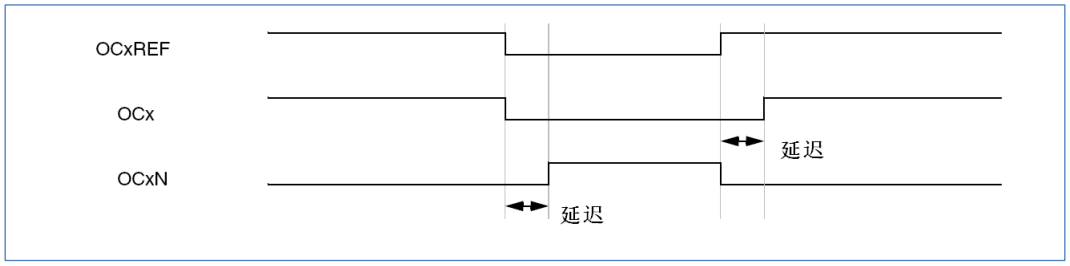


图 12-35 带死区插入的互补输出

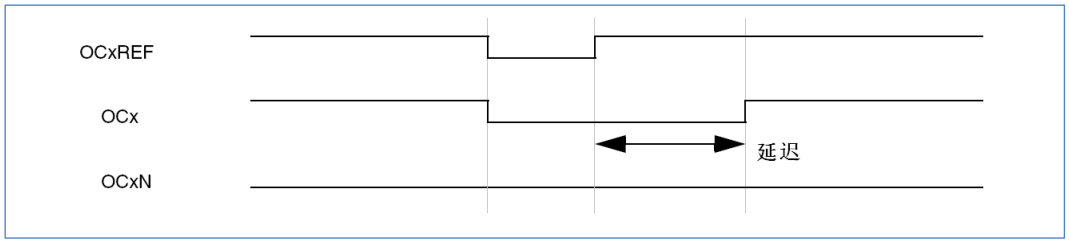


图 12-36 死区波形延迟大于负脉冲

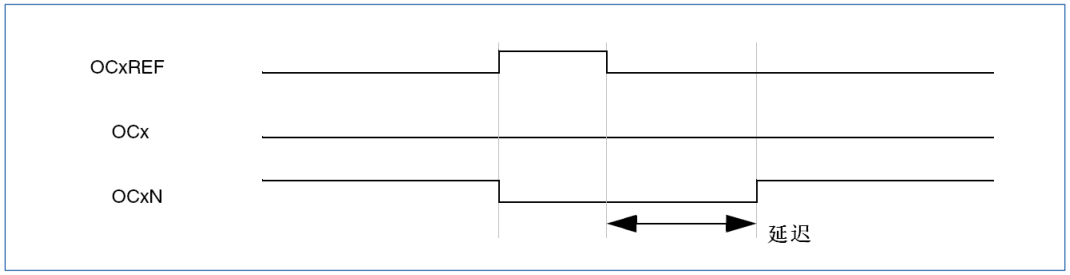


图 12-37 死区波形延迟大于正脉冲

每一个通道的死区延时都是相同的，是由 TIM1_BDTR 寄存器中的 DTG 位编程配置。参见 TIM1 刹车和死区寄存器（TIM1_BDTR）中的延时计算。

重定向 OCxREF 到 OCx 或 OCxN

在输出模式下（强置、输出比较或 PWM），通过配置 TIM1_CCER 寄存器的 CCxE 和 CCxNE 位，OCxREF 可以被重定向到 OCx 或者 OCxN 的输出。

这个功能可以在互补输出处于无效电平时，在某个输出上送出一个特殊的波形（例如 PWM 或者静态有效电平）。另一个作用是，让两个输出同时处于无效电平，或处于有效电平和带死区的互补输出。

说明：当只使能 OCxN（CCxE=0、CCxNE=1）时，它不会反相，当 OCxREF 有效时立即变高。例如，如果 CCxNP=0，则 OCxN=OCxREF。另一方面，当 OCx 和 OCxN 都被使能时（CCxE=CCxNE=1），当 OCxREF 为高时 OCx 有效；而 OCxN 相反，当 OCxREF 低时 OCxN 变为有效。

12.2.12 使用刹车功能

当使用刹车功能时，依据相应的控制位（TIM1_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位，TIM1_CR2 寄存器中的 OISx 和 OISxN 位），输出使能信号和无效电平都会被修改。但无论何时，OCx 和 OCxN 输出不能在同一时间同时处于有效电平上，参见表 12-4。

刹车源既可以是刹车输入引脚又可以是一个时钟失败事件。时钟失败事件由复位时钟控制器中的时钟安全系统产生。

系统复位后，刹车电路被禁止，MOE 位为低。设置 TIM1_BDTR 寄存器中的 BKE 位可以使能刹车功能，刹车输入信号的极性可以通过配置同一个寄存器中的 BKP 位选择。BKE 和 BKP 可以同时被修改。当写入 BKE 和 BKP 位时，在真正写入之前会有 1 个 APB 时钟周期的延迟，因此需要等待一个 APB 时钟周期之后，才能正确地读回写入的位。

因为 MOE 下降沿可以是异步的，在实际信号（作用在输出端）和同步控制位（在 TIM1_BDTR 寄存器中）之间设置了一个再同步电路。这个再同步电路会在异步信号和同步信号之间产生延迟。特别的，如果当它为低时写 MOE=1，则读出它之前必须先插入一个延时（空指令）才能读到正确的值。这是因为写入的是异步信号而读的是同步信号。

当发生刹车时（在刹车输入端出现选定的电平），有下述动作：

- MOE 位被异步地清除，将输出置于无效状态、空闲状态或者复位状态（由 OSSI 位选择）。这个特性在 MCU 的振荡器关闭时依然有效。
- 一旦 MOE=0，每一个输出通道输出由 TIM1_CR2 寄存器中的 OISx 位设定的电平。如果 OSSI=0，则定时器释放使能输出，否则使能输出始终为高。
- 当使用互补输出时：
  - 输出首先被置于复位状态即无效的状态（取决于极性）。这是异步操作，即使定时器没有时钟时，此功能也有效。
  - 如果定时器的时钟依然存在，死区生成器将会重新生效，在死区之后根据 OISx 和 OISxN 位指示的电平驱动输出端口。即使在这种情况下，OCx 和 OCxN 也不能被同时驱动到有效的电平。注意；因为重新同步 MOE，死区时间比通常情况下长一些（大约 2 个 ck_tim 的时钟周期）。
  - 如果 OSSI=0，定时器释放使能输出，否则保持使能输出；或一旦 CCxE 与 CCxNE 之一变高时，使能输出变为高。
- 如果设置了 TIM1_DIER 寄存器中的 BIE 位，当刹车状态标志（TIM1_SR 寄存器中的 BIF 位）为’1’时，则产生一个中断。
- 如果设置了 TIM1_BDTR 寄存器中的 AOE 位，在下一个更新事件 UEV 时 MOE 位被自动置位；例如，这可以用来进行整形。否则，MOE 始终保持低直到被再次置’1’；此时，这个特性可以被用在安全方面，你可以把刹车输入连到电源驱动的报警输出、热敏传感器或者其他安全器件上。

说明：刹车输入为电平有效。所以，当刹车输入有效时，不能同时（自动地或者通过软件）设置 MOE。同时，状态标志 BIF 不能被清除。

刹车由 BRK 输入产生，它的有效极性是可编程的，且由 TIM1_BDTR 寄存器中的 BKE 位开启。除了刹车输入和输出管理，刹车电路中还实现了写保护以保证应用程序的安全。它允许用户冻结几个配置参数（死区长度，OCx/OCxN 极性和被禁止的状态，OCxM 配置，刹车使能和极性）。用户可以通过 TIM1_BDTR 寄存器中的 LOCK 位，从三级保护中选择一种，参见 TIM1 刹车和死区寄存器（TIM1_BDTR）。在 MCU 复位后 LOCK 位只能被修改一次。

下图显示响应刹车的输出实例。

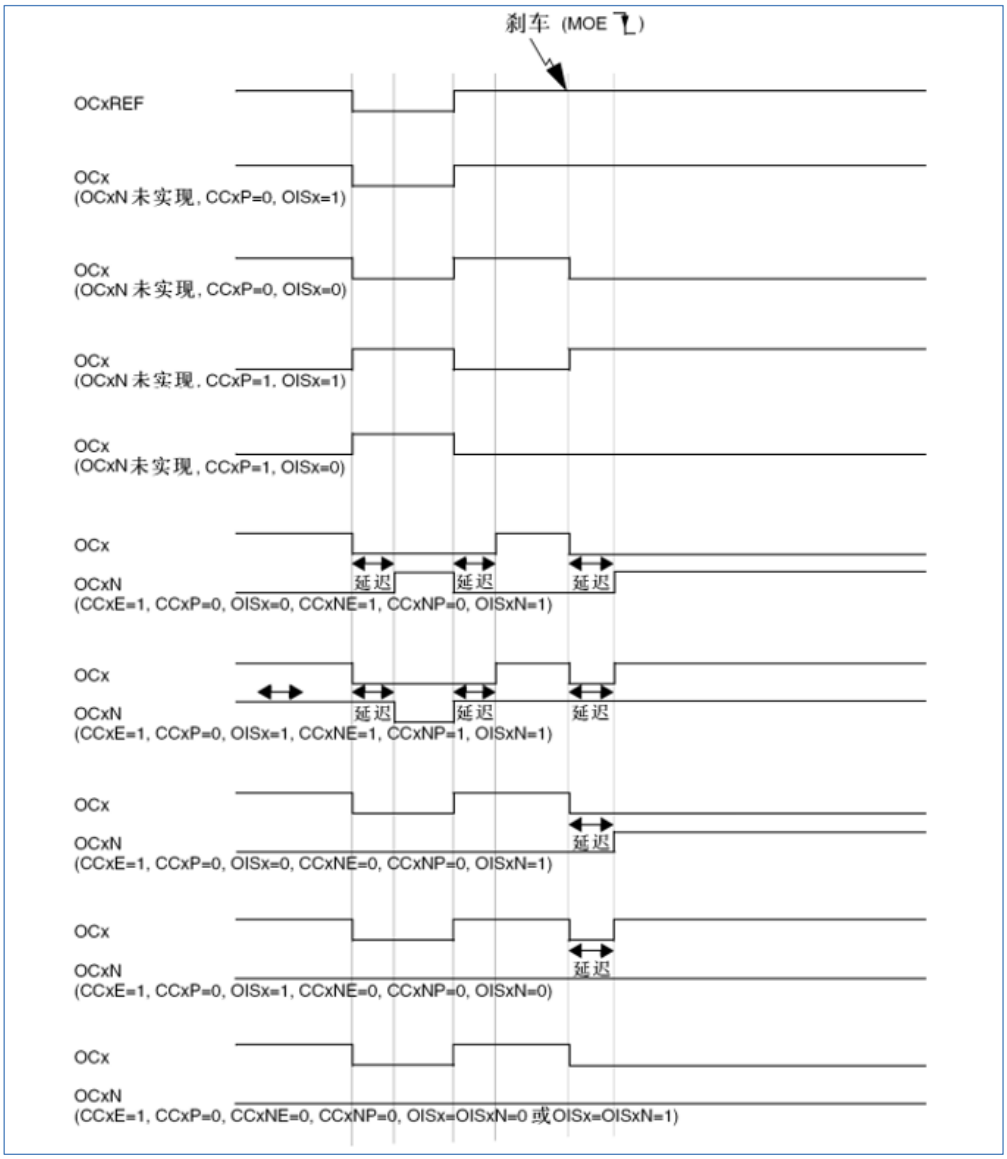


图 12-38 响应刹车的输出

12.2.13 在外部事件时清除 OCxREF 信号

对于一个给定的通道，设置 TIM1_CCMRx 寄存器中对应的 OCxCE 位为'1'，能够用 ETRF 输入端的高电平把 OCxREF 信号拉低，OCxREF 信号将保持为低直到发生下一次的更新事件 UEV。

该功能只能用于输出比较和 PWM 模式，而不能用于强置模式。

例如，OCxREF 信号可以联到一个比较器的输出，用于控制电流。这时，ETR 必须配置如下：

1. 外部触发预分频器必须处于关闭：TIM1_SMCR 寄存器中的 ETPS[1:0]=00。
2. 必须禁止外部时钟模式 2：TIM1_SMCR 寄存器中的 ECE=0。
3. 外部触发极性（ETP）和外部触发滤波器（ETF）可以根据需要配置。

下图显示了当 ETRF 输入变为高时，对应不同 OCxCE 的值，OCxREF 信号的动作。在这个例子中，定时器 TIM1 被置于 PWM 模式。

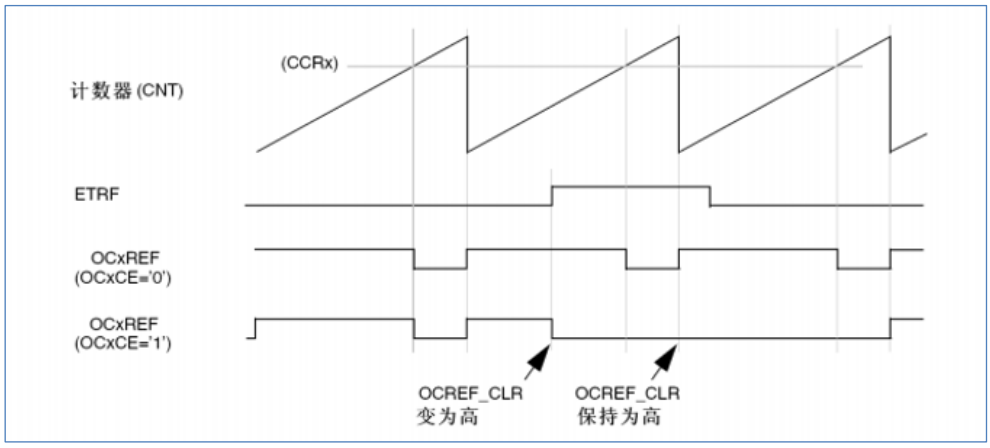


图 12-39 清除 TIM1 的 OCxREF

12.2.14 产生六步 PWM 输出

当在一个通道上需要互补输出时，预装载位有 OCxM、CCxE 和 CCxNE。在发生 COM 换相事件时，这些预装载位被传送到影子寄存器位。这样用户就可以预先设置好下一步配置，并在同一个时刻同时修更改所有通道的配置。COM 可以通过设置 TIM1_EGR 寄存器的 COM 位由软件产生，或在 TRGI 上升沿由硬件产生。

当发生 COM 事件时会设置一个标志位（TIM1_SR 寄存器中的 COMIF 位），这时如果已设置了 TIM1_DIER 寄存器的 COMIE 位，则产生一个中断。

下图显示当发生 COM 事件时，三种不同配置下 OCx 和 OCxN 输出。

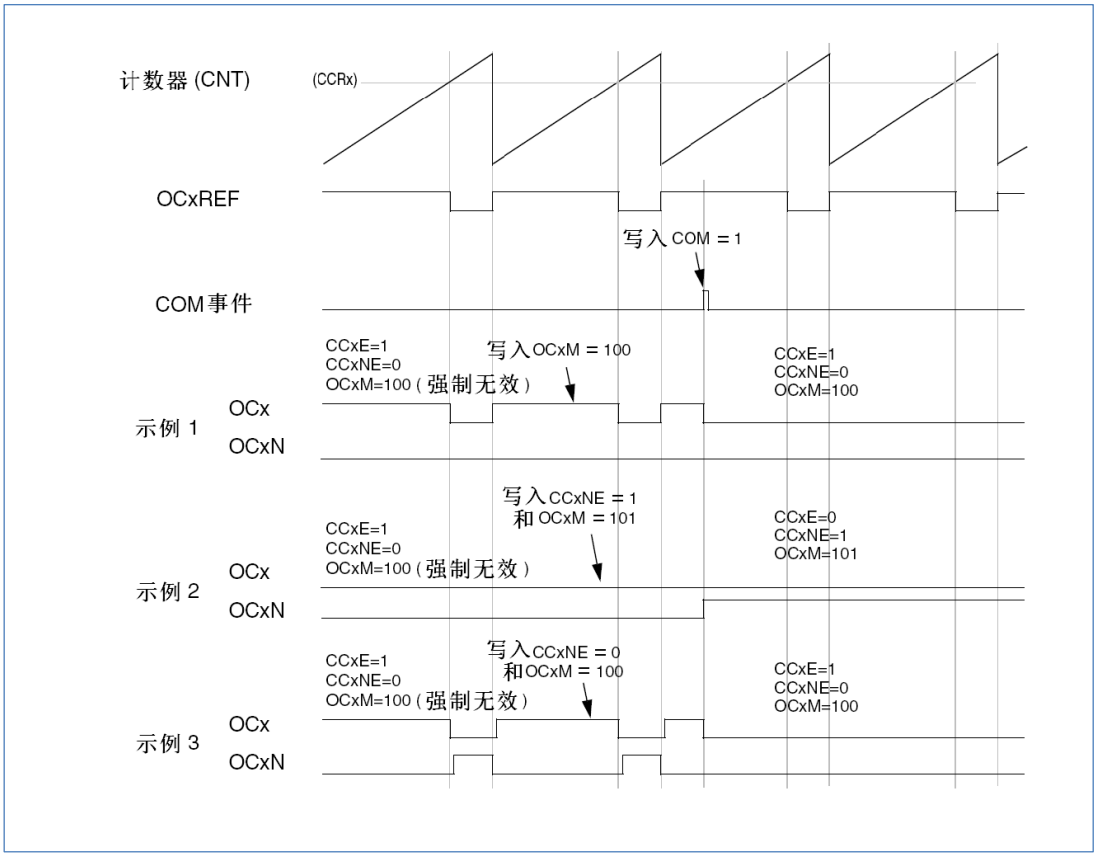


图 12-40 产生六步 PWM，使用 COM 的例子（OSSR=1）

12.2.15 单脉冲模式

单脉冲模式（OPM）是前述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程

序可控的延时之后产生一个脉宽可程序控制的脉冲。

可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIM1_CR1 寄存器中的 OPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件 UEV 时停止。

仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前（当定时器正在等待触发），必须如下配置：

- 向上计数方式：计数器 CNT < CCRx ≤ ARR（特别地 0 < CCRx）；
- 向下计数方式：计数器 CNT > CCRx。

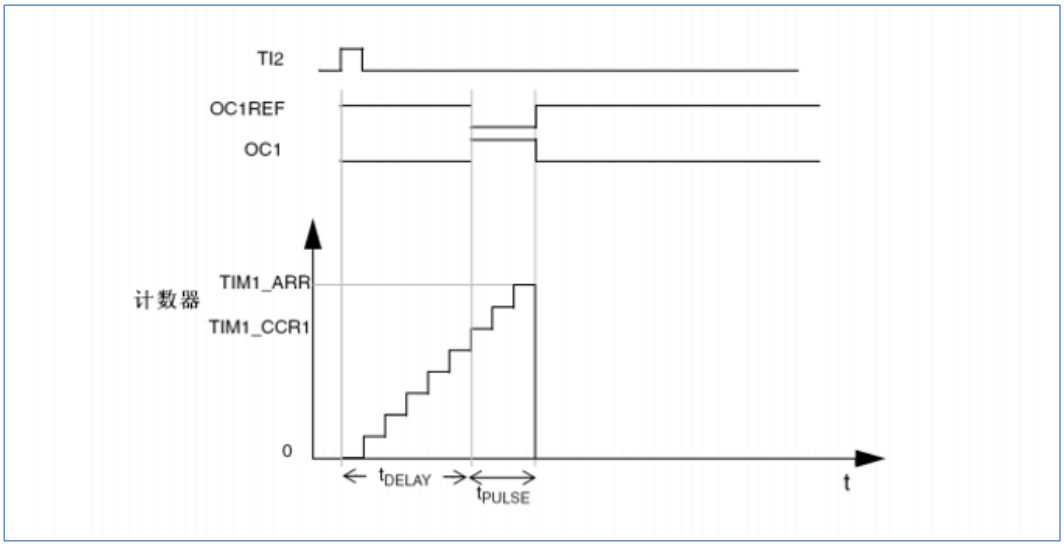


图 12-41 单脉冲模式的例子

例如，你需要在从 TI2 输入脚上检测到一个上升沿开始，延迟 t_{DELAY} 之后，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

假定 TI2FP2 作为触发 1：

- 置 TIM1_CCMR1 寄存器中的 CC2S=01，把 TI2FP2 映射到 TI2。
- 置 TIM1_CCER 寄存器中的 CC2P=0，使 TI2FP2 能够检测上升沿。
- 置 TIM1_SMCR 寄存器中的 TS=110，TI2FP2 作为从模式控制器的触发（TRGI）。
- 置 TIM1_SMCR 寄存器中的 SMS=110（触发模式），TI2FP2 被用来启动计数器。

OPM 的波形由写入比较寄存器的数值决定（要考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由 TIM1_CCR1 寄存器中的值定义。
- t_{PULSE} 由自动装载值和比较值之间的差值定义（TIM1_ARR-TIM1_CCR1）。
- 假定当发生比较匹配时要产生从 0 到 1 的波形，当计数器达到预装载值时要产生一个从 1 到 0 的波形：首先要置 TIM1_CCMR1 寄存器的 OC1M=111，进入 PWM 模式 2；根据需要有选择地使能预装载寄存器：置 TIM1_CCMR1 中的 OC1PE=1 和 TIM1_CR1 寄存器中的 ARPE；然后在 TIM1_CCR1 寄存器中填写比较值，在 TIM1_ARR 寄存器中填写自动装载值，设置 UG 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P=0。

在这个例子中，TIM1_CR1 寄存器中的 DIR 和 CMS 位应该置低。

因为只需要一个脉冲，所以必须设置 TIM1_CR1 寄存器中的 OPM=1，在下一个更新事件（当计数器从自动装载值翻转到 0）时停止计数。

特殊情况：OCx 快速使能：

在单脉冲模式下，在 TIx 输入脚的边沿检测逻辑设置 CEN 位以启动计数器。然后计数器和比较值间的比较操作产生了输出的转换。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时

t_{DELAY}。

如果要以最小延时输出波形，可以设置 TIM1_CCMRx 寄存器中的 OCxFE 位；此时 OCxREF（和 OCx）直接响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCxFE 只在通道配置为 PWM1 和 PWM2 模式时起作用。

12.2.16 编码器接口模式

选择编码器接口模式的方法是：如果计数器只在 TI2 的边沿计数，则置 TIM1_SMCR 寄存器中的 SMS=001；如果只在 TI1 边沿计数，则置 SMS=010；如果计数器同时在 TI1 和 TI2 边沿计数，则置 SMS=011。

通过设置 TIM1_CCER 寄存器中的 CC1P 和 CC2P 位，可以选择 TI1 和 TI2 极性；如果需要，还可以对输入滤波器编程。

两个输入 TI1 和 TI2 被用来作为增量编码器的接口。参看表 12-2，假定计数器已经启动（TIM1_CR1 寄存器中的 CEN=1），则计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和变相，则 TI1FP1=TI1，TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对 TIM1_CR1 寄存器的 DIR 位进行相应的设置。不管计数器是依靠 TI1 计数、依靠 TI2 计数或者同时依靠 TI1 和 TI2 计数，在任一输入端（TI1 或者 TI2）的跳变都会重新计算 DIR 位。

编码器接口模式基本上相当于使用了一个带有方向选择的外部时钟。这意味着计数器只在 0 到 TIM1_ARR 寄存器的自动装载值之间连续计数（根据方向，或是 0 到 ARR 计数，或是 ARR 到 0 计数）。所以在开始计数之前必须配置 TIM1_ARR；同样，捕获器、比较器、预分频器、重复计数器、触发输出特性等仍工作如常。编码器模式和外部时钟模式 2 不兼容，因此不能同时操作。

在这个模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的组合，假设 TI1 和 TI2 不同时变换。

表 12-2 计数方向与编码器信号的关系

有效边沿	相对信号的电平（TI1FP1 对应 TI2、TI2FP2 对应 TI1）	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
仅在 TI1 计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在 TI2 计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在 TI1 和 TI2 上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

一个外部的增量编码器可以直接与 MCU 连接而不需要外部接口逻辑。但是，一般会使用比较器将编码器的差分输出转换到数字信号，这大大增加了抗噪声干扰能力。编码器输出的第三个信号表示机械零点，可以把它连接到一个外部中断输入并触发一个计数器复位。

下图是一个计数器操作的实例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。在这个例子中，我们假定配置如下：

- CC1S=’01’（TIM1_CCMR1 寄存器，IC1FP1 映射到 TI1）
- CC2S=’01’（TIM1_CCMR2 寄存器，IC2FP2 映射到 TI2）
- CC1P=‘0’（TIM1_CCER 寄存器，IC1FP1 不反相，IC1FP1=TI1）
- CC2P=‘0’（TIM1_CCER 寄存器，IC2FP2 不反相，IC2FP2=TI2）
- SMS=’011’（TIM1_SMCR 寄存器，所有的输入均在上升沿和下降沿有效）
- CEN=‘1’（TIM1_CR1 寄存器，计数器使能）

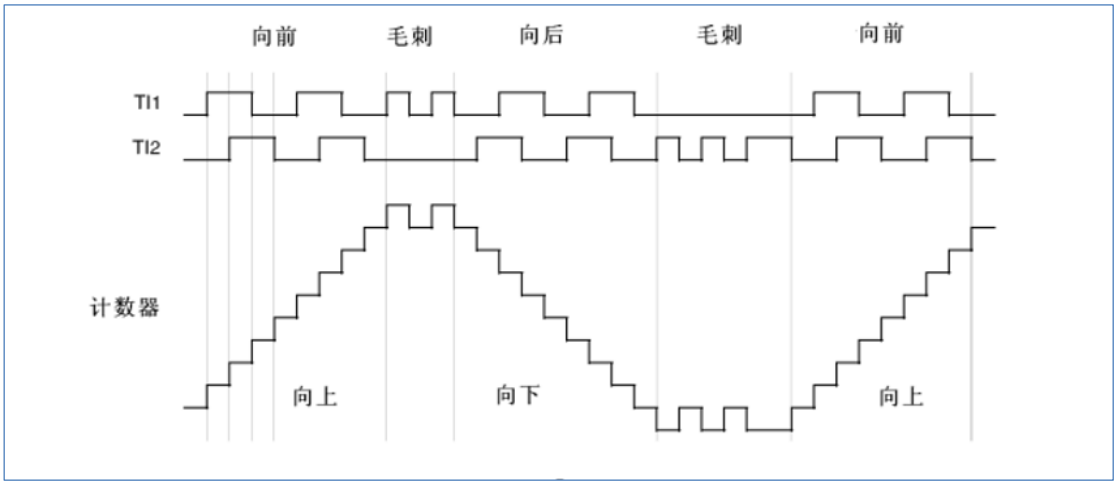


图 12-42 编码器模式下的计数器操作实例

下图为当 IC1FP1 极性反相时计数器的操作实例（CC1P=‘1’，其他配置与上例相同）。

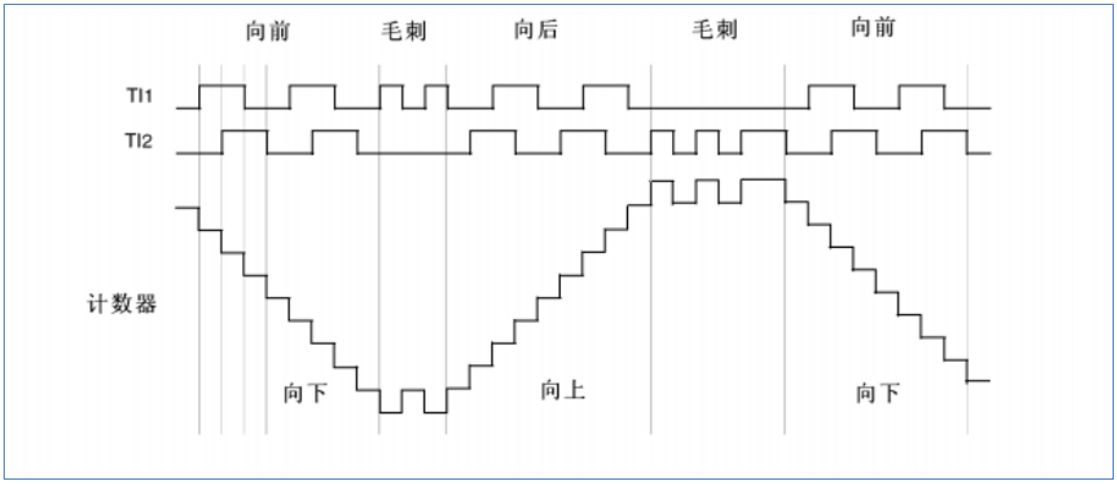


图 12-43 IC1FP1 反相的编码器接口模式实例

当定时器配置成编码器接口模式时，提供传感器当前位置的信息。使用第二个配置在捕获模式的定时器，可以测量两个编码器事件的间隔，获得动态的信息（速度，加速度，减速度）。指示机械零点的编码器输出可被用做此目的。根据两个事件间的间隔，可以按照固定的时间读出计数器。如果可能的话，用户可以把计数器的值锁存到第三个输入捕获寄存器（捕获信号必须是周期的并且可以由另一个定时器产生）。

12.2.17 定时器输入异或功能

TIM1_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIM1_CH1、TIM1_CH2 和 TIM1_CH3。

异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。下节给出了此特性用于连接霍尔传感器的例子。

12.2.18 与霍尔传感器的接口

使用高级控制定时器 TIM1 产生 PWM 信号驱动马达时，可以用另一个通用 TIM2 定时器作为“接口定时器”来连接霍尔传感器，见图 12-44。3 个定时器输入脚（CC1、CC2、CC3）通过一个异或门连接到 TI1 输入通道（通过设置 TIM1_CR2 寄存器中的 TI1S 位来选择），“接口定时器”捕获这个信号。

从模式控制器被配置于复位模式，从输入是 TI1F_ED。每当 3 个输入之一变化时，计数器重新从 0 开始计数。这样产生一个由霍尔输入端的任何变化而触发的时间基准。

“接口定时器”上的捕获/比较通道 1 配置为捕获模式，捕获信号为 TRC（图 12-27）。捕获值反映了两个输入变化间的时间延迟，给出了马达速度的信息。

“接口定时器”可以用来在输出模式产生一个脉冲，这个脉冲可以（通过触发一个 COM 事件）用于改变高级定时器 TIM1 各个通道的属性，而高级控制定时器产生 PWM 信号驱动马达。因此“接口定时器”通道必须编程为在一个指定的延时（输出比较或 PWM 模式）之后产生一个正脉冲，这个脉冲通过 TRGO 输出被送到高级控制定时器 TIM1。

举例：霍尔输入连接到 TIM1 定时器，要求每次任一霍尔输入上发生变化之后的一个指定的时刻，改变高级控制定时器 TIM1 的 PWM 配置。

- 置 TIM1_CR2 寄存器的 TI1S 位为’1’，配置三个定时器输入逻辑或到 TI1 输入。
- 时基编程：置 TIM1_ARR 为其最大值（计数器必须通过 TI1 的变化清零）。设置预分频器得到一个最大的计数器周期，它长于传感器上的两次变化的时间间隔。
- 设置通道 1 为捕获模式（选中 TRC）：置 TIM1_CCMR1 寄存器中 CC1S=01，如果需要，还可以设置数字滤波器。
- 设置通道 2 为 PWM2 模式，并具有要求的延时：置 TIM1_CCMR1 寄存器中的 OC2M=111 和 CC2S=00。
- 选择 OC2REF 作为 TRGO 上的触发输出：置 TIM1_CR2 寄存器中的 MMS=101。

在高级控制寄存器 TIM1 中，正确的 ITR 输入必须是触发器输入，定时器被编程为产生 PWM 信号，捕获/比较控制信号为预装载的（TIM1_CR2 寄存器中 CCPC=1），同时触发输入控制 COM 事件（TIM1_CR2 寄存器中 CCUS=1）。在一次 COM 事件后，写入下一步的 PWM 控制位（CCxE、OCxM），这可以在处理 OC2REF 上升沿的中断子程序里实现。

下图显示了这个实例。

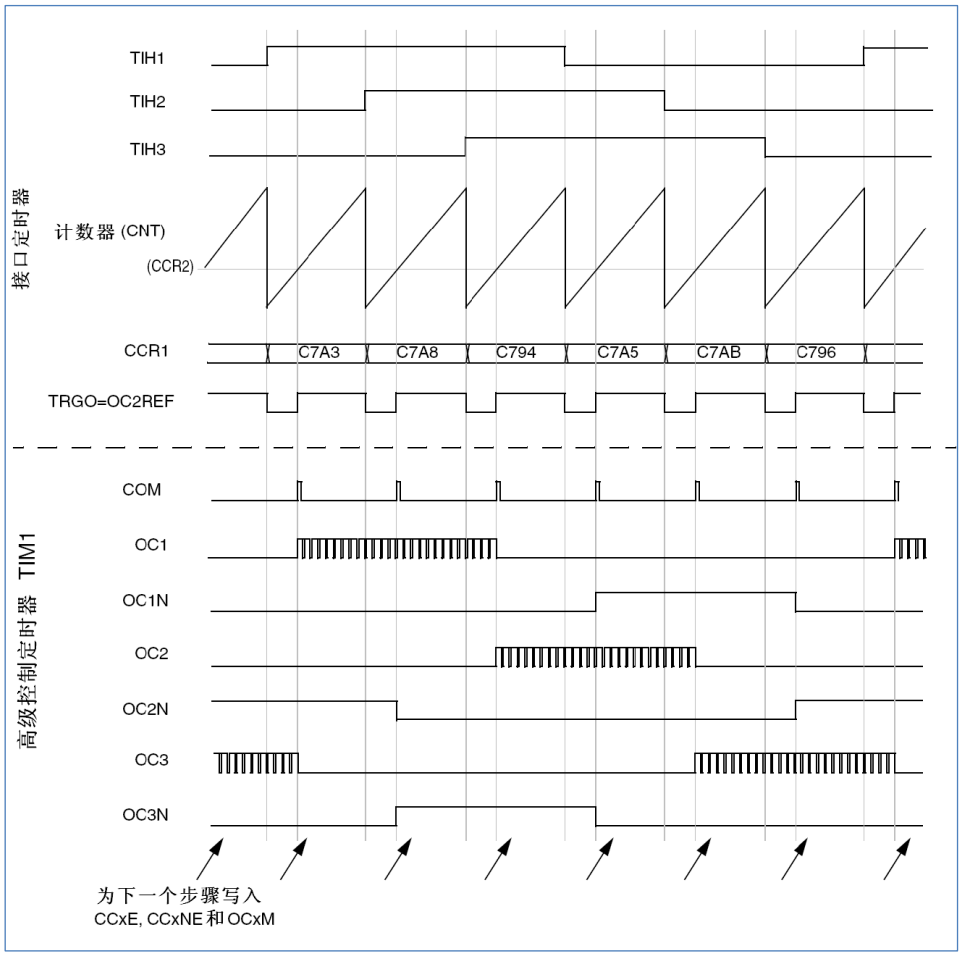


图 12-44 霍尔传感器接口的实例

12.2.19 TIM1 定时器和外部触发的同步

TIM1 定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

12.2.19.1 从模式：复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIM1_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器（TIM1_ARR，TIM1_CCRx）都被更新了。

在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽（在本例中，不需要任何滤波器，因此保持 IC1F=0000）。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位只选择输入捕获源，即 TIM1_CCMR1 寄存器中 CC1S=01。置 TIM1_CCER 寄存器中 CC1P=0 以确定极性（只检测上升沿）。
- 置 TIM1_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIM1_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIM1_CR1 寄存器中 CEN=1，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志（TIM1_SR 寄存器中的 TIF 位）被设置，根据 TIM1_DIER 寄存器中 TIE（中断使能）位的设置，产生一个中断请求。

下图显示当自动重装载寄存器 TIM1_ARR=0x36 时的动作。在 TI1 上升沿和计数器的实际复位之间的延时取决于 TI1 输入端的重同步电路。

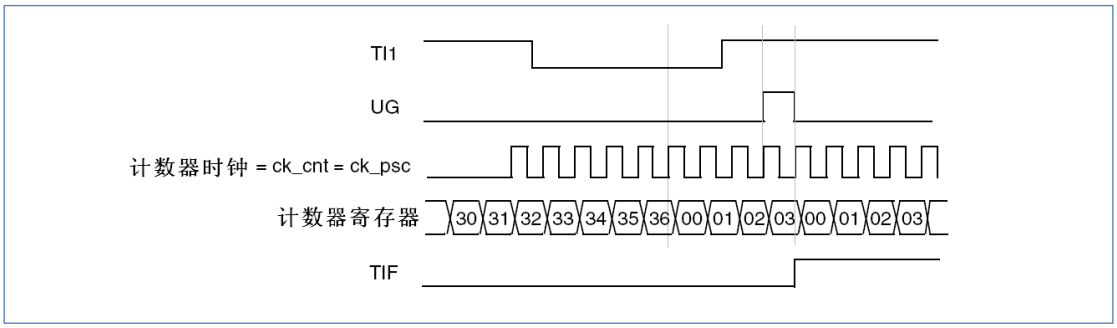


图 12-45 复位模式下的控制电路

12.2.19.2 从模式：门控模式

按照选中的输入端电平使能计数器。

在如下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽（本例中，不需要滤波，所以保持 IC1F=0000）。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIM1_CCMR1 寄存器中 CC1S=01。置 TIM1_CCER 寄存器中 CC1P=1 以确定极性（只检测低电平）。
- 置 TIM1_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIM1_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIM1_CR1 寄存器中 CEN=1，启动计数器。在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何。

只要 TI1 为低，计数器开始依据内部时钟计数，一旦 TI1 变高则停止计数。当计数器开始或停止时都设置 TIM1_SR 中的 TIF 标置。

TI1 上升沿和计数器实际停止之间的延时取决于 TI1 输入端的重同步电路。

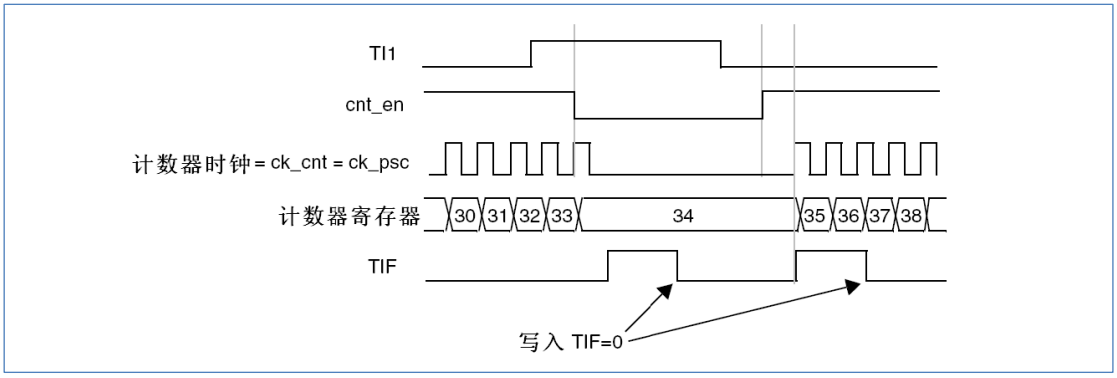


图 12-46 门控模式下的控制电路

12.2.19.3 从模式：触发模式

输入端上选中的事件使能计数器。

在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽（本例中，不需要任何滤波器，保持 IC2F=0000）。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIM1_CCMR1 寄存器中 CC2S=01。置 TIM1_CCER 寄存器中 CC2P=1 以确定极性（只检测低电平）。
- 置 TIM1_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIM1_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

当 TI2 出现一个上升沿时，计数器开始在内部时钟驱动下计数，同时设置 TIF 标志。

TI2 上升沿和计数器启动计数之间的延时，取决于 TI2 输入端的重同步电路。

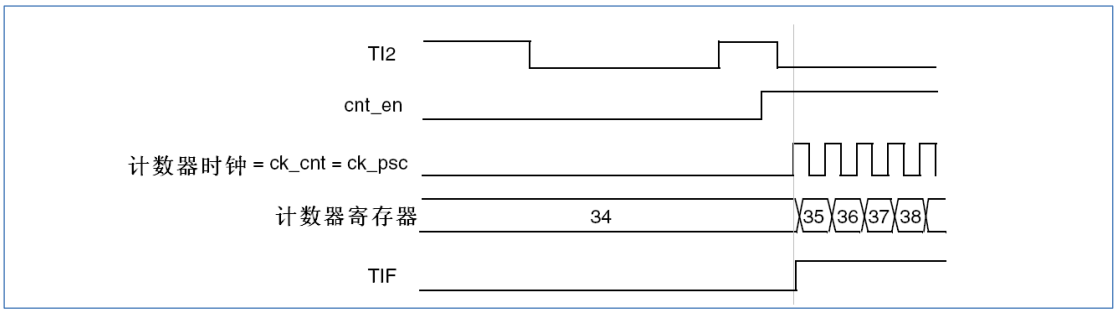


图 12-47 触发器模式下的控制电路

12.2.19.4 从模式：外部时钟模式 2＋触发模式

外部时钟模式 2 可以与另一种从模式（外部时钟模式 1 和编码器模式除外）一起使用。这时，ETR 信号被用作外部时钟的输入，在复位模式、门控模式或触发模式可以选择另一个输入作为触发输入。不建议使用 TIM1_SMCR 寄存器的 TS 位选择 ETR 作为 TRGI。

在下面的例子中，一旦在 TI1 上出现一个上升沿，计数器即在 ETR 的每一个上升沿向上计数一次：

1. 通过 TIM1_SMCR 寄存器配置外部触发输入电路：
   - ETF=0000：没有滤波
   - ETPS=00：不用预分频器
   - ETP=0：检测 ETR 的上升沿，置 ECE=1 使能外部时钟模式 2。
2. 按如下配置通道 1，检测 TI 的上升沿：
   - IC1F=0000：没有滤波
   - 触发操作中不使用捕获预分频器，不需要配置。
   - 置 TIM1_CCMR1 寄存器中 CC1S=01，选择输入捕获源。
   - 置 TIM1_CCER 寄存器中 CC1P=0 以确定极性（只检测上升沿）。
3. 置 TIM1_SMCR 寄存器中 SMS=110，配置定时器为触发模式。置 TIM1_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。

当 TI1 上出现一个上升沿时，TIF 标志被设置，计数器开始在 ETR 的上升沿计数。

ETR 信号的上升沿和计数器实际复位间的延时，取决于 ETRP 输入端的重同步电路。

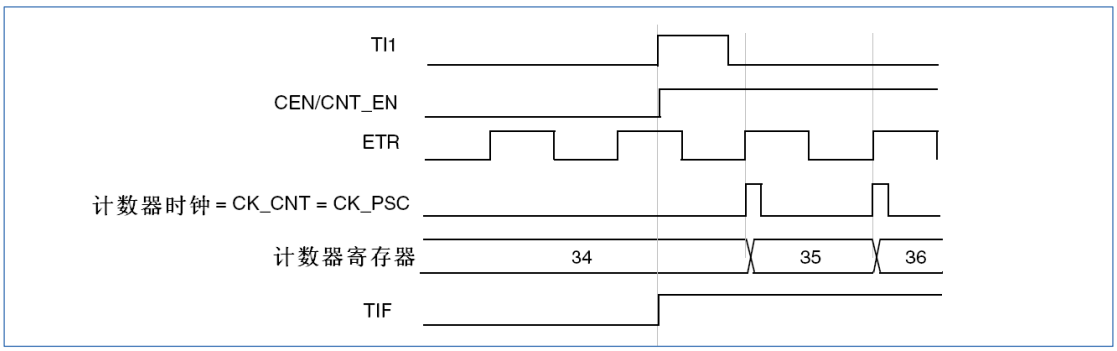


图 12-48 外部时钟模式 2＋触发模式下的控制电路

12.2.20 定时器同步

所有 TIM 定时器在内部相连，用于定时器同步或链接。参见 13.2.15 定时器同步。

12.2.21 调试模式

当 MCU 进入调试模式时（Cortex-M0 内核停止），根据 DBG 模块中 DBG_TIM1_Stop 的设置，TIM1 计数器可以或者继续正常操作，或者停止。参见“23.8.2 对定时器、看门狗和 I2C 的调试支持”。

12.3 TIM1 寄存器

基地址：0x4001 2C00

空间大小：0x400

12.3.1 TIM1 控制存器 1（TIM1_CR1）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CC4_ADC_SEL	Res														
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res						CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

位 31	CC4_ADC_SEL：ADC 触发信号产生源 • 0：默认值；ADC 的 CC4 触发信号由端口 CH4 信号产生。 • 1：ADC 的 CC4 触发信号改由定时器内部 OC4REF 信号产生。
位 30:10	Res：保留 必须保持复位值。
位 9:8	CKD[1:0]：时钟分频因子 该位域定义了定时器时钟频率（CK_INT）与死区发生器和数字滤波器（ETR, TIx）所使用的采样时钟（t_{DTS}）之间的分频比例。 • 00：$t_{DTS} = t_{CK\_INT}$ • 01：$t_{DTS} = 2 \times t_{CK\_INT}$ • 10：$t_{DTS} = 4 \times t_{CK\_INT}$ • 11：保留，不使用该配置
位 7	ARPE：自动重装载预装载允许 • 0：TIMx_ARR 寄存器没有缓冲 • 1：TIMx_ARR 寄存器有缓冲
位 6:5	CMS[1:0]：选择中央对齐模式 • 00：边沿对齐模式 计数器依据方向位（DIR）向上或向下计数。 • 01：中央对齐模式1 计数器交替地向上、向下计数。配置为输出通道的输出比较中断标志位，仅在计数器向下计数时设置。配置 TIMx_CCMRx 寄存器的 CCxS=00，则为输出通道。

	• 10：中央对齐模式2 计数器交替地向上、向下计数。配置为输出通道的输出比较中断标志，仅在计数器向上计数时设置。 • 11：中央对齐模式3 计数器交替地向上、向下计数。配置为输出通道的输出比较中断标志位，在计数器向上和向下计数时均可设置。
位 4	DIR：方向 • 0：计数器向上计数 • 1：计数器向下计数
位 3	OPM：单脉冲模式 • 0：在发生更新事件时，计数器不停止 • 1：在发生下一次更新事件（清除 CEN 位）时，计数器停止。
位 2	URS：更新请求源 软件通过该位选择更新事件（UEV）的源。 • 0：如果使能了更新中断请求，则下述任一事件可产生更新中断请求： o 计数器溢出或下溢 o 设置 UG 位 o 从模式控制器产生的更新 • 1：如果使能了更新中断请求，则只有计数器溢出/下溢才产生更新中断请求。
位 1	UDIS：禁止更新 软件通过该位允许/禁止 UEV 的产生。 • 0：允许 UEV 更新（UEV）事件由下述任一事件产生： o 计数器溢出/下溢 o 设置 UG 位 o 从模式控制器产生的更新，具有缓存的寄存器被装入它们的预装载值。 • 1：禁止 UEV 不产生更新事件，影子寄存器（ARR、PSC、CCRx）保持原来的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。
位 0	CEN：使能计数器 • 0：禁止计数器 • 1：使能计数器

12.3.2 TIM1 控制寄存器 2（TIM1_CR2）

偏移地址：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			Res	CCUS	Res	CCPC

	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw		rw

位 15	Res：保留 必须保持复位值。
位 14	OIS4：输出空闲状态4（OC4 输出）
位 13	OIS3N：输出空闲状态3（OC3N 输出）
位 12	OIS3：输出空闲状态3（OC3 输出）
位 11	OIS2N：输出空闲状态2（OC2N 输出）
位 10	OIS2：输出空闲状态2（OC2 输出）
位 9	OIS1N：输出空闲状态1（OC1N 输出） 当MOE=0 时，OIS1N 为： • 0：死区后OC1N=0 • 1：死区后OC1N=1
位 8	OIS1：输出空闲状态1（OC1 输出） 当 MOE=0 时，如果完成了 OC1N，OIS1 为： • 0：死区后 OC1=0 • 1：死区后 OC1=1
位 7	TI1S：TI1 选择 • 0：TIMx_CH1 引脚连到TI1 输入。 • 1：TIMx_CH1、TIMx_CH2 和TIMx_CH3 引脚经异或运算后连到TI1 输入。
位 6:4	MMS[2:0]：主模式选择 该位域用于选择在主模式下送到从定时器的同步信息（TRGO）。 • 000：复位 TIMx_EGR 寄存器的UG 位被作为触发输出（TRGO）。如果是触发输入产生的复位（模式控制器处于复位模式），则TRGO 上的信号相对实际的复位会有延迟。 • 001：使能 计数器使能信号CNT_EN 被作为触发输出（TRGO），用于同时启动多个定时器或控制在一段时间内使能从定时器。在门控模式下，计数器使能信号是 CEN 控制位和的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式。 • 010：更新 更新事件被选为触发输入（TRGO）。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 • 011：比较脉冲 在发生一次捕获或一次比较成功时，当要设置CC1IF 标志（即使它为高），触发输出送出一个正脉冲（TRGO）。

	• 100：比较 OC1REF 信号被用于作为触发输出（TRGO）。 • 101：比较 OC2REF 信号被用于作为触发输出（TRGO）。 • 110：比较 OC3REF 信号被用于作为触发输出（TRGO）。 • 111：比较 OC4REF 信号被用于作为触发输出（TRGO）。
位 3	Res：保留 必须保持复位值。
位 2	CCUS：捕获/比较控制更新选择 • 0：如果捕获/比较控制位是预装载的（CCPC=1），只能通过设置COMG 位更新捕获比较寄存器。 • 1：如果捕获/比较控制位是预装载的（CCPC=1），可以通过设置COMG 位或TRGI 上的一个上升沿来更新捕获比较寄存器。
位 1	Res：保留 必须保持复位值。
位 0	CCPC：捕获/比较预装载控制位 • 0：CCxE、CCxNE 和 OCxM 位不是预装载的。 • 1：CCxE、CCxNE 和 OCxM 位是预装载的。设置该位后，只能在发生捕获/比较更新事件（COMG 设置或 TRGI 是检测到上升沿，取决于 CCUS 位）时被更新。

12.3.3 TIM1 从模式控制寄存器（TIM1_SMCR）

偏移地址：0x08

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			OCCS	SMS[2:0]		
rw	rw	rw		rw				rw	rw			rw	rw		

位 15	ETP：外部触发极性 该位选择 ETR 还是 ETR 的反相来作为触发操作。 • 0：ETR 不反相，高电平或上升沿有效。 • 1：ETR 被反相，低电平或下降沿有效。
位 14	ECE：外部时钟使能位 该位启用外部时钟模式 2。 • 0：禁止外部时钟模式 2 • 1：使能外部时钟模式 2，计数器由 ETRF 信号上的任意有效边沿驱动。

位 13:12	ETPS[1:0]：外部触发预分频 外部触发信号 ETRP 的频率最高是 TIMxCLK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETRP 的频率。 • 00：关闭预分频 • 01：ETRP 频率/2 • 10：ETRP 频率/4 • 11：ETRP 频率/8
位 11:8	ETF[3:0]：外部触发滤波 该位域定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。 • 0000：无滤波器，以 $f_{SAMPLING}=f_{DTS}$ 采样 • 0001：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=2 • 0010：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=4 • 0011：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=8 • 0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6 • 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 • 0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6 • 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 • 1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6 • 1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8 • 1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5 • 1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6 • 1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8 • 1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5 • 1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6 • 1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8
位 7	MSM：主/从模式 • 0：无作用 • 1：触发输入（TRGI）上的事件被延迟，以允许在当前定时器与它的从定时器间的同步（通过 TRGO）。这对于要求把几个定时器同步到一个单一的外部事件时是非常有用的。
位 6:4	TS[2:0]：触发选择 该位域选择同步计数器的触发输入。 • 000：内部触发 0（ITR0） • 001：内部触发 1（ITR1） • 010：内部触发 2（ITR2） • 011：内部触发 3（ITR3） • 100：TI1 的边沿检测器（TI1F_ED） • 101：滤波后的定时器输入 1（TI1FP1）

	• 110：滤波后的定时器输入 2（TI2FP2） • 111：外部触发输入（ETRF）
位 3	OCCS：OCREF 清除选择 该位选择 OCREF 清除的源。 • 0：OCREF_CLR_INT 被连接到 OCREF_CLR 输入。 • 1：OCREF_CLR_INT 被连接到 ETRF。
位 2:0	SMS[2:0]：从模式选择 当选择了外部信号，触发信号（TRGI）的有效边沿与选中的外部输入极性相关。 • 000：关闭从模式 如果 CEN=1，则预分频器直接由内部时钟驱动。 • 001：编码器模式 1 根据 TI1FP1 的电平，计数器在 TI2FP2 的边沿向上/向下计数。 • 010：编码器模式 2 根据 TI2FP2 的电平，计数器在 TI1FP1 的边沿向上/向下计数。 • 011：编码器模式 3 根据另一个信号的输入电平，计数器在 TI1FP1 和 TI2FP2 的边沿向上/向下计数。 • 100：复位模式 选中的触发输入信号（TRGI）的上升沿重新初始化计数器，并且触发一次更新寄存器。 • 101：门控模式 当触发输入（TRGI）为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止（不复位）。计数器的启动和停止都是受控的。 • 110：触发模式 计数器在触发输入 TRGI 的上升沿启动（不复位），只有计数器的启动是受控的。 • 111：外部时钟模式 1 选中的触发输入（TRGI）的上升沿驱动计数器。

表 12-3 TIM1 和 TIM2 内部触发连接

从定时器	ITR0（TS=000）	ITR1（TS=001）	ITR2（TS=010）
TIM1	TIM2	TIM6	ADC_AWD
TIM2	TIM1	TIM6	ADC_AWD

12.3.4 TIM1 中断使能寄存器（TIM1_DIER）

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
								rw	rw	rw	rw	rw	rw	rw	rw

位 15:8	Res：保留 必须保持复位值。
位 7	BIE：刹车中断使能 • 0：刹车中断禁用 • 1：刹车中断允许
位 6	TIE：触发中断使能 • 0：触发中断禁用 • 1：触发中断允许
位 5	COMIE：COM 中断使能 • 0：COM 中断禁用 • 1：COM 中断允许
位 4	CC4IE：捕捉/比较 4 中断使能 • 0：CC4 中断禁用 • 1：CC4 中断允许
位 3	CC3IE：捕捉/比较 3 中断使能 • 0：CC3 中断禁用 • 1：CC3 中断允许
位 2	CC2IE：捕捉/比较 2 中断使能 • 0：CC2 中断禁用 • 1：CC2 中断允许
位 1	CC1IE：捕捉/比较 1 中断使能 • 0：CC1 中断禁用 • 1：CC1 中断允许
位 0	UIE：更新中断使能 • 0：更新中断禁用 • 1：更新中断允许

12.3.5 TIM1 状态寄存器（TIM1_SR）

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res			CC4OF	CC3OF	CC2OF	CC1OF	Res	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位 15:13	Res：保留 必须保持复位值。
位 12	CC4OF：捕捉/比较4 重复捕捉标志 *说明：rc_w0 表示可读；可通过对该位写 0 清除。对该位写 1，该位的值无变化。*
位 11	CC3OF：捕捉/比较3 重复捕捉标志
位 10	CC2OF：捕捉/比较2 重复捕捉标志
位 9	CC1OF：捕捉/比较1 重复捕捉标志 仅当相应的通道被配置为输入捕获时，该位由硬件置1。软件写0 可清除该位。 • 0：无重复捕获产生。 • 1：当CC1IF 的状态已经为 1，计数器的值被捕获到TIMx_CCR1 寄存器。
位 8	Res：保留 必须保持复位值。
位 7	BIF：刹车中断标志 一旦刹车输入有效，由硬件对该位置 1。 如果刹车输入无效，则该位由软件清 0。 • 0：无刹车事件产生。 • 1：刹车输入信号上检测到有效电平。
位 6	TIF：触发器中断标志 当发生触发事件（从模式控制器处于门控模式外的其他模式时 TRGI 输入端检测到有效边沿，或者在门控模式下 TRGI 输入端检测到任一边沿）时由硬件对该位置 1。 该位由软件清 0。 • 0：无触发器事件产生 • 1：触发中断等待响应
位 5	COMIF：COM 中断标志 一旦产生 COM 事件（当捕获/比较控制位：CCxE、CCxNE、OCxM 被更新） 该位由硬件置 1 或软件清 0。 • 0：无COM 事件产生 • 1：COM 中断等待响应
位 4	CC4IF：捕捉/比较4 中断标志
位 3	CC3IF：捕捉/比较 3 中断标志
位 2	CC2IF：捕捉/比较 2 中断标志
位 1	CC1IF：捕捉/比较1 中断标志

	• 如果通道CC1 配置为输出模式：当计数器值与比较值匹配时，该位由硬件置1；但在中心对称模式下除外。 该位由软件清 0。 ○ 0：无匹配发生。 ○ 1：TIMx_CNT 的值与TIMx_CCR1 的值匹配。 当 TIMx_CCR1 > TIMx_ARR 时，满足以下任一条件，CC1IF 变为 1： - 在向上或向上/向下计数模式下，计数器溢出。 - 向下计数模式下，计数器溢出。 • 如果通道CC1 配置为输入模式：当捕获事件发生时，该位由硬件置 1。通过软件或读TIMx_CCR1 寄存器清 0。 ○ 0：无输入捕获产生； ○ 1：计数器值被捕获至 TIMx_CCR1（ 在 IC1 上检测到与所选极性相同的边沿）。
位 0	UIF：更新中断标志 当产生更新事件时，该位由硬件置 1。该位由软件清 0。 • 0：无更新事件产生； • 1：更新中断等待响应。 当发生以下更新寄存器事件时，该位由硬件置 1。 • 若 TIMx_CR1 寄存器的 UDIS=0，当重复计数器数值上溢或下溢时（重复计数器=0 时，产生更新事件）。 • 若 TIMx_CR1 寄存器的 URS=0 且 UDIS=0，当设置 TIMx_EGR 寄存器的 UG=1 时产生更新事件，通过软件对计数器 CNT 重新初始化。 • 若 TIMx_CR1 寄存器的 URS=0 且 UDIS=0，当计数器 CNT 被触发事件重新初始化时。

12.3.6 TIM1 事件产生寄存器（TIM1_EGR）

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
								w	w	w	w	w	w	w	w

位 15:8	Res：保留 必须保持复位值。
位 7	BG：产生刹车事件 该位由软件置 1，用于产生一个刹车事件。该位由硬件自动清 0。 • 0：无动作 • 1：产生一个刹车事件。此时 MOE=0、BIF=1，若开启对应的中断，则产生相应的中断。
位 6	TG：触发产生 该位由软件置 1，用于产生一个事件，由硬件自动清 0。 • 0：无动作

	• 1：TIMx_SR 中 TIF=1，若开启对应的中断，则产生相应的中断。
位 5	COMG：捕捉/比较控制更新产生 该位由软件置 1，由硬件自动清 0。 • 0：无动作 • 1：当 CCPC=1，允许更新 CCxE、CCxNE、OCxM 位。
位 4	CC4G：捕捉/比较 4 发生
位 3	CC3G：捕捉/比较 3 发生
位 2	CC2G：捕捉/比较 2 发生

位 1	CC1G：捕捉/比较 1 发生
	该位由软件置 1，用于产生一个捕获/比较事件，由硬件自动清 0。
	• 0：无动作
	• 1：在通道 1 上产生一个捕获/比较事件。
	  ○ 若通道 CC1 配置为输出：设置 CC1IF=1，若开启对应的中断，则产生相应的中断。
	  ○ 若通道 CC1 配置为输入：当前的计数器值被捕获至 TIMx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断，则产生相应的中断。若 CC1IF 已经为 1，则设置 CC1OF=1。

位 0	UG：产生更新事件 该位由软件置 1，由硬件自动清 0。 • 0：无动作； • 1：重新初始化计数器，并产生一个（寄存器）更新事件。 *注意：* *预分频器的计数器清 0，但预分频系数不变。若在中心对称模式下或 DIR=0（向上计数），则计数器被清 0；若 DIR=1（向下计数），则计数器取 TIMx_ARR 的值。*

12.3.7 TIM1 捕捉/比较模式寄存器 1（TIM1_CCMR1）

偏移地址：0x18

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			
rw	rw			rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 15	OC2CE：允许输出比较 2 清 0
位 14:12	OC2M[2:0]：输出比较模式 2

位 11	OC2PE：允许预装输出比较 2
位 10	OC2FE：允许快速输出比较 2
位 9:8	CC2S[1:0]：捕捉/比较 2 选择 该位域定义通道的方向（输入/输出），及选择输入信号。 • 00：CC2 通道配置为输出。 • 01：CC2 通道配置为输入，IC2 映射在 TI2 上。 • 10：CC2 通道配置为输入，IC2 映射在 TI1 上。 • 11：CC2 通道配置为输入，IC2 映射在 TRC 上。此模式仅在内部触发器输入被选中时工作（由 TIMx_SMCR 寄存器的 TS 位选择）。
位 7	OC1CE：允许输出比较 1 清零 • 0：OC1REF 不受 ETRF 输入的影响。 • 1：一旦检测到 ETRF 输入高电平，OC1REF 清 0。

位 6:4	OC1M[2:0]：输出比较模式 1（Output Compare 1 mode）
	该位域定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1 和 OC1N 的值。OC1REF 是高电平有效，而 OC1、OC1N 的有效电平分别取决于 CC1P、CC1NP。
	• 000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较，对 OC1REF 不起作用。
	• 001：匹配时设置通道 1 为有效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1（TIMx_CCR1）相同时，强制 OC1REF 为高。
	• 010：匹配时设置通道 1 为无效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1（TIMx_CCR1）相同时，强制 OC1REF 为低。
	• 011：翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。
	• 100：强制为无效电平。强制 OC1REF 为低。
	• 101：强制为有效电平。强制 OC1REF 为高。
	• 110：PWM 模式 1
	  ○ 在向上计数时若 TIMx_CNT < TIMx_CCR1，则通道 1 为有效电平，否则为无效电平。
	  ○ 在向下计数时若 TIMx_CNT > TIMx_CCR1，则通道 1 为无效电平（OC1REF=0），否则为有效电平（OC1REF=1）。
	• 111：PWM 模式 2
	  ○ 在向上计数时若 TIMx_CNT < TIMx_CCR1，则通道 1 为无效电平，否则为有效电平。
	  ○ 在向下计数时若 TIMx_CNT > TIMx_CCR1，则通道 1 为有效电平，否则为无效电平。

位 3	OC1PE：使能输出比较 1 预装载（Output Compare 1 preload enable） • 0：禁用 TIMx_CCR1 寄存器的预装载功能。可随时写入 TIMx_CCR1 寄存器，并且新写入的数值立即生效。 • 1：开启 TIMx_CCR1 寄存器的预装载功能。读写操作仅针对预装载寄存器，

	TIMx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。
位 2	OC1FE：输出比较 1 快速使能（Output Compare 1 fast enable） 该位可加快 CC 输出对触发输入事件的响应。 • 0：CC1 的正常操作依赖于计数器与 CCR1 的值，即使触发器处于工作状态。当触发器的输入产生一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期。 • 1：输入到触发器的有效沿的作用就像发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。OCFE 只在通道被配置成 PWM1 或 PWM2 模式时生效。
位 1:0	CC1S[1:0]：捕捉/比较 1 选择（Capture/Compare 1 selection） 该位域定义通道的方向（输入/输出），及所选择的输入信号。 • 00：CC1 通道被配置为输出。 • 01：CC1 通道被配置为输入，IC1 映射在 TI1 上。 • 10：CC1 通道被配置为输入，IC1 映射在 TI2 上。 • 11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅在内部触发器输入被选中时（通过 TIMx_SMCR 寄存器的 TS 位选择）工作。

输入捕捉模式

位 15:12	IC2F[3:0]：输入捕捉 2 滤波器
位 11:10	IC2PSC[1:0]：输入捕捉 2 预分频器
位 9:8	CC2S[1:0]：捕捉/比较 2 选择 该位域定义通道的方向（输入/输出），及输入信号的选择： • 00：CC2 通道被配置为输出； • 01：CC2 通道被配置为输入，IC2 映射在 TI2 上； • 10：CC2 通道被配置为输入，IC2 映射在 TI1 上； • 11：CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时（由 TIMx_SMCR 寄存器的 TS 位选择）。
位 7:4	IC1F[3:0]：输入捕捉 1 滤波器 该位域定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变： • 0000：无滤波器，以 f_{DTS} 采样 • 0001：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=2 • 0010：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=4 • 0011：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=8 • 0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6 • 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 • 0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6 • 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8

	• 1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6 • 1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8 • 1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5 • 1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6 • 1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8 • 1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5 • 1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6 • 1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8
位 3:2	IC1PSC[1:0]：输入捕捉 1 预分频器 该位域定义了 CC1 输入（IC1）的预分频系数。一旦 CC1E=0（在 TIMx_CCER 寄存器中），则预分频器复位。 • 00：无预分频器，捕获输入口上检测到的每一个边沿都会触发一次捕获。 • 01：每 2 个事件触发一次捕获。 • 10：每 4 个事件触发一次捕获。 • 11：每 8 个事件触发一次捕获。
位 1:0	CC1S[1:0]：捕捉/比较 1 选择 该位域定义通道的方向（输入/输出），及输入信号的选择： • 00：CC1 通道被配置为输出。 • 01：CC1 通道被配置为输入，IC1 映射在 TI1 上。 • 10：CC1 通道被配置为输入，IC1 映射在 TI2 上。 • 11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅在内部触发器输入被选中时（由 TIMx_SMCR 寄存器的 TS 位选择）工作。

12.3.8 TIM1 捕捉/比较模式寄存器 2（TIM1_CCMR2）

偏移地址：0x1C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
rw	rw			rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 15	OC4CE：输出比较4 清除使能
位 14:12	OC4M[2:0]：输出比较4 模式
位 11	OC4PE：输出比较4 预分频使能
位 10	OC4FE：输出比较4 快速使能
位 9:8	CC4S[1:0]：捕捉/比较4 选择

	该位域定义通道的方向（输入/输出），及输入信号的选择： • 00：CC4 通道被配置为输出。 • 01：CC4 通道被配置为输入，IC4 映射在TI4 上。 • 10：CC4 通道被配置为输入，IC4 映射在TI3 上。 • 11：CC4 通道被配置为输入，IC4 映射在TRC 上。此模式仅在内部触发器输入被选中时（由TIMx_SMCR 寄存器的TS 位选择）工作。
位 7	OC3CE：输出比较3 清除使能
位 6:4	OC3M[2:0]：输出比较3 模式
位 3	OC3PE：输出比较3 预分频使能
位 2	OC3FE：输出比较3 快速使能
位 1:0	CC3S[1:0]：捕捉/比较3 选择 该位域定义通道的方向（输入/输出），及输入信号的选择： • 00：CC3 通道被配置为输出。 • 01：CC3 通道被配置为输入，IC4 映射在TI3 上。 • 10：CC3 通道被配置为输入，IC4 映射在TI4 上。 • 11：CC3 通道被配置为输入，IC4 映射在TRC 上。

输入捕捉模式

位 15:12	IC4F[3:0]：输入捕捉4 滤波器
位 11:10	IC4PSC[1:0]：输入捕捉4 预分频器
位 9:8	CC4S[1:0]：捕捉/比较4 选择 该位域定义通道的方向（输入/输出），及输入信号的选择： • 00：CC4 通道被配置为输出。 • 01：CC4 通道被配置为输入，IC3 映射在TI4 上。 • 10：CC4 通道被配置为输入，IC3 映射在TI3 上。 • 11：CC4 通道被配置为输入，IC3 映射在TRC 上。
位 7:4	IC3F[3:0]：输入捕捉3 滤波器
位 3:2	IC3PSC[1:0]：输入比较3 预分频器
位 1:0	CC3S[1:0]：捕捉/比较3 选择 该位域定义通道的方向（输入/输出），及输入信号的选择： • 00：CC3 通道被配置为输出。 • 01：CC3 通道被配置为输入，IC3 映射在TI3 上。 • 10：CC3 通道被配置为输入，IC3 映射在TI4 上。

	• 11：CC3 通道被配置为输入，IC3 映射在TRC 上。

12.3.9 TIM1 捕捉/比较使能寄存器（TIM1_CCER）

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res		CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:14	Res：保留 必须保持复位值。
位 13	CC4P：捕捉/比较 4 输出极性
位 12	CC4E：捕捉/比较 4 输出使能
位 11	CC3NP：捕捉/比较 3 互补输出极性
位 10	CC3NE：捕捉/比较 3 互补输出使能
位 9	CC3P：捕捉/比较 3 输出极性
位 8	CC3E：捕捉/比较 3 输出使能
位 7	CC2NP：捕捉/比较 2 互补输出极性
位 6	CC2NE：捕捉/比较 2 互补输出使能
位 5	CC2P：捕捉/比较 2 输出极性
位 4	CC2E：捕捉/比较 2 输出使能
位 3	CC1NP：捕捉/比较 1 互补输出极性
位 2	CC1NE：捕捉/比较1 互补输出使能 • 0：关闭 OC1N 禁止输出，因此OC1N 的电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N 和CC1E 位的值。 • 1：开启 OC1N 信号输出到对应的输出引脚，其输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N 和CC1E 位的值。
位 1	CC1P：捕捉/比较1 输出极性 • CC1 通道配置为输出：

	○ 0：OC1 高电平有效。 ○ 1：OC1 低电平有效。 • CC1 通道配置为输入： CC1NP/CC1P 位选择在触发或捕捉模式下TI1FP1 和TI2FP1 的有效极性。 ○ 00：非反相/上升沿 电路作用于TIxFP1 的上升沿（在复位、外部时钟或触发模式下的捕捉或触发操作），TIxFP1 非反相。 ○ 01：反相/下降沿 电路作用于TIxFP1 的下降沿（在复位、外部时钟或触发模式下的捕捉或触发操作）TIxFP1 反相。 ○ 00：保留不用 ○ 11：非反相/上升或下降沿 电路作用于 TIxFP1 的上升沿和下降沿（在复位、外部时钟或触发模式下的捕捉或触发操作），TIxFP1 非反相（在门控模式）。在编码模式下不能使用此配置。
位 0	CC1E：捕捉/比较1 输出使能 • CC1 通道配置为输出： ○ 0：关闭 OC1N 禁止输出，因此OC1N 的电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。 ○ 1：开启 OC1N 信号输出到对应的输出引脚，其输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N 和CC1E 位的值。 • CC1 通道配置为输入： 该位决定是否将一个定时器值的捕捉装载到捕捉/比较寄存器1（TIMx_CCR1）。 ○ 0：捕捉禁止 ○ 1：捕捉使能

表 12-4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位

控制位					输出状态（1）	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	0	0	0	输出禁止（与定时器断开） OCx=0，OCx_EN=0	输出禁止（与定时器断开） OCxN=0，OCxN_EN=0
		0	0	1	输出禁止（与定时器断开） OCx=0，OCx_EN=0	OCxREF + 极性， OCxN= OCxREF xor CCxNP， OCxN_EN=1
		0	1	0	OCxREF + 极性， OCx= OCxREF xor CCxP， OCx_EN=1	输出禁止（与定时器断开） OCxN=0，OCxN_EN=0
		0	1	1	OCxREF + 极性 + 死区， OCx_EN=1	OCxREF 反相 + 极性 + 死区， OCxN_EN=1

控制位					输出状态（1）	
		1	0	0	输出禁止（与定时器断开） OCx=CCxP，OCx_EN=0	输出禁止（与定时器断开） OCxN=CCxNP，OCxN_EN=0
		1	0	1	关闭状态（输出使能且为无效电平）OCx=CCxP，OCx_EN=1	OCxREF + 极性， OCxN= OCxREF xor CCxNP，OCxN_EN=1
		1	1	0	OCxREF + 极性， OCx= OCxREF xor CCxP，OCx_EN=1	关闭状态（输出使能且为无效电平）OCxN=CCxNP，OCxN_EN=1
		1	1	1	OCxREF + 极性 + 死区， OCx_EN=1	OCxREF 反相 + 极性 + 死区， OCxN_EN=1
0	0	X	0	0	输出禁止（与定时器断开） 异步地：OCx=CCxP，OCx_EN=0，OCxN=CCxNP，OCxN_EN=0； 若时钟存在：经过 一个死区时间后 OCx=OISx ， OCxN=OISxN，假设 OISx 与 OISxN 并不都对应 OCx 和 OCxN 的 有效电平。	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0	关闭状态（输出使能且为无效电平） 异步地： OCx=CCxP ， OCx_EN=1 ， OCxN=CCxNP ， OCxN_EN=1； 若时钟存 在：经过 一个死区 时间后 OCx=OISx ， OCxN=OISxN，假设 OISx 与 OISxN 并不都对应 OCx 和 OCxN 的 有效电平。	
	1		0	1		
	1		1	0		
	1		1	1		

1. 如果一个通道的 2 个输出都没有使用（CCxE = CCxNE = 0），那么 OISx, OISxN, CCxP 和 CCxNP 都必须清零。

说明：引脚连接到互补的 OCx 和 OCxN 通道的外部 I/O 引脚的状态，取决于 OCx 和 OCxN 通道状态和 GPIO 以及 AFIO 寄存器。

12.3.10 TIM1 计数器（TIM1_CNT）

偏移地址：0x24

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

位15:0	CNT[15:0]：计数器值

12.3.11 TIM1 预分频器（TIM1_PSC）

偏移地址：0x28

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															

rw

位 15:0	PSC[15:0]：预分频值 计数器的时钟频率（CK_CNT）等于 f_{CK_PSC}/（PSC[15:0]+1）。每次当更新事件产生时，PSC 的值被装入当前预分频器寄存器。 更新事件包括计数器被TIM_EGR 的UG 位清 0 或被工作在复位模式的从控制器清 0。

12.3.12 TIM1 自动重装载寄存器（TIM1_ARR）

偏移地址：0x2C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw															

位 15:0	ARR[15:0]：自动重装载的值 ARR 包含了将要装载实际的自动重装载寄存器的值。

12.3.13 TIM1 重复计数寄存器（TIM1_RCR）

偏移地址：0x30

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								REP[7:0]							
								rw							

位 15:8	Res：保留 必须保持复位值。
位 7:0	REP[7:0]：重复计数器的值 使能预装载寄存器后，该位域允许用户设置比较寄存器的更新速率（即周期性地从预装载寄存器传输到当前寄存器）；如果允许产生更新中断，则会同时影响产生更新中断的速率。 每次向下计数器REP_CNT 达到0，会产生一个更新事件并且计数器REP_CNT 重新从REP 值开始计数。由于REP_CNT 只有在周期更新事件U_RC 发生时才重载 REP 值，因此对TIMx_RCR 寄存器写入的新值仅在下次周期更新事件发生时才起作用。这意味着在PWM 模式中，（REP+1）对应着： • 在边沿对齐模式下，PWM 周期的数目。 • 在中心对称模式下，PWM 半周期的数目。

12.3.14 TIM1 捕捉/比较寄存器 1（TIM1_CCR1）

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw															

位 15:0	CCR1[15:0]：捕捉/比较通道1 的值 • 若CC1 通道配置为输出： CCR1 决定了装入当前捕获/比较1 寄存器的值（预装载值）。如果在TIMx_CCMR1 寄存器（OC1PE 位）中未选择预装载功能，写入的数值会立即传输至 TIM1_CCR1 寄存器。否则只有当更新事件发生时，此预装载值才传输至 TIM1_CCR1 寄存器。TIM1_CCR1 寄存器参与同计数器TIMx_CNT 的比较，并在 OC1 端口上产生输出信号。 • 若CC1 通道配置为输入： CCR1 包含了由上一次输入捕获1 事件（IC1）传输的计数器值。

12.3.15 IM1 捕捉/比较寄存器 2（TIM1_CCR2）

偏移地址：0x38

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw															

位 15:0	CCR2[15:0]：捕捉/比较通道 2 的值 • 若 CC2 通道配置为输出： CCR2 决定了装入 TIM_CCR2 寄存器的值（预装载值）。如果在 TIMx_CCMR2 寄存器（OC2PE 位）中未选择预装载功能，写入的数值会立即传输至 TIM_CCR2 寄存器。否则只有当更新事件发生时，此预装载值才传输至 TIM_CCR2 寄存器。TIM_CCR2 寄存器参与同计数器 TIMx_CNT 的比较，并在 OC2 端口上产生输出信号。 • 若 CC2 通道配置为输入： CCR2 包含了由上一次输入捕获 2 事件（IC2）传输的计数器值。

12.3.16 TIM1 捕捉/比较寄存器 3（TIM1_CCR3）

偏移地址：0x3C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw															

位 15:0	CCR3[15:0]：捕捉/比较通道 3 的值 • 若 CC3 通道配置为输出： CCR3 决定了装入 TIM1_CCR3 寄存器的值（预装载值）。如果在 TIMx_CCMR3 寄存器（OC4PE 位）中未选择预装载功能，写入的数值会立即传输至 TIM1_CCR3 寄存器。否则只有当更新事件发生时，此预装载值才传输至 TIM1_CCR3 寄存器。TIM1_CCR3 寄存器参与同计数器 TIMx_CNT 的比较，并在 OC3 端口上产生输出信号。 • 若 CC3 通道配置为输入： CCR3 包含了由上一次输入捕获 3 事件（IC3）传输的计数器值。

12.3.17 TIM1 捕捉/比较寄存器 4（TIM1_CCR4）

偏移地址：0x40

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw															

位 15:0	CCR4[15:0]：捕捉/比较通道 4 的值 • 若 CC4 通道配置为输出： CCR4 决定了装入 TIM1_CCR4 寄存器的值（预装载值）。如果在 TIMx_CCMR4 寄存器（OC4PE 位）中未选择预装载功能，写入的数值会立即传输至 TIM1_CCR4 寄存器。否则只有当更新事件发生时，此预装载值才传输至 TIM1_CCR4 寄存器。TIM1_CCR4 寄存器参与同计数器 TIMx_CNT 的比较，并在 OC4 端口上产生输出信号。 • 若 CC4 通道配置为输入： CCR4 包含了由上一次输入捕获 4 事件（IC4）传输的计数器值。

12.3.18 TIM1 刹车和死区寄存器（TIM1_BDTR）

偏移地址：0x44

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

位 15	MOE：主输出使能（Main output enable） 一旦刹车输入有效，该位被硬件异步清 0。根据AOE 位的值，该位可由软件清 0 或被自动置1。它仅对配置为输出的通道有效。 • 0：禁止OC 和OCN 输出或强制为空闲状态。 • 1：如果设置了相应的使能位（TIMx_CCER 寄存器的CCxE、CCxNE 位），则开启 OC 和 OCN 输出。
位 14	AOE：自动输出使能（Automatic output enable） • 0：MOE 只能被软件置 1。 • 1：MOE 能被软件置 1 或在下一个更新事件被自动置 1（如果刹车输入无效）。
位 13	BKP：刹车输入极性（Break polarity） • 0：刹车输入低电平有效 • 1：刹车输入高电平有效
位 12	BKE：刹车使能（Break enable） • 0：刹车输入禁止（BRK 和CCS 时钟失效事件） • 1：刹车输入允许（BRK 和CCS 时钟失效事件）
位 11	OSSR：运行模式下“关闭状态”选择（Off-state selection for Run mode）

	该位用于当MOE=1 且通道为互补输出时，没有互补输出的定时器中不存在OSSR位的情况。 • 0：当定时器不工作时，禁止OC/OCN 输出（OC/OCN 使能输出信号=0） • 1：当定时器不工作时，一旦CCxE=1 或CCxNE=1，OC/OCN 使能并输出无效电平，然后置OC/OCN 使能输出信号=1
位 10	OSSI：运行模式下“空闲状态”选择（Off-state selection for Idle mode） 该位用于当MOE=0 时通道为输出。 • 0：当定时器不工作时，禁止OC/OCN 输出（OC/OCN 使能输出信号=0）。 • 1：当定时器不工作时，一旦CCxE=1 或CCxNE=1，OC/OCN 被强制输出空闲电平，且置OC/OCN 使能输出信号=1。
位 9:8	LOCK[1:0]：锁定设置（Lock configuration） 该位域为防止软件错误而提供写保护。 • 00：锁定关闭，寄存器无写保护。 • 01：锁定级别1，不能写入TIMx_BDTR 寄存器的DTG、BKE、BKP、AOE 位和TIMx_CR2 寄存器的OISx/OISxN 位。 • 10：锁定级别2，不能写入锁定级别1 中的各位，也不能写入CC 极性位（一旦相关通道通过CCxS 位设为输出，CC 极性位是TIMx_CCER 寄存器的CCxP/CCNxP 位）以及OSSR/OSSI 位。 • 11：锁定级别3，不能写入锁定级别2 中的各位，也不能写入CC 控制位（一旦相关通道通过CCxS位设为输出，CC 控制位是TIMx_CCMRx寄存器的OCxM/OCxPE位）（
位 7:0	DTG[7:0]：死区发生器设置（Dead-time generator setup） 该位域定义了插入互补输出之间的死区持续时间。

13 通用定时器（TIM2）

通用定时器是一个由可编程预分频器驱动的 32 位自动装载计数器构成。它适用于多种场合，包括测量输入信号的脉冲宽度（输入捕获）或者产生输出波形（输出比较和 PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，脉冲宽度和波形周期可以在几个微秒到几个毫秒间调整。

每个通用定时器都是完全独立的，没有互相共享任何资源。它们可以一起同步操作，TIM2 的特性如下：

表 13-1 TIM2 特性

符号	参数	条件	最小值	典型值	最大值	单位
$t_{res（TIM）}$	定时器分辨时间	$f_{TIMxCLK}$=32MHz	-	31.2	-	ns
f_{EXT}	定时器的 CH1 至 CH4，外部输入的时钟频率	-	-	$f_{TIMxCLK}/2$	-	MHz
		$f_{TIMxCLK}$=32MHz	-	16	-	MHz
t_{MAX_COUNT}	当选择内部时钟时，32 位计数器的时钟周期		-	2^{32}	-	$t_{TIMxCLK}$
		$f_{TIMxCLK}$=32MHz	-	134.22	-	s

13.1 TIM2 主要功能

通用 TIM2 定时器功能包括：

- 四路输入通道都支持下降沿触发和双沿触发功能
- 32 位向上、向下、向上/向下自动装载计数器
- 16 位可编程（可实时修改）预分频器，计数器时钟频率的分频系数为 1~65536 之间的任意数值。
- 4 个独立通道：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿或中央对齐模式）
  - 单脉冲模式输出
- 使用外部信号控制定时器和定时器互连的同步电路
- 以下事件发生时产生中断：
  - 更新：计数器向上溢出/向下溢出，计数器初始化（通过软件或者内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或者由内部/外部触发计数）
  - 输入捕获
  - 输出比较
- 支持针对定位的增量（正交）编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理

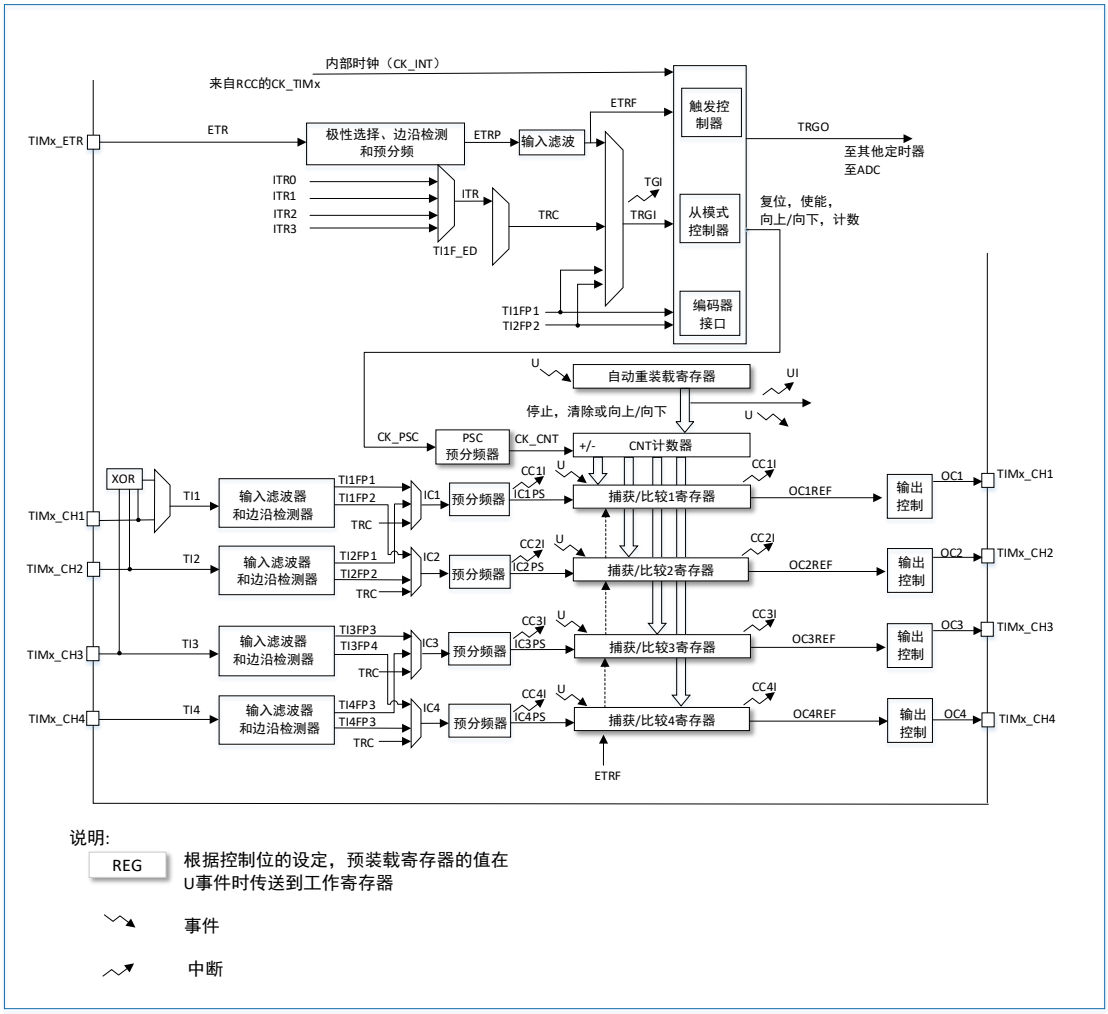


图 13-1 通用定时器框图

13.2 TIM2 功能描述

13.2.1 时基单元

可编程通用定时器的主要部分是一个 32 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数、向下计数或者向上向下双向计数。此计数器时钟由预分频器分频得到。

计数器、自动装载寄存器和预分频器寄存器可以由软件读写，且在计数器运行时仍可以读写。

时基单元包含：

- 计数器寄存器（TIM2_CNT）
- 预分频器寄存器（TIM2_PSC）
- 自动装载寄存器（TIM2_ARR）

自动装载寄存器是预先装载的，写或读自动重装载寄存器将访问预装载寄存器。根据在 TIM2_CR1 寄存器中的自动装载预装载使能位（ARPE）的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 到来时传送到影子寄存器。当计数器达到溢出条件（向下计数时的下溢条件）并当 TIM2_CR1 寄存器中的 UDIS 位等于'0'时，产生更新事件。更新事件也可以由软件产生。随后会详细描述每一种配置下更新事件的产生。

计数器由预分频器的时钟输出 CK_CNT 驱动，仅当设置了计数器 TIM2_CR1 寄存器中的计数器使能位（CEN）时，CK_CNT 才有效。

说明：真正的计数器使能信号 CNT_EN 是在 CEN 的一个时钟周期后被设置。

预分频器描述

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个（在 TIM2_PSC 寄存器中的）16 位寄存器控制的 16 位计数器。这个控制寄存器带有缓冲器，它能够在工作时被改变。新的预分频器参数在下一次更新事件到来时被采用。

下面两个图给出了在预分频器运行时，更改计数器参数的例子。

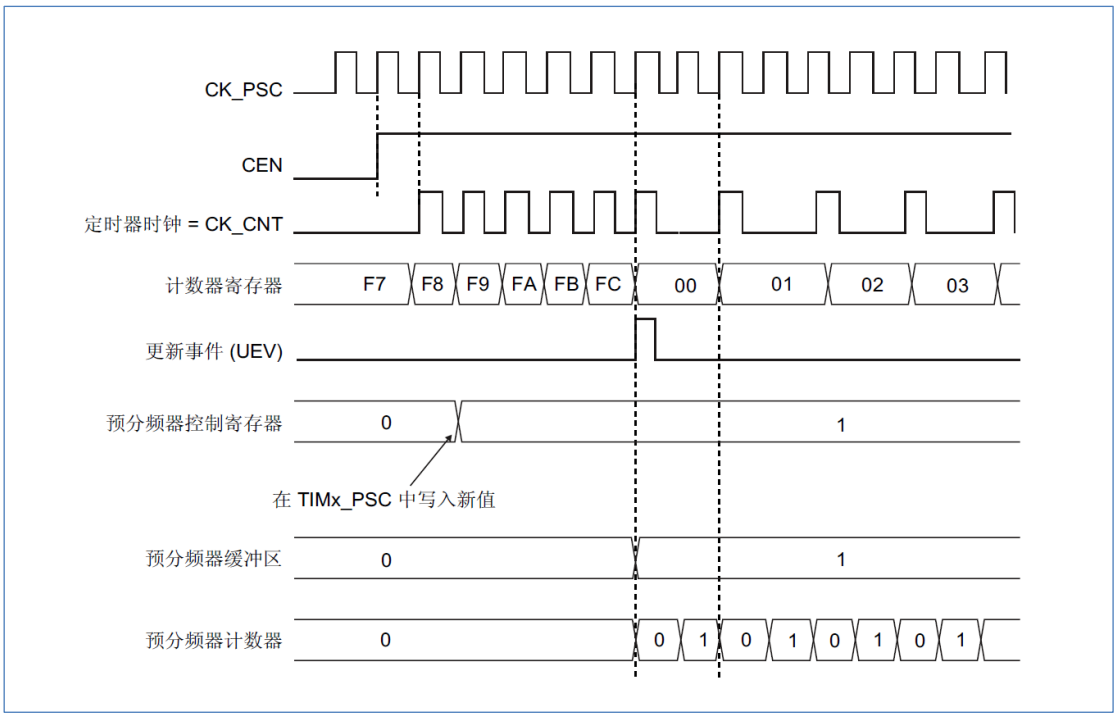


图 13-2 当预分频器的参数从 1 变到 2 时，计数器的时序图

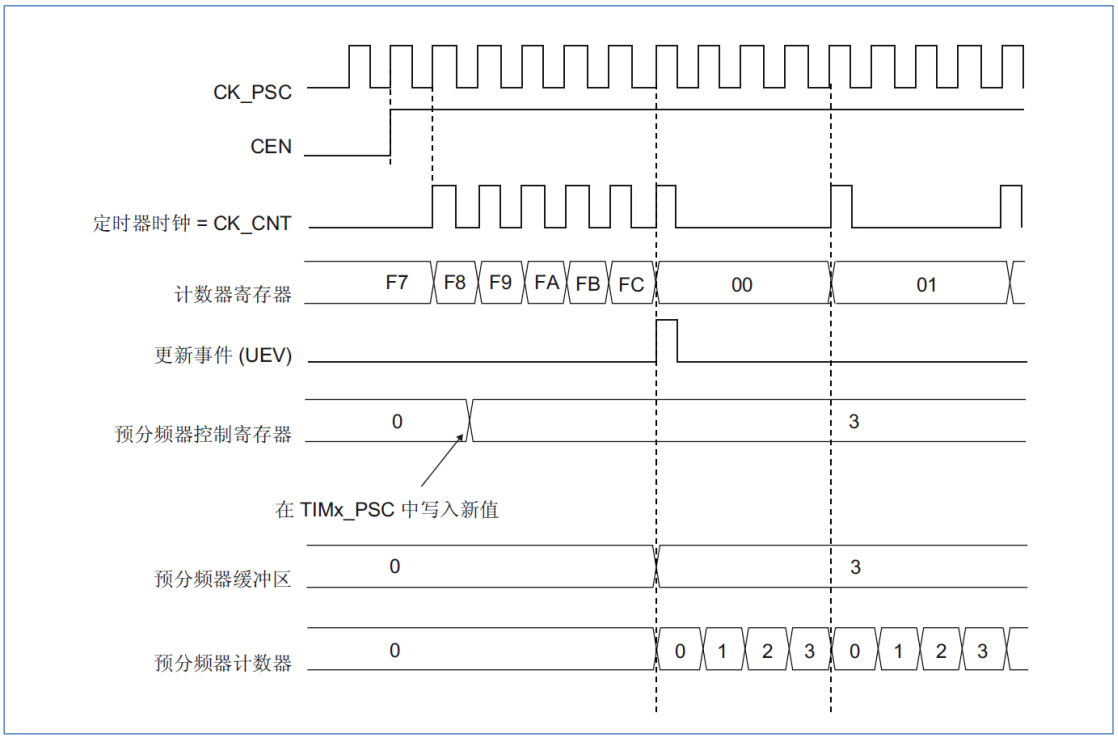


图 13-3 当预分频器的参数从 1 变到 4 时，计数器的时序图

13.2.2 计数器模式

13.2.2.1 向上计数模式

在向上计数模式中，计数器从 0 计数到自动加载值（TIM2_ARR 计数器的内容），然后重新从 0 开始计数并且产生一个计数器溢出事件。

每次计数器溢出时可以产生更新事件，在 TIM2_EGR 寄存器中（通过软件方式或者使用从模式控制器）设置 UG 位也同样可以产生一个更新事件。

设置 TIM2_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清'0'之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清'0'，同时预分频器的计数也清零（但预分频系数不变）。此外，如果设置了 TIM2_CR1 寄存器中的 URS 位（选择更新请求），设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志（即不产生中断）；这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时（依据 URS 位）设置更新标志位（TIM2_SR 寄存器中的 UIF 位）。

- 预分频器的缓冲区被置入预装载寄存器的值（TIM2_PSC 寄存器的内容）。
- 自动装载影子寄存器被重新置入预装载寄存器的值（TIM2_ARR）。

下图给出一些例子，当 TIM2_ARR=0x36 时计数器在不同时钟频率下的动作。

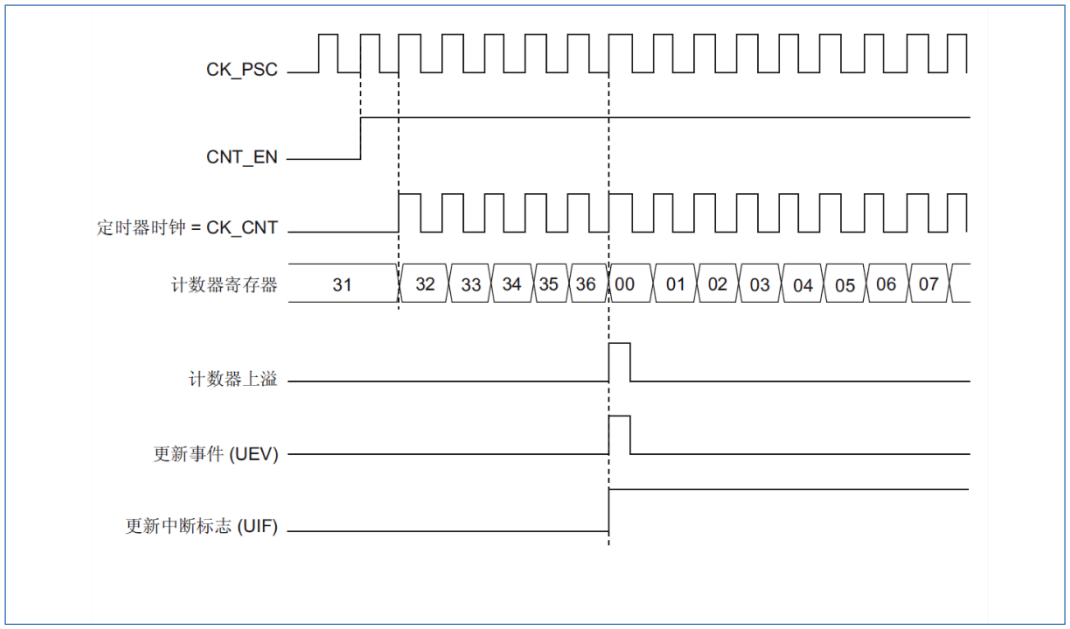


图 13-4 计数器时序图，内部时钟分频因子为 1

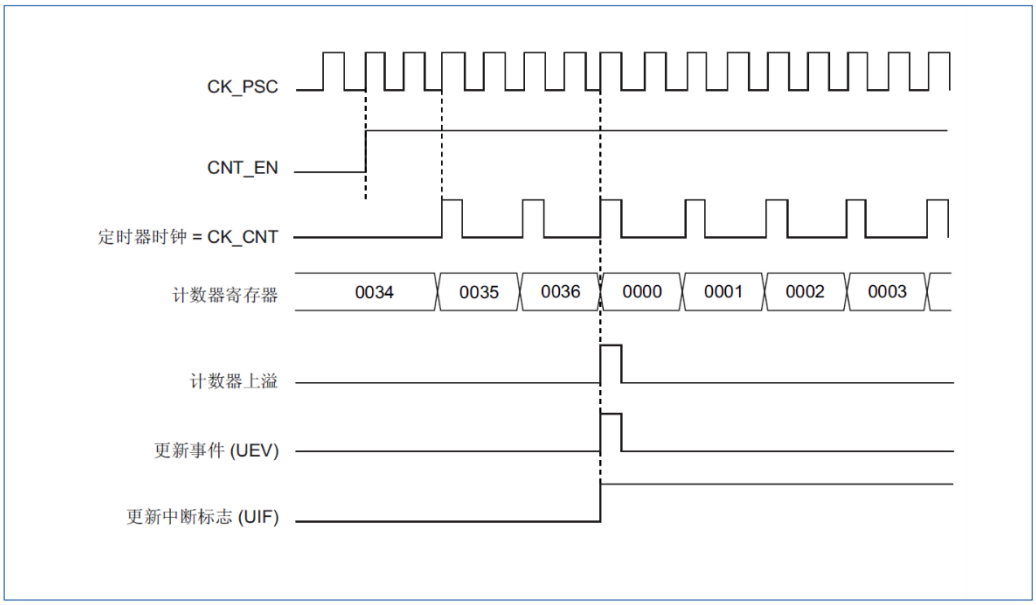


图 13-5 计数器时序图，内部时钟分频因子为 2

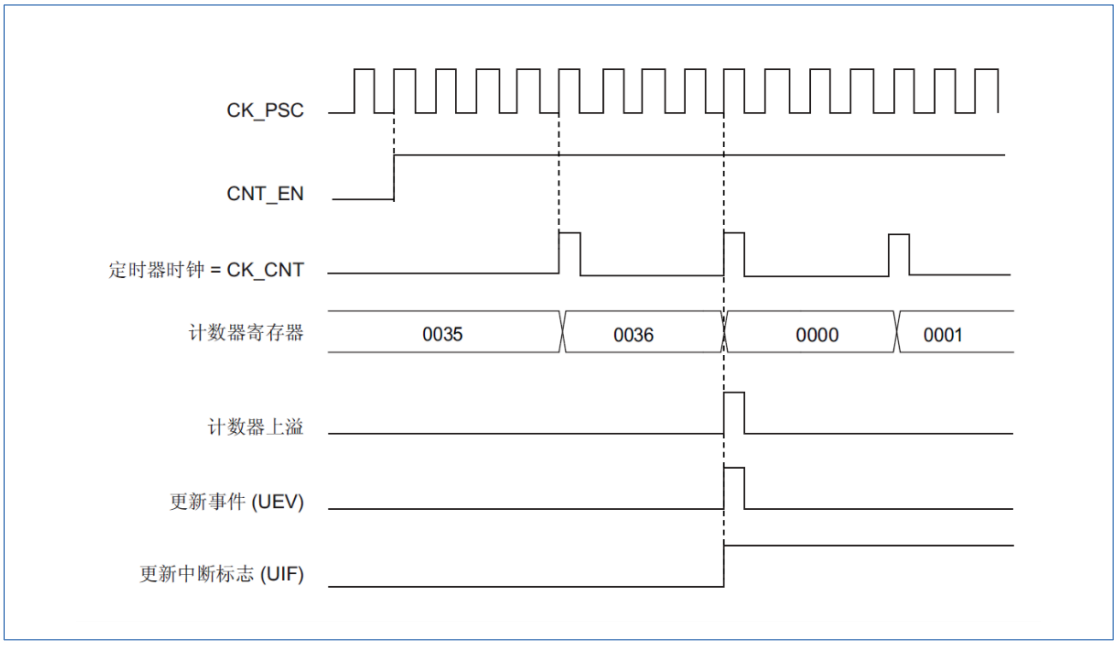


图 13-6 计数器时序图，内部时钟分频因子为 4

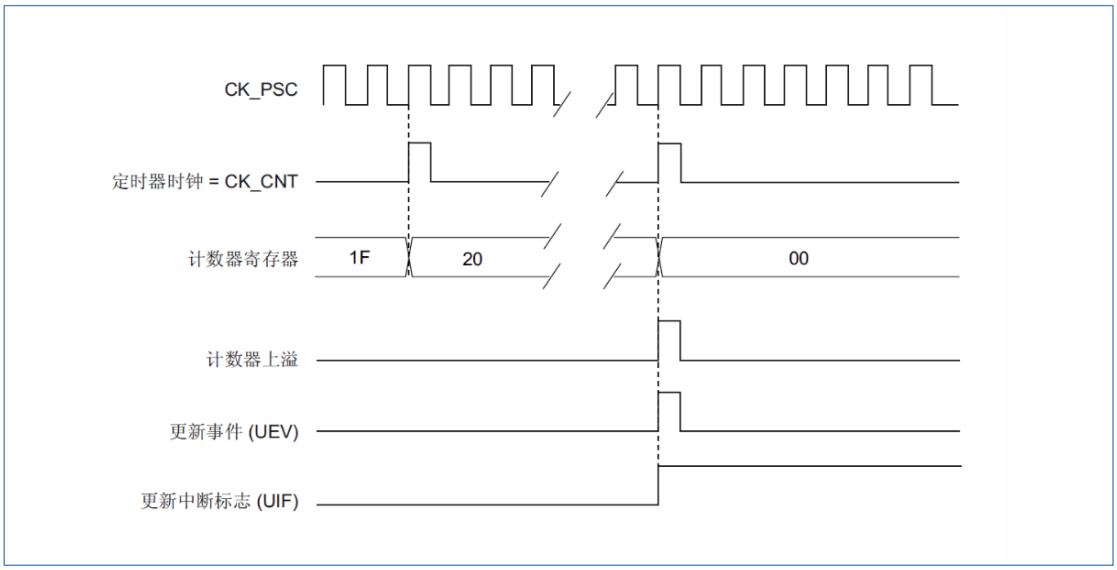


图 13-7 计数器时序图，内部时钟分频因子为 N

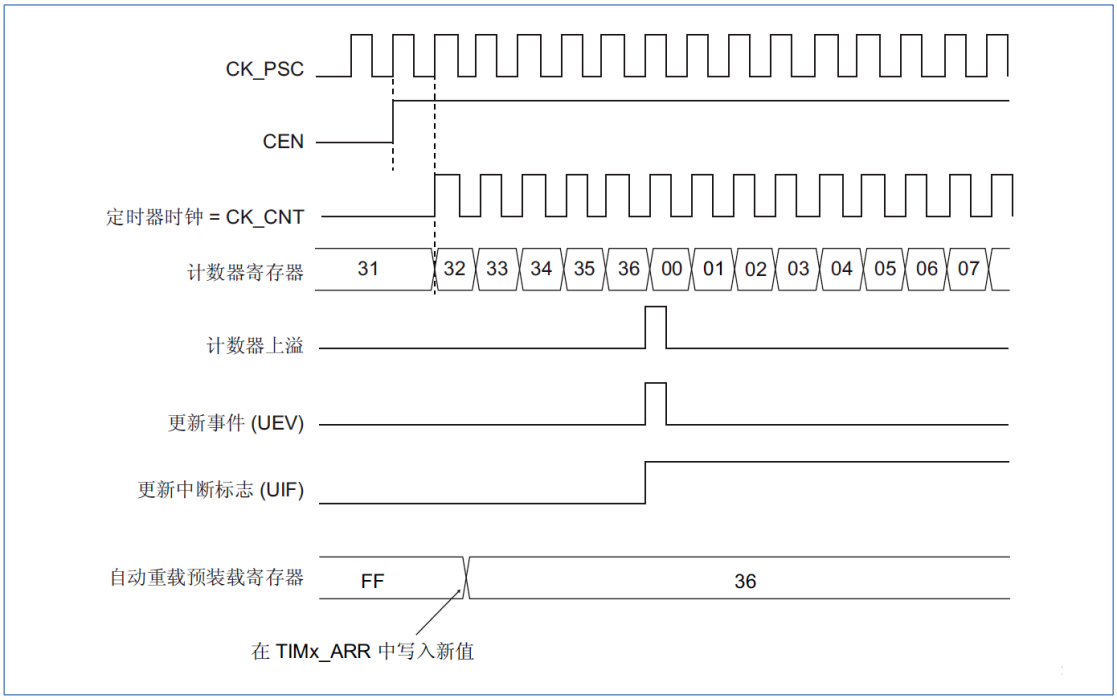


图 13-8 计数器时序图，当 ARPE=0 时的更新事件（TIM2_ARR 没有预装入）

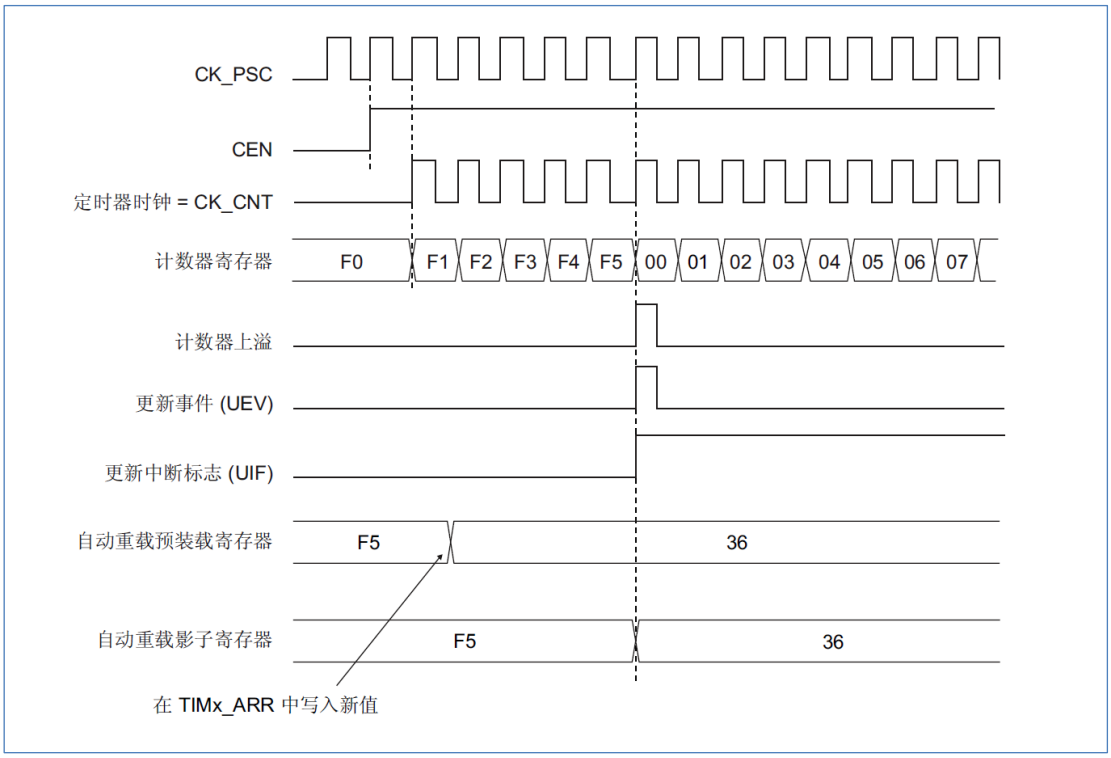


图 13-9 计数器时序图，当 ARPE=1 时的更新事件（预装入了 TIM2_ARR）

13.2.2.2 向下计数模式

在向下模式中，计数器从自动装入的值（TIM2_ARR 计数器的值）开始向下计数到 0，然后从自动装入的值重新开始并且产生一个计数器向下溢出事件。

每次计数器溢出时可以产生更新事件，在 TIM2_EGR 寄存器中（通过软件方式或者使用从模式控制器）设置 UG 位，也同样可以产生一个更新事件。

设置 TIM2_CR1 寄存器的 UDIS 位可以禁止 UEV 事件。这样可以避免向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为'0'之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，同时预分频器的计数器重新从 0 开始（但预分频系数不变）。

此外，如果设置了 TIM2_CR1 寄存器中的 URS 位（选择更新请求），设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志（因此不产生中断），这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且（根据 URS 位的设置）更新标志位（TIM2_SR 寄存器中的 UIF 位）也被设置。

- 预分频器的缓存器被置入预装载寄存器的值（TIM2_PSC 寄存器的值）。
- 当前的自动加载寄存器被更新为预装载值（TIM2_ARR 寄存器中的内容）。

说明：自动装载在计数器重载入之前被更新，因此下一个周期将是预期的值。

以下是一些当 TIM2_ARR=0x36 时，计数器在不同时钟频率下的操作例子。

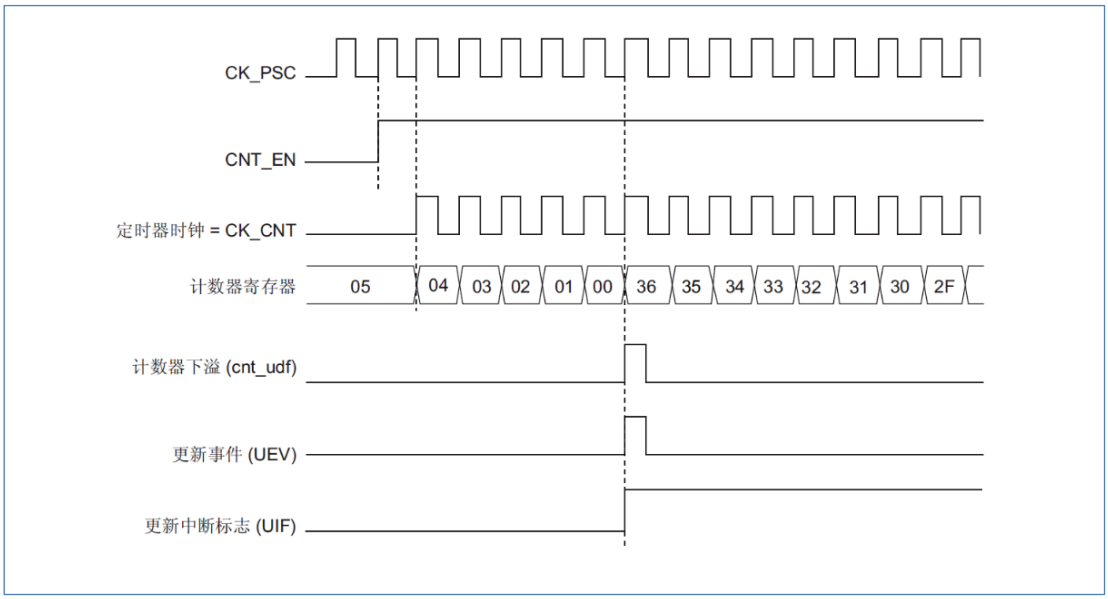


图 13-10 计数器时序图，内部时钟分频因子为 1

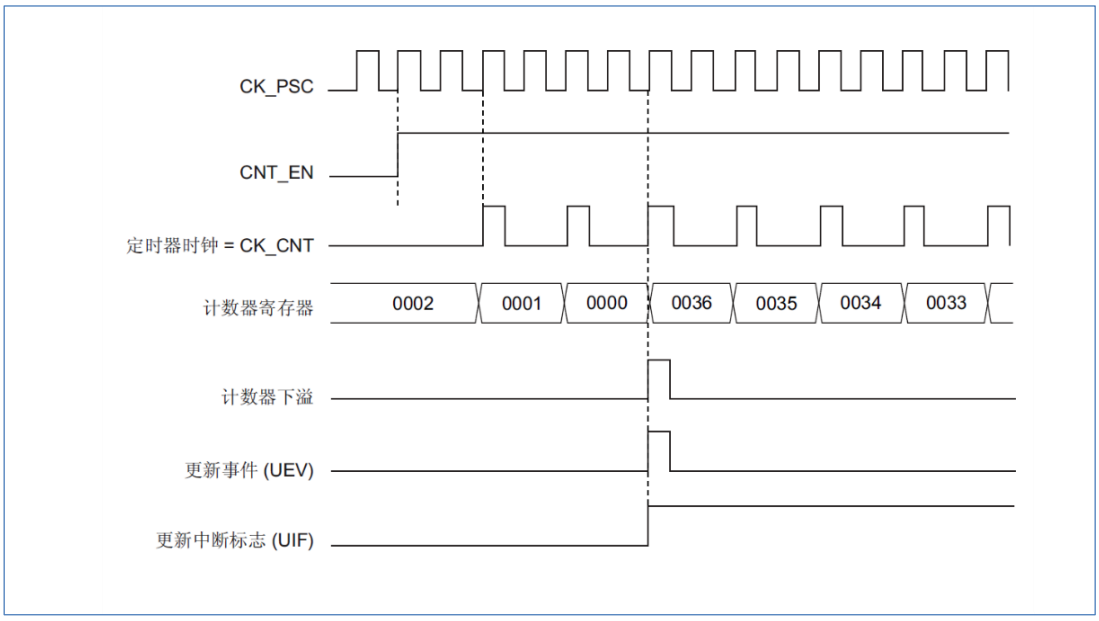


图 13-11 计数器时序图，内部时钟分频因子为 2

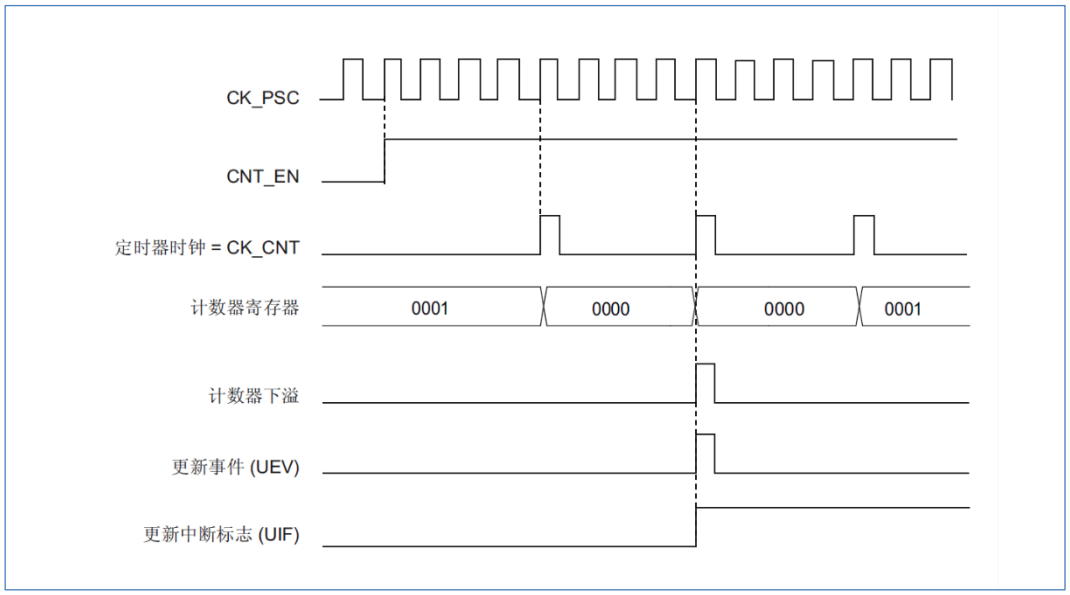


图 13-12 计数器时序图，内部时钟分频因子为 4

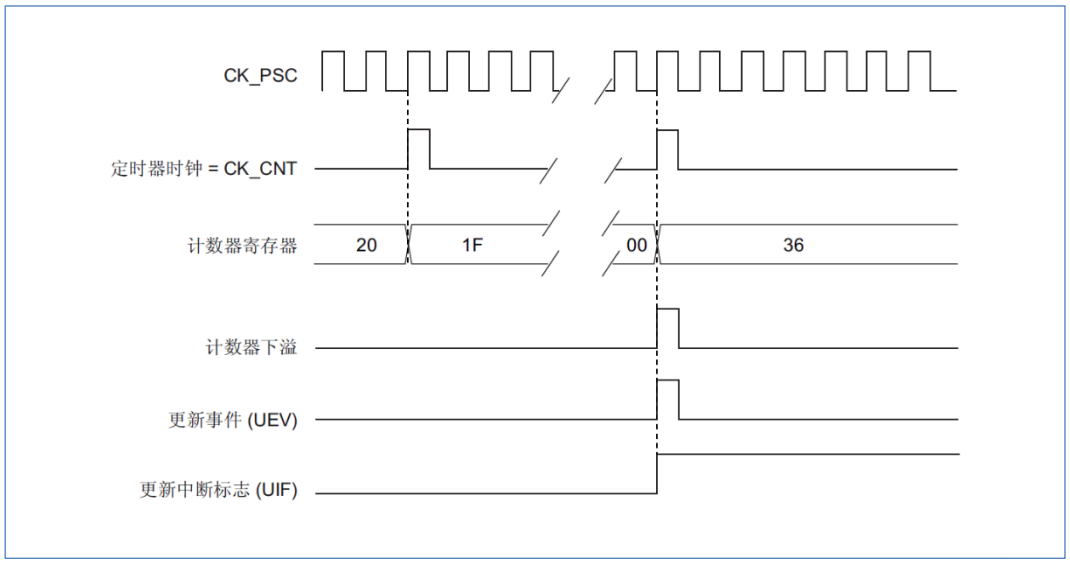


图 13-13 计数器时序图，内部时钟分频因子为 N

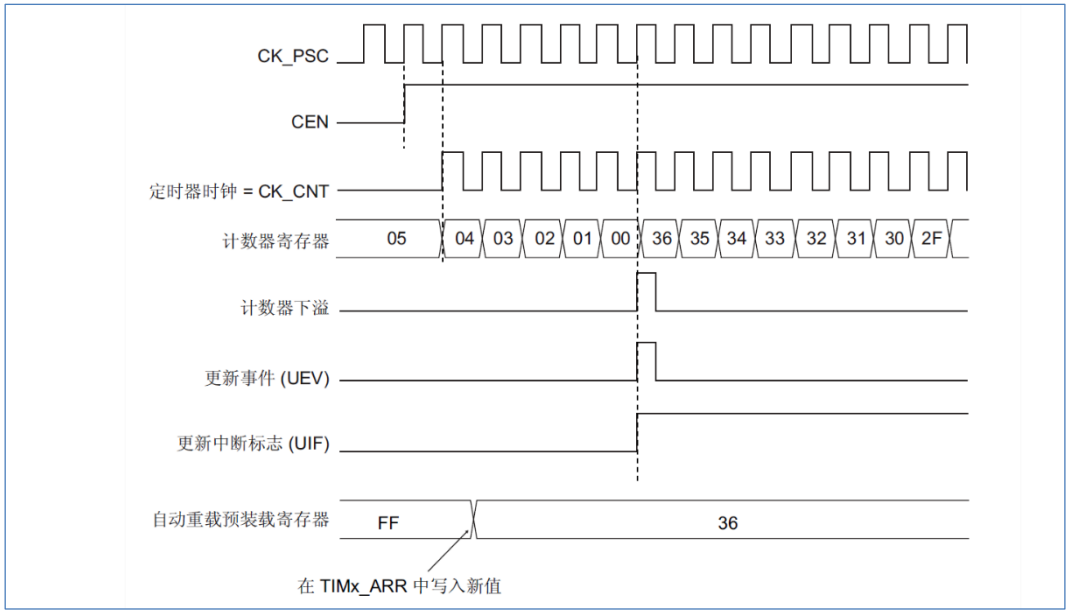


图 13-14 计数器时序图，当没有使用重复计数器时的更新事件

13.2.2.3 中央对齐模式（向上/向下计数）

在中央对齐模式，计数器从 0 开始计数到自动加载的值（TIM2_ARR 寄存器）减 1，产生一个计数器溢出事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。

在这个模式，不能写入 TIM2_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

可以在每次计数上溢和每次计数下溢时产生更新事件；也可以通过（软件或者使用从模式控制器）设置 TIM2_EGR 寄存器中的 UG 位产生更新事件。然后，计数器重新从 0 开始计数，预分频器也重新从 0 开始计数。

设置 TIM2_CR1 寄存器中的 UDIS 位可以禁止 UEV 事件。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为'0'之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

此外，如果设置了 TIM2_CR1 寄存器中的 URS 位（选择更新请求），设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志（因此不产生中断），这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且（根据 URS 位的设置）更新标志位（TIM2_SR 寄存器中的 UIF 位）也被设置。

- 预分频器的缓存器被加载为预装载（TIM2_PSC 寄存器）的值。
- 当前的自动加载寄存器被更新为预装载值（TIM2_ARR 寄存器中的内容）。

说明：如果因为计数器溢出而产生更新，自动重装载将在计数器重载入之前被更新，因此下一个周期将是预期的值（计数器被装载为新的值）。

以下是一些计数器在不同时钟频率下的操作的例子：

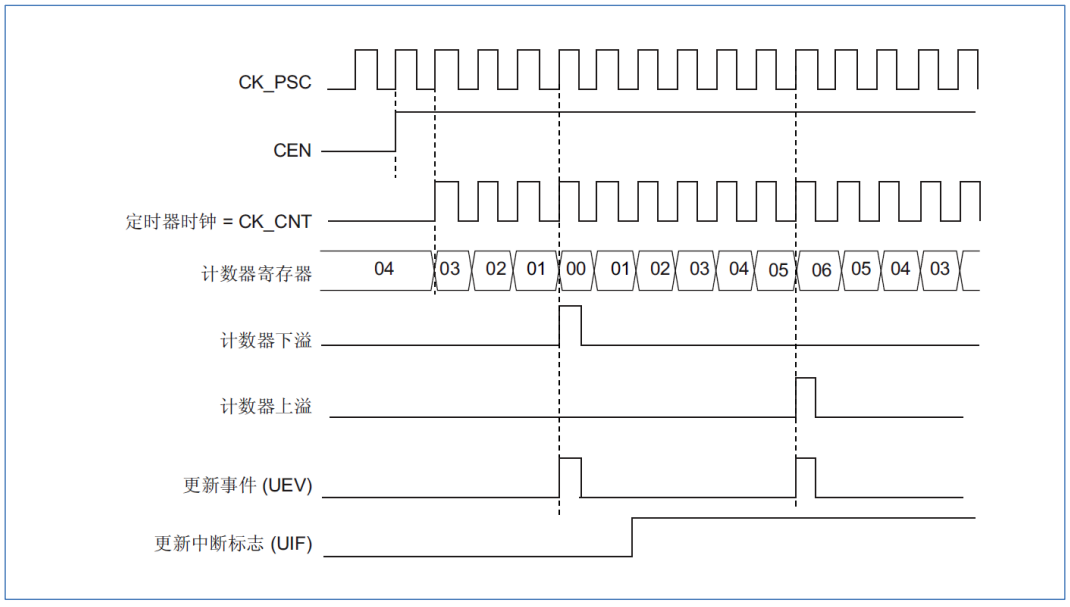


图 13-15 计数器时序图，内部时钟分频因子为 1，TIM2_ARR=0x6

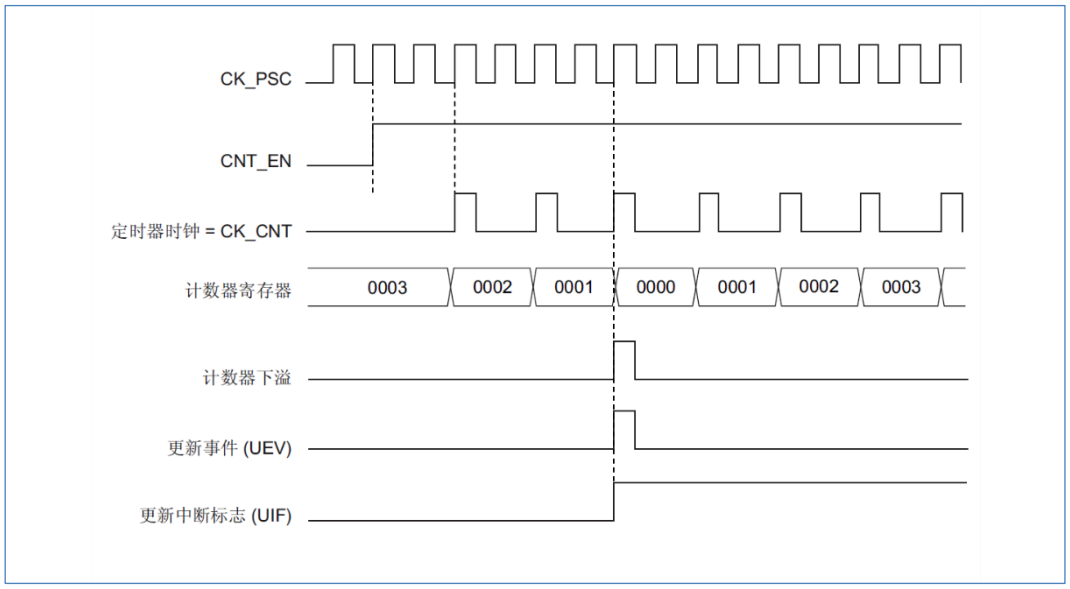


图 13-16 计数器时序图，内部时钟分频因子为 2

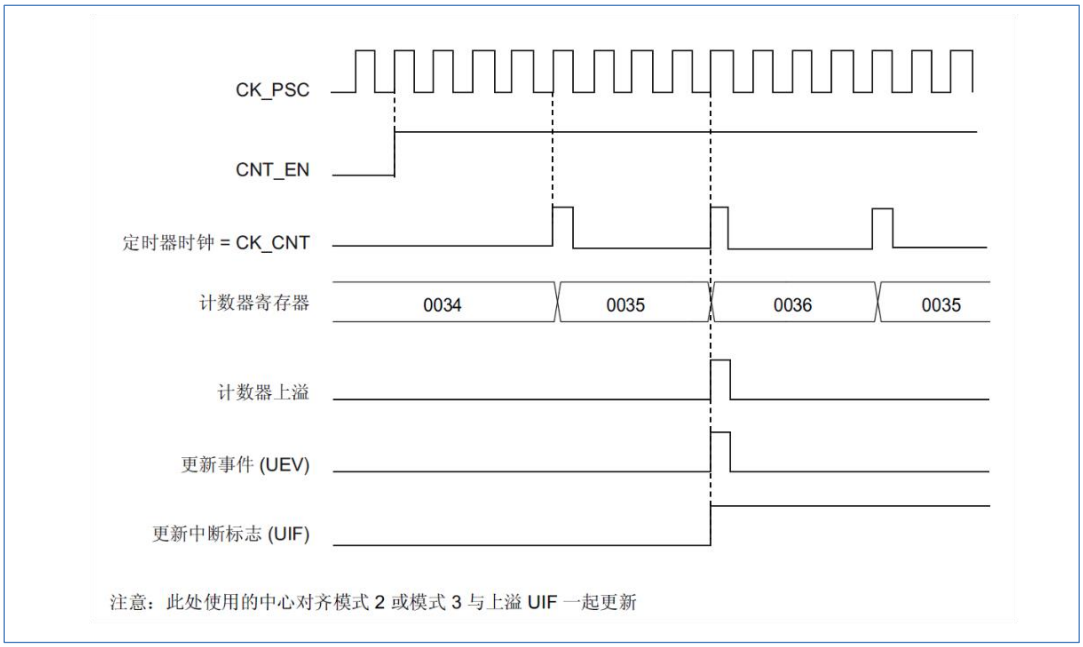


图 13-17 计数器时序图，内部时钟分频因子为 4，TIM2_ARR=0x36

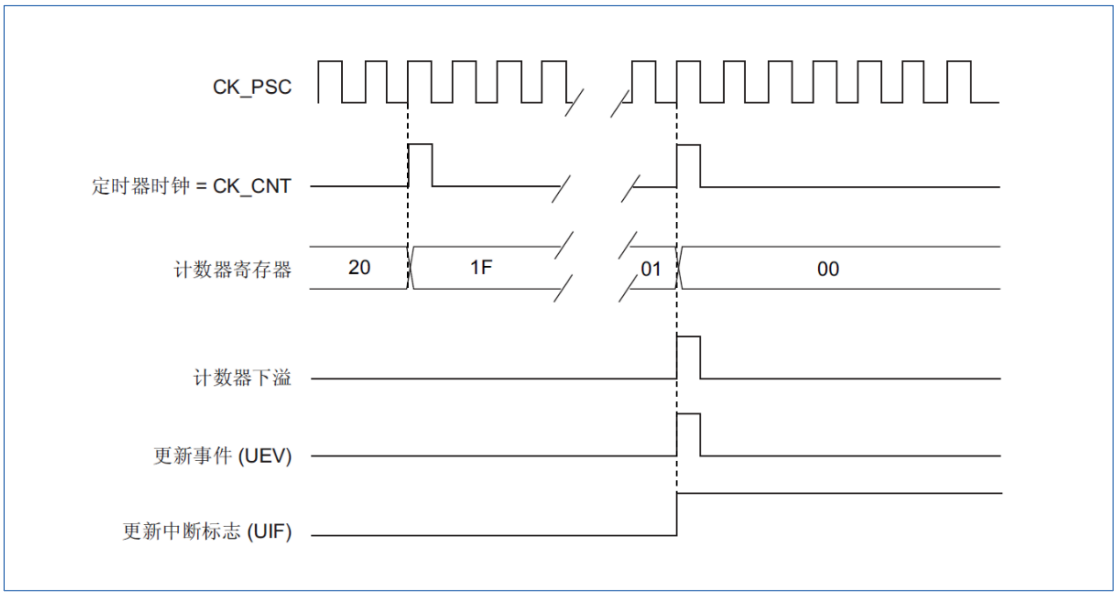


图 13-18 计数器时序图，内部时钟分频因子为 N

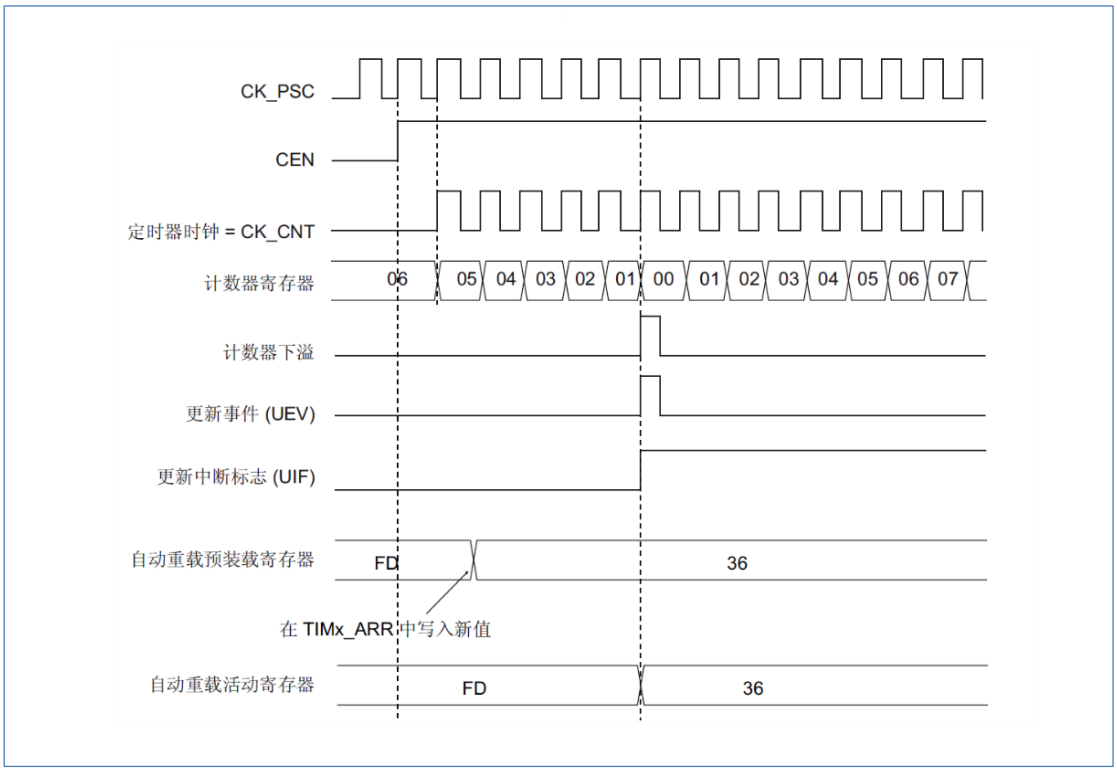


图 13-19 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

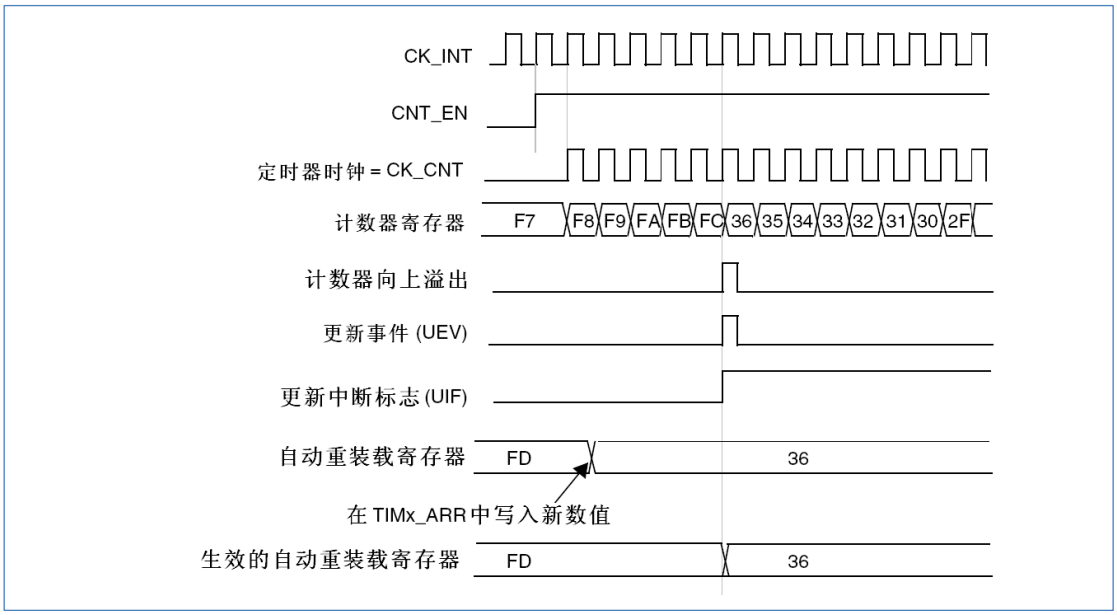


图 13-20 计数器时序图，ARPE=1 时的更新事件（计数器溢出）

13.2.3 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟（CK_INT）
- 外部时钟模式 1：外部输入脚（TIx）
- 外部时钟模式 2：外部触发输入（ETR）
- 内部触发输入（ITRx）：使用一个定时器作为另一个定时器的预分频器，如可以配置一个定时器 Timer1 作为另一个定时器 Timer2 的预分频器。详细信息，参见“13.2.15.1 使用一个定时器作为另一个定时器的预分频器”。

内部时钟源（CK_INT）

如果禁止了从模式控制器（TIM2_SMCR 寄存器的 SMS=000），则 CEN、DIR（TIM2_CR1 寄存器）和 UG 位（TIM2_EGR 寄存器）是实际的控制位，并且只能被软件修改（UG 位仍被自动清除）。只要 CEN 位被写成‘1’，预分频器的时钟就由内部时钟 CK_INT 提供。

下图显示了控制电路和向上计数器在一般模式下，不带预分频器时的操作。

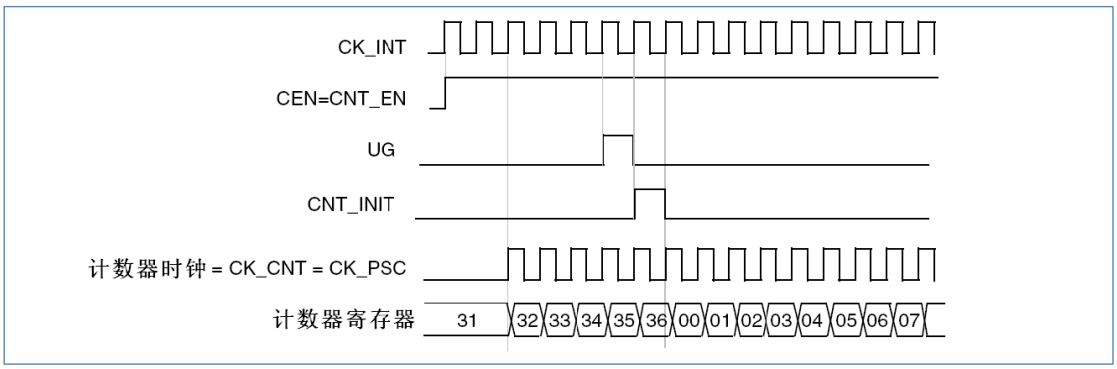


图 13-21 一般模式下的控制电路，内部时钟分频因子为 1

外部时钟源模式 1

当 TIM2_SMCR 寄存器的 SMS=111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

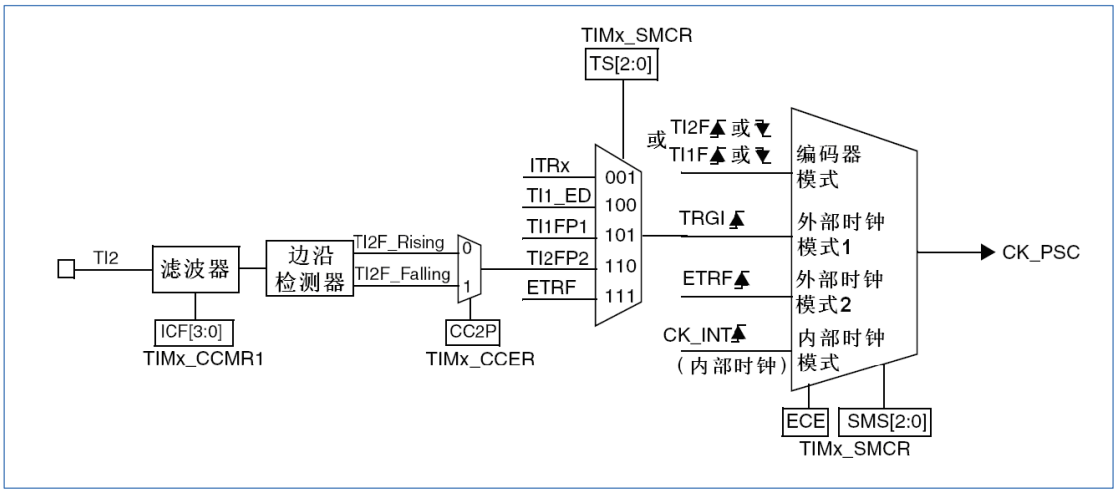


图 13-22 TI2 外部时钟连接例子

例如，配置向上计数器在 T12 输入端的上升沿计数，使用下列步骤：

1. 配置 TIM2_CCMR1 寄存器 CC2S=’01’，配置通道 2 检测 TI2 输入的上升沿。
2. 配置 TIM2_CCMR1 寄存器的 IC2F[3:0]，选择输入滤波器带宽（如果不需要滤波器，保持 IC2F=0000）。

   说明：捕获预分频器不用作触发，所以不需要对它进行配置。
3. 配置 TIM2_CCER 寄存器的 CC2P=‘0’，选定上升沿极性。
4. 配置 TIM2_SMCR 寄存器的 SMS=’111’，选择定时器外部时钟模式 1。
5. 配置 TIM2_SMCR 寄存器中的 TS=’110’，选定 TI2 作为触发输入源。
6. 设置 TIM2_CR1 寄存器的 CEN=‘1’，启动计数器。

当上升沿出现在 TI2，计数器计数一次，且 TIF 标志被设置。

在 TI2 的上升沿和计数器实际时钟之间的延时，取决于在 TI2 输入端的重新同步电路。

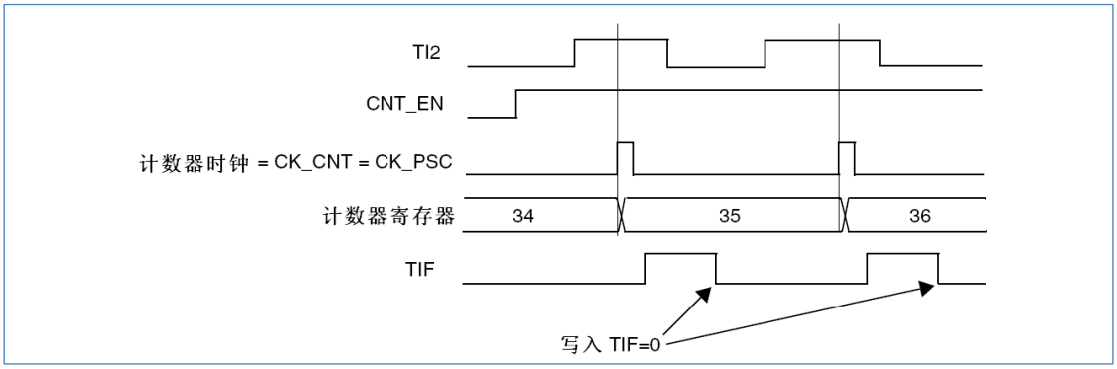


图 13-23 外部时钟模式 1 下的控制电路

外部时钟源模式 2

选定此模式的方法为：令 TIM2_SMCR 寄存器中的 ECE=1。计数器能够在外部触发 ETR 的每一个上升沿或下降沿计数。

下图是外部触发输入的框图。

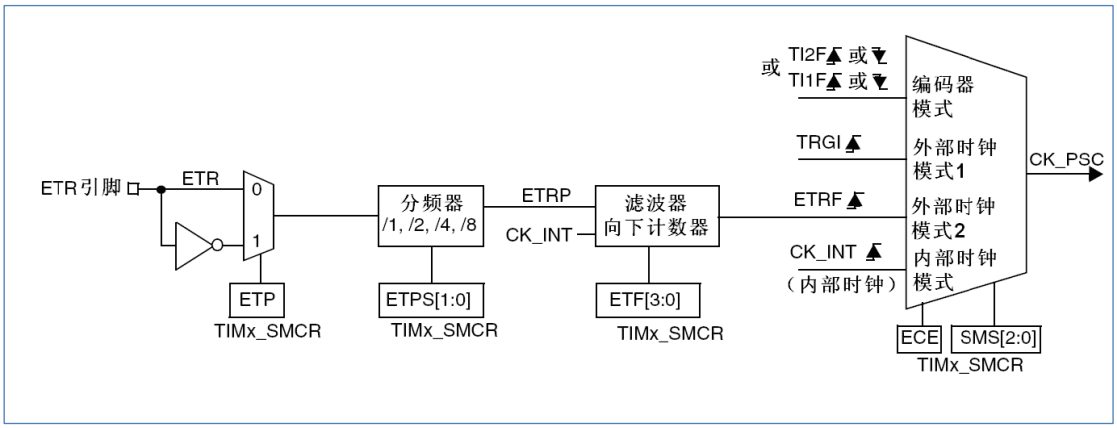


图 13-24 外部触发输入框图

例如，配置在 ETR 上每 2 个上升沿计数一次的向上计数器，使用下列步骤：

1. 本例中不需要滤波器，置 TIM2_SMCR 寄存器中的 ETF[3:0]=0000。
2. 设置预分频器，置 TIM2_SMCR 寄存器中的 ETPS[1:0]=01。
3. 设置在 ETR 的上升沿检测，置 TIM2_SMCR 寄存器中的 ETP=0。
4. 开启外部时钟模式 2，置 TIM2_SMCR 寄存器中的 ECE=1。
5. 启动计数器，置 TIM2_CR1 寄存器中的 CEN=1。

计数器在每 2 个 ETR 上升沿计数一次。在 ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的重新同步电路。

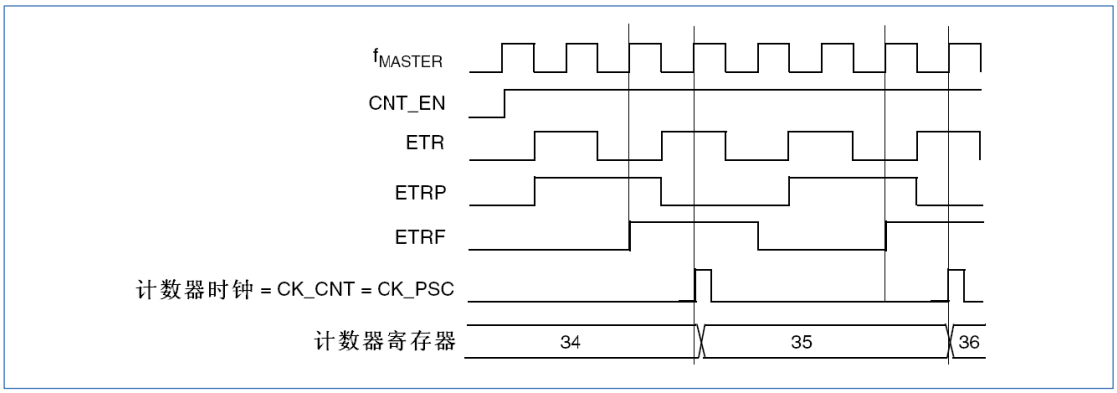


图 13-25 外部时钟模式 2 下的控制电路

13.2.4 捕获/比较通道

每一个捕获/比较通道都是围绕着一个捕获/比较寄存器（包含影子寄存器），包括捕获的输入部分（数字滤波、多路复用和预分频器），和输出部分（比较器和输出控制）。

下面几张图是一个捕获/比较通道概览。

输入部分对相应的 TIx 输入信号采样，并产生一个滤波后的信号 TIxF。然后，一个带极性选择的边沿检测器产生一个信号（TIxFPx），它可以作为从模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器（ICxPS）。

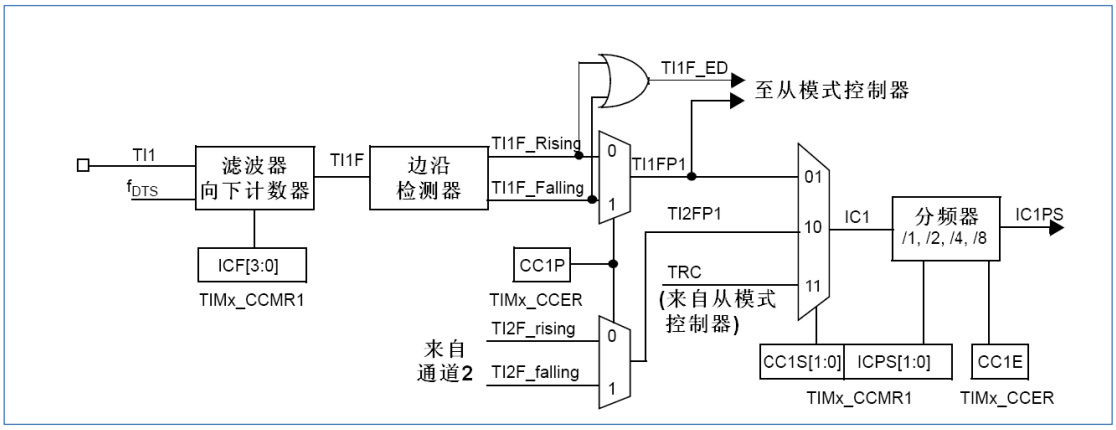


图 13-26 捕获/比较通道（如：通道 1 输入部分）

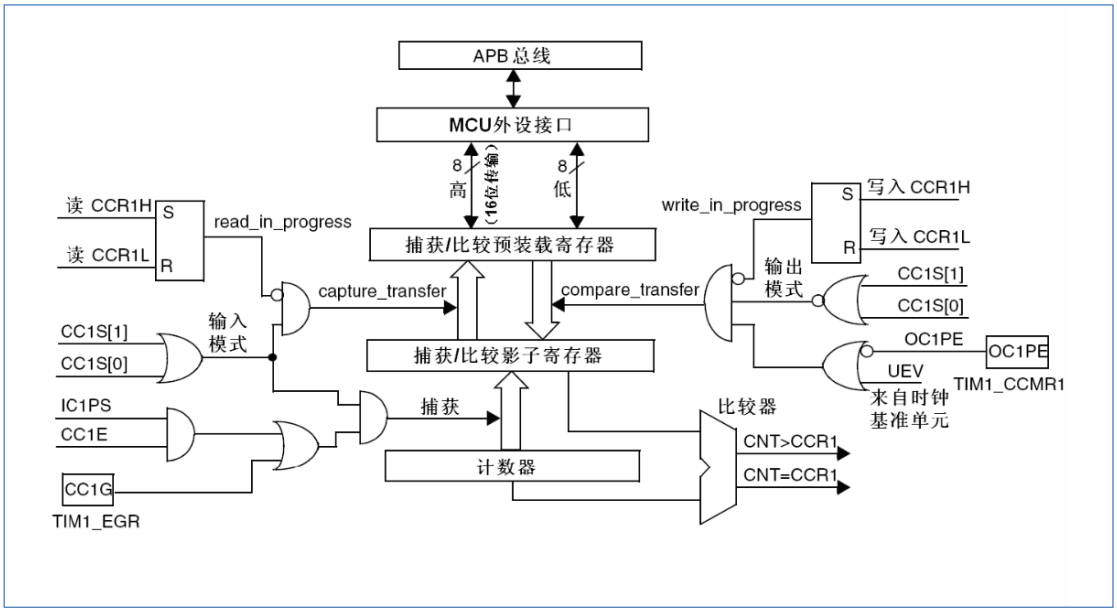


图 13-27 捕获/比较通道 1 的主电路

输出部分产生一个中间波形 OCxRef（高有效）作为基准，链的末端决定最终输出信号的极性。

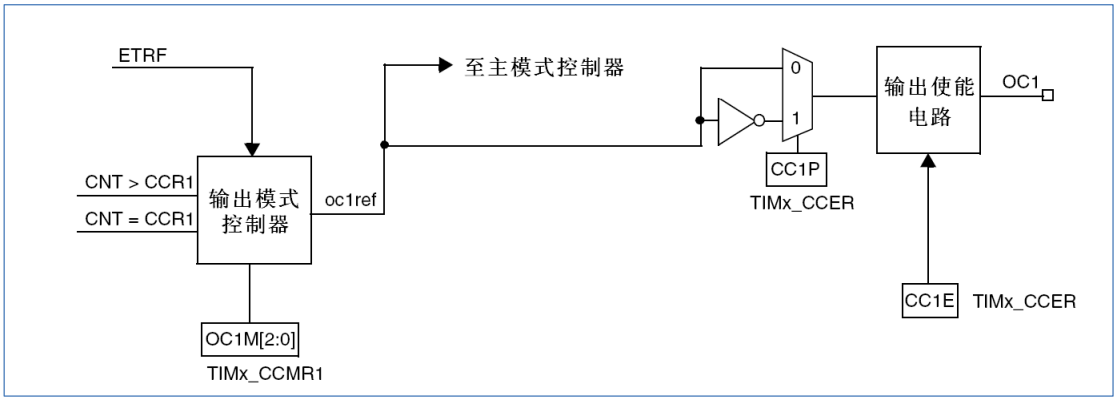


图 13-28 捕获/比较通道的输出部分（通道 1）

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。读写过程仅操作预装载寄存器。

在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

13.2.5 输入捕获模式

在输入捕获模式下，当检测到 ICx 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（TIM2_CCRx）中。当捕获事件发生时，相应的 CCxIF 标志（TIM2_SR 寄存器）被置'1'，如果使能了中断，则将产生中断。如果捕获事件发生时 CCxIF 标志已经为高，那么重复捕获标志 CCxOF（TIM2_SR 寄存器）被置'1'。写 CCxIF=0 可清除 CCxIF，或读取存储在 TIM2_CCRx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIM2_CCR1 寄存器中，步骤如下：

1. 选择有效输入端：TIM2_CCR1 必须连接到 TI1 输入，所以写入 TIM2_CCMR1 寄存器中的 CC1S=01。只要 CC1S 不为'00'，通道将被配置为输入并且 TIM2_CCR1 寄存器变为只读。
2. 根据输入信号的特点，配置输入滤波器为所需的带宽（即输入为 TIx 时，输入滤波器控制位是 TIM2_CCMRx 寄存器中的 ICxF 位）。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽大于 5 个时钟周期。因此我们可以（以 fDTS 频率）连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIM2_CCMR1 寄存器中写入 IC1F=0011。
3. 选择 TI1 通道的有效转换边沿，在 TIM2_CCER 寄存器中写入 CC1P=0（上升沿）。
4. 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止（写 TIM2_CCMR1 寄存器的 IC1PS=00）。
5. 设置 TIM2_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
6. 如果需要，通过设置 TIM2_DIER 寄存器中的 CC1IE 位允许相关中断请求。

当发生一个输入捕获时：

- 产生有效的电平转换时，计数器的值被传送到 TIM2_CCR1 寄存器。
- CC1IF 标志被设置（中断标志）。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置'1'。
- 如设置了 CC1IE 位，则会产生一个中断。

为了处理捕获溢出，建议在读出捕获溢出标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

说明：设置 TIM2_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断。

13.2.6 PWM 输入模式

该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

1. 两个 ICx 信号被映射至同一个 TIx 输入。
2. 这 2 个 ICx 信号为边沿有效，但是极性相反。
3. 其中一个 TIxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。
4. 例如，你需要测量输入到 TI1 上的 PWM 信号的长度（TIM2_CCR1 寄存器）和占空比（TIM2_CCR2 寄存器），具体步骤如下（取决于 CK_INT 的频率和预分频器的值）。
5. 选择 TIM2_CCR1 的有效输入：置 TIM2_CCMR1 寄存器的 CC1S=01（选择 TI1）。
6. 选择 TI1FP1 的有效极性（用来捕获数据到 TIM2_CCR1 中和清除计数器）：置 CC1P=0（上升沿有效）。
7. 选择 TIM2_CCR2 的有效输入：置 TIM2_CCMR1 寄存器的 CC2S=10（选择 TI1）。

8. 选择 TI1FP2 的有效极性（捕获数据到 TIM2_CCR2）：置 CC2P=1（下降沿有效）。

9. 选择有效的触发输入信号：置 TIM2_SMCR 寄存器中的 TS=101（选择 TI1FP1）。

10. 配置从模式控制器为复位模式：置 TIM2_SMCR 中的 SMS=100。

11. 使能捕获：置 TIM2_CCER 寄存器中 CC1E=1 且 CC2E=1。

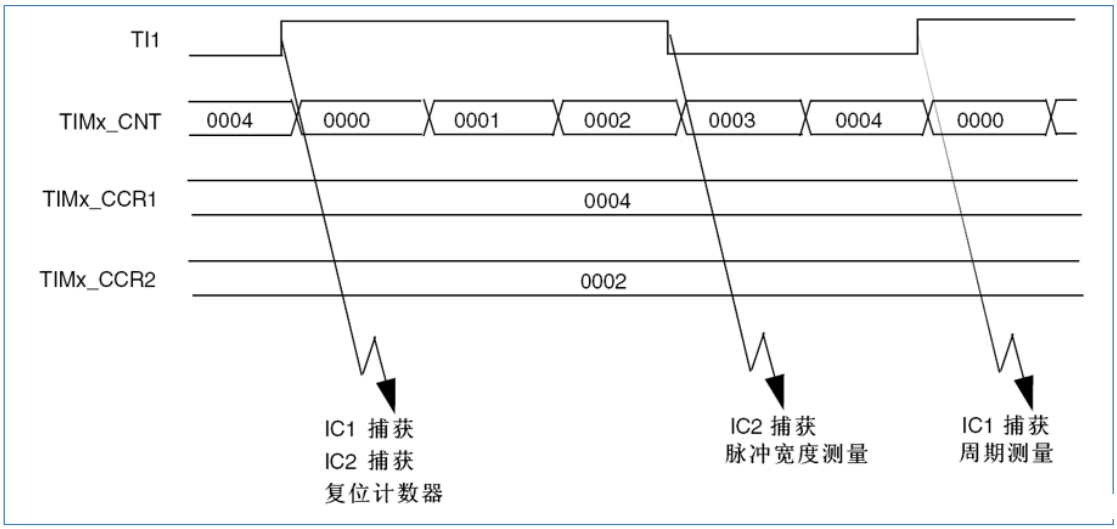


图 13-29 PWM 输入模式时序

由于只有 TI1FP1 和 TI2FP2 连到了从模式控制器，所以 PWM 输入模式只能使用 TIM2_CH1/TIM2_CH2 信号。

13.2.7 强置输出模式

在输出模式（TIM2_CCMRx 寄存器中 CCxS=00）下，输出比较信号（OCxREF 和相应的 OCx）能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

置 TIM2_CCMRx 寄存器中相应的 OCxM=101，即可强置输出比较信号（OCxREF/OCx）为有效状态。这样 OCxREF 被强置为高电平（OCxREF 始终为高电平有效），同时 OCx 得到 CCxP 极性位相反的值。

例如：CCxP=0（OCx 高电平有效），则 OCx 被强置为高电平。

置 TIM2_CCMRx 寄存器中的 OCxM=100，可强置 OCxREF 信号为低。

该模式下，在 TIM2_CCRx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断。这将会在下面的输出比较模式一节中介绍。

13.2.8 输出比较模式

此项功能是用来控制一个输出波形，或者指示一段给定的时间已经到时。当计数器与捕获/比较寄存器的内容相同时，输出比较功能按如下操作：

将输出比较模式（TIM2_CCMRx 寄存器中的 OCxM 位）和输出极性（TIM2_CCER 寄存器中的 CCxP 位）定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平（OCxM=000）、被设置成有效电平（OCxM=001）、被设置成无效电平（OCxM=010）或进行翻转（OCxM=011）。

- 设置中断状态寄存器中的标志位（TIM2_SR 寄存器中的 CCxIF 位）。
- 若设置了相应的中断屏蔽（TIM2_DIER 寄存器中的 CCxIE 位），则产生一个中断。

TIM2_CCMRx 中的 OCxPE 位选择 TIM2_CCRx 寄存器是否需要使用预装载寄存器。

在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。

同步的精度可以达到计数器的一个计数周期。输出比较模式（在单脉冲模式下）也能用来输出一个

单脉冲。

输出比较模式的配置步骤：

1. 选择计数器时钟（内部、外部、预分频器）。
2. 将相应的数据写入 TIM2_ARR 和 TIM2_CCRx 寄存器中。
3. 如果要产生一个中断请求，设置 CCxIE 位。
4. 选择输出模式，例如当计数器 CNT 与 CCRx 匹配时翻转 OCx 的输出引脚，CCRx 预装载未用，开启 OCx 输出且高电平有效，则必须设置 OCxM=’011’、OCxPE=‘0’、CCxP=‘0’和 CCxE=‘1’。
5. 设置 TIM2_CR1 寄存器的 CEN 位启动计数器。

TIM2_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器（OCxPE=‘0’，否则 TIM2_CCRx 影子寄存器只能在发生下一次更新事件时被更新）。下图给出了一个例子。

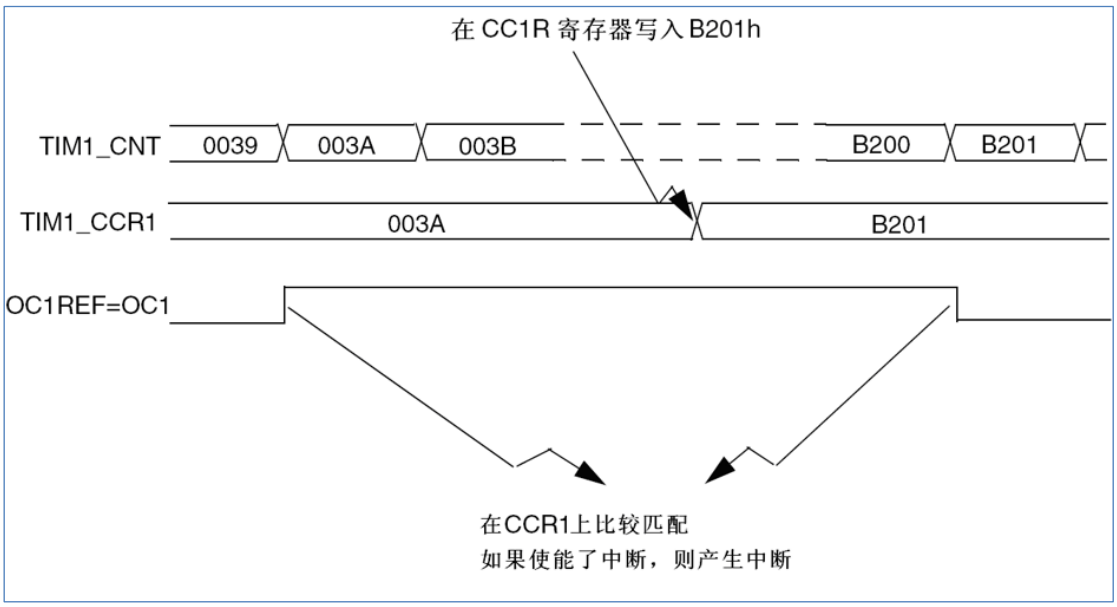


图 13-30 输出比较模式，翻转 OC1

13.2.9 PWM 模式

脉冲宽度调制模式可以产生一个由 TIM2_ARR 寄存器确定频率、由 TIM2_CCRx 寄存器确定占空比的信号。

在 TIM2_CCMRx 寄存器中的 OCxM 位写入’110’（PWM 模式 1）或’111’（PWM 模式 2），能够独立地设置每个 OCx 输出通道产生一路 PWM。必须设置 TIM2_CCMRx 寄存器 OCxPE 位以使能相应的预装载寄存器，最后还要设置 TIM2_CR1 寄存器的 ARPE 位，（在向上计数或中心对称模式中）使能自动重装载的预装载寄存器。

仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 TIM2_EGR 寄存器中的 UG 位来初始化所有的寄存器。OCx 的极性可以通过软件在 TIM2_CCER 寄存器中的 CCxP 位设置，它可以设置为高电平有效或低电平有效。TIM2_CCER 寄存器中的 CCxE 位控制 OCx 输出使能。参见 TIM2 捕捉/比较使能寄存器（TIM2_CCER）的描述。

在 PWM 模式（模式 1 或模式 2）下，TIM2_CNT 和 TIM2_CCRx 始终在进行比较，（依据计数器的计数方向）以确定是否符合 TIM2_CCRx≤TIM2_CNT 或者 TIM2_CNT≤TIM2_CCRx。然而为了与 OCREF_CLR 的功能（在下一个 PWM 周期之前，ETR 信号上的一个外部事件能够清除 OCxREF）一致，OCxREF 信号只能

在下述条件下产生：

- 当比较的结果改变
- 当输出比较模式（TIM2_CCMRx 寄存器中的 OCxM 位）从“冻结”（无比较，OCxM=’000’）切换到某个 PWM 模式（OCxM=’110’或’111’）。

这样在运行中可以通过软件强置 PWM 输出。

根据 TIM2_CR1 寄存器中 CMS 位的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

13.2.9.1 PWM 边沿对齐模式

向上计数配置

当 TIM2_CR1 寄存器中的 DIR 位为低的时候执行向上计数。参看章节计数器模式。

下面是一个 PWM 模式 1 的例子。当 TIM2_CNT<TIM2_CCRx 时 PWM 信号参考 OCxREF 为高，否则为低。如果 TIM2_CCRx 中的比较值大于自动重装载值（TIM2_ARR），则 OCxREF 保持为’1’。如果比较值为 0，则 OCxREF 保持为’0’。下图为 TIM2_ARR=8 时边沿对齐的 PWM 波形实例。

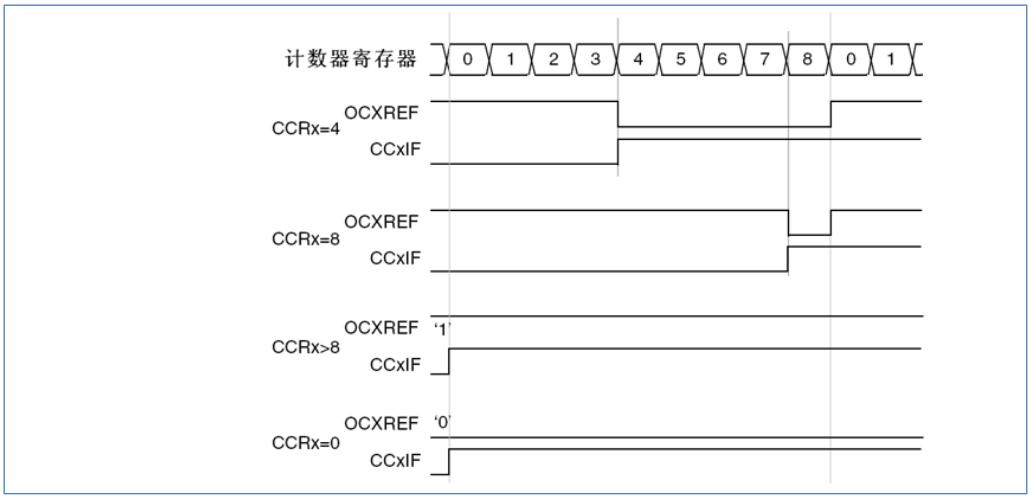


图 13-31 边沿对齐的 PWM 波形（ARR=8）

向下计数的配置

当 TIM2_CR1 寄存器的 DIR 位为高时执行向下计数。参看章节计数器模式节。

在 PWM 模式 1，当 TIM2_CNT>TIM2_CCRx 时参考信号 OCxREF 为低，否则为高。如果 TIM2_CCRx 中的比较值大于 TIM2_ARR 中的自动重装载值，则 OCxREF 保持为’1’。该模式下不能产生 0%的 PWM 波形。

13.2.9.2 PWM 中央对齐模式

当 TIM2_CR1 寄存器中的 CMS 位不为’00’时，为中央对齐模式（所有其他的配置对 OCxREF/OCx 信号都有相同的作用）。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时被置’1’、在计数器向下计数时被置’1’、或在计数器向上和向下计数时被置’1’。TIM2_CR1 寄存器中的计数方向位（DIR）由硬件更新，不用软件修改。参见“13.2.2 计数器模式的中央对齐模式”。

下图给出了一些中央对齐的 PWM 波形的例子。

- TIM2_ARR=8
- PWM 模式 1
- TIM2_CR1 寄存器中的 CMS=01，在中央对齐模式 1 时，当计数器向下计数时设置比较标志。

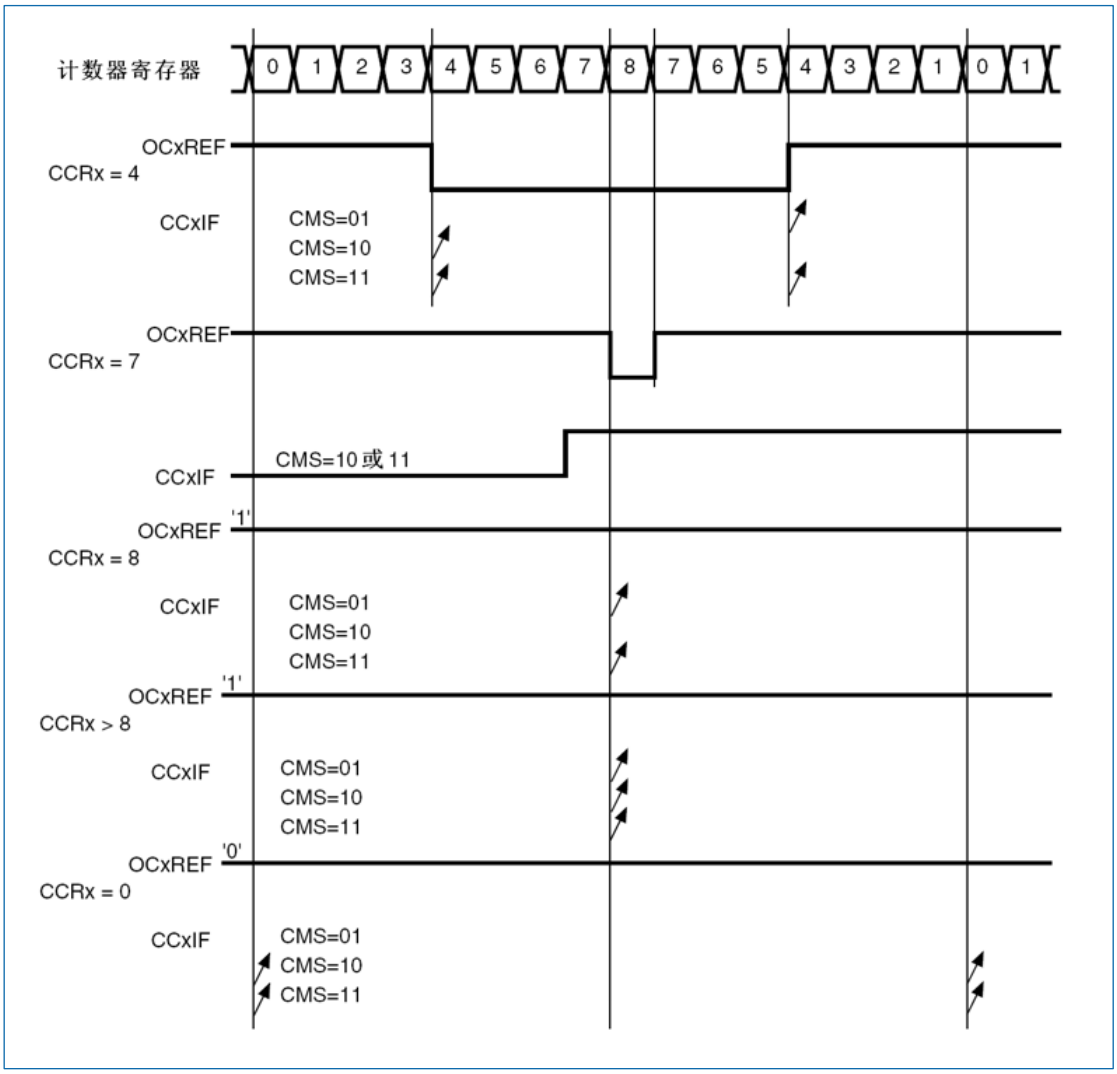


图 13-32 中央对齐的 PWM 波形（APR=8）

使用中央对齐模式的提示：

- 进入中央对齐模式时，使用当前的向上/向下计数配置；这就意味着计数器向上还是向下计数取决于 TIM2_CR1 寄存器中 DIR 位的当前值。此外，软件不能同时修改 DIR 和 CMS 位。
- 不推荐当运行在中央对齐模式时改写计数器，因为这会产生不可预知的结果。特别是：
  - 如果写入计数器的值大于自动重加载的值（TIM2_CNT>TIM2_ARR），则方向不会被更新。例如，如果计数器正在向上计数，它就会继续向上计数。
  - 如果将 0 或者 TIM2_ARR 的值写入计数器，方向被更新，但不产生更新事件 UEV。
- 使用中央对齐模式最保险的方法，就是在启动计数器之前产生一个软件更新（设置 TIM2_EGR 位中的 UG 位），不要在计数进行时修改计数器的值。

13.2.10 单脉冲模式

单脉冲模式（OPM）是前述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后，产生一个脉宽可程序控制的脉冲。

可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIM2_CR1 寄存器中的 OPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件 UEV 时停止。

仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前（当定时器正在等待触发），必须如下配置：

- 向上计数方式：CNT<CCRx≤ARR（特别地，0<CCRx）

- 向下计数方式：CNT>CCRx

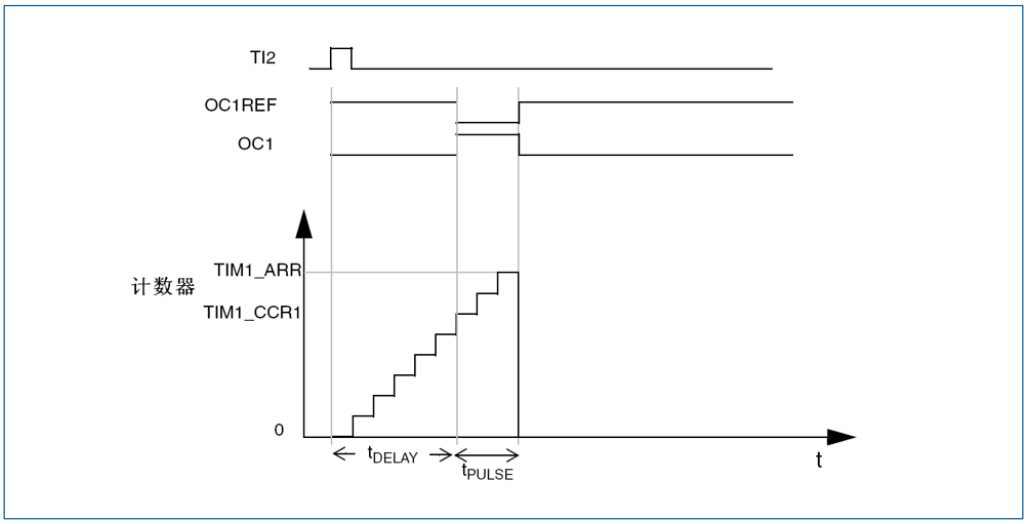


图 13-33 单脉冲模式的例子

例如，你需要在从 TI2 输入脚上检测到一个上升沿开始，延迟 t_{DELAY} 之后，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

假定 TI2FP2 作为触发 1：

- 置 TIM2_CCMR1 寄存器中的 CC2S='01'，把 TI2FP2 映射到 TI2。
- 置 TIM2_CCER 寄存器中的 CC2P=‘0’，使 TI2FP2 能够检测上升沿。
- 置 TIM2_SMCR 寄存器中的 TS='110'，TI2FP2 作为从模式控制器的触发（TRGI）。
- 置 TIM2_SMCR 寄存器中的 SMS='110'（触发模式），TI2FP2 被用来启动计数器。

OPM 波形由写入比较寄存器的数值决定（要考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIM2_CCR1 寄存器中的值定义。
- t_{PULSE} 由自动装载值和比较值之间的差值定义（TIM2_ARR-TIM2_CCR1）。
- 假定当发生比较匹配时要产生从'0'到'1'的波形，当计数器到达预装载值时要产生一个从'1'到'0'的波形；首先要置 TIM2_CCMR1 寄存器的 OC1M='111'，进入 PWM 模式 2；根据需要有选择地使能预装载寄存器：置 TIM2_CCMR1 中的 OC1PE=‘1’和 TIM2_CR1 寄存器中的 ARPE；然后在 TIM2_CCR1 寄存器中填写比较值，在 TIM2_ARR 寄存器中填写自动装载值，修改 UG 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P=‘0’。

在这个例子中，TIM2_CR1 寄存器中的 DIR 和 CMS 位应该置低。

因为只需一个脉冲，所以必须设置 TIM2_CR1 寄存器中的 OPM=‘1’，在下一个更新事件（当计数器从自动装载值翻转到 0）时停止计数。

特殊情况：OCx 快速使能：

在单脉冲模式下，在 TIx 输入脚的边沿检测逻辑设置 CEN 位以启动计数器。然后计数器和比较值间的比较操作产生了输出的转换。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 t_{DELAY}。

如果要以最小延时输出波形，可以设置 TIM2_CCMRx 寄存器中的 OCxFE 位；此时 OCxREF（和 OCx）被强制响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCxFE 只在通道配置为 PWM1 和 PWM2 模式时起作用。

13.2.11 在外部事件时清除 OCxREF 信号

对于一个给定的通道，设置 TIM2_CCMRx 寄存器中对应的 OCxCE 位为'1'，能够用 ETRF 输入端的高

电平把 OCxREF 信号拉低，OCxREF 信号将保持为低直到发生下一次的更新事件 UEV。

该功能只能用于输出比较和 PWM 模式，而不能用于强置模式。

例如，OCxREF 信号可以联到一个比较器的输出，用于控制电流。这时，ETR 必须配置如下：

1. 外部触发预分频器必须处于关闭：TIM2_SMCR 寄存器中的 ETPS[1:0]=’00’。
2. 必须禁止外部时钟模式 2：TIM2_SMCR 寄存器中的 ECE=‘0’。
3. 外部触发极性（ETP）和外部触发滤波器（ETF）可以根据需要配置。

下图显示了当 ETRF 输入变为高时，对应不同 OCxCE 的值，OCxREF 信号的动作。在这个例子中，定时器 TIM2 被置于 PWM 模式。

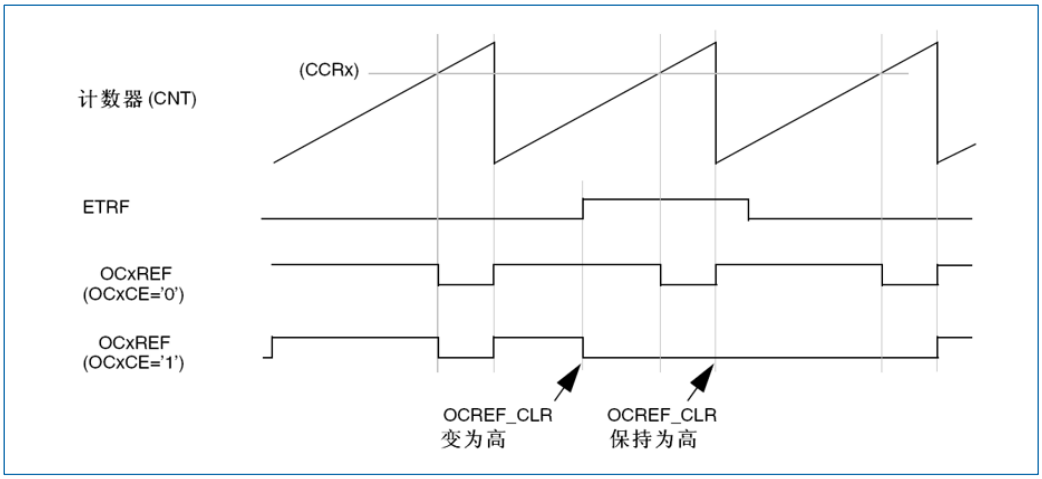


图 13-34 清除 TIM2 的 OCxREF

13.2.12 编码器接口模式

选择编码器接口模式的方法是：如果计数器只在 TI2 的边沿计数，则置 TIM2_SMCR 寄存器中的 SMS=001；如果只在 TI1 边沿计数，则置 SMS=010；如果计数器同时在 TI1 和 TI2 边沿计数，则置 SMS=011。

通过设置 TIM2_CCER 寄存器中的 CC1P 和 CC2P 位，可以选择 TI1 和 TI2 极性；如果需要，还可以对输入滤波器编程。

两个输入 TI1 和 TI2 被用来作为增量编码器的接口。参见表 13-2，假定计数器已经启动（TIM2_CR1 寄存器中的 CEN=‘1’），计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和变相，则 TI1FP1=TI1，TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对 TIM2_CR1 寄存器的 DIR 位进行相应的设置。不管计数器是依靠 TI1 计数、依靠 TI2 计数或者同时依靠 TI1 和 TI2 计数。在任一输入端（TI1 或者 TI2）的跳变都会重新计算 DIR 位。

编码器接口模式基本上相当于使用了一个带有方向选择的外部时钟。这意味着计数器只在 0 到 TIM2_ARR 寄存器的自动装载值之间连续计数（根据方向，或是 0 到 ARR 计数，或是 ARR 到 0 计数）。所以在开始计数之前必须配置 TIM2_ARR；同样，捕获器、比较器、预分频器、触发输出特性等仍工作如常。

在这个模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的组合，假设 TI1 和 TI2 不同时变换。

表 13-2 计数方向与编码器信号的关系

有效边沿	相对信号的电平（TI1FP1 对应 TI2、TI2FP2 对应 TI1）	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
仅在 TI1 计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在 TI2 计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在 TI1 和 TI2 上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

一个外部的增量编码器可以直接与 MCU 连接而不需要外部接口逻辑。但是，一般会使用比较器将编码器的差分输出转换到数字信号，这大大增加了抗噪声干扰能力。编码器输出的第三个信号表示机械零点，可以把它连接到一个外部中断输入并触发一个计数器复位。

下图是一个计数器操作的实例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。在这个例子中，我们假定配置如下：

- CC1S=’01’（TIM2_CCMR1 寄存器，IC1FP1 映射到 TI1）
- CC2S=’01’（TIM2_CCMR2 寄存器，IC2FP2 映射到 TI2）
- CC1P=‘0’（TIM2_CCER 寄存器，IC1FP1 不反相，IC1FP1=TI1）
- CC2P=‘0’（TIM2_CCER 寄存器，IC2FP2 不反相，IC2FP2=TI2）
- SMS=’011’（TIM2_SMCR 寄存器，所有的输入均在上升沿和下降沿有效）.
- CEN=‘1’（TIM2_CR1 寄存器，计数器使能）

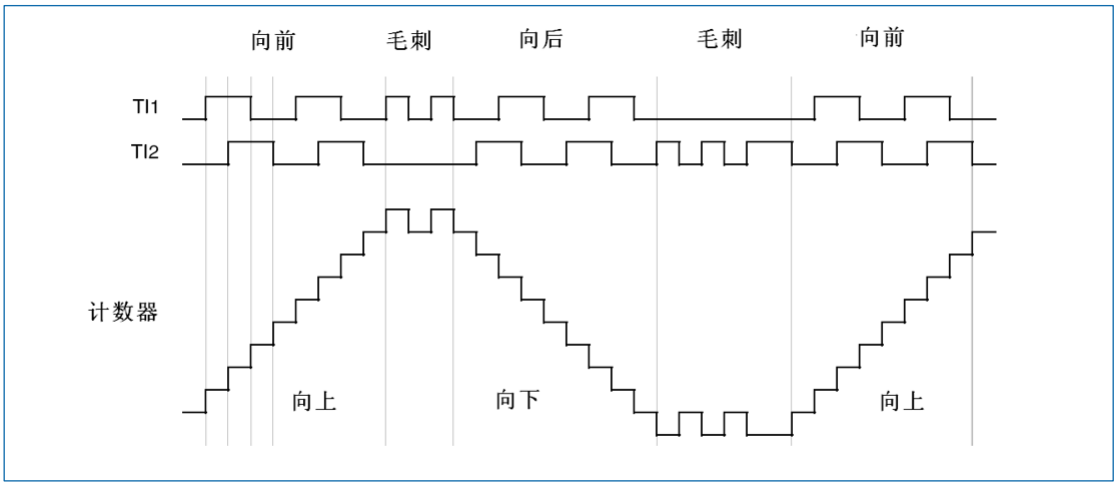


图 13-35 编码器模式下的计数器操作实例

下图为当 IC1FP1 极性反相时计数器的操作实例（CC1P=‘1’，其他配置与上例相同）。

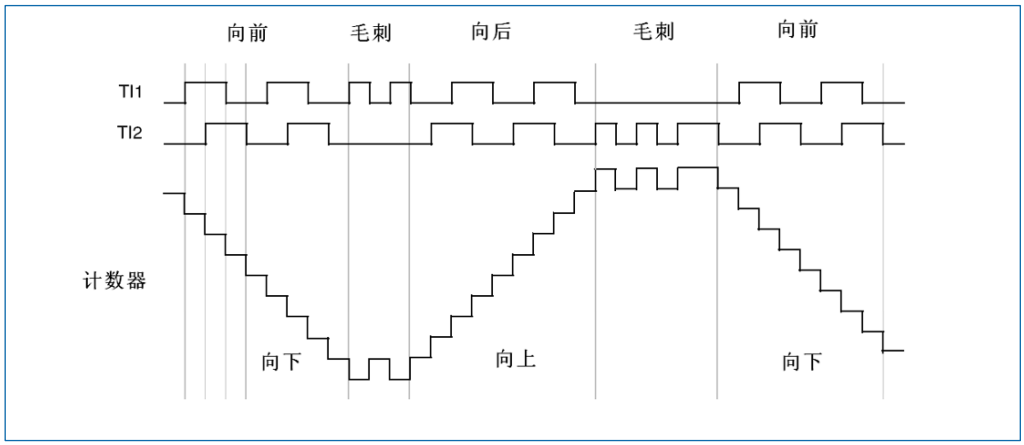


图 13-36 IC1FP1 反相的编码器接口模式实例

当定时器配置成编码器接口模式时，提供传感器当前位置的信息。使用第二个配置在捕获模式的定时器，可以测量两个编码器事件的间隔，获得动态的信息（速度，加速度，减速度）。指示机械零点的编码器输出可被用做此目的。根据两个事件间的间隔，可以按照固定的时间读出计数器。如果可能的话，你可以把计数器的值锁存到第三个输入捕获寄存器（捕获信号必须是周期的并且可以由另一个定时器产生）。

13.2.13 定时器输入异或功能

TIM2_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIM2_CH1、TIM2_CH2 和 TIM2_CH3。

异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。

13.2.14 定时器和外部触发的同步

TIM2 定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

13.2.14.1 从模式：复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIM2_CR1 寄存器的 URS 位为低，还会产生一个更新事件 UEV；然后所有的预装载寄存器（TIM2_ARR，TIM2_CCRx）都会被更新。

在下面的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽（在本例中，不需要任何滤波器，因此保持 IC1F=0000）。触发操作中不使用捕获预分频器，所以不需要配置它。CC1S 位只选择输入捕获源，即 TIM2_CCMR1 寄存器中 CC1S=01。置 TIM2_CCER 寄存器中 CC1P=0 以确定极性（只检测上升沿）。
- 置 TIM2_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIM2_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIM2_CR1 寄存器中 CEN=1，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志（TIM2_SR 寄存器中的 TIF 位）被设置，根据 TIM2_DIER 寄存器中 TIE（中断使能）位的设置，产生一个中断请求。

下图显示当自动重装载寄存器 TIM2_ARR=0x36 时的动作。在 TI1 上升沿和计数器的实际复位之间的延时，取决于 TI1 输入端的重同步电路。

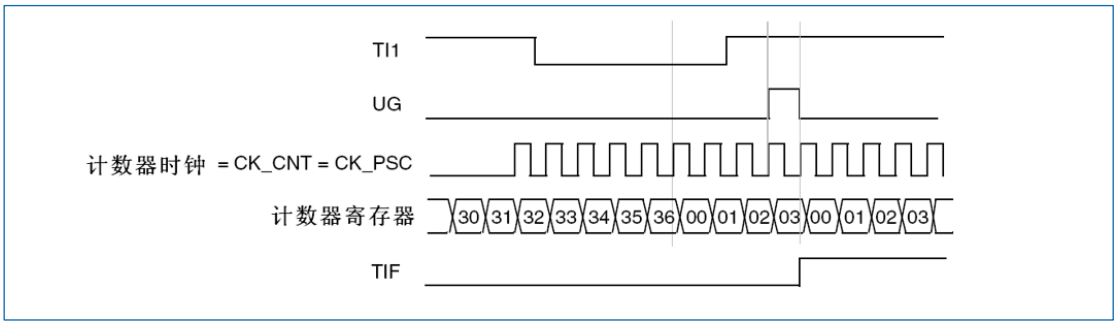


图 13-37 复位模式下的控制电路

13.2.14.2 从模式：门控模式

按照选中的输入端电平使能计数器。

在如下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽（本例中，不需要滤波，所以保持 IC1F=0000）。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIM2_CCMR1 寄存器中 CC1S=01。置 TIM2_CCER 寄存器中 CC1P=1 以确定极性（只检测低电平）。
- 置 TIM2_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIM2_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIM2_CR1 寄存器中 CEN=1，启动计数器。在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何。

只要 TI1 为低，计数器开始依据内部时钟计数，在 TI1 变高时停止计数。当计数器开始或停止时都设置 TIM2_SR 中的 TIF 标置。

TI1 上升沿和计数器实际停止之间的延时，取决于 TI1 输入端的重同步电路。

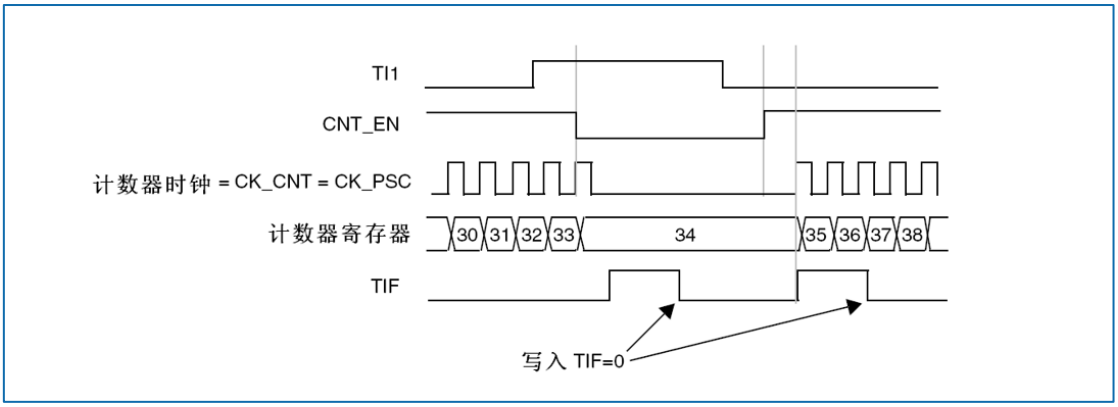


图 13-38 门控模式下的控制电路

13.2.14.3 从模式：触发模式

输入端上选中的事件使能计数器。

在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽（本例中，不需要任何滤波器，保持 IC2F=0000）。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIM2_CCMR1 寄存器中 CC2S=01。置 TIM2_CCER 寄存器中 CC2P=1 以确定极性（只检测低电平）。
- 置 TIM2_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIM2_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

- 当 TI2 出现一个上升沿时，计数器开始在内部时钟驱动下计数，同时设置 TIF 标志。

TI2 上升沿和计数器启动计数之间的延时，取决于 TI2 输入端的重同步电路。

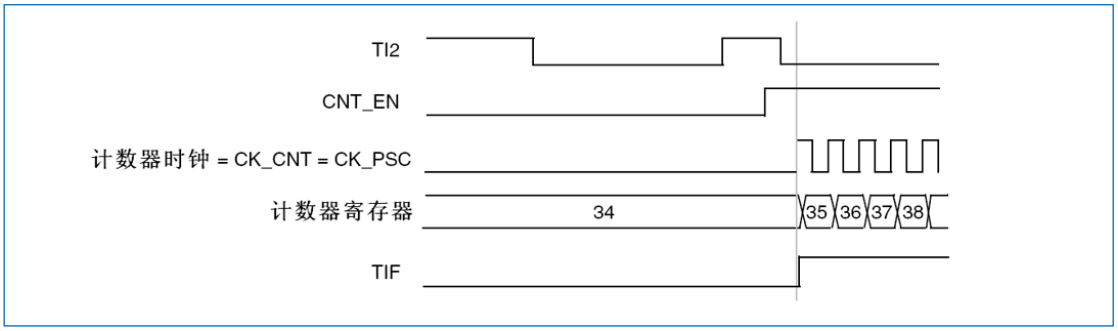


图 13-39 触发器模式下的控制电路

13.2.14.4 从模式：外部时钟模式 2＋触发模式

外部时钟模式 2 可以与另一种从模式（外部时钟模式 1 和编码器模式除外）一起使用。这时，ETR 信号被用作外部时钟的输入，在复位模式、门控模式或触发模式时可以选择另一个输入作为触发输入。不建议使用 TIM2_SMCR 寄存器的 TS 位选择 ETR 作为 TRGI。

下面的例子中，TI1 上出现一个上升沿之后，计数器即在 ETR 的每一个上升沿向上计数一次：

- 通过 TIM2_SMCR 寄存器配置外部触发输入电路：
  - ETF=0000：没有滤波
  - ETPS=00：不用预分频器
  - ETP=0：检测 ETR 的上升沿，置 ECE=1 使能外部时钟模式。
- 按如下配置通道 1，检测 TI 的上升沿：
  - IC1F=0000：没有滤波
  - 触发操作中不使用捕获预分频器，不需要配置。
  - 置 TIM2_CCMR1 寄存器中 CC1S=01，选择输入捕获源。
  - 置 TIM2_CCER 寄存器中 CC1P=0 以确定极性（只检测上升沿）。
- 置 TIM2_SMCR 寄存器中 SMS=110，配置定时器为触发模式。置 TIM2_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。

当 TI1 上出现一个上升沿时，TIF 标志被设置，计数器开始在 ETR 的上升沿计数。

ETR 信号的上升沿和计数器实际复位间的延时，取决于 ETRP 输入端的重同步电路。

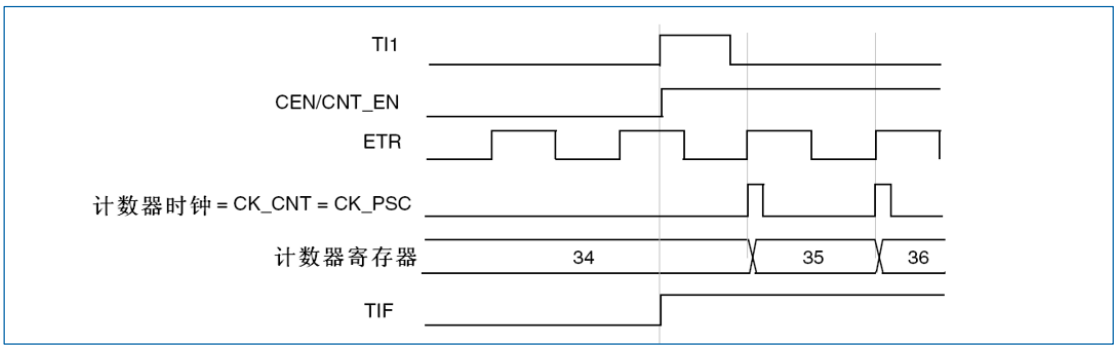


图 13-40 外部时钟模式 2＋触发模式下的控制电路

13.2.15 定时器同步

所有 TIM2 定时器在内部相连，用于定时器同步或链接。当一个定时器处于主模式时，它可以对另

一个处于从模式的定时器的计数器进行复位、启动、停止或提供时钟等操作。

下图显示了触发选择和主模式选择模块的概况。

13.2.15.1 使用一个定时器作为另一个定时器的预分频器

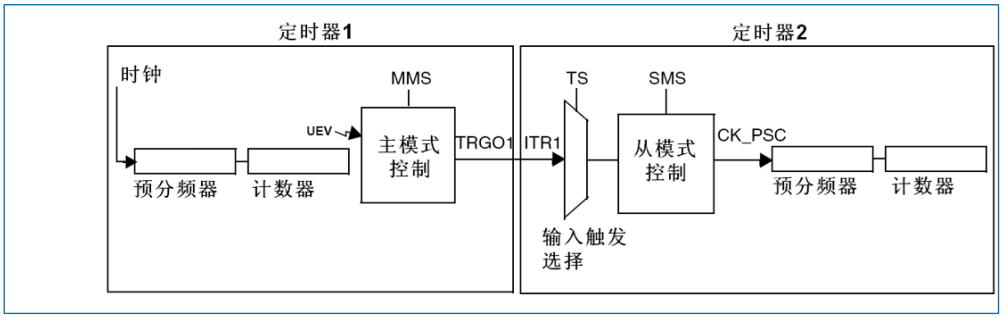


图 13-41 主/从定时器的例子

如：可以配置定时器 1 作为定时器 2 的预分频器。参考上图，进行下述操作：

1. 配置定时器 1 为主模式，它可以在每一个更新事件 UEV 时输出一个周期性的触发信号。在 TIM1_CR2 寄存器的 MMS=’010’时，每当产生一个更新事件时在 TRGO1 上输出一个上升沿信号。

2. 连接定时器 1 的 TRGO1 输出至定时器 2，设置 TIM2_SMCR 寄存器的 TS=’000’，配置定时器 2 为使用 ITR1 作为内部触发的从模式。

3. 然后把从模式控制器置于外部时钟模式 1（TIM2_SMCR 寄存器的 SMS=111）；这样定时器 2 即可由定时器 1 周期性的上升沿（即定时器 1 的计数器溢出）信号驱动。

4. 最后，必须设置相应（TIM2_CR1 寄存器）的 CEN 位分别启动两个定时器。

说明：如果 OCx 已被选中为定时器 1 的触发输出（MMS=1xx），它的上升沿用于驱动定时器 2 的计数器。

13.2.15.2 使用一个定时器使能另一个定时器

在这个例子中，定时器 2 的使能由定时器 1 的输出比较控制。参考图主/从定时器的例子的连接。仅当定时器 1 的 OC1REF 为高时，定时器 2 才对分频后的内部时钟计数。两个定时器的时钟频率都是由预分频器对 CK_INT 除以 3（fCK_CNT=fCK_INT/3）得到。

1. 配置定时器 1 为主模式，送出它的输出比较参考信号（OC1REF）为触发输出（TIM1_CR2 寄存器的 MMS=100）。

2. 配置定时器 1 的 OC1REF 波形（TIM1_CCMR1 寄存器）。

3. 配置定时器 2 从定时器 1 获得输入触发（TIM2_SMCR 寄存器的 TS=000）。

4. 配置定时器 2 为门控模式（TIM2_SMCR 寄存器的 SMS=101）。

5. 置 TIM2_CR1 寄存器的 CEN=1 以使能定时器 2。

6. 置 TIM1_CR1 寄存器的 CEN=1 以启动定时器 1。

说明：定时器 2 的时钟不与定时器 1 的时钟同步，这个模式只影响定时器 2 计数器的使能信号。

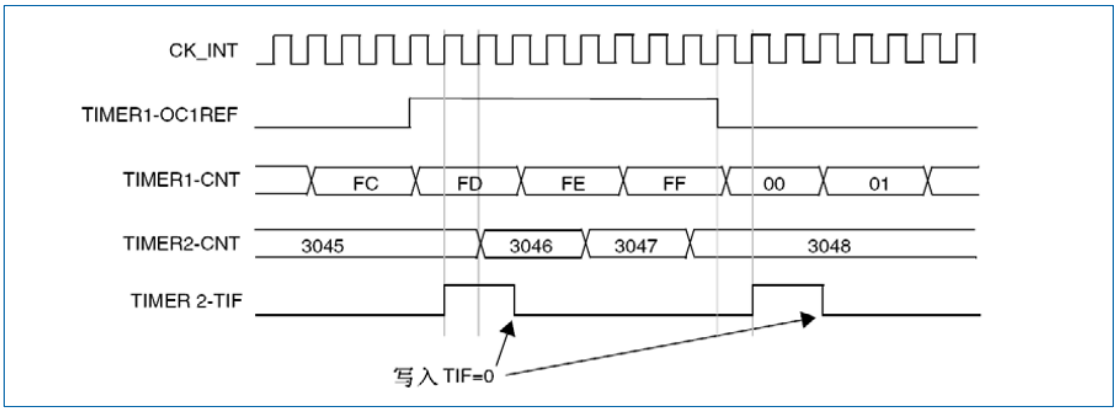


图 13-42 定时器 1 的 OC1REF 控制定时器 2

在上图的例子中，在定时器 2 启动之前，它们的计数器和预分频器未被初始化，因此它们从当前的数值开始计数。可以在启动定时器 1 之前复位 2 个定时器，使它们从给定的数值开始，即在定时器计数器中写入需要的任意数值。写 TIM2_EGR 寄存器的 UG 位即可复位定时器。

在下一个例子中，需要同步定时器 1 和定时器 2。定时器 1 是主模式并从 0 开始，定时器 2 是从模式并从 0xE7 开始；2 个定时器的预分频器系数相同。写'0'到 TIM1_CR1 的 CEN 位将禁止定时器 1，定时器 2 随即停止。

1. 配置定时器 1 为主模式，送出输出比较 1 参考信号（OC1REF）做为触发输出（TIM1_CR2 寄存器的 MMS=100）。
2. 配置定时器 1 的 OC1REF 波形（TIM1_CCMR1 寄存器）。
3. 配置定时器 2 从定时器 1 获得输入触发（TIM2_SMCR 寄存器的 TS=000）
4. 配置定时器 2 为门控模式（TIM2_SMCR 寄存器的 SMS=101）
5. 置 TIM1_EGR 寄存器的 UG=‘1’，复位定时器 1。
6. 置 TIM2_EGR 寄存器的 UG=‘1’，复位定时器 2。
7. 写'0xE7'至定时器 2 的计数器（TIM2_CNTL），初始化它为 0xE7。
8. 置 TIM2_CR1 寄存器的 CEN=‘1’以使能定时器 2。
9. 置 TIM1_CR1 寄存器的 CEN=‘1’以启动定时器 1。
10. 置 TIM1_CR1 寄存器的 CEN=‘0’以停止定时器 1。

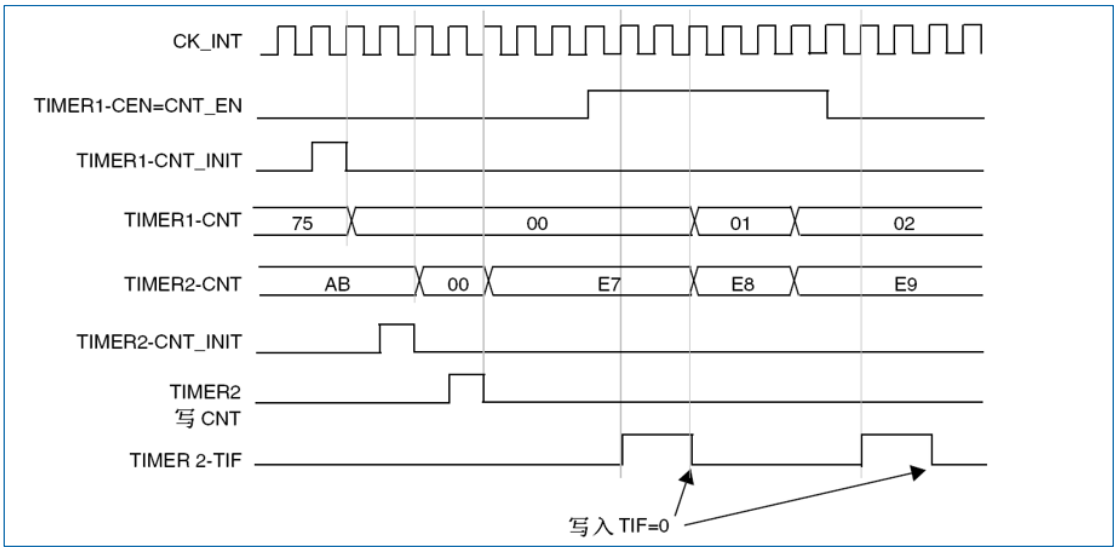


图 13-43 通过使能定时器 1 可以控制定时器 2

13.2.15.3 使用一个定时器去启动另一个定时器

在这个例子中，使用定时器 1 的更新事件使能定时器 2。参考图主/从定时器的例子的连接。一旦定时器 1 产生更新事件，定时器 2 即从它当前的数值（可以是非 0）按照分频的内部时钟开始计数。在收到触发信号时，定时器 2 的 CEN 位被自动地置'1'，同时计数器开始计数直到写'0'到 TIM2_CR1 寄存器的 CEN 位。两个定时器的时钟频率都是由预分频器对 CK_INT 除以 3（fCK_CNT=fCK_INT/3）。

1. 配置定时器 1 为主模式，送出它的更新事件（UEV）做为触发输出（TIM1_CR2 寄存器的 MMS=010）。
2. 配置定时器 1 的周期（TIM1_ARR 寄存器）。
3. 配置定时器 2 从定时器 1 获得输入触发（TIM2_SMCR 寄存器的 TS=000）。
4. 配置定时器 2 为触发模式（TIM2_SMCR 寄存器的 SMS=110）。
5. 置 TIM1_CR1 寄存器的 CEN=1 以启动定时器 1。

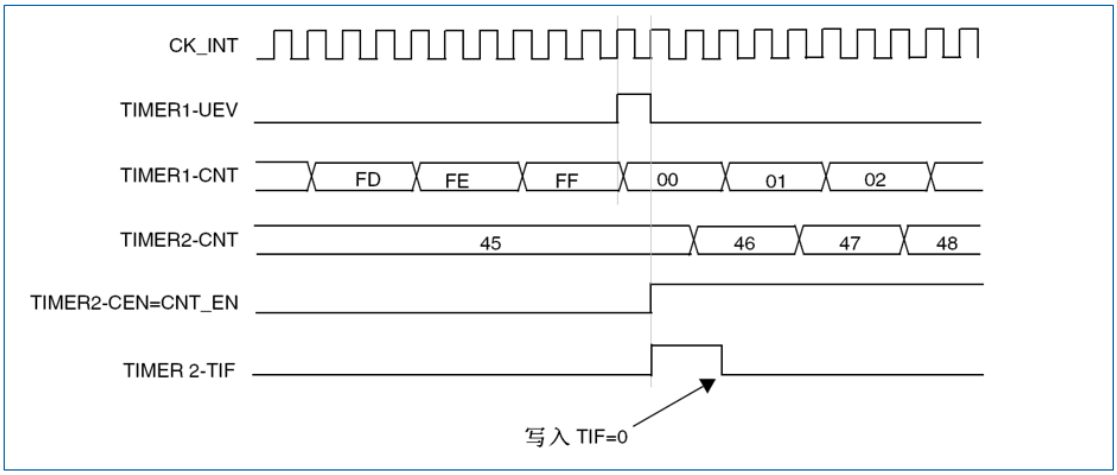


图 13-44 使用定时器 1 的更新触发定时器 2

在上一个例子中，可以在启动计数之前初始化两个计数器。下图显示在与 0 相同配置情况下，使用触发模式而不是门控模式（TIM2_SMCR 寄存器的 SMS=110）的动作。

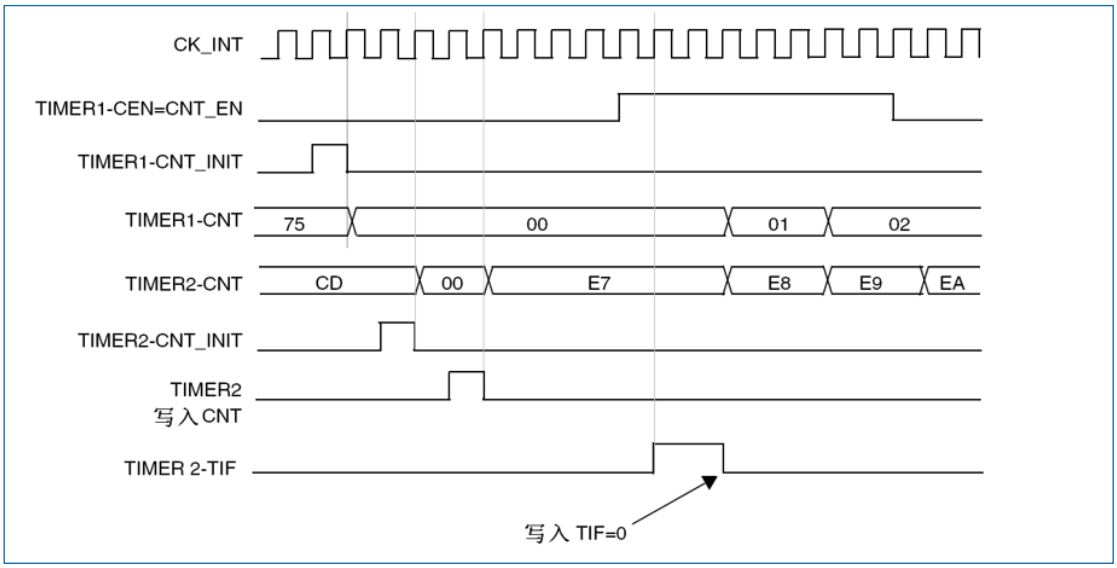


图 13-45 利用定时器 1 的使能触发定时器 2

13.2.15.4 使用一个外部触发同步地启动 2 个定时器

这个例子中当定时器 1 的 TI1 输入上升时使能定时器 1，使能定时器 1 的同时使能定时器 2，参见图 13-41。为保证计数器的对齐，定时器 1 必须配置为主/从模式（对应 TI1 为从，对应定时器 2 为主）：

1. 配置定时器 1 为主模式，送出它的使能做为触发输出（TIM1_CR2 寄存器的 MMS='001'）
2. 配置定时器 1 为从模式，从 TI1 获得输入触发（TIM1_SMCR 寄存器的 TS='100'）。
3. 配置定时器 1 为触发模式（TIM1_SMCR 寄存器的 SMS='110'）。
4. 配置定时器 1 为主/从模式，TIM1_SMCR 寄存器的 MSM= '1'。
5. 配置定时器 2 从定时器 1 获得输入触发（TIM2_SMCR 寄存器的 TS=000）
6. 配置定时器 2 为触发模式（TIM2_SMCR 寄存器的 SMS='110'）。

当定时器 1 的 TI1 上出现一个上升沿时，两个定时器同步地按照内部时钟开始计数，两个 TIF 标志也同时被设置。

说明：在这个例子中，在启动之前两个定时器都被初始化（设置相应的 UG 位），两个计数器都从 0 开始，但可以通过写入任意一个计数器寄存器（TIM2_CNT）在定时器间插入一个偏移。下图中能看到主/从模式下在定时器 1 的 CNT_EN 和 CK_PSC 之间有个延迟。

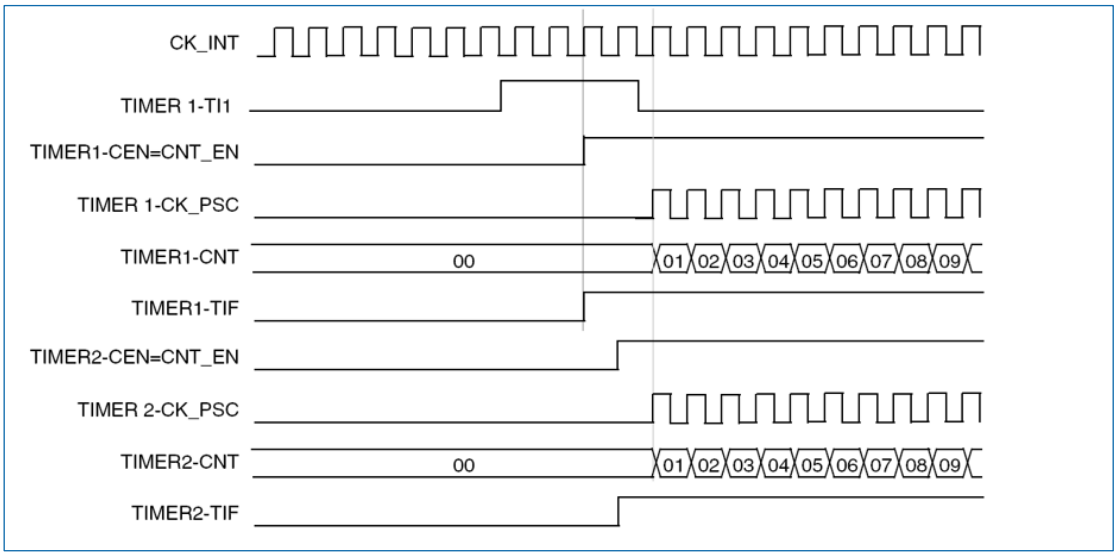


图 13-46 使用定时器 1 的 TI1 输入触发定时器 1 和定时器 2

13.2.16 调试模式

当 MCU 进入调试模式（Cortex-M3 内核停止），根据 DBG 模块中 DBG_TIM2_Stop 的设置，TIM2 计数器继续正常操作或者停止。参见“23.8.2 对定时器、看门狗和 I2C 的调试支持”。

13.3 TIM2 寄存器

基地址：0x4000 0000

空间大小：0x0400

13.3.1 TIM2 控制寄存器 1（TIM2_CR1）

地址偏移：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res						CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

位 15:10	Res：保留 必须保持复位值。
位 9:8	CKD[1:0]：时钟分频因子（Clock division） 这2 位定义在定时器时钟（CK_INT）频率与数字滤波器（ETR，TIx）使用的采样频率之间的分频比例。 • 00：$t_{DTS} = t_{CK\_INT}$ • 01：$t_{DTS} = 2 \times t_{CK\_INT}$ • 10：$t_{DTS} = 4 \times t_{CK\_INT}$ • 11：保留，未使用
位 7	ARPE：自动重装载预装载允许（Auto-reload preload enable） • 0：TIMx_ARR 寄存器没有缓冲 • 1：TIMx_ARR 寄存器有缓冲

位 6:5	CMS[1:0]：选择中央对齐模式（Center-aligned mode selection） • 00：边沿对齐模式 计数器依据方向位（DIR）向上或向下计数。 • 01：中央对齐模式1 计数器交替地向上、向下计数。配置为输出的通道（TIMx_CCMRx 寄存器中 CCxS=00）的输出比较中断标志位，只在计数器向下计数时被设置。 • 10：中央对齐模式2 计数器交替地向上、向下计数。配置为输出的通道（TIMx_CCMRx 寄存器中 CCxS=00）的输出比较中断标志位，只在计数器向上计数时被设置。 • 11：中央对齐模式3 计数器交替地向上、向下计数。配置为输出的通道（TIMx_CCMRx 寄存器中 CCxS=00）的输出比较中断标志位，在计数器向上和向下计数时均被设置。
位 4	DIR：方向（Direction） • 0：计数器向上计数 • 1：计数器向下计数
位 3	OPM：单脉冲模式（One pulse mode） • 0：在发生更新事件时，计数器不停止。 • 1：在发生下一次更新事件（清除 CEN 位）时，计数器停止。
位 2	URS：更新请求源（Update request source） 软件通过该位选择 UEV 事件的源。 • 0：如果使能了更新中断请求，则下述任一事件将产生更新中断请求： o 计数器溢出/下溢 o 设置 UG 位 o 从模式控制器产生的更新 • 1：如果使能了更新中断请求，则只有计数器溢出/下溢才产生更新中断请求。
位 1	UDIS：禁止更新（Update disable） 软件通过该位允许/禁止 UEV 事件的产生。 • 0：允许 UEV 更新（UEV）事件由下述任一事件产生： o 计数器溢出/下溢 o 设置 UG 位 o 从模式控制器产生的更新，具有缓存的寄存器被装入它们的预装载值 • 1：禁止 UEV 不产生更新事件，影子寄存器（ARR、PSC、CCRx）保持它们的值。如果设置了 UG 位或从模式控制器产出了一个硬件复位，则计数器和预分频器被重新初始化。
位 0	CEN：使能计数器（Counter enable）

	• 0：禁止计数器 • 1：使能计数器

13.3.2 TIM2 控制寄存器 2（TIM2_CR2）

地址偏移：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								TI1S	MMS[2:0]			Res			
								rw	rw						

位 15:8	Res：保留 必须保持复位值。
位 7	TI1S：TI1 选择 • 0：TIMx_CH1 引脚连到 TI1 输入； • 1：TIMx_CH1、TIMx_CH2 和 TIMx_CH3 引脚经异或后连到 TI1 输入。
位 6:4	MMS[2:0]：主模式选择 该位域可选择在主模式下送到从定时器的同步信息（TRGO）。 • 000：复位 TIMx_EGR 寄存器的 UG 位被用于作为触发输出（TRGO）。如果是触发输入产生的复位（从模式控制器处于复位模式），则 TRGO 上的信号相对实际的复位会有一个延迟。 • 001：使能 计数器使能信号 CNT_EN 被作为触发输出（TRGO）。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式。 • 010：更新 更新事件被选为触发输入（TRGO）。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 • 011：比较脉冲 在发生次捕获或一次比较成功时，当要设置 CC1IF 标志时（即使它已经为高），触发输出送出一个正脉冲（TRGO）。 • 100：比较 OC1REF 信号被用于作为触发输出（TRGO）。 • 101：比较 OC2REF 信号被用于作为触发输出（TRGO）。 • 110：比较 OC3REF 信号被用于作为触发输出（TRGO）。 • 111：比较 OC4REF 信号被用于作为触发输出（TRGO）。

位 3:0	Res：保留 必须保持复位值，读该位始终为 0。

13.3.3 TIM2 从模式控制寄存器（TIM2_SMCR）

地址偏移：0x08

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Res	SMS[2:0]		
rw	rw	rw		rw				rw	rw				rw		

位 15	ETP：外部触发极性 该位选择 ETR 或 ETR 的反相来作为触发操作。 • 0：ETR 不反相，高电平或上升沿有效。 • 1：ETR 被反相，低电平或下降沿有效。
位 14	ECE：外部时钟使能位 该位启用外部时钟模式2。 • 0：禁止外部时钟模式2 • 1：使能外部时钟模式2 计数器由ETRF 信号的任意有效边沿驱动。
位 13:12	ETPS[1:0]：外部触发预分频 外部触发信号ETRP 的频率最多是CK_INT 频率的1/4。当输入较快的外部时钟时，可以使用预分频降低ETRP 的频率。 • 00：关闭预分频 • 01：ETRP 频率/2 • 10：ETRP 频率/4 • 11：ETRP 频率/8
位 11:8	ETF[3:0]：外部触发滤波 该位域定义了采样 ETRP 信号的频率和对 ETRP 信号数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到N 个事件后会产生一个输出的跳变。 • 0000：无滤波器，以 f_{DTS} 采样 • 0001：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=2 • 0010：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=4 • 0011：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=8 • 0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6 • 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 • 0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6 • 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 • 1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6 • 1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8

	• 1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5 • 1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6 • 1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8 • 1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5 • 1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6 • 1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8
位 7	MSM：主/从模式 • 0：无作用 • 1：触发输入（TRGI）上的事件被延迟，以允许当前定时器（通过 TRGO）与它的从定时器之间的完美同步。这适用于多个定时器需要同步到一个单一的外部事件的情况。
位 6:4	TS[2:0]：触发选择 该位域选择用于同步计数器的触发输入。 • 000：内部触发0（ITR0） • 001：内部触发1（ITR1） • 010：内部触发2（ITR2） • 011：内部触发3（ITR3） • 100：TI1 的边沿检测器（TI1F_ED） • 101：滤波后的定时器输入1（TI1FP1） • 110：滤波后的定时器输入2（TI2FP2） • 111：外部触发输入（ETRF）
位 3	Res：保留 必须保持复位值。
位 2:0	SMS[2:0]：从模式选择 当选择了外部信号，触发信号（TRGI）的有效边沿与选中的外部输入极性相关。 • 000：关闭从模式 如果CEN=1，则预分频器直接由内部时钟驱动。 • 001：编码器模式1 根据TI1FP1 的电平，计数器在TI2FP2 的边沿向上/下计数。 • 010：编码器模式2 根据TI2FP2 的电平，计数器在TI1FP1 的边沿向上/下计数。 • 011：编码器模式3 根据另一个信号的输入电平，计数器在TI1FP1 和TI2FP2 的边沿向上/下计数。 • 100：复位模式 选中的触发输入（TRGI）的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。 • 101：门控模式 当触发输入（TRGI）为高时，计数器的时钟开启。一旦触发输入变为低，则计数

	器停止（但不复位）。计数器的启动和停止都是受控的。 • 110：触发模式 计数器在触发输入TRGI 的上升沿启动（但不复位），只有计数器的启动是受控的。 • 111：外部时钟模式1 选中的触发输入（TRGI）的上升沿驱动计数器。

表 13-3 TIM1 和 TIM2 内部触发连接

从定时器	ITR0（TS=000）	ITR1（TS=001）	ITR2（TS=010）
TIM1	TIM2	TIM6	ADC_AWD
TIM2	TIM1	TIM6	ADC_AWD

13.3.4 TIM2 中断允许寄存器（TIM2_DIER）

地址偏移：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res									TIE	Res	CC4IE	CC3IE	CC2IE	CC1IE	UIE
									rw		rw	rw	rw	rw	rw

位 15:7	Res：保留 必须保持复位值。
位 6	TIE：触发中断使能 • 0：触发中断禁用 • 1：触发中断允许
位 5	Res：保留 必须保持复位值。
位 4	CC4IE：捕捉/比较 4 中断使能 • 0：CC4 中断禁用 • 1：CC4 中断允许
位 3	CC3IE：捕捉/比较 3 中断使能 • 0：CC3 中断禁用 • 1：CC3 中断允许
位 2	CC2IE：捕捉/比较 2 中断使能 • 0：CC2 中断禁用 • 1：CC2 中断允许
位 1	CC1IE：捕捉/比较 1 中断使能

	• 0：CC1 中断禁用 • 1：CC1 中断允许
位 0	UIE：更新中断使能 • 0：更新中断禁用 • 1：更新中断允许

13.3.5 TIM2 状态寄存器（TIM2_SR）

地址偏移：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res			CC4OF	CC3OF	CC2OF	CC1OF	Res		TIF	Res	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位 15:13	Res：保留 始终读为 0。
位 12	CC4OF：捕捉/比较4 重复捕捉标志
位 11	CC3OF：捕捉/比较3 重复捕捉标志
位 10	CC2OF：捕捉/比较2 重复捕捉标志
位 9	CC1OF：捕捉/比较1 重复捕捉标志（Capture/Compare 1 overcapture flag） 仅当相应的通道被配置为输入捕获时，该标记可由硬件置 1。 该位由软件清 0。 • 0：无重复捕获产生。 • 1：当计数器的值被捕获到TIMx_CCR1 寄存器时，CC1IF 的状态已经为 1。
位 8:7	Res：保留 始终读为 0。
位 6	TIF：触发器中断标记（Tigger interrupt flag） 当发生触发事件（当从模式控制器处于除门控模式外的其它模式时，在TRGI 输入端检测到有效边沿、或门控模式下的任一边沿）时由硬件对该位置 1。该位由软件清 0。 • 0：无触发器事件产生。 • 1：触发器中断等待响应。
位 5	Res：保留 始终读为 0。
位 4	CC4IF：捕捉/比较 4 中断标志

位 3	CC3IF：捕捉/比较 3 中断标志
位 2	CC2IF：捕捉/比较 2 中断标志
位 1	CC1IF：捕捉/比较 1 中断标志（Capture/Compare 1 interrupt flag） • 如果通道 CC1 配置为输出模式： 当计数器值与比较值匹配时该位由硬件置 1，但在中心对称模式下除外。该位由软件清 0。 ○ 0：无匹配发生 ○ 1：TIMx_CNT 的值与 TIMx_CCR1 的值匹配 当 TIMx_CCR1 > TIMx_APR 时，在计数器溢出（向上、向上/向下计数模式）或计数器下溢出时（向下计数模式）时 CC1IF 位变高。 • 如果通道 CC1 配置为输入模式： 当捕获事件发生时，该位由硬件置 1。该位由软件清 0 或通过读 TIMx_CCR1 清 0。 ○ 0：无输入捕获产生； ○ 1：计数器值已被捕获（拷贝）至 TIMx_CCR1（在 IC1 上检测到与所选极性相同的边沿）。
位 0	UIF：更新中断标记（Update interrupt flag） 当产生以下更新事件时，该位由硬件置 1。 • 若 TIMx_CR1 寄存器的 UDIS=0 且 URS=0，当 TIMx_EGR 寄存器的 UG=1 时产生更新事件（软件对计数器 CNT 重新初始化）。 • 若 TIMx_CR1 寄存器的 UDIS=0 且 URS=0，当计数器 CNT 被触发事件重初始化时产生。 该位由软件清 0。 • 0：无更新事件产生； • 1：更新中断等待响应。

13.3.6 TIM2 事件产生寄存器（TIM2_EGR）

地址偏移：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res									TG	Res	CC4G	CC3G	CC2G	CC1G	UG
									w		w	w	w	w	w

位 15:7	Res：保留 必须保持复位值。
位 6	TG：产生触发事件（Trigger generation） 该位由软件置 1，用于产生一个触发事件。该位由硬件自动清 0。 • 0：无动作 • 1：TIMx_SR 寄存器的 TIF=1，若开启对应的中断，则产生相应的中断。

位 5	Res：保留 必须保持复位值。
位 4	CC4G：捕捉/比较 4 产生
位 3	CC3G：捕捉/比较 3 产生
位 2	CC2G：捕捉/比较 2 产生
位 1	CC1G：捕捉/比较 1 产生 • 若通道 CC1 配置为输出： 设置 CC1IF=1，若开启对应的中断，则产生相应的中断。 该位由软件置 1，用于产生一个捕获/比较事件，由硬件自动清 0。 ○ 0：无动作； ○ 1：在通道 CC1 上产生一个捕获/比较事件： • 若通道 CC1 配置为输入： 当前的计数器值捕获至 TIMx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断，则产生相应的中断。若 CC1IF 已经为 1，则设置 CC1OF=1。
位 0	UG：产生更新事件（Update generation） 该位由软件置 1，由硬件自动清 0。 • 0：无动作； • 1：重新初始化计数器，并产生一个更新事件。 *注意：若 UG=1，预分频器的计数器也被清 0（但是预分频系数不变）。若在中心对称模式下或 DIR=0（向上计数）则计数器被清 0，若 DIR=1（向下计数）则计数器取 TIMx_ARR 的值。*

13.3.7 TIM2 捕捉/比较模式寄存器 1（TIM2_CCMR1）

地址偏移：0x18

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			
rw	rw			rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	

输出比较模式

位 15	OC2CE：输出比较 2 清除使能
位 14:12	OC2M[2:0]：输出比较 2 模式
位 11	OC2PE：输出比较 2 预装载使能
位 10	OC2FE：输出比较 2 快速使能

位 9:8	CC2S[1:0]：捕捉/比较 2 选择（Capture/Compare 2 selection） 该位域定义通道的方向（输入/输出），及输入信号的选择： • 00：CC2 通道配置为输出。 • 01：CC2 通道配置为输入，IC2 映射在 TI2 上。 • 10：CC2 通道配置为输入，IC2 映射在 TI1 上。 • 11：CC2 通道配置为输入，IC2 映射在 TRC 上。
位 7	OC1CE：输出比较 1 清除使能 • 0：OC1REF 不受 ETRF 输入的影响。 • 1：一旦检测到 ETRF 输入高电平，清除 OC1REF=0。
位 6:4	OC1M[2:0]：输出比较 1 模式（Output compare 1 mode） 该位域定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1 和 OC1N 的值。OC1REF 是高电平有效，而 OC1、OC1N 的有效电平取决于 CC1P、CC1PN 位。 • 000：冻结 输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较，对 OC1REF 不起作用； • 001：匹配时设置通道 1 为有效电平 当计数器 TIMx_CNT 的值与捕获/比较寄存器 1（TIMx_CCR1）相同时，强制 OC1REF 为高。 • 010：匹配时设置通道 1 为无效电平 当计数器 TIMx_CNT 的值与捕获/比较寄存器 1（TIMx_CCR1）相同时，强制 OC1REF 为低。 • 011：翻转 当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。 • 100：强制为无效电平 强制 OC1REF 为低。 • 101：强制为有效电平 强制 OC1REF 为高。 • 110：PWM 模式 1 o 在向上计数时，一旦 TIMx_CNT<TIMx_CCR1 则通道 1 为有效电平，否则为无效电平。 o 在向下计数时，一旦 TIMx_CNT>TIMx_CCR1 则通道 1 为无效电平（OC1REF=0），否则为有效电平（OC1REF=1）。 • 111：PWM 模式 2 o 在向上计数时，一旦 TIMx_CNT < TIMx_CCR1 时通道 1 为无效电平，否则为有效电平。 o 在向下计数时，一旦 TIMx_CNT > TIMx_CCR1 时通道 1 为有效电平，否则为无效电平。
位 3	OC1PE：输出比较 1 预装载使能（Output compare 1 preload enable） • 0：禁止 TIMx_CCR1 寄存器的预装载功能，可随时写入 TIMx_CCR1 寄存器，并且新

	写入的数值立即生效。 • 1：开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅针对预装载寄存器，TIMx_CCR1 的预装载值在更新事件到来时被传送至当前寄存器中。
位 2	OC1FE：输出比较 1 快速使能（Output compare 1 fast enable） 该位加快 CC 输出对触发器输入事件的响应。 • 0：CC1 的正常操作依赖于计数器与 CCR1 的值，即使触发器是打开的。当触发器的输入出现一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期。 • 1：输入到触发器的有效沿的作用就像发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。该位只在通道被配置成 PWM1 或 PWM2 模式时生效。
位 1:0	CC1S[1:0]：捕捉/比较 1 选择（Capture/Compare 1 selection） 该位域定义通道的方向（输入/输出），及输入信号的选择： • 00：CC1 通道被配置为输出。 • 01：CC1 通道被配置为输入，IC1 映射在 TI1 上。 • 10：CC1 通道被配置为输入，IC1 映射在 TI2 上。 • 11：CC1 通道被配置为输入，IC1 映射在 TRC 上。

输入捕捉模式

位 15:12	IC2F[3:0]：输入捕捉 2 滤波器
位 11:10	IC2PSC[1:0]：输入捕捉 2 预分频
位 9:8	CC2S[1:0]：捕捉/比较 2 选择（Capture/compare 2 selection） 该位域定义通道的方向（输入/输出），及输入信号的选择： • 00：CC2 通道被配置为输出。 • 01：CC2 通道被配置为输入，IC2 映射在 TI2 上。 • 10：CC2 通道被配置为输入，IC2 映射在 TI1 上。 • 11：CC2 通道被配置为输入，IC2 映射在 TRC 上。
位 7:4	IC1F：输入捕获 1 滤波器（Input capture 1 filter） 该位域定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变： • 0000：无滤波器，以 f_{DTS} 采样 • 0001：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=2 • 0010：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=4 • 0011：采样频率 $f_{SAMPLING}= f_{CK\_INT}$，N=8 • 0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6 • 0101：采样频率 $f_{SAMPLING}= f_{DTS}/2$，N=8 • 0110：采样频率 $f_{SAMPLING}= f_{DTS}/4$，N=6 • 0111：采样频率 $f_{SAMPLING}= f_{DTS}/4$，N=8

	• 1000：采样频率 $f_{SAMPLING=}$ f_{DTS}/8，N=6 • 1001：采样频率 $f_{SAMPLING=}$ f_{DTS}/8，N=8 • 1010：采样频率 $f_{SAMPLING=}$ f_{DTS}/16，N=5 • 1011：采样频率 $f_{SAMPLING=}$ f_{DTS}/16，N=6 • 1100：采样频率 $f_{SAMPLING=}$ f_{DTS}/16，N=8 • 1101：采样频率 $f_{SAMPLING=}$ f_{DTS}/32，N=5 • 1110：采样频率 $f_{SAMPLING=}$ f_{DTS}/32，N=6 • 1111：采样频率 $f_{SAMPLING=}$ f_{DTS}/32，N=8
位 3:2	IC1PSC[1:0]：输入捕捉 1 预分频 该位域定义了 CC1 输入（IC1）的预分频系数。一旦 CC1E=0（TIMx_CCER 寄存器中），则预分频器复位。 • 00：无预分频器，捕获输入口上检测到的每一个边沿都会触发一次捕获。 • 01：每 2 个事件触发一次捕获。 • 10：每 4 个事件触发一次捕获。 • 11：每 8 个事件触发一次捕获。
位 1:0	CC1S[1:0]：捕捉/比较 1 选择 该位域定义通道的方向（输入/输出），及输入信号的选择： • 00：CC1 通道被配置为输出。 • 01：CC1 通道被配置为输入，IC1 映射在 TI1 上。 • 10：CC1 通道被配置为输入，IC1 映射在 TI2 上。 • 11：CC1 通道被配置为输入，IC1 映射在 TRC 上。

13.3.8 TIM2 捕捉/比较模式寄存器 2（TIM2_CCMR2）

地址偏移：0x1C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
rw	rw			rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	

输出比较模式

位 15	OC4CE：输出比较4 清除使能
位 14:12	OC4M[2:0]：输出比较4 模式
位 11	OC4PE：输出比较4 预分频使能
位 10	OC4FE：输出比较4 快速使能
位 9:8	CC4S[1:0]：捕捉/比较4 选择 该位域定义通道的方向（输入/输出），及输入信号的选择：

	• 00：CC4 通道被配置为输出。 • 01：CC4 通道被配置为输入，IC4 映射在TI4 上。 • 10：CC4 通道被配置为输入，IC4 映射在TI3 上。 • 11：CC4 通道被配置为输入，IC4 映射在TRC 上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCR 寄存器的TS 位选择）。
位 7	OC3CE：输出比较3 清除使能
位 6:4	OC3M[2:0]：输出比较3 模式
位 3	OC3PE：输出比较3 预分频使能
位 2	OC3FE：输出比较3 快速使能
位 1:0	CC3S[1:0]：捕捉/比较3 该位域定义通道的方向（输入/输出），及输入信号的选择： • 00：CC3 通道被配置为输出。 • 01：CC3 通道被配置为输入，IC4 映射在TI3 上。 • 10：CC3 通道被配置为输入，IC4 映射在TI4 上。 • 11：CC3 通道被配置为输入，IC4 映射在TRC 上。

输入捕捉模式

位 15:12	IC4F[3:0]：输入捕捉4 滤波器
位 11:10	IC4PSC[1:0]：输入捕捉4 预分频器
位 9:8	CC4S：捕捉/比较4 选择 该位域定义通道的方向（输入/输出），及输入信号的选择： • 00：CC4 通道被配置为输出。 • 01：CC4 通道被配置为输入，IC3 映射在TI4 上。 • 10：CC4 通道被配置为输入，IC3 映射在TI3 上。 • 11：CC4 通道被配置为输入，IC3 映射在TRC 上。
位 7:4	IC3F[3:0]：输入捕捉3 滤波器
位 3:2	IC3PSC[1:0]：输入比较3 预分频器
位 1:0	CC3S[1:0]：捕捉/比较3 选择 该位域定义通道的方向（输入/输出），及输入信号的选择： • 00：CC3 通道被配置为输出。 • 01：CC3 通道被配置为输入，IC3 映射在TI3 上。 • 10：CC3 通道被配置为输入，IC3 映射在TI4 上。 • 11：CC3 通道被配置为输入，IC3 映射在TRC 上。

13.3.9 TIM2 捕捉/比较使能寄存器（TIM2_CCER）

地址偏移：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res	CC4P	CC4E	CC3NP	Res	CC3P	CC3E	CC2NP	Res	CC2P	CC2E	CC1NP	Res	CC1P	CC1E
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

位 15	CC4NP：捕捉/比较 4 输出极性
位 14	Res：保留 始终读为 0。
位 13	CC4P：捕捉/比较 4 输出极性
位 12	CC4E：捕捉/比较 4 输出使能
位 11	CC3NP：捕捉/比较 3 输出极性
位 10	Res：保留 始终读为 0。
位 9	CC3P：捕捉/比较 3 输出极性
位 8	CC3E：捕捉/比较 3 输出使能
位 7	CC2NP：捕捉/比较 2 输出极性
位 6	Res：保留 始终读为 0。
位 5	CC2P：捕捉/比较 2 输出极性
位 4	CC2E：捕捉/比较 2 输出使能
位 3	CC1NP：捕捉/比较 1 输出极性 • 通道 CC1 配置为输出时，CC1NP 必须清 0 并保持 CC1NP＝0。 • 通道 CC1 配置为输入时，CC1NP 与 CC1P 联合控制 TI1FP1/TI2FP1 的极性。
位 2	Res：保留 始终读为 0。
位 1	CC1P：捕捉/比较 1 输出极性 • 通道 CC1 配置为输出:

	○ 0：OC1 高有效 ○ 1：OC1 低有效 • 通道 CC1 配置为输入: CC1NP/CC1P 用于选择作为触发或捕获的信号 TI1FP1 和 TI2FP1 的极性是 IC1 还是 IC1 的反相信号。 ○ 00：不反相/上升沿 捕获发生在 TIxFP1 的上升沿（复位、外部时钟或触发模式的捕捉或触发），TIxFP1 不反相（在门控、编码器模式下的触发）。 ○ 01：反相/下降沿 捕获发生在 TIxFP1 的下降沿（复位、外部时钟或触发模式的捕捉或触发），TIxFP1 反相（在门控、编码器模式下的触发）。 ○ 10：保留，不使用此配置 ○ 11：不反相/上升和下降沿 捕获发生在 TIxFP1 的上升沿和下降沿（复位、外部时钟或触发模式的捕捉或触发），TIxFP1 不反相（门控模式的触发）。此配置不能用于编码器模式。
位 0	CC1E：捕捉/比较 1 输出使能 • CC1 通道配置为输出： ○ 0：关闭，OC1 禁止输出。 ○ 1：开启，OC1 信号输出到对应的输出引脚。 • CC1 通道配置为输入： 该位决定了计数器的值是否能捕获入 TIMx_CCR1 寄存器。 ○ 0：捕获禁止 ○ 1：捕获使能

13.3.10 TIM2 计数器（TIM2_CNT）

地址偏移：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

位 31:16	CNT[31:16]：高16 位值
位 15:0	CNT[15:0]：低16 位值

13.3.11 TIM2 预分频（TIM2_PSC）

地址偏移：0x28

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PSC[15:0]
rw

位 15:0	PSC[15:0]：预分频值 计数器的时钟频率CK_CNT 等于f_{CK_PSC}/（PSC[15:0]+1）。PSC 包含了当更新事件产生时装入当前预分频器寄存器的值。

13.3.12 TIM2 自动重装寄存器（TIM2_ARR）

地址偏移：0x2C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARR[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw															

位 31:16	ARR[31:16]：自动重装高16 位值
位 15:0	ARR[15:0]：自动重装低16 位值 ARR 包含了将传送至实际的自动重装载寄存器的数值。当自动重装载的值为空时，计数器不工作。

13.3.13 TIM2 捕捉/比较寄存器 1（TIM2_CCR1）

地址偏移：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw															

位 31:16	CCR1[31:16]：捕捉/比较1 高16 位值
位 15:0	CCR1[15:0]：捕捉/比较 1 低 16 位值 • 若CC1 通道配置为输出： CCR1 包含了装入当前捕获/比较1 寄存器的值。如果在 TIMx_CCMR1 寄存器（OC1PE 位）中未选择预装载特性，写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较1 寄存器中。 当前捕获/比较寄存器参与同计数器 TIMx_CNT 的比较，并在OC1 端口上产生输出信号。 • 若CC1 通道配置为输入： CCR1 包含了由上一次输入捕获1 事件（IC1）传输的计数器值。

13.3.14 TIM2 捕捉/比较寄存器 2（TIM2_CCR2）

地址偏移：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw															

位 31:16	CCR2[31:16]：捕捉/比较2 高16 位值
位 15:0	CCR2[15:0]：捕捉/比较 2 低16 位值 • 若CC2 通道配置为输出： CCR2 包含了装入 TIMx_CCR2 寄存器的值（预装载值）。 如果在 TIMx_CCMR2 寄存器（OC2PE 位）中未选择预装载特性，写入的数值会被立即传输至 TIMx_CCR2 寄存器中。否则只有当更新事件发生时，此预装载值才传输至 TIMx_CCR2 寄存器。 TIMx_CCR2 寄存器参与同计数器 TIMx_CNT 的比较，并在OC2 端口上产生输出信号。 • 若CC2 通道配置为输入： CCR2 包含了由上一次输入捕获 2 事件（IC2）传输的计数器值。

13.3.15 TIM2 捕捉/比较寄存器 3（TIM2_CCR3）

地址偏移：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw															

位 31:16	CCR3[31:16]：捕捉/比较3 高16 位值
位 15:0	CCR3[15:0]：捕捉/比较 3 低16 位值 • 若CC3 通道配置为输出： CCR3 包含了装入 TIMx_CCR3 寄存器的值（预装载值）。如果在 TIMx_CCMR3 寄存器（OC3PE 位）中未选择预装载特性，写入的数值会被立即传输至 TIMx_CCR3 寄存器中。否则只有当更新事件发生时，此预装载值才传输至 TIMx_CCR3 寄存器。 TIMx_CCR3 寄存器参与同计数器 TIMx_CNT 的比较，并在OC3 端口上产生输出信号。 • 若CC3 通道配置为输入：

	CCR3 包含了由上一次输入捕获3 事件（IC3）传输的计数器值。

13.3.16 TIM2 捕捉/比较寄存器 4（TIM2_CCR4）

地址偏移：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw															

位 31:16	CCR4[31:16]：捕捉/比较4 高16 位值
位 15:0	CCR4[15:0]：捕捉/比较 4 低16 位值 • 若CC4 通道配置为输出： CCR4 包含了装入当前 TIMx_CCR4 寄存器的值（预装载值）。 如果在 TIMx_CCMR4 寄存器（OC4PE 位）中未选择预装载特性，写入的数值会被立即传输至当前寄存器。否则只有当更新事件发生时，此预装载值才传输至 TIMx_CCR4 寄存器。 TIMx_CCR4 寄存器参与同计数器 TIMx_CNT 的比较，并在OC4 端口上产生输出信号。 • 若CC4 通道配置为输入： CCR4 包含了由上一次输入捕获4 事件（IC4）传输的计数器值。

14 基本定时器（TIM6）

基本定时器 TIM6 包含一个 16 位自动重装载计数器，由它的编程预分频器驱动。

TIM6 可以作为通用定时器提供时间基准。

表 14-1 TIM6 特性

符号	参数	条件	最小值	典型值	最大值	单位
$t_{res(TIM)}$	定时器分辨时间	$f_{TIMxCLK}$=32MHz	-	31.2	-	ns
f_{EXT}	定时器的 CH1 至 CH4，外部输入的时钟频率	-	-	$f_{TIMxCLK}/2$	-	MHz
		$f_{TIMxCLK}$=32MHz	-	16	-	MHz
t_{MAX_COUNT}	当选择内部时钟时，16 位计数器的时钟周期	-	-	2^{16}	-	$t_{TIMxCLK}$
		$f_{TIMxCLK}$=32MHz	-	2048	-	us

14.1 TIM6 主要功能

- 16 位自动重装载累加计数器
- 16 位可编程（可实时修改）预分频器，用于对输入的时钟按系数为 1～65536 之间的任意数值分频。
- 在更新事件（计数器溢出）时产生中断。

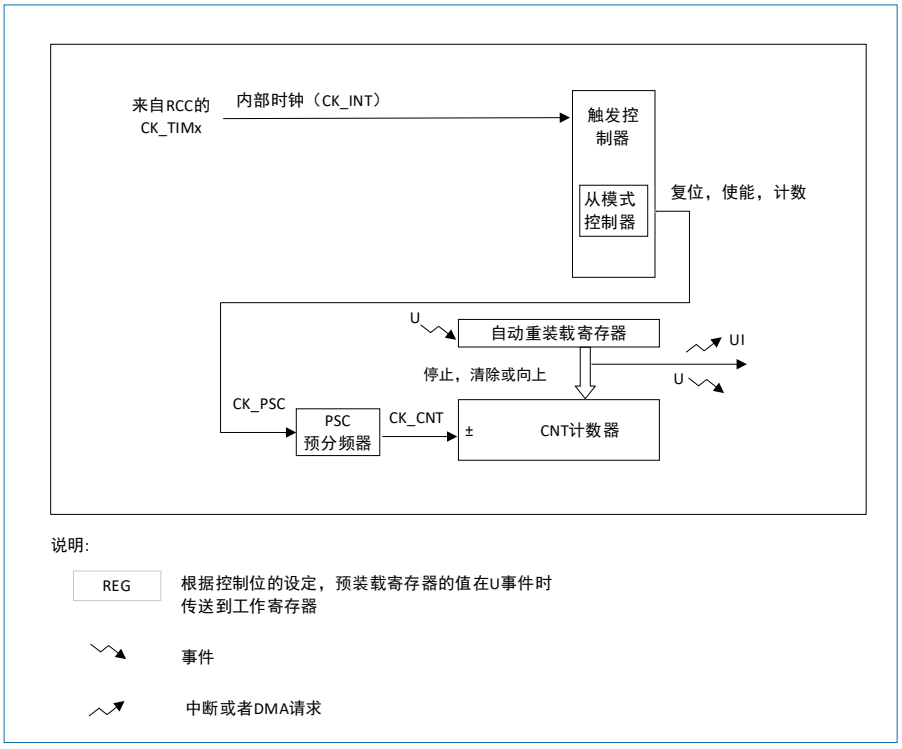


图 14-1 基本寄存器框图

14.2 TIM6 功能描述

14.2.1 时基单元

这个可编程定时器的主要部分是一个带有自动重装载的 16 位累加计数器。计数器的时钟通过一个

预分频器得到。

软件可以读写计数器、自动重装载寄存器和预分频寄存器，即使计数器运行时也可以操作。

时基单元包含：

- 计数器寄存器（TIM6_CNT）
- 预分频寄存器（TIM6_PSC）
- 自动重装载寄存器（TIM6_ARR）

自动重装载寄存器是预加载的。每次读写自动重装载寄存器时，实际上是通过读写预加载寄存器实现。根据 TIM6_CR1 寄存器中的自动重装载预加载使能位（ARPE），写入预加载寄存器的内容能够立即或在每次更新事件时，传送到它的影子寄存器。当 TIM6_CR1 寄存器的 UDIS 位为’0’，则每当计数器达到溢出值时，硬件发出更新事件；软件也可以产生更新事件；关于更新事件的产生，随后会有详细的介绍。

计数器由预分频输出 CK_CNT 驱动，设置 TIM6_CR1 寄存器中的计数器使能位（CEN）使能计数器计数。

说明：实际的设置计数器使能信号 CNT_EN 相对于 CEN 滞后一个时钟周期。

预分频器

预分频能以系数介于 1 至 65536 之间的任意数值对计数器时钟分频。它是通过一个 16 位寄存器（TIM6_PSC）的计数实现分频。因为 TIM6_PSC 控制寄存器具有缓冲，可以在运行过程中改变它的数值，新的预分频数值将在下一个更新事件时起作用。

以下两图是在运行过程中改变预分频系数的例子。

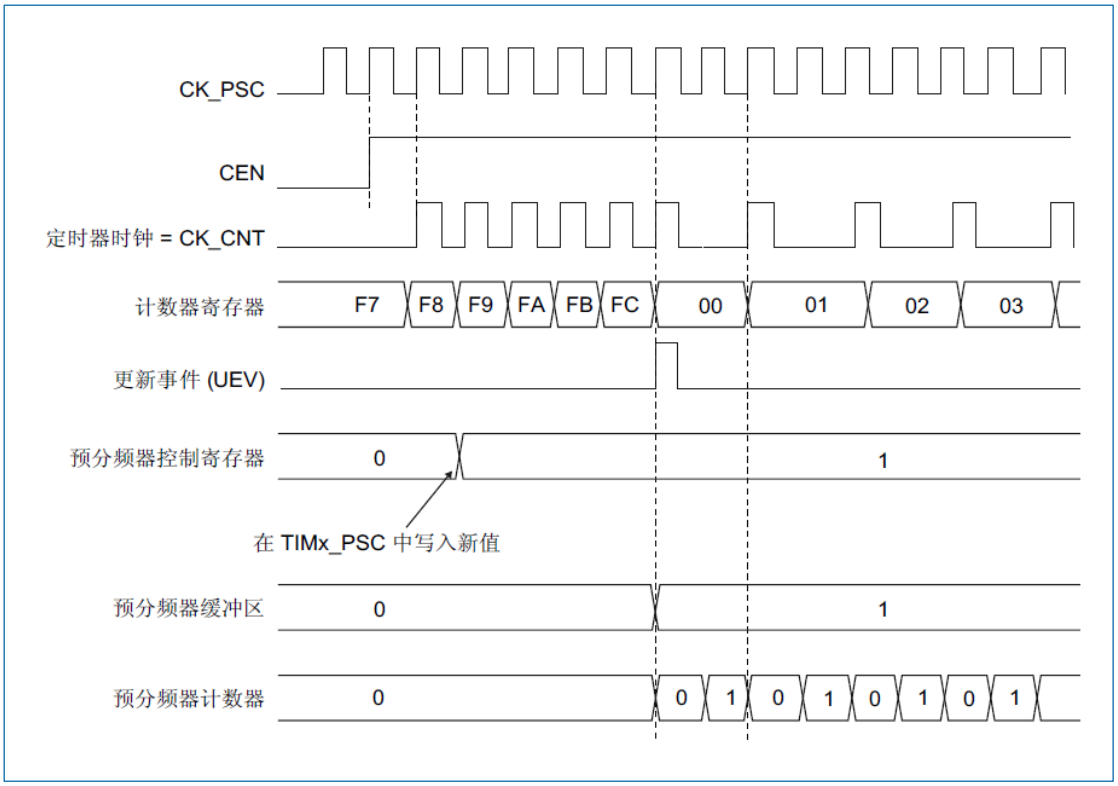


图 14-2 预分频系数从 1 变到 2 的计数器时序图

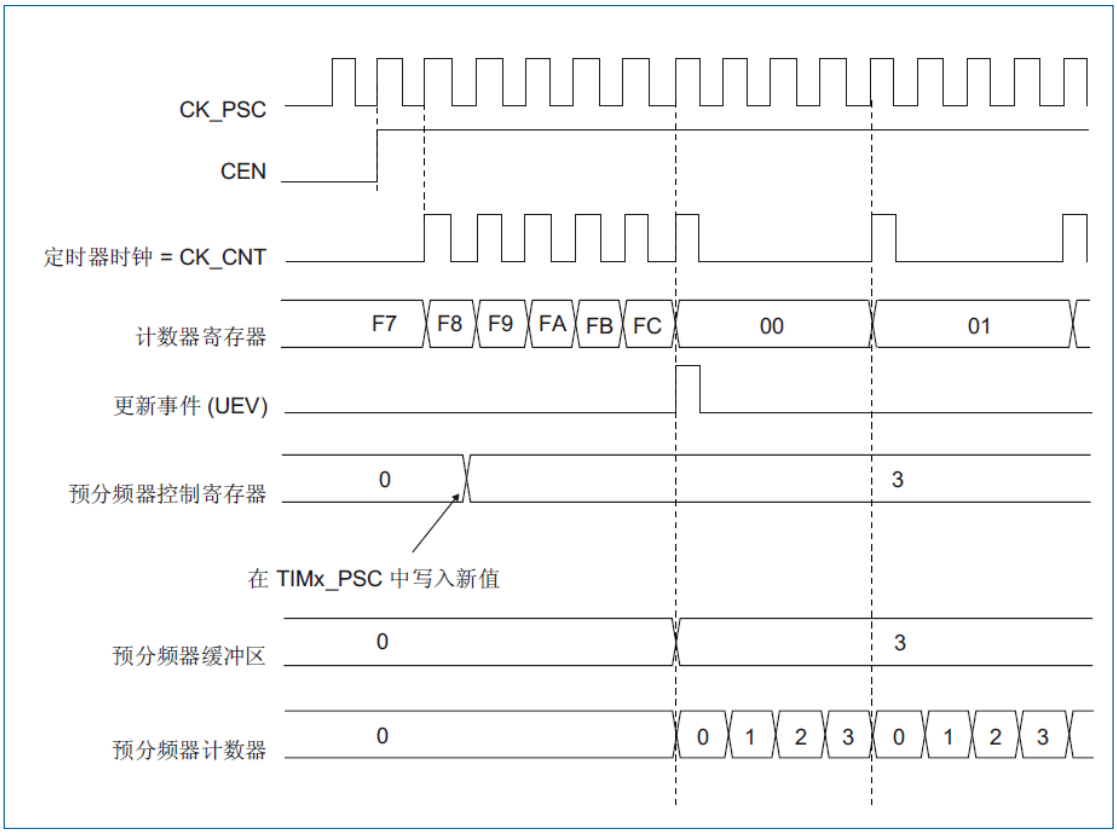


图 14-3 预分频系数从 1 变到 4 的计数器时序图

14.2.2 计数模式

向上计数模式

计数器从 0 累加计数到自动重装载数值（TIM6_ARR 寄存器），然后重新从 0 开始计数并产生一个计数器溢出事件。

每次计数器溢出时可以产生更新事件；（通过软件或使用从模式控制器）设置 TIM6_EGR 寄存器的 UG 位也可以产生更新事件。

设置 TIM6_CR1 中的 UDIS 位可以禁止产生 UEV 事件，这可以避免在写入预加载寄存器时更改影子寄存器。在清除 UDIS 位为’0’之前，将不再产生更新事件，但计数器和预分频器依然会在应产生更新事件时重新从 0 开始计数（但预分频系数不变）。另外，如果设置了 TIM6_CR1 寄存器中的 URS（选择更新请求），设置 UG 位可以产生一次更新事件 UEV，但不设置 UIF 标志（即没有中断）。

当发生一次更新事件时，所有寄存器会被更新并（根据 URS 位）设置更新标志（TIM6_SR 寄存器的 UIF 位）：

- 传送预装载值（TIM6_PSC 寄存器的内容）至预分频器的缓冲区。
- 自动重装载影子寄存器被更新为预装载值（TIM6_ARR）。

以下是一些在 TIM6_ARR=0x36 时不同时钟频率下计数器工作的图示例。

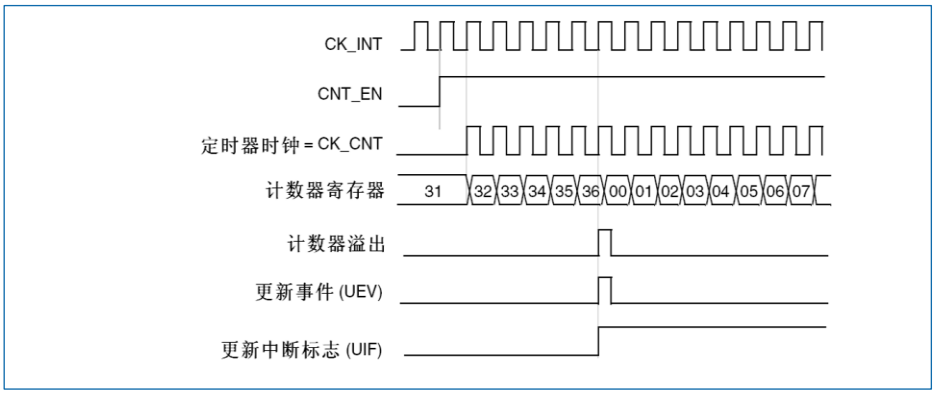


图 14-4 计数器时序图，内部时钟分频系数为 1

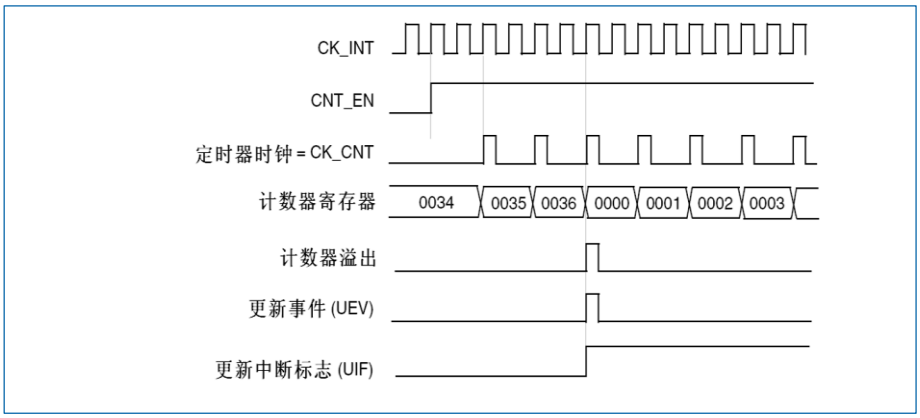


图 14-5 计数器时序图，内部时钟分频系数为 2

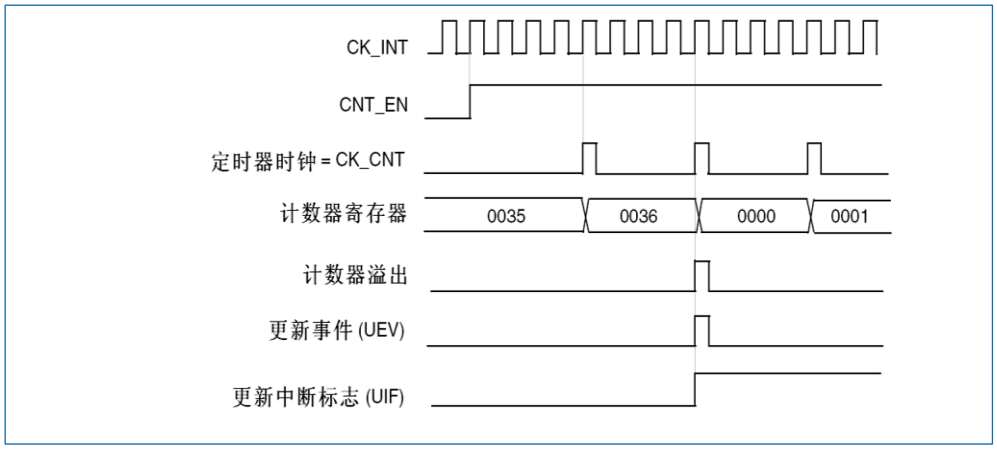


图 14-6 计数器时序图，内部时钟分频系数为 4

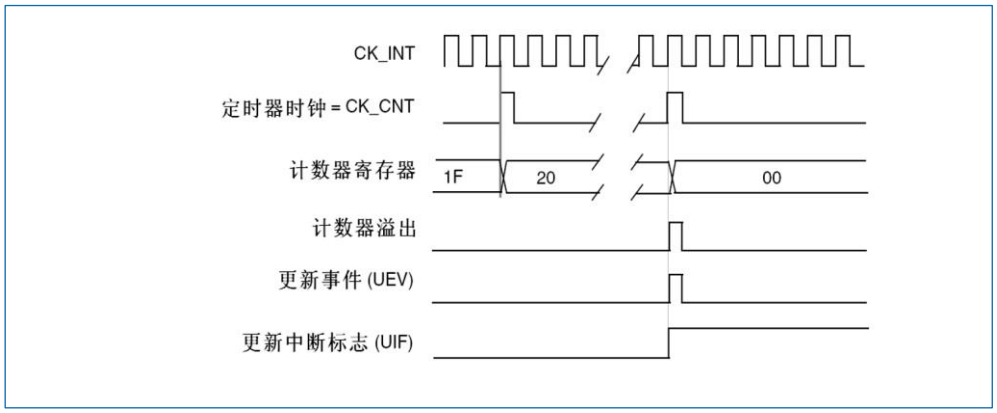


图 14-7 计数器时序图，内部时钟分频系数为 N

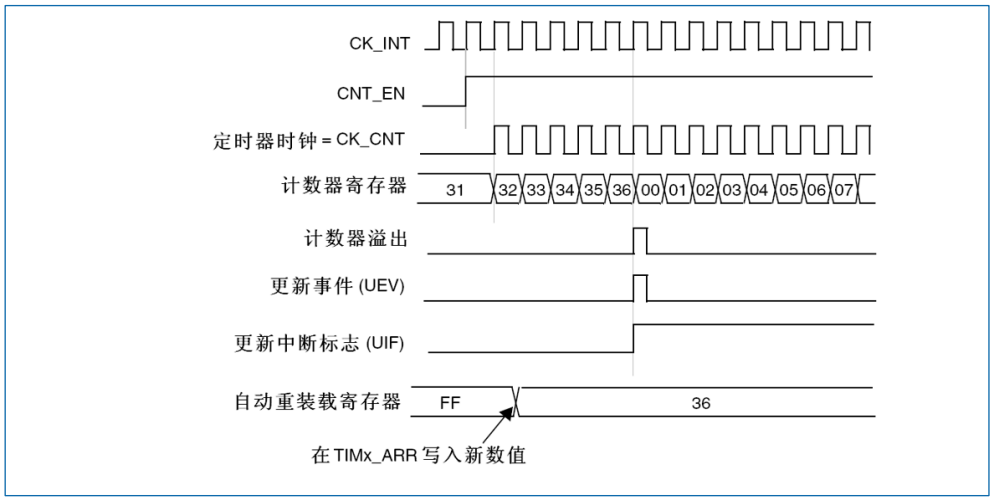


图 14-8 计数器时序图，当 ARPE=1 时的更新事件（无预装载 TIM6_ARR）

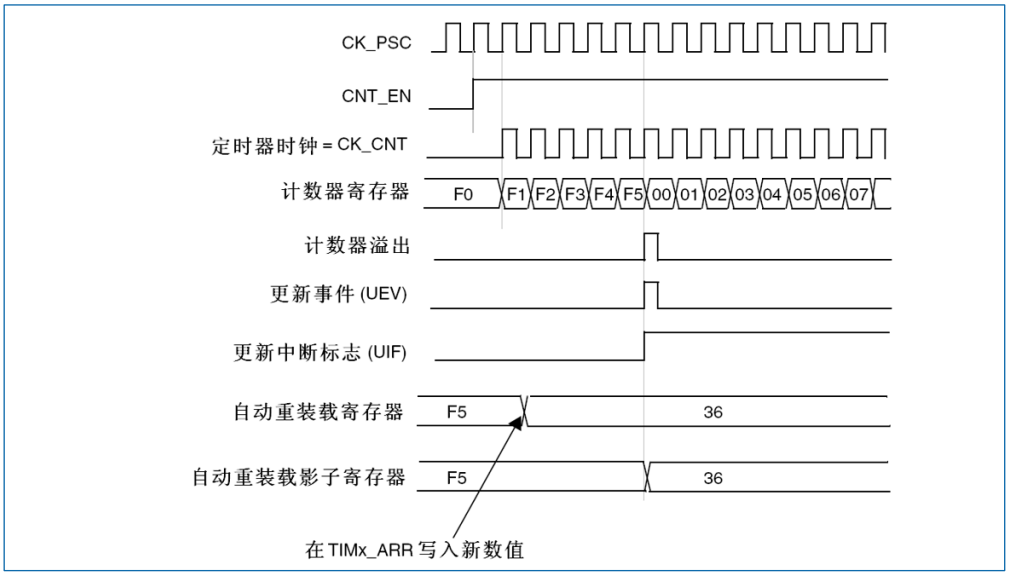


图 14-9 计数器时序图，当 ARPE=1 时的更新事件（预装载 TIM6_ARR）

14.2.3 时钟源

计数器的时钟由内部时钟（CK_INT）提供。

TIM6_CR1 寄存器的 CEN 位和 TIM6_EGR 寄存器的 UG 位是实际的控制位，（除了 UG 位被自动清除外）只能通过软件改变它们。一旦置 CEN 位为’1’，内部时钟即向预分频器提供时钟。

下图示出控制电路和向上计数器在普通模式下，没有预分频器时的操作。

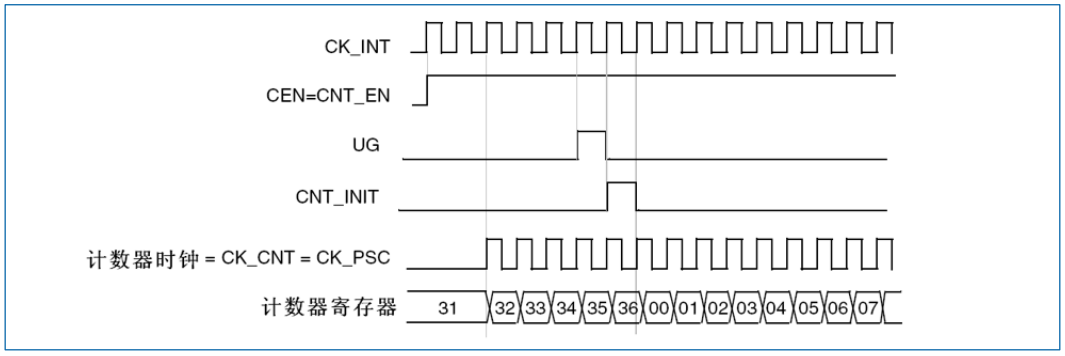


图 14-10 普通模式时序图，内部时钟分频系数为 1

14.2.4 调试模式

当 MCU 进入调试模式（Cortex-M0 内核停止）时，根据 DBG 模块中的配置位 DBG_TIM6_Stop 的设置，TIM6 计数器或者继续计数或者停止工作。

14.3 TIM6 寄存器

基地址：0x4000 1000

空间大小：0x400

14.3.1 TIM6 控制寄存器 1（TIM6_CR1）

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								ARPE	Res		DIR	OPM	URS	UDIS	CEN
								rw			rw	rw	rw	rw	rw

位 15:8	Res：保留 必须保持复位值。
位 7	ARPE：自动重装预装载使能（Auto-reload preload enable） • 0：TIMx_ARR 寄存器无缓冲器 • 1：TIMx_ARR 寄存器有缓冲器
位 6:5	Res：保留 必须保持复位值。
位 4	DIR：方向（Direction） • 0：计数器向上计数 • 1：计数器向下计数
位 3	OPM：单脉冲模式 • 0：在发生更新事件时，计数器不停止。 • 1：在发生下次更新事件时，计数器停止计数（清除CEN 位）。

位 2	URS：更新请求源（Update request source）
	该位由软件置 1 和清 0，以选择UEV 事件的请求源。
	• 0：如果使能了中断请求，以下任一事件可以产生一个更新中断请求：
	  ○ 计数器上溢或下溢
	  ○ 设置UG 位
	  ○ 通过从模式控制器产生的更新
	• 1：如果使能了中断请求，只有计数器上溢或下溢才产生更新中断请求。

位 1	UDIS：禁止更新（Update disable）

	该位由软件置 1 和清 0，以使能或禁止 UEV 事件的产生。
	• 0：UEV 使能。更新事件（UEV）可以由下列事件产生：
	  ○ 计数器上溢或下溢
	  ○ 设置UG 位
	  ○ 通过从模式控制器产生的更新生成更新事件后，带缓冲的寄存器被加载为预加载数值。
	• 1：禁止UEV
	不产生更新事件，影子寄存器保持它的内容（ARR、PSC）。但是如果设置了UG 位或从模式控制器产生了一个硬件复位，则计数器和预分频器将被重新初始化。

位 0	CEN：计数器使能（Counter enable） • 0：计数器禁止 • 1：计数器使能 在单脉冲模式下，当产生更新事件时CEN 被自动清除。

14.3.2 TIM6 控制寄存器 2（TIM6_CR2）

偏移地址：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res									MMS[2:0]			Res			
									rw						

位 15:7	Res：保留 必须保持复位值。
位 6:4	MMS[2:0]：主模式选择（Master mode selection） 这些位用于选择在主模式下向从定时器发送的同步信息（TRGO）。 • 000：复位 使用TIMx_EGR 寄存器的UG 位作为触发输出（TRGO）。如果触发输入产生了复位（从模式控制器配置为复位模式），则相对于实际的复位信号，TRGO 上的信号有一定的延迟。 • 001：使能 计数器使能信号CNT_EN 被用作为触发输出（TRGO）。它可用于在同一时刻启动多个定时器，或控制使能从定时器的时机。计数器使能信号是通过CEN 控制位和配置为门控模式时的触发输入的’逻辑或’产生。当计数器使能信号是通过触发输入控制时，在TRGO 输出上会有一些延迟，除非选择了主/从模式。 • 010：更新 更新事件被用作为触发输出（TRGO）。例如：一个主定时器可以作为从定时器的预分频器使用。
位 3:0	Res：保留 必须保持复位值。

14.3.3 TIM6 中断使能寄存器（TIM6_DIER）

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res															UIE
															rw

位 15:1	Res：保留 必须保持复位值。
位 0	UIE：更新中断使能 • 0：更新中断禁用 • 1：更新中断使能

14.3.4 TIM6 状态寄存器（TIM6_SR）

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res															UIF
															rc_w0

位 15:1	Res：保留 必须保持复位值。

位 0	UIF：更新中断标志（Update interrupt flag）
	硬件在更新中断时设置该位，它由软件清除。
	• 0：没有产生更新。
	• 1：产生了更新中断。满足以下情况，该位由硬件置位：
	  ○ 计数器产生上溢或下溢并且TIMx_CR1 中的UDIS=0。
	  ○ 如果TIMx_CR1 中的URS=0 并且UDIS=0，当使用TIMx_EGR 寄存器的UG 位重新初始化计数器CNT 时。

14.3.5 TIM6 事件产生寄存器（TIM6_EGR）

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res															UG
															w

位 15:1	Res：保留 必须保持复位值。
位 0	UG：产生更新事件（Update generation）

	该位由软件置 1，由硬件自动清 0。 • 0：无作用 • 1：重新初始化定时器的计数器，并更新寄存器。

14.3.6 TIM6 定时器（TIM6_CNT）

偏移地址：0x24

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

位 15:0	CNT[15:0]：计数器值

14.3.7 TIM6 预分频器（TIM6_PSC）

偏移地址：0x28

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

位 15:0	PSC[15:0]：预分频器值 计数器的时钟频率CK_CNT 等于f_{CK_PSC}/（PSC[15:0]+1）。在每一次更新事件时，PSC 的数值被传送到实际的预分频寄存器。

14.3.8 TIM6 自动重装寄存器（TIM6_ARR）

偏移地址：0x2C

复位值：0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw															

位 15:0	ARR[15:0]：自动重装载数值（Prescaler value） ARR 的数值将传送到实际的自动重装载寄存器。如果自动重装载数值为空，则计数器停止。

15 自动唤醒定时器（AWU）

AWU 用于在 MCU 停机模式（Stop）下计时并产生中断唤醒 MCU；AWU 内置超低功耗 22 位定时器，工作时钟可配置为 GPIO 外部输入时钟或 114kHz 片内 LSI 慢速时钟；定时器使用向下计数的方式计数。

15.1 AWU 寄存器

基地址：0x4000 7800

空间大小：0x400

15.1.1 控制寄存器（AWU_CR）

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RLR_WBUSY	Res								AWU_RLR[21:15]						
rw									rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWU_RLR[14:0]															AWU_CKSEL
rw															rw

位 31	RLR_WBUSY：AWU_RLR 是否正在被 APB 总线写操作 • 0：AWU_RLR 寄存器未被 APB 总线写操作。 • 1：AWU_RLR 寄存器正在被 APB 总线写操作。 （当 RLR_WBUSY=1 时，软件禁止再次对 AWU_RLR[21:0]寄存器进行写操作。）
位 30:23	Res：保留 必须保持复位值。
位 22:1	AWU_RLR[21:0]：AWU 计数器转载值 在 MCU 进入 Stop 模式时，此装载值将被自动加载到 AWU 内部的 22 位计时器中并开始计时。 *注意：* *当 AWU_RLR[21:0]=22`d0 或者 22`d1 时，装载行为将不会发生且 AWU 将不会启动工作。* *当 AWU_WBUSY=1 时，对 AWU-RLR 寄存器的写操作将无效。*
位 0	AWU_CKSEL：选择 AWU 定时器的计时时钟 • 0：选择 114 kHz LSI 作为 AWU 定时器的计时时钟。 • 1：选择 GPIO 外部输入时钟作为 AWU 定时器的计时时钟。

15.1.2 控制寄存器（AWU_SR）

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res															AWU_BUSY
															rw

位 31:1	Res：保留 必须保持复位值。
位 0	AWU_BUSY：（只读寄存器） • 0：AWU 计时器未工作。 • 1：AWU 计时器正在工作。 除了 AWU，MCU 在 STOP 模式还可被其它事件唤醒；当第一次从 STOP 模式被其它事件唤醒时，在一段延迟时间内 AWU 仍然在继续计时；在这段延迟时间内，如果 MCU 进入第二次 STOP 模式后，有可能被这段延迟时间内产生的 AWUCNT_INT 事件意外的唤醒。 为了解决第二次 STOP 模式被意外唤醒的问题，需要软件在执行第二个 WFI/WFE 指令前读取 AWU_BUSY 状态标志位，以判断 AWU 是否还在工作。 如果判断 AWU 还在工作，需要等待 AWU_BSY 状态标志位变为 0 后，探测并清除 AWU_INT 中断后，再执行第二个 WFI/WFE 指令。 AWU_INT 中断标志的探测和清除，通过软件访问 EXTI 模块相应寄存器完成。

16 独立看门狗（IWDG）

独立看门狗用于检测和解决由软件失效引起的故障，并在计数器达到指定的超时值时触发系统复位。独立看门狗由专用的低速时钟（LSI）驱动，即使主时钟发生故障它也仍然有效。

IWDG 适用于看门狗能完全独立在主程序之外工作，并且对时间精度要求较低的场合。

16.1 IWDG 主要功能

- 自由运行的递减计数器
- 时钟由独立的 RC 振荡器提供（可在停机模式下工作）
- 复位条件
  - 当递减计数器值小于 0x000 时复位（如果看门狗已激活）
  - 在窗口之外重载递减计数器时复位（如果看门狗已激活）
- IWDG 计数器复位初始值可由 Flash 选项字 IWDG_RL_IV 设置（请参见“3.3 Flash 选项字节”），以确保在芯片复位后若程序跑飞，IWDG 复位能在较短时间内复位。
- 通过 Flash 选项字 LSI_LP_CTL 控制芯片在进入停机模式下 LSI 的状态。（请参见“3.3 Flash 选项字节”）。

16.2 IWDG 功能描述

在关键字寄存器（IWDG_KR）中写入 0xCCCC，开始启用独立看门狗；此时计数器开始从其复位值 0xFFF 递减计数。当计数器计数到末尾 0x000 时，会产生一个复位信号（IWDG_RESET）。

无论何时，只要在键寄存器 IWDG_KR 中写入 0xAAAA，IWDG_RLR 中的值就会被重新加载到计数器，从而避免产生看门狗复位。

支持看门狗窗口模式，详细的描述请参考“16.3.5 窗口寄存器（IWDG_WINR）”。

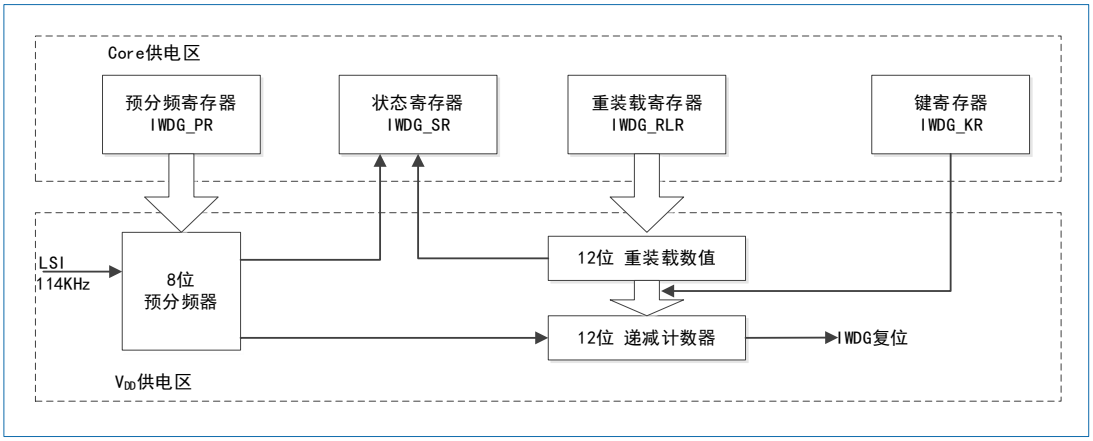


图 16-1 IWDG 框图

说明：看门狗功能处于 V_{DD} 供电区，即在停机模式时仍能正常工作。

计算超时的公式为：

$$T_{IWDG}= ((1/ f_{LSI})/PR)* (RL+1)$$

其中：

- T_{IWDG}：IWDG 超时时间
- f_{LSI}：LSI 的频率

表 16-1 IWDG 超时时间表（114kHz 的输入时钟（LSI））

预分频系数	PR[2:0]位	最短时间（ms） RL[11:0]=0x000	最长时间（ms） RL[11:0]=0xFFF
/4	0	0.035	143.719
/8	1	0.070	287.439
/16	2	0.140	574.877
/32	3	0.281	1149.754
/64	4	0.561	2299.509
/128	5	1.123	4599.018
/256	6 或 7	2.246	9198.035

注意:

这些时间是按照 114 kHz 时钟给出。实际上，MCU 内部的 RC 频率会在 114kHz(1-10%)到 114kHz*(1+10%)之间变化。此外，即使 RC 振荡器的频率是精确的，确切的时序仍然依赖于 APB 接口时钟与 RC 振荡器时钟之间的相位差，因此总会有一个完整的 RC 周期是不确定的。*

通过对 LSI 进行校准可获得相对精确的看门狗超时时间。

16.2.1 窗口选项

通过在 IWDG_WINR 寄存器中设置合适的窗口，IWDG 也可以用作窗口看门狗。当计数器值大于窗口寄存器（IWDG_WINR）中存储的值时，如果执行重载操作，则会产生复位。IWDG_WINR 的默认值为 0x0000 0FFF，如果不更新默认值，将会禁用窗口选项。窗口值一经更改，便会执行重载操作，以便将递减计数器的值复位为 IWDG_RLR 值，方便计算周期数以生成下一次重载。

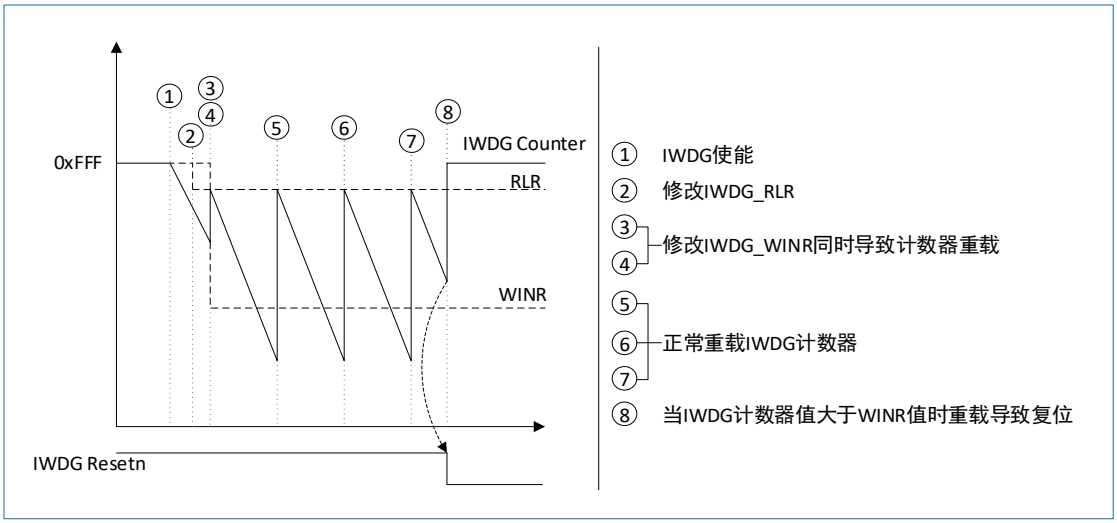


图 16-2 IWDG 使能窗口选项时的工作说明

使能窗口选项时，配置 IWDG 的流程如下：

1. 向 IWDG_KR 寄存器写入 0x0000 CCCC 来使能 IWDG。
2. 向 IWDG_KR 寄存器写入 0x0000 5555 来使能寄存器访问。
3. 修改 IWDG_PR 寄存器的值为 0~7 中的值，以配置 IWDG 的预分频器。
4. 写重载寄存器 IWDG_RLR。

5. 等待 IWDG_RLR 寄存器值更新完成（IWDG_SR=0x0000 0000）。
6. 写窗口寄存器 IWDG_WINR。该操作会导致 IWDG 计数器自动更新为 IWDG_RLR 中的值。

说明：当 IWDG_SR 设置为 0x0000 0000 时，才能写 IWDT_WINR 寄存器，否则 IWDG 计数器可能不会正确的重载为 IWDG_RLR 中的值。

禁用窗口选项时，配置 IWDG 的流程如下：

1. 向 IWDG_KR 寄存器写入 0x0000 CCCC 来使能 IWDG。
2. 向 IWDG_KR 寄存器写入 0x0000 5555 来使能寄存器访问。
3. 修改 IWDG_PR 寄存器的值为 0~7 中的值，以配置 IWDG 的预分频器。
4. 写重载寄存器 IWDG_RLR。
5. 等待 IWDG_RLR 寄存器值更新完成（IWDG_SR=0x0000 0000）。
6. 刷新计数器值为 IWDG_RLR 值（IWDG_KR=0x0000 AAAA）。

16.2.2 硬件看门狗

如果用户在选择字节中启用了“硬件看门狗”功能，在系统上电复位后，看门狗会自动开始运行；在计数器计数结束前，若软件没有向 IWDG_KR 寄存器写入 0x0000 AAAA 以重载 IWDG 计数器，则系统会产生复位。

16.2.3 寄存器访问保护

IWDG_PR、IWDG_RLR 和 IWDG_WINR 寄存器具有写保护功能。若需修改这两个寄存器的值，必须先向 IWDG_KR 寄存器中写入 0x0000 5555。以不同的值写入这个寄存器将会打乱操作顺序，寄存器将重新被保护。重装载操作（即写入 0x0000 AAAA）也会启动写保护功能。

状态寄存器（IWDG_SR）指示预分频值和递减计数器是否正在被更新。

16.2.4 调试模式

当 MCU 进入调试模式时（Cortex-M0 内核停止），根据调试模块中的 DBG_IWDG_Stop 配置位的状态，IWDG 的计数器能够继续或停止工作。参见“23.9.3 调试 MCU APB1 冻结寄存器（DBGMCU_APB1_FZ）”。

16.3 IWDG 寄存器

基地址：0x4000 3000

空间大小：0x400

16.3.1 关键字寄存器（IWDG_KR）

地址偏移：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
w															

位 31:16	Res：保留

	必须保持复位值。
位 15:0	KEY[15:0]：关键值（只写，读的值为 0x0000） 该位域必须周期性的由软件写入 0xAAAA，否则当计数器向下计数到 0 的时候会产生硬件复位请求。写入 0x5555 会使能 IWWDG_PR，IWWDG_RLR 和 IWWDG_WINR 寄存器的访问。写入 0xCCCC 会启动看门狗（除非已经由选项字节在上电时启动看门狗）。

16.3.2 预分频寄存器（IWDG_PR）

地址偏移：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res													PR[2:0]		
													rw		

位 31:3	Res：保留 必须保持复位值。
位 2:0	PR[2:0]：预分频器 该位域平时处于写保护状态，可选择输入时钟的预分频系数。为了能够改变预分频器的分频系数，IWWDG_SR 寄存器中的 PVU 位必须先清零。 • 000：4 分频 • 001：8 分频 • 010：16 分频 • 011：32 分频 • 100：64 分频 • 101：128 分频 • 110：256 分频 • 111：256 分频

16.3.3 重加载寄存器（IWDG_RLR）

地址偏移：0x08

复位值：0x0000 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				RL[11:0]											
				rw											

位 31:12	Res：保留 必须保持复位值。

位 11:0	RL[11:0]：看门狗计数器重置值 该位域平时处于写保护状态，通过软件来设置的。每次向 IWWDG_KR 寄存器写入 0xAAAA，这个值会被更新到看门狗计数器。如需更长的延时，则将该值配置更大。因为看门狗计数器是从该值开始向下计数。定时的长度是由该值和预分频器的设置值来共同决定的。

16.3.4 状态寄存器（IWDG_SR）

地址偏移：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res													WVU	RVU	PVU
													r		

位 31:3	Res：保留 必须保持复位值。
位 2	WVU：看门狗计数器窗口值更新 该位由硬件置位，表示正在更新窗口值。当 VDD 供电区域中完成了看门狗定时器的重加载时，该位会被硬件清零（这需要 5 个 114 kHz 的 RC 振荡周期）。只有当 WVU 的值为零时，才能再次改写窗口值。该位只在窗口功能打开的时候有效。
位 1	RVU：看门狗计数器重置值更新 该位由硬件置位，表示正在更新定时器重置值。当 VDD 供电区域中完成了看门狗定时器的重加载时，该位会被硬件清零（这需要 5 个 114 kHz 的 RC 振荡周期）。只有当 RVU 的值为零时，才能再次改写重置值。
位 0	PVU：看门狗预分频器设置更新 该位由硬件置位，表示正在更新预分频器设置。当 VDD 供电区域中完成了看门狗定时器的重加载时，该位会被硬件清零（这需要 5 个 114 kHz 的 RC 振荡周期）。只有当 WVU 的值为零时，才能再次改写预分频器的值。

16.3.5 窗口寄存器（IWDG_WINR）

地址偏移：0x10

复位值：0x0000 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				WIN[11:0]											
				rw											

位 31:12	Res：保留

	必须保持复位值。
位 11:0	WIN[11:0]：看门狗计数器窗口值

17 系统窗口看门狗（WWDG）

系统窗口看门狗（WWDG）通常被用来监测由外部干扰或不可预见的逻辑条件造成的应用程序背离正常的运行次序而产生的软件故障。除非程序在 T6 位变成 0 前刷新递减计数器的值，否则看门狗电路在达到预置的时间周期时，会产生一个 MCU 复位。如果在递减计数器达到窗口寄存器值之前刷新控制寄存器中的7位递减计数器值，也会产生MCU复位。这意味着必须在限定的时间窗口内刷新计数器。

窗口看门狗时钟由 APB 时钟分频得到，它具有可配置的时间窗口。该窗口可编程，以检测异常过晚或过早的应用行为。

窗口看门狗适用于那些需要看门狗工作在准确的时间窗口下的应用。

17.1 WWDG 主要特性

- 可编程的自由运行递减计数器
- 条件复位
  - 当递减计数器的值小于 0x40，（若看门狗被启动）则产生复位。
  - 当递减计数器在窗口外被重新装载，（若看门狗被启动）则产生复位。
- 如果启动了看门狗并且允许中断，当递减计数器等于 0x40 时产生早期唤醒中断（Early wakeup interrupt，EWI），它可以用于重装载计数器以避免 WWDG 复位。

17.2 WWDG 功能描述

如果启动了看门狗（WWDG_CR 寄存器中的 WDGA 位被置’1’），并且当 7 位（T[6:0]）递减计数器从 0x40 递减到 0x3F（T6 位清零）时，则产生复位信号。如果在计数器值大于窗口寄存器中的数值时，软件重载计数器，则会产生复位。

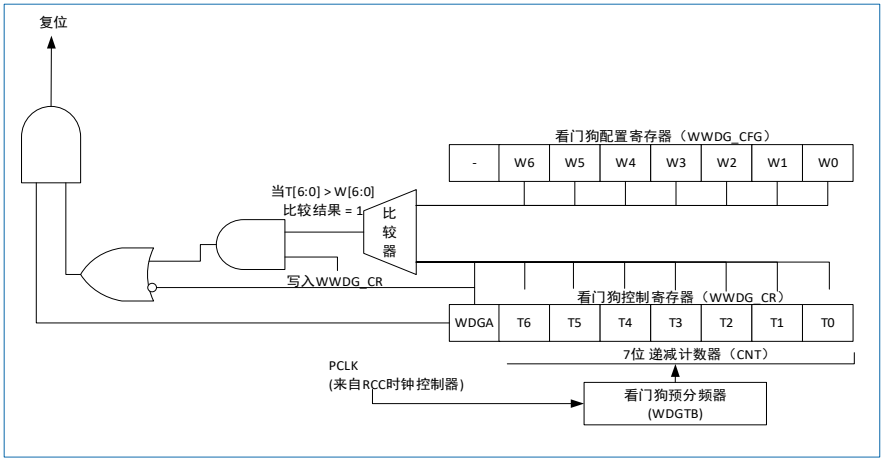


图 17-1 WWDG 框图

应用程序在正常运行过程中必须定期地写入 WWDG_CR 寄存器以防止 MCU 发生复位。只有当计数器值小于窗口寄存器的值时，才能进行写操作。储存在 WWDG_CR 寄存器中的数值必须在 0xFF 和 0xC0 之间。

17.2.1 启动看门狗

在系统复位后，看门狗总是处于关闭状态。设置 WWDG_CR 寄存器的 WDGA 位能够开启看门狗，随后它不能再被关闭，除非发生复位。

17.2.2 控制递减计数器

递减计数器处于自由运行状态，即使禁止看门狗，递减计数器仍继续向下计数。当看门狗被启用时，T6 位必须被置 1，以防止立即复位。

T[5:0]位包含了看门狗产生复位之前的计时数目；复位前的延时时间在一个最小值和一个最大值之间变化，这是因为写入 WWDG_CR 寄存器时，预分频值是未知的。

配置寄存器（WWDG_CFR）中包含窗口的上限值：避免产生复位，递减计数器必须在其值小于窗口寄存器的数值并且大于 0x3F 时被重载，图 17-2 介绍了窗口看门狗的工作过程。

说明：可以用 T6 产生一个软件复位（设置 WDGA 位为 ‘1’，T 位为 ‘0’）。

17.2.3 看门狗中断高级特性

如果在产生实际复位之前必须执行特定的安全操作或数据记录，则可使用提前唤醒中断（EWI）。通过设置 WWDG_CFR 寄存器中的 WEI 位使能 EWI 中断。当递减计数器的值为 0x40 时，将生成 EWI 中断。在复位器件之前，可以使用相应的中断服务程序（ISR）来触发待定操作（例如通信或数据记录）。

在某些应用中，可以使用 EWI 中断来管理软件系统检查和/或系统恢复/功能退化，而不会生成 WWDG 复位。在这种情况下，相应的中断服务程序（ISR）可用来重载 WWDG 以避免 WWDG 复位，然后再触发所需操作。

通过将 0 写入 WWDG_SR 寄存器中的 EWIF 位来清除 EWI 中断。

说明：当由于在更高优先级任务中有系统锁定而无法使用 EWI 中断时，最终会产生 WWDG 复位。

17.2.4 如何编写看门狗超时程序

可以使用下图中提供的公式计算 WWDG 的超时时间。

警告：当写入 WWDG_CR 寄存器时，请始终置 T6 位为‘1’以避免立即产生复位。

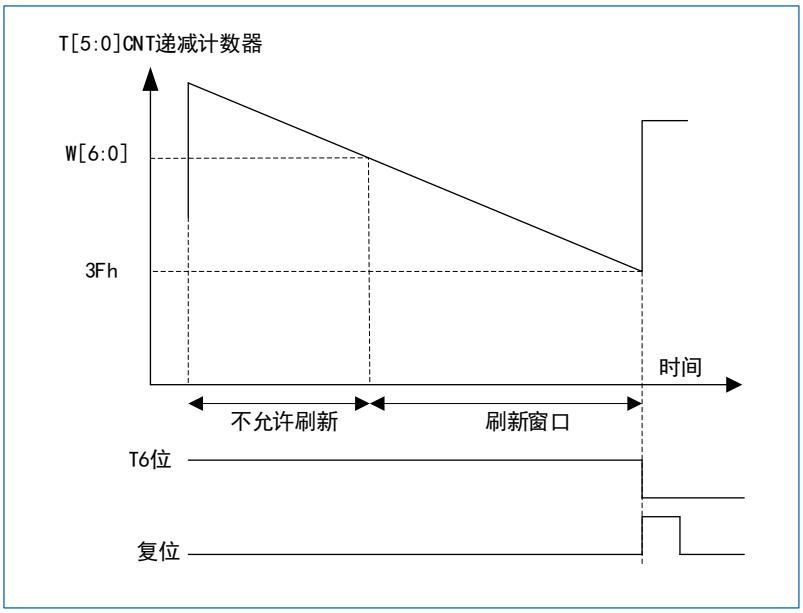


图 17-2 WWDG 工作时间计算说明

如图 17-2 所示，计算超时的公式为：

$T_{WWDG}=T_{PCLK} \times 4096 \times 2^{WDGTB[1:0]} \times (T[5:0]+1)$ （ms）

其中：

- T_{WWDG}：WWDG 超时时间；

- T_{PCLK}：以 ms 为单位的 APB 时钟周期。
- 4096：对应于内部分频器的值
- WDGTB[1:0]：分频器分频系数

例如：假设 APB 频率等于 32 MHz，将 WDGTB[1:0]设置为 3，将 T[5:0]设置为 63：

T_{WWDG} = = 1/ 32000 × 4096 × 2^3 × (63 +1) = 65.54 ms

表 17-1 在 PCLK=32MHz 时的最小~最大超时时间表

WDGTB	最小超时值(µs)	最大超时值(ms)
0	128	8.192
1	256	16.384
2	512	32.768
3	1024	65.536

17.2.5 调试模式

当 MCU 进入调试模式时（Cortex-M0 内核停止），WWDG 计数器会根据 DBG 模块中的 DBG_WWDG_STOP 配置位选择继续正常工作或者停止工作。

17.3 WWDG 寄存器

基地址：0x4000 2C00

空间大小：0x400

17.3.1 控制寄存器（WWDG_CR）

地址偏移：0x00

复位值：0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								WDGA	T[6:0]						
								rs	rw						

位 31:8	Res：保留 必须保持复位值。
位 7	WDGA：激活位 该位由软件置 1，仅能由硬件在复位后清 0。当WDGA=1 时，看门狗可以产生复位。 • 0：禁止看门狗 • 1：启用看门狗
位 6:0	T[6:0]：7 位计数器（MSB 到 LSB） 该位域用于存储看门狗的计数器值。每（4096 x 2^{WDGTB}）个PCLK1 周期减1。当计数

	器值从0x40 变为0x3F 时（T6 变成 0），产生看门狗复位。

17.3.2 配置寄存器（WWDG_CFR）

地址偏移：0x04

复位值：0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res						EWI	WDGTB[1:0]		W[6:0]						
						rs	rw		rw						

位 31:10	Res：保留 必须保持复位值。
位 9	EWI：提前唤醒中断 此位若置 1，则当计数器值达到0x40，即产生中断。此中断只能由硬件在复位后清除。
位 8:7	WDGTB[1:0]：时基预分频器 • 00：CK 计时器时钟（PCLK1/4096）1 分频 • 01：CK 计时器时钟（PCLK1/4096）2 分频 • 10：CK 计时器时钟（PCLK1/4096）4 分频 • 11：CK 计时器时钟（(PCLK1/4096）8 分频
位 6:0	W[6:0]：7 位窗口值 该位域包含了用来与递减计数器进行比较用的窗口值。

17.3.3 状态寄存器（WWDG_SR）

地址偏移：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res															EWIF
															rc_w0

位 31:1	Res：保留 必须保持复位值。
位 0	EWIF：提前唤醒中断标志 当计数器的值达到0x40 时，此位由硬件置 1。它必须通过软件写 0 来清除。对此位写 1 无效。若中断未被启用，此位也会被置 1。

18 内部集成电路接口（I2C）

内部集成电路（I2C）总线接口处理 MCU 与串行 I2C 总线间的通信。它遵循 I2C 规范，支持标准模式（Standard mode，Sm）、快速模式（Fast mode，Fm）和超快速模式（Fast mode pluse，Fm+）。

它与系统管理总线（System management bus，SMBus）和电源管理总线（Power management bus，PMBus）兼容。

18.1 I2C 主要特性

- 兼容 I2C 总线规范第 03 版
  - 支持从模式和主模式
  - 支持多主设备
  - 支持标准速度模式（高达 100 kHz）
  - 支持快速模式（高达 400 kHz）
  - 支持超速模式（高达 1 MHz）
  - 支持 7 位和 10 位寻址模式
  - 支持多个 7 位从地址（2 个从设备地址寄存器，1 个具有可配置的掩码位段）
  - 所有 7 位地址应答模式
  - 支持广播呼叫
  - 支持总线上的数据建立和保持时间可软件配置
  - 支持事件管理
  - 支持时钟延展功能
  - 支持软件复位
- 可编程模拟和数字噪声滤波器
- 兼容 SMBus 规范第 2.0 版
  - 支持硬件数据包错误校验（Packet Error Checking，PEC）生成和验证
  - 支持命令和数据应答控制
  - 支持地址解析协议（Address Resolution Protocol，ARP）
  - 支持主机和从设备
  - 支持 SMBus 报警
  - 支持超时和空闲条件检测
- 兼容 PMBus 第 1.1 版标准
- 独立时钟：选择独立时钟源可使 I2C 通信速度不受 PCLK 时钟频率更改的影响
- 地址匹配时从停机模式唤醒

18.2 I2C 功能说明

该接口通过数据引脚（SDA）和时钟引脚（SCL）连接到 I2C 总线。它可以连接到标准速度（高达 100 kHz）、快速（高达 400 kHz）或超速模式（高达 1 MHz）I2C 总线。

该接口也可通过数据引脚（SDA）和时钟引脚（SCL）连接到 SMBus。还可使用额外的 SMBus 报警引脚（SMBA）。

18.2.1 I2C 框图

I2C 接口的框图如图 18-1 所示。

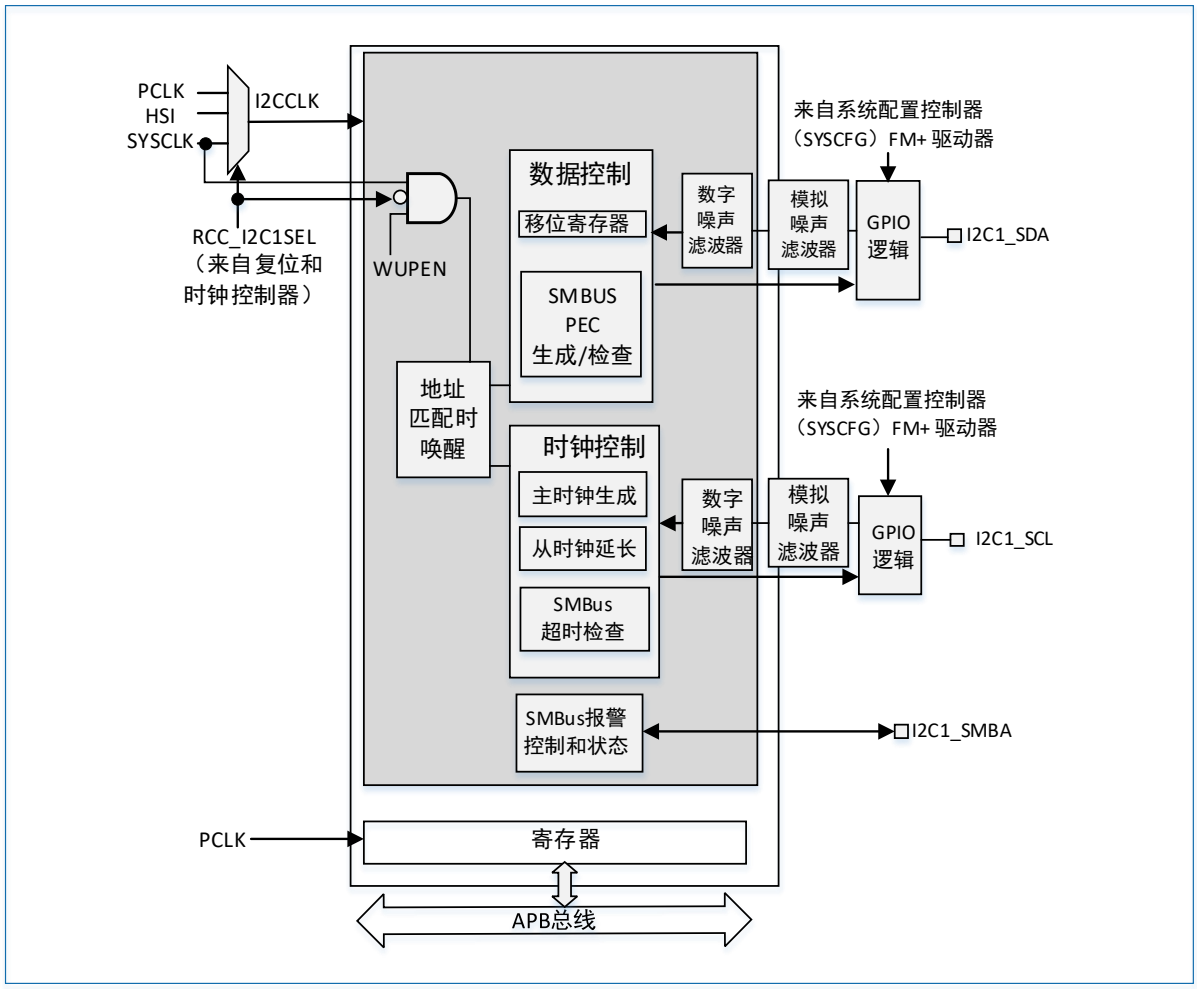


图 18-1 I2C 框图

I2C 的时钟由独立时钟源提供，这使得 I2C 能够独立于 PCLK 频率工作。

该时钟源可以是以下任一时钟源：

- PCLK：APB 时钟（默认值）
- HSI：内部 RC 振荡器
- SYSCLK：系统时钟

更多详细信息，请参见“6 复位和时钟控制（RCC）”。

18.2.2 I2C 时钟要求

I2C 内核的时钟由 I2CCLK 提供。

I2CCLK 周期 t_{I2CCLK} 必须遵循以下条件：

$$t_{I2CCLK} < （t_{LOW} - t_{filters}）/4 \text{ 且 } t_{I2CCLK} < t_{HIGH}$$

其中：

- t_{LOW}：SCL 低电平时间
- t_{HIGH}：SCL 高电平时间
- $t_{filters}$：滤波器使能时，该值为模拟滤波器和数字滤波器引入的延时总和。模拟滤波器延时最大

值为 260 ns。数字滤波器延时为 DNF x t_{I2CCLK}。

PCLK 时钟周期 t_{PCLK} 必须遵循以下条件：

$$t_{PCLK}<4/3t_{SCL}$$

其中，t_{SCL} 表示 SCL 周期。

说明：当 I2C 内核的时钟由 PCLK 提供时，PCLK 必须遵循 t_{I2CCLK} 的条件。

18.2.3 模式选择

该接口在工作时可选用以下四种模式之一：

- 从发送器
- 从接收器
- 主发送器
- 主接收器

默认工作模式是从模式。

通信流程

在主模式下，I2C 接口会启动数据传输并生成时钟信号。串行数据传输始终是在出现起始位时开始，在出现停止位时结束。在主模式下，起始条件 START 和地址由软件触发，停止位可由软件生成，也可由硬件自动生成。

在从模式下，该接口能够识别其自身地址（7 或 10 位）以及广播呼叫地址。广播呼叫地址检测可由软件使能或禁止。SMBus 总线中的主机地址、器件默认地址、报警地址也可由软件使能是否进行响应。

数据和地址均以 8 位（字节）传输，MSB 在前。起始位后紧随地址字节（7 位地址占据一个字节；10 位地址占据两个字节）。地址始终由主设备发出。

在传输一个字节所需的 8 个时钟周期后是第 9 个时钟脉冲。在第 9 个时钟期间，接收器必须向发送器发送一个应答位（ACK），如下图所示。

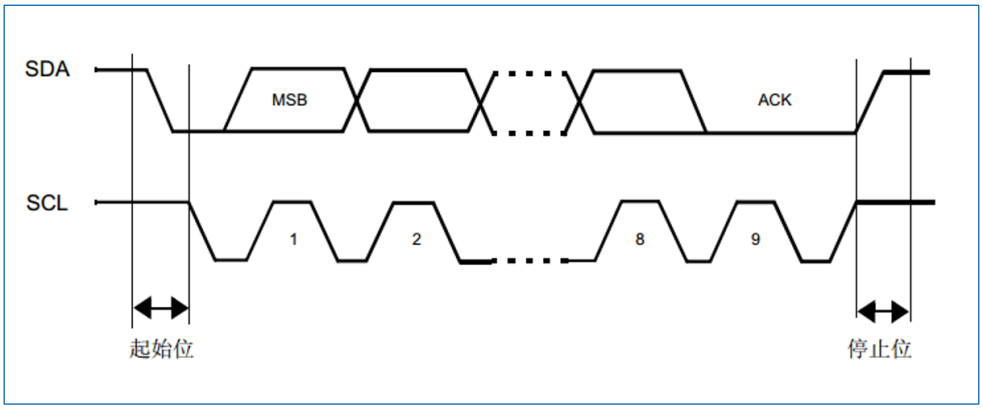


图 18-2 I2C 总线协议

ACK 信号可由软件使能或禁止。I2C 接口地址可通过软件进行选择。

18.2.4 I2C 初始化

使能和禁止外设

I2C 外设时钟必须在时钟控制器中进行配置和使能（请参见“6 复位和时钟控制（RCC）”）。通过将 I2C_CR1 寄存器中的 PE 位置 1 可使能 I2C。

当禁止 I2C（PE=0）时，I2C 将执行软件复位。更多详细信息，请参见“18.2.5 软件复位”。

噪声滤波器

SDA 和 SCL 输入上集成了模拟和数字噪声滤波器。如果用户要使用滤波器，必须在使能 I2C 模块之前完成滤波器的配置。

模拟滤波器符合 I2C 规范，此规范要求在快速模式下对脉宽在 50 ns 以下的脉冲都要抑制。模块默认使能模拟滤波器，用户可通过将 ANFOFF 位置 1 来禁止该模拟滤波器。

数字滤波器通过配置 I2C_CR1 寄存器中的 DNF[3:0]位可实现对尖峰脉宽 1 到 15 个 I2CCLK 的噪声抑制。使能数字滤波器时，SCL 或 SDA 线的电平只有在电平稳定时间超过 DNFxI2CCLK 个周期后才会发生内部变化。

表 18-1 模拟滤波器与数字滤波器对比

	模拟滤波器	数字滤波器
抑制尖峰的脉冲宽度	≥ 50ns	可编程长度为 1 到 15 个 I2C 外设时钟
优点	停机模式下仍可用	• 可编程长度：额外的滤波能力与标准要求 • 稳定的长度
缺点	随温度、电压和过程变化会发生变化	当使能数字滤波器后，无法在地址匹配时从停机模式唤醒。

说明：使能 I2C 时，不允许更改滤波器配置。

I2C 时序

必须配置时序，以保证主模式和从模式下使用正确的数据保持和建立时间。配置方法是编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLDEL[3:0]和 SDADEL[3:0]位。

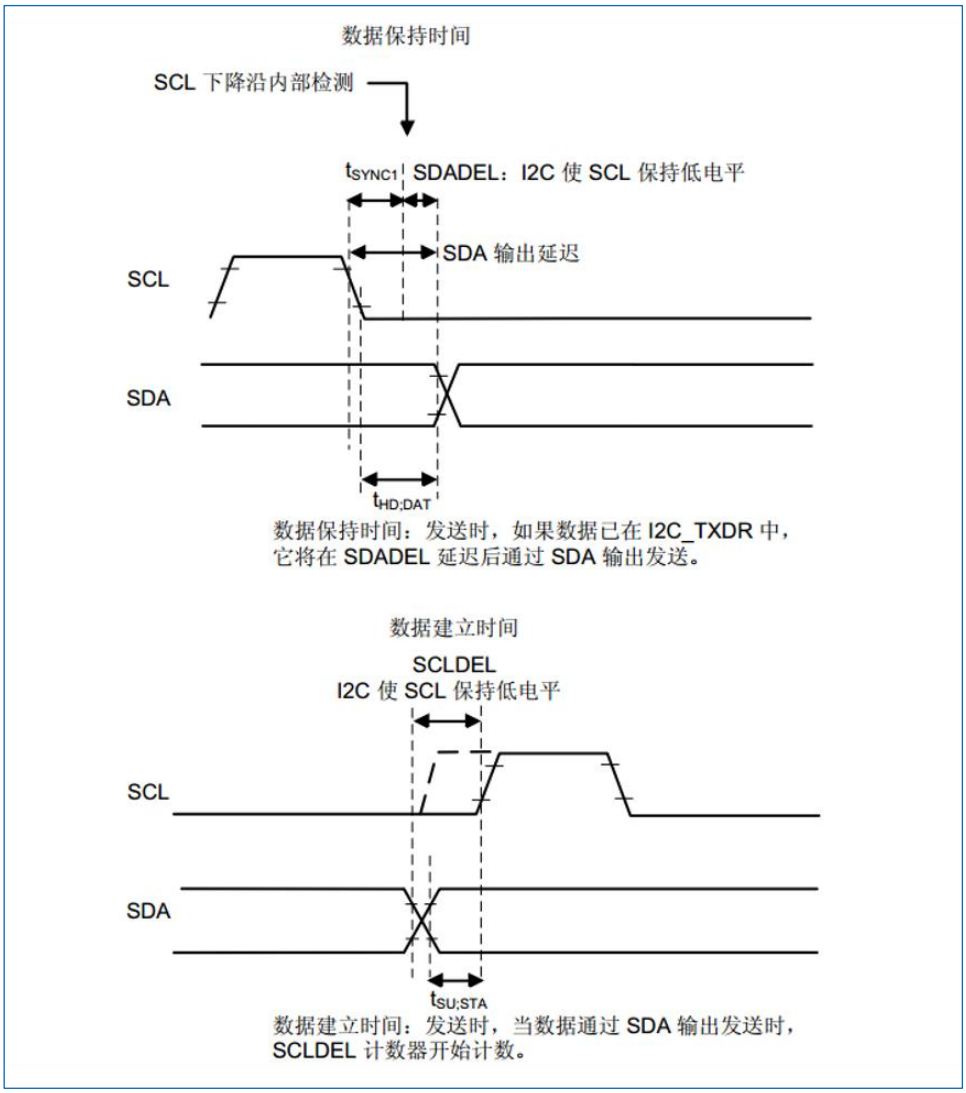


图 18-3 建立和保持时序

- 当内部检测到 SCL 下降沿时，会在发送 SDA 输出之前插入一段延时。该延时为 t_{SDADEL}= *SDADEL x* t_{PRESC} + t_{I2CCLK}，其中 t_{PRESC} =（PRESC+1）x t_{I2CCLK}

  TSDADEL 会影响保持时间 $t_{HD;DAT}$。

- SDA 总输出延时为：

  t_{SYNC1}+{[SDADEL x (PRESC+1)+1] x t_{I2CCLK}}

  t_{SYNC1} 持续时间取决于以下参数：

  - SCL 下降斜率
  - 模拟滤波器（使能时）引入的输入延时：t_{AF}（min）<T_{AF}<T_{AF}（max）ns
  - 数字滤波器（使能时）引入的输入延时：t_{DNF}=DNF x t_{I2CCLK}
  - SCL 与 I2CCLK 时钟建立同步而产生的延时（2 到 3 个 I2CCLK 周期）

为了桥接 SCL 下降沿的未定义区域，用户编程 SDADEL 时必须遵循以下条件：

$$\{t_{f(max)}+t_{HD;DAT(min)}-t_{AF(min)}-[(DNF+3)xt_{I2CCLK}]\}/\{(PRESC+1)xt_{I2CCLK}\} \leq SDADEL$$

$$SDADEL \leq \{t_{HD;DAT(max)}-t_{AF(max)}-[(DNF+4)xt_{I2CCLK}]\}/\{(PRESC+1)xt_{I2CCLK}\}$$

说明：只有使能模拟滤波器时，公式中才包含 t_{AF}（min）/t_{AF}（max）。有关 t_{AF} 值的信息，请参见 HK32F030M 数据手册。

标准模式和快速模式下的 $T_{HD;DAT}$ 最大值分别可达 3.45 μs、0.9 μs 和 0.45 μs，但必须小于 $T_{VD;DAT}$ 最

大值（差值为跳变时间）。只有器件未延长 SCL 信号的低电平周期（t_{LOW}）时，才必须满足该最大值条件。如果时钟延长 SCL，数据必须在建立时间内保持有效，之后才能释放时钟。

SDA 上升沿通常为最坏情况，因此在这种情况下，上述公式变成如下形式：

$$SDADEL \leq \{t_{VD;DAT（max）}-t_{R（max）}-260ns-[（DNF+4）xt_{I2CCLK}]\}/\{（PRESC+1）x\ t_{I2CCLK}\}$$

说明：NOSTRETCH=0 时会违反该条件，这是因为器件会根据 SCLDEL 值来延长 SCL 低电平时间，以保证建立时间。

有关 t_f、tr、$t_{HD;DAt}$ 和 $t_{VD;DAt}$ 标准值的信息，请参见表 18-2。

- 在 t_{SDADEL} 延时后，或在因数据未写入 I2C_TXDR 寄存器而导致从器件必须延长时钟的情况下发送 SDA 输出后，SCL 线会在建立时间内保持低电平。该建立时间为 $t_{SCLDEL}=（SCLDEL+1）x\ t_{PRESC}$，其中 $t_{PRESC}=（PRESC+1）xt_{I2CCLK}$。

t_{SCLDEL} 会影响建立时间 $t_{SU;DAT}$。

为了桥接 SDA 跳变（上升沿通常为最坏情况）的未定义区域，编程 SCLDEL 时必须遵循以下条件：

$$\{[t_R（max）+t_{SU;DAt}（min）]/[（PRESC+1）]xt_{I2CCLK}\}-1<=SCLDEL$$

有关 tr 和 $t_{SU;DAt}$ 标准值的信息，请参见表 18-2。

将使用的 SDA 和 SCL 跳变时间值就是应用中的值。使用最大值而非标准值会增加 SDADEL 和 SCLDEL 计算的约束条件，但能够确保任意应用的特性。

说明：在发送和接收模式下，对于每个时钟脉冲，检测到 SCL 下降沿后，I2C 主器件或从器件会至少在[(SDADEL+SCLDEL+1) x (PRESC+1) + 1] x t_{I2CCLK} 期间内延长 SCL 低电平时间。在发送模式下，如果 SDADEL 计数器计数结束后数据还未写入 I2C_TXDR，则 I2C 会继续延长 SCL 低电平时间，直到写入下一个数据。随后，会将新数据 MSB 发送到 SDA 输出，SCLDEL 计数器将开始计数，同时会继续延长 SCL 低电平时间以确保提供充足的数据建立时间。

如果从模式下 NOSTRETCH=1，则 SCL 不会延长。因此，编程 SDADEL 时还必须确保提供充足的建立时间。

表 18-2 I2C-SMBus 规范数据建立和保持时间

符号	参数	标准模式（Sm）		快速模式（Fm）		超快速模式（Fm+）		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
$t_{HD;DAT}$	数据保持时间	0	-	0	-	0	-	0.3	-	μs
$t_{VD;DAT}$	数据有效时间	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	数据建立时间	250	-	100	-	50	-	250		ns
tr	SDA 和 SCL 信号的上升时间	-	1000	-	300	-	120	-	1000	
tf	SDA 和 SCL 信号的下	-	300	-	300	-	120	-	300	

符号	参数	标准模式（Sm）		快速模式（Fm）		超快速模式（Fm+）		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
	降时间									

此外，在主模式下，必须通过编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLH[7:0]和 SCLL[7:0]位来配置 SCL 时钟的高电平和低电平。

- 当内部检测到 SCL 下降沿时，会在释放 SCL 输出之前插入一段延时。该延时为 t_{SCLL}=（SCLL+1）x t_{PRESC}，其中 t_{PRESC}=（PRESC+1）x t_{I2CCLK}。t_{SCLL} 会影响 SCL 低电平时间 t_{LOW}。
- 当内部检测到 SCL 上升沿时，会在将 SCL 输出强制为低电平之前插入一段延时。该延时为 t_{SCLH}=（SCLH+1）x t_{PRESC}，其中 t_{PRESC}=（PRESC+1）xt_{I2CCLK}。t_{SCLH} 会影响 SCL 高电平时间 t_{HIGH}。

更多详细信息，请参见“18.2.8 主模式”中的“I2C 主模式初始化”。

说明：使能 I2C 后，不允许更改时序配置。

此外，还必须在使能从设备前，对 NOSTRETCH 进行配置。更多详细信息，请参见“18.2.7 从模式”中的“I2C 从模式初始化”。

说明：使能 I2C 后，不允许更改 NOSTRETCH 配置。

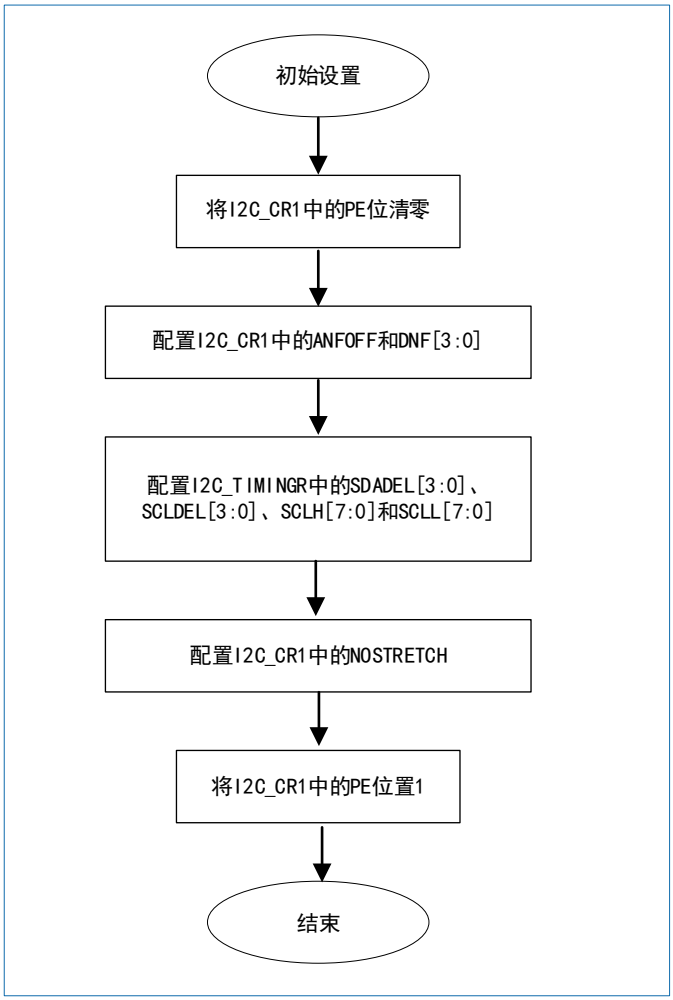


图 18-4 I2C 初始化流程图

18.2.5 软件复位

可通过将 I2C_CR1 寄存器中的 PE 位清零来执行软件复位。在这种情况下，I2C 线 SCL 和 SDA 被释

放。内部状态机复位，通信控制位和状态位重置为复位值。配置寄存器不受影响。

下面列出了受影响的寄存器位：

- I2C_CR2 寄存器：START、STOP 和 NACK
- I2C_ISR 寄存器：BUSY、TXE、TXIS、RXNE、ADDR、NACKF、TCR、TC、StopF、BERR、ARLO 和 OVR

支持 SMBus 功能时还会影响到以下寄存器位：

- I2C_CR2 寄存器：PECBYTE
- I2C_ISR 寄存器：PECERR、TIMEOUT 和 ALERT

由于寄存器存在跨时钟域设计，必须使 PE 保持低电平持续至少 3 个 APB 时钟周期，才能成功执行软件复位。写入以下软件序列可确保这一点：

1. 写入 PE=0；
2. 检查 PE 位，直到 PE=0；
3. 写入 PE=1。

18.2.6 数据传输

数据传输由发送和接收数据寄存器以及移位寄存器来管理。

接收

在 SDA 上接收的数据被填充到内部移位寄存器。在第 8 个 SCL 脉冲后，数据已全部接收到移位寄存器中。如果 I2C_RXDR 寄存器为空（RXNE=0），则移位寄存器的内容会复制到 I2C_RXDR 寄存器中。

如果 RXNE=1，意味着尚未读取上一次接收到的数据字节。

作为 I2C Master 时，硬件将延长第 8 和第 9 个 SCL 脉冲之间的低电平，直到软件读取 I2C_RXDR 为止。

作为 I2C Slave 时，如果 NOSTRETCH=0，将延长 SCL 线的低电平时间，直到读取了 I2C_RXDR 为止。如果 NOSTRETCH=1，则不会延长时钟，但会产生溢出错误。

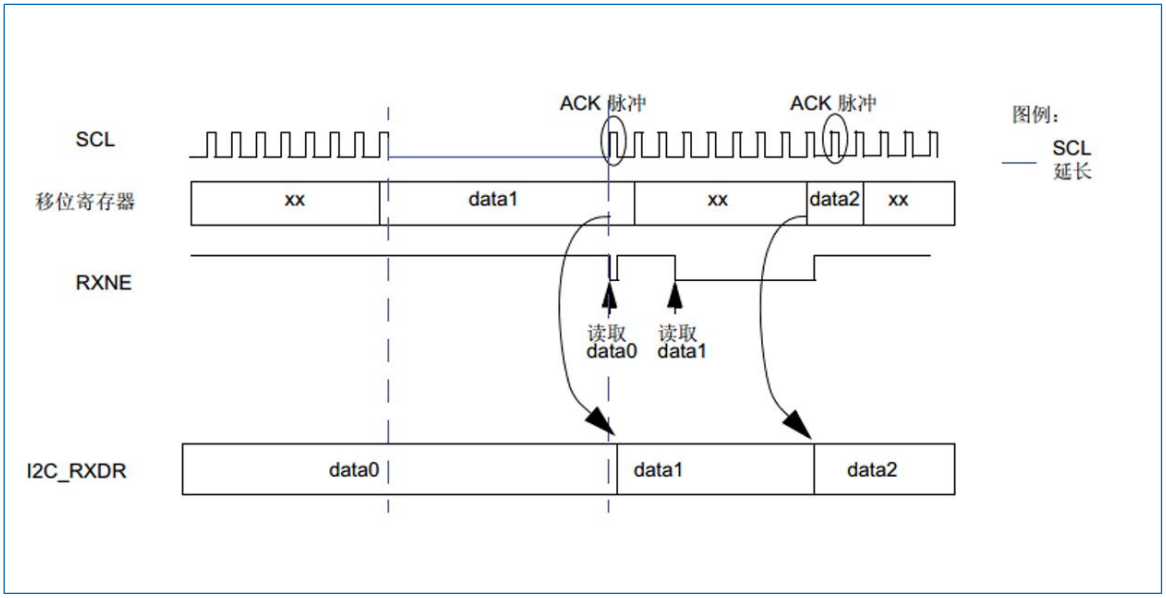


图 18-5 数据接收

发送

如果 I2C_TXDR 寄存器不为空（TXE=0），则其内容会在第 9 个 SCL 脉冲（ACK 脉冲）后复制到移位寄存器中。然后移位寄存器的内容会移出到 SDA 线上。

如果 TXE=1，意味着 I2C_TXDR 内尚未写入任何数据。

作为 I2C Master，硬件将延长第 9 个 SCL 脉冲后的低电平，直到写入了 I2C_TXDR 为止。

作为 I2C Slave，如果 NOSTRETCH=0，将延长时钟，直到软件写 I2C_TXDR；如果 NOSTRETCH=1，则不会延长时钟，但会产生溢出错误。

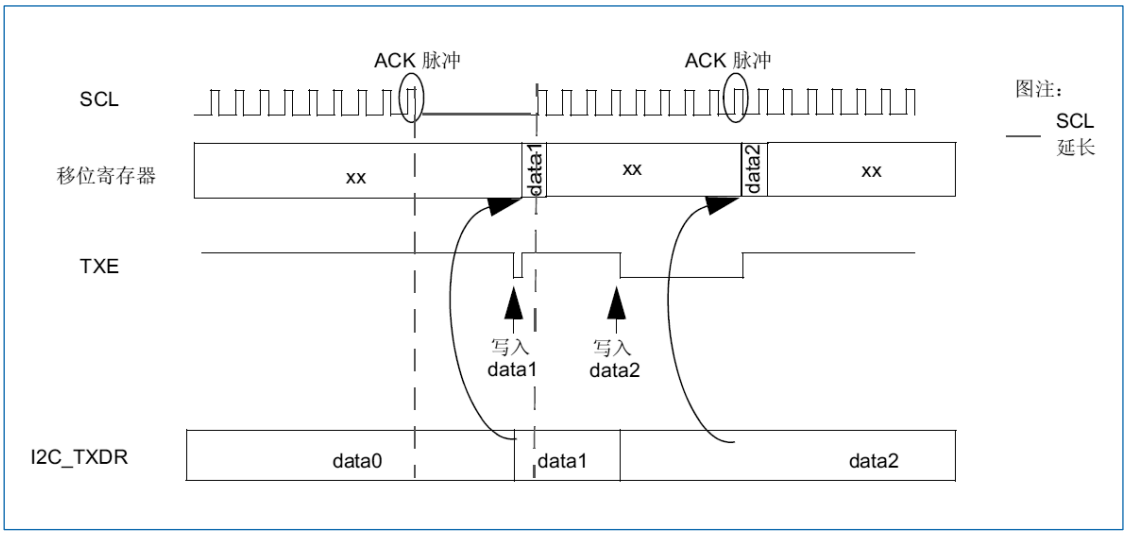


图 18-6 数据发送

硬件传输管理

I2C 在硬件中内置了字节计数器，以便在下列各种模式下管理字节传输和结束通信：

- 主模式下生成 NACK、Stop 和 ReSTART
- 从接收器模式下控制回复 ACK/NACK
- SMBus 模式下生成 PEC/对接收的数据进行 PEC 校验

字节计数器在主模式下默认开启。在从模式下，字节计数器默认为关闭状态，但可以通过软件来使能，方法是将 I2C_CR1 寄存器中的 SBC（从字节控制）位置 1。

待传输的字节数在 I2C_CR2 寄存器的 NBYTES[7:0]位域中进行编程。如果待传输的字节数（NBYTES）大于 255，或者接收方希望控制是否对接收到的数据字节进行应答，则必须选择重载模式，方法是将 I2C_CR2 寄存器的 RELOAD 位置 1。在该模式下，完成 NBYTES 中所编程字节数的数据传输之后，TCR 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TCR 标志置 1，SCL 便会延长。当往 NBYTES 写入一个非零值时，TCR 由软件清零。

在向 NBYTE 中设置最后一次传输的字节数前，必须把 RELOAD 位清零。

在主模式下，RELOAD=0 时，字节计数器功能如下：

- 自动结束模式（I2C_CR2 寄存器中的 AUTOEND="1"）。在该模式下，一旦完成 NBYTES[7:0]位域中所编程字节数的数据传输，主器件便会自动发送停止位。
- 软件结束模式（I2C_CR2 寄存器中的 AUTOEND="0"）在该模式下，一旦完成 NBYTES[7:0]位域中所编程字节数的数据传输，TC 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TC 标志置 1，SCL 信号便会延长，需要软件介入操作。当软件把 I2C_CR2 寄存器中的起始位或停止位置 1 时，TC 标志将被清零。当主器件要发送重复起始位时，必须使用该模式。

说明：当 RELOAD 位置 1 时，AUTOEND 位将不起作用。

表 18-3 I2C 配置表

功能	SBC 位	RELOAD 位	AUTOEND 位
主 Tx/Rx+NBYTES+Stop	x	0	1
主 Tx/Rx+NBYTES+RESTART	x	0	0
从 Tx/Rx 接收的所有字节都要回复应答	0	x	x
具有 ACK 控制的从 Rx	1	1	x

18.2.7 从模式

I2C 从模式初始化

在从模式下工作，用户必须至少使能一个从地址。可使用 I2C_OAR1 和 I2C_OAR2 这两个寄存器来编程自身从地址 OA1 和 OA2。

- OA1 既可配置为 7 位寻址模式（默认），也可通过将 I2C_OAR1 寄存器 OA1MODE 位置 1 配置为 10 位寻址模式。

  通过将 I2C_OAR1 寄存器中的 OA1EN 位置 1 来使能 OA1。

- 如果需要额外的从地址，可配置第 2 个从地址 OA2。将 I2C_OAR2 寄存器的 OA2MSK[2:0]位置 1 最多可屏蔽 7 个 OA2LSB。因此，当 OA2MSK 配置为 1 到 6 时，将分别只有 OA2[7:2]、OA2[7:3]、OA2[7:4]、OA2[7:5]、OA2[7:6]或 OA2[7]与接收到的地址作比较。只要 OA2MSK 不等于 0，OA2 的地址比较器便会排除 I2C 保留地址（0000XXX 和 1111XXX），这些地址将不会得到应答。如果 OA2MSK=7，接收到的所有 7 位地址（保留地址除外）均得到应答。OA2 始终为 7 位地址。

  如果这些保留地址在 I2C_OAR1 或 I2C_OAR2 寄存器中进行了编程并且 OA2MSK=0，则它们可以在通过特定使能位使能后得到应答。

  通过将 I2C_OAR2 寄存器中的 OA2EN 位置 1 来使能 OA2。

- 通过将 I2C_CR1 寄存器中的 GCEN 位置 1 来使能广播呼叫地址。

当通过 I2C 的其中一个使能地址来寻址到该 I2C 设备时，ADDR 中断状态标志将置 1，并且 ADDRIE 位置 1 时将生成中断。

默认情况下，从器件使用其时钟延长功能（即必要时延长 SCL 信号的低电平时间）来为软件操作的执行提供时机。如果主器件不支持时钟延长，则必须对 I2C 进行如下配置：将 I2C_CR1 寄存器中 NOSTRETCH 位置 1。

接收到 ADDR 中断后，如果使能多个地址，则用户必须读取 I2C_ISR 寄存器中的 ADDCODE[6:0]位，以确定是哪个地址匹配。还必须检查 DIR 标志，以获悉传输方向。

带时钟延长的从模式（NOSTRETCH=0）

在默认模式下，I2C 从器件会在以下情况下延长 SCL 时钟：

- ADDR 标志置 1 时：接收到的地址与其中一个使能的从地址匹配。通过软件将 ADDRCF 位置 1 以清零 ADRR 标志时，将释放该时钟延展。
- 发送时，前一次数据传输已完成但 I2C_TXDR 寄存器中未写入任何新数据，或者 ADDR 标志清零（TXE=1）时未写入第一个数据字节。往 I2C_TXDR 寄存器中写入数据时，将释放该时钟延展。
- 接收时，尚未读取 I2C_RXDR 寄存器但新的数据接收已完成。读取 I2C_RXDR 时，将释放该时钟延展。
- 当从器件字节控制模式和重载模式（SBC=1 且 RELOAD=1）下 TCR=1 时，这意味着最后一个数

据字节已完成传输。通过向 NBYTES[7:0]字段写入一个非零值以将 TCR 清零时，将释放该时钟延展。

- 在 SCL 下降沿检测之后，I2C 会延长 SCL 的低电平时间（不超过[（SDADEL+SCLDEL+1）x（PRESC+1）+1] x t_{I2CCLK}）。

不带时钟延长的从模式（NOSTRETCH=1）

当 I2C_CR1 寄存器中的 NOSTRETCH=1 时，I2C 从器件不会延长 SCL 信号。

- ADDR 标志置 1 时，不会延长 SCL 时钟。
- 发送时，必须在与发送数据对应的第一个 SCL 脉冲出现之前，向 I2C_TXDR 寄存器写入数据。否则，会发生下溢，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。当第一次数据发送开始而 STOPF 位仍置 1（尚未清零）时，OVR 标志也将置 1。因此，如果写入下一次传输要发送的第一个数据后才清零上一次传输的 STOPF 标志，则应提供 OVR 状态，甚至对于待发送的第一个数据也是如此。
- 接收时，必须在下一个数据字节的第 9 个 SCL 脉冲（ACK 脉冲）出现之前，从 I2C_RXDR 寄存器读取数据。否则，会发生上溢，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

从器件字节控制模式

要在从接收模式下实现字节 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从器件字节控制模式。这样符合 SMBus 标准。

要在从接收模式下实现字节 ACK 控制，必须选择重载模式（RELOAD=1）。要控制每个字节，必须在 ADDR 中断子程序中将 NBYTES 初始化为 0x1，并在每接收一个字节后将 NBYTE 重载为 0x1。接收到字节后，TCR 位将置 1，从而延长 SCL 信号的第 8 个和第 9 个脉冲之间的低电平时间。用户可以从 I2C_RXDR 寄存器中读取数据，然后通过配置 I2C_CR2 寄存器中的 ACK 位来决定是否应答。通过将 NBYTES 编程为非零值来释放 SCL 延长：发送应答或不应答信号，然后可继续接收下一个字节。

NBYTES 可加载大于 0x1 的值，在这种情况下，接收流在 NBYTES 个数据接收期间是连续的。

说明：

- *SBC 位只能在 I2C 被禁止时、从器件不被寻址时或 ADDR=1 时配置。*
- *ADDR=1 或 TCR=1 时，可以更改 RELOAD 位的值。*
- *从器件字节控制模式与 NOSTRETCH 模式不兼容。不允许在 NOSTRETCH=1 时将 SBC 位置 1。*

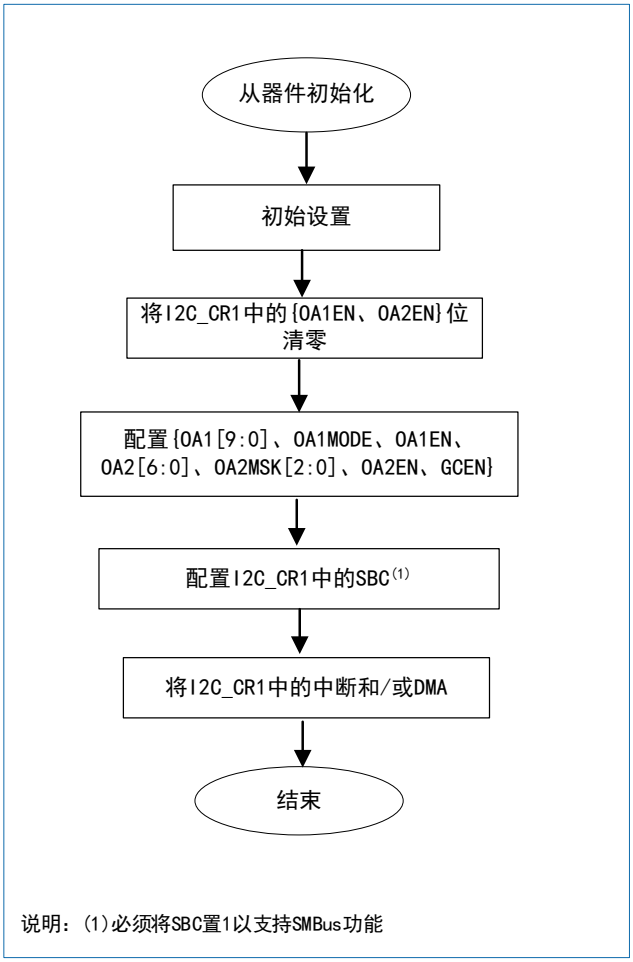


图 18-7 从器件初始化流程图

从发送器

当 I2C_TXDR 寄存器为空时，将生成发送中断状态（TXIS）。如果 I2C_CR1 寄存器中的 TXIE 位置 1，将生成中断。

I2C_TXDR 寄存器中写入待发送的下一个数据字节时，TXIS 位将被清零。

接收到 NACK 时，I2C_ISR 寄存器中的 NACKF 位将置 1，如果 I2C_CR1 寄存器中的 NACKIE 位置 1，还将生成中断。从器件自动释放 SCL 和 SDA 线，以使主器件执行停止或重复起始位的发送。收到 NACK 时，TXIS 位不会置 1。

当接收到停止位且 I2C_CR1 寄存器中的 STOPIE 位置 1 时，I2C_ISR 寄存器中的 STOPF 标志将置 1 并且会生成中断。在大多数应用中，SBC 位通常编程为“0”。在这种情况下，如果接收到从地址（ADDR=1）时 TXE=0，用户可以选择发送 I2C_TXDR 寄存器的内容作为第一个数据字节，也可以选择通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器以编程新的数据字节。

在从器件字节控制模式（SBC=1）下，必须在地址匹配中断子程序（ADDR=1）中向 NBYTE 写入待发送数据的个数。在这种情况下，传输期间 TXIS 事件的数量对应于 NBYTES 中编程的值。

说明：如果 NOSTRETCH=1，当 ADDR 标志置 1 时不会延长 SCL 时钟，因此用户无法在 ADDR 子程序中刷新 I2C_TXDR 寄存器的内容，从而编程第一个数据字节。必须在 I2C_TXDR 寄存器中预编程待发送的第一个数据字节：

- 该数据可以是前一个传输消息的最后一个 TXIS 事件中写入的数据。
- 如果该数据字节不是待发送的数据字节，可通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器，从而编程新的数据字节。必须仅在执行完这些操作后再清零 STOPF 位，以确保在地址应答之后，第一次数据传输开始之前执行这些操作。

如果第一次数据传输开始时 STOPF 仍置 1，则将生成下溢错误（OVR 标志置 1）。

如果需要 TXIS 事件（发送中断或发送 DMA 请求），用户必须将 TXE 位和 TXIS 位均置 1，以便生成 TXIS 事件。

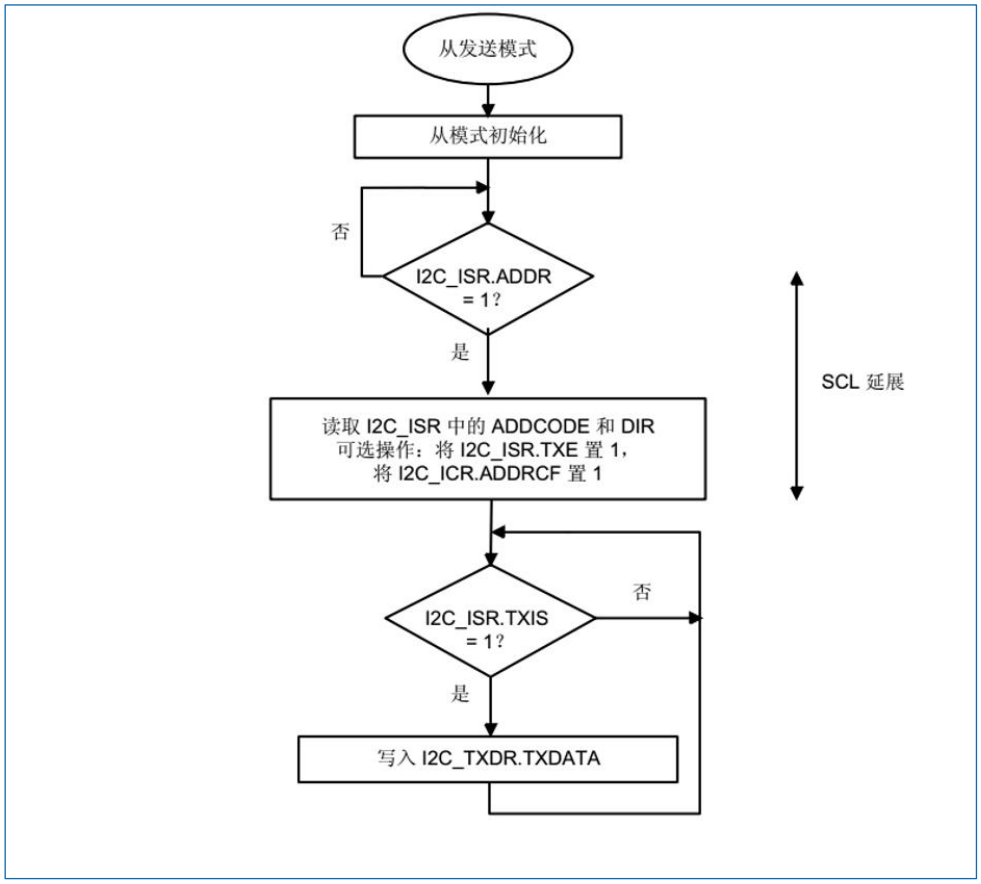


图 18-8 I2C 从发送器的传输序列流程图（NOSTRETCH=0）

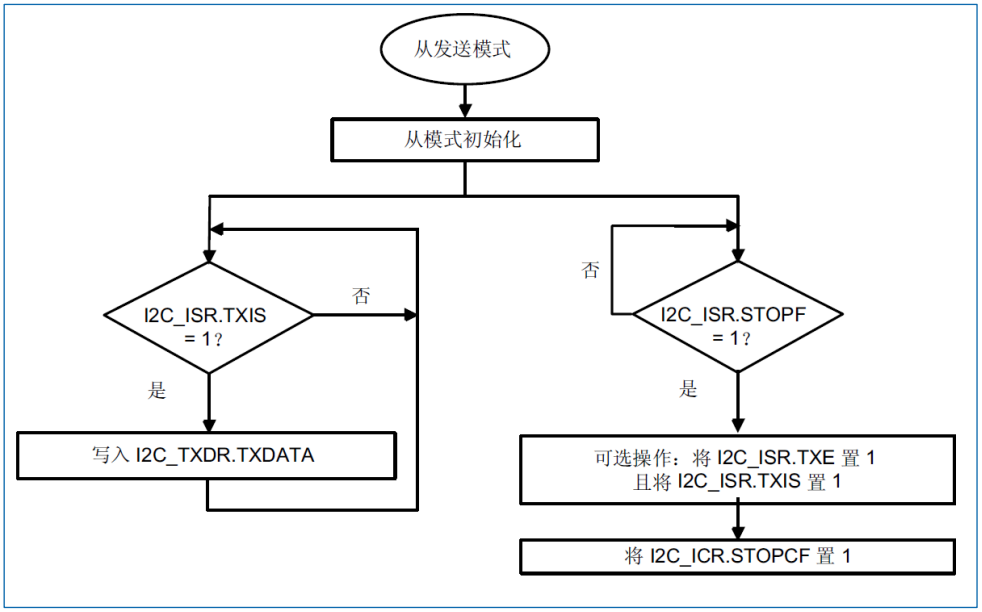


图 18-9 I2C 从发送器的传输序列流程图（NOSTRETCH=1）

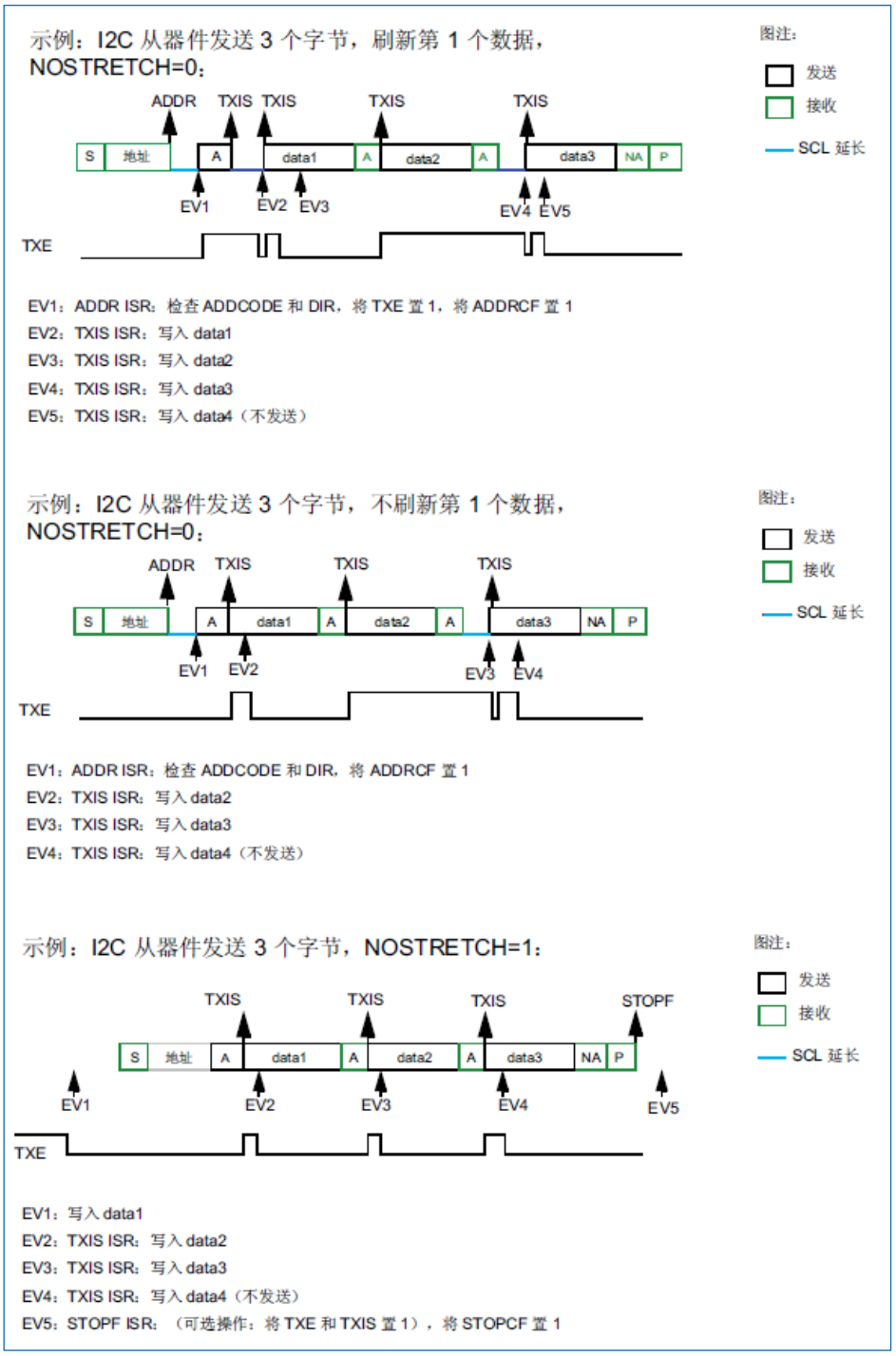


图 18-10 从发送器的传输总线图

从接收器

当 I2C_RXDR 满时，I2C_ISR 中的 RXNE 将置 1，如果 I2C_CR1 中的 RXIE 置 1，还将生成中断。读取 I2C_RXDR 时，将清零 RXNE。

接收到停止条件且 I2C_CR1 寄存器中的 STOPIE 置 1 时，I2C_ISR 中的 STOPF 将置 1 并且会生成中断。

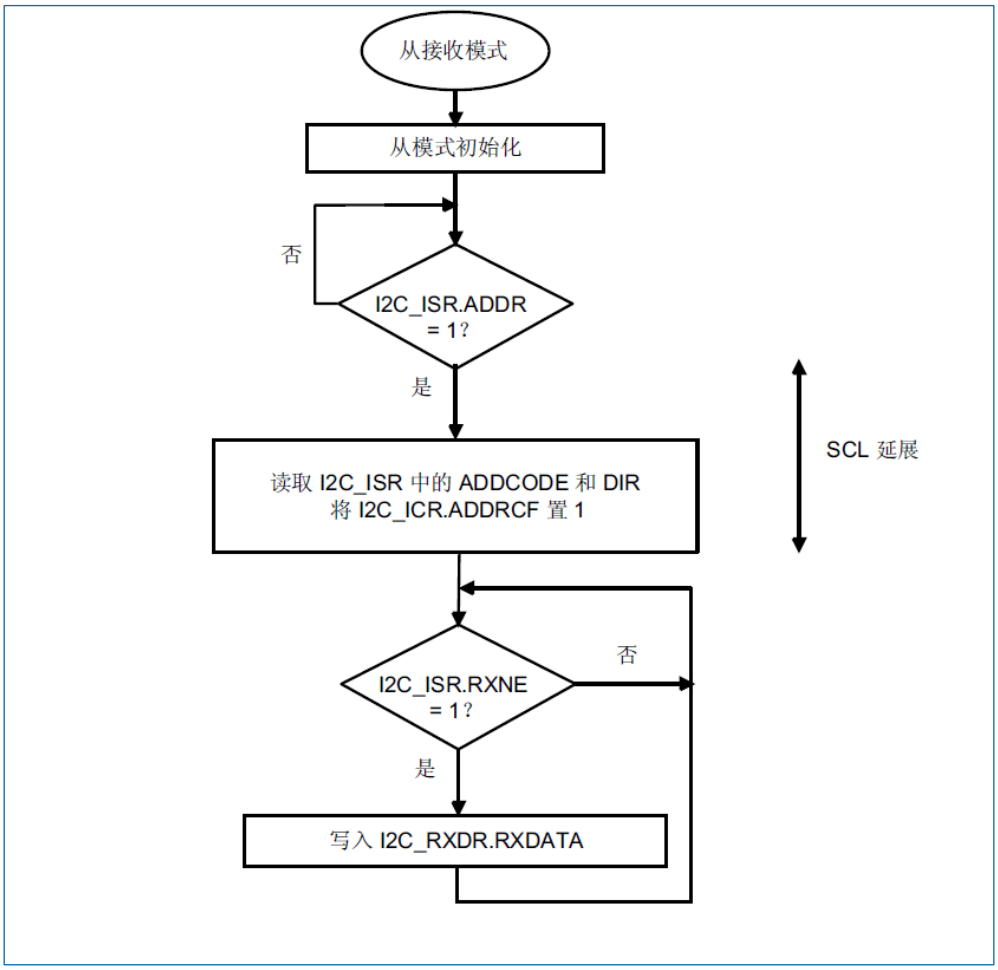


图 18-11 从接收器的传输序列流程图（NOSTRETCH=0）

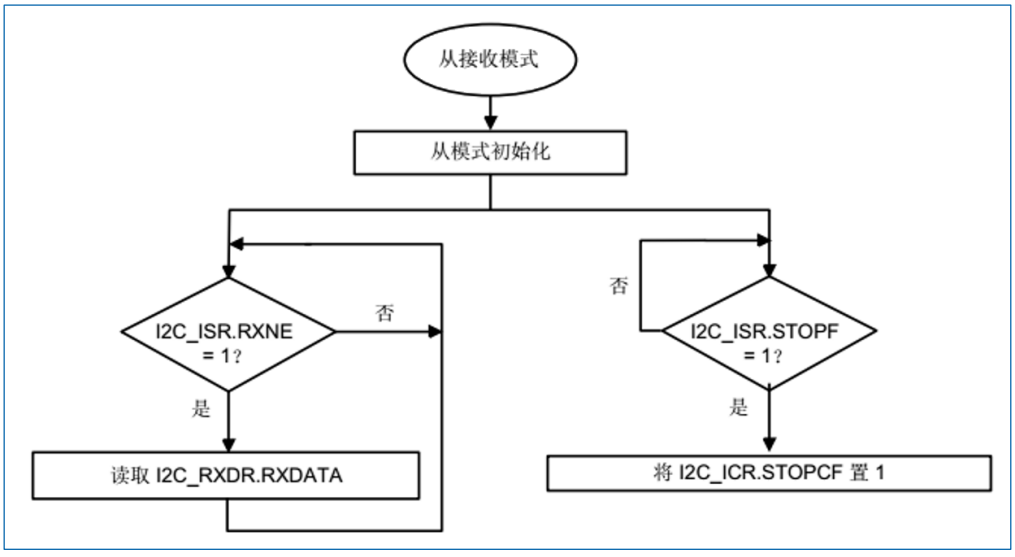


图 18-12 从接收器的传输序列流程图（NOSTRETCH=1）

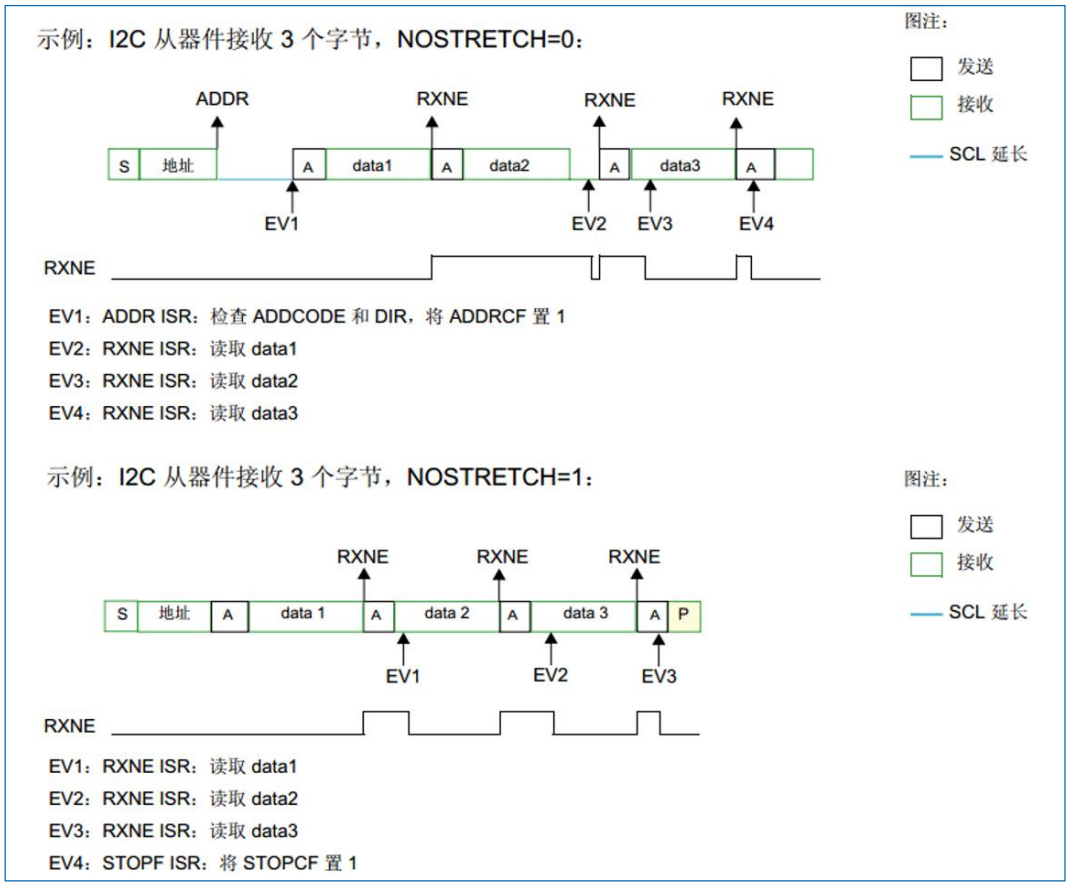


图 18-13 I2C 从接收器的传输总线图

18.2.8 主模式

I2C 主模式初始化

使能外设前，必须通过设置 I2C_TIMINGR 寄存器中的 SCLH 和 SCLL 位来配置 I2C 主时钟。

为了支持多主环境和从时钟延长，I2C 实现了时钟同步机制。

为了实现时钟同步，需执行以下操作：

- 使用 SCLL 计数器，从 SCL 低电平内部检测开始对时钟的低电平进行计数。
- 使用 SCLH 计数器，从 SCL 高电平内部检测开始对时钟的高电平进行计数。

I2C 经过 T_{SYNC1} 延时后，检测其自身的 SCL 低电平，该延时取决于 SCL 下降沿、SCL 输入噪声滤波器（模拟+数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLL 计数器达到 I2C_TIMINGR 寄存器的 SCLL[7:0]位中编程的值，I2C 便会将 SCL 释放为高电平。

I2C 经过 T_{SYNC2} 延时后检测其自身的 SCL 高电平，该延时取决于 SCL 上升沿、SCL 输入噪声滤波器（模拟+数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLH 计数器达到 I2C_TIMINGR 寄存器的 SCLH[7:0]位中编程的值，I2C 便会使 SCL 变为低电平。

因此，主时钟周期为：

$$t_{SCL}=T_{SYNC1}+T_{SYNC2}+\{[（SCLH+1）+（SCLL+1）]x（PRESC+1）x\ t_{I2CCLK}\}$$

t_{SYNC1} 的持续时间取决于以下参数：

- SCL 下降斜率
- 模拟滤波器（使能时）引入的输入延时
- 数字滤波器（使能时）引入的输入延时：DNF x t_{I2CCLK}

- SCL 与 I2CCLK 时钟建立同步而产生的延时（2 到 3 个 I2CCLK 周期）

tSYNC2 的持续时间取决于以下参数：

- SCL 上升斜率
- 模拟滤波器（使能时）引入的输入延时
- 数字滤波器（使能时）引入的输入延时：$DNFxt_{I2CCLK}$
- SCL 与 I2CCLK 时钟建立同步而产生的延时（2 到 3 个 I2CCLK 周期）

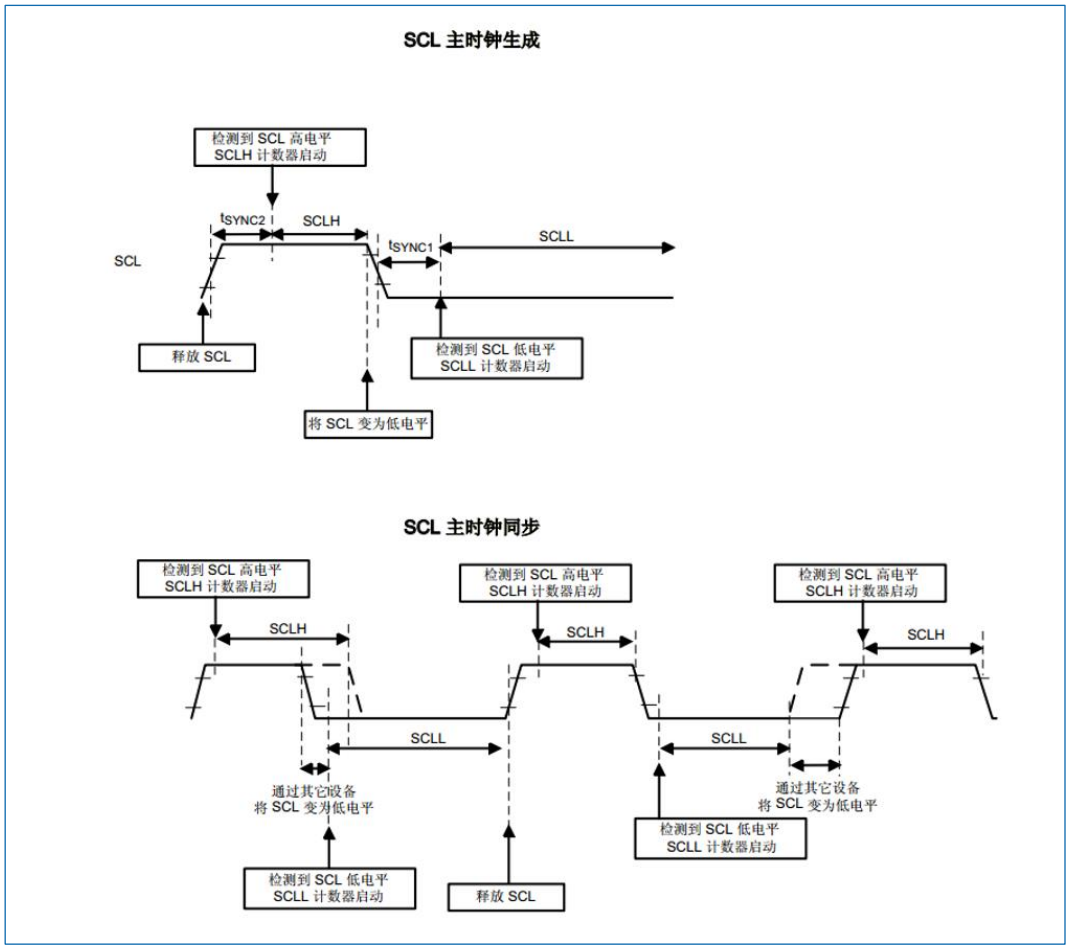


图 18-14 主时钟生成

说明：为了符合 I2C 或 SMBus 规范，主时钟必须遵循下表中给出的时序：

表 18-4 I2C-SMBus 规范时钟时序

符号	参数	标准模式（Sm）		快速模式（Fm）		超快速模式（Fm+）		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
f_{SCL}	SCL 时钟频率		100		400		1000		100	kHz
$t_{HD:STA}$	（重复）起始条件的保持时间	4.0	-	0.6		0.26		4.0	-	μs
$t_{SU:STA}$	重复起始条件的建立时间	4.7	-	0.6		0.26		4.7	-	μs

符号	参数	标准模式（Sm）		快速模式（Fm）		超快速模式（Fm+）		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
$t_{SU:STO}$	停止条件的建立时间	4.0	-	0.6	-	0.26	-	4.0	-	μs
t_{BUF}	停止条件和起始条件之间的总线空闲时间	4.7	-	1.3	-	0.5	-	4.7	-	μs
t_{LOW}	SCL时钟的低电平周期	4.7	-	1.3	-	0.5	-	4.7	-	μs
t_{HIGH}	SCL时钟的高电平周期	4.0	-	0.6	-	0.26	-	4.0	50	μs
t_r	SDA和SCL信号的上升时间	-	1000	-	300	-	120	-	1000	μs
t_f	SDA和SCL信号的下降时间	-	300	-	300	-	120	-	300	μs

说明：

(1). SCLL 还用于生成 t_{BUF} 和 $t_{SU;STA}$ 时序。

(2). SCLH 还用于生成 $t_{HD;STA}$ 和 $t_{SU;STO}$ 时序。

主模式通信初始化（地址阶段）

要发起通信，用户必须在 I2C_CR2 寄存器中为寻址的从器件编程以下参数：

- 寻址模式（7 位或 10 位）：ADD10
- 待发送的从地址：SADD[9:0]
- 传输方向：RD_WRN
- 读取 10 位地址时：HEAD10R 位。必须对 HEAD10R 进行相应配置，以指示传输方向变化时必须发送完整的地址序列，还是只发送地址头。
- 待传输的字节数：NBYTES[7:0]。如果字节数等于或大于 255，则初始化时必须将 NBYTES[7:0]填充为 0xFF。然后，用户必须将 I2C_CR2 寄存器中的 START 位置 1。START 位置 1 时，不允许更改上述所有位。

之后，当主器件检测到总线空闲（BUSY=0）时，它会在经过 T_{BUF} 的延时后自动发送起始位，随后发出从器件地址。

仲裁丢失时，主器件将自动切换回从模式，如果作为从器件被寻址，还可对其自身地址进行应答。

说明：无论接收到的应答值为何，只要已在总线上发送从地址，START 位便会由硬件复位。如果仲裁丢失，START 位也会由硬件复位。如果当 START 位置 1 时，I2C 作为从器件（ADDR=1）被寻址，则 I2C 将切换为从模式，START 位将在 ADDRCF 位置 1 时清零。

说明：该步骤同样适用于重复起始位。在这种情况下，BUSY=1。

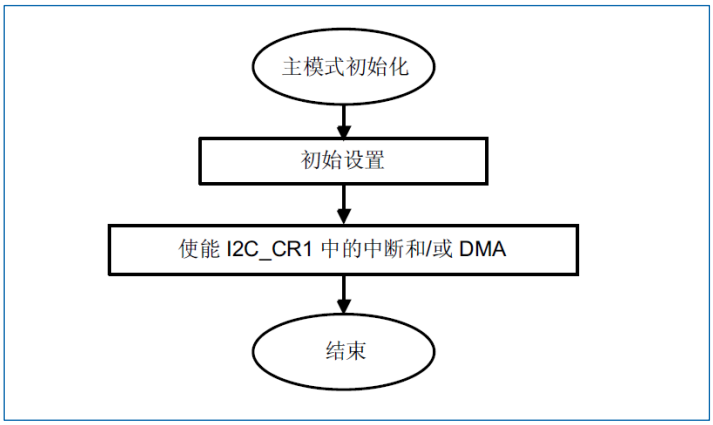


图 18-15 主模式初始化流程图

主接收器寻址 10 位地址从器件的初始化过程

- 如果从地址采用 10 位格式，用户可选择将 I2C_CR2 寄存器中的 HEAD10R 位清零来发送完整的读序列。在这种情况下，主器件会在 START 位置 1 后自动发送以下完整序列：（重复）起始位+带写方向的从器件 10 位地址头字节+从器件地址第 2 个字节+重复起始位+带读方向的从器件 10 位地址头字节

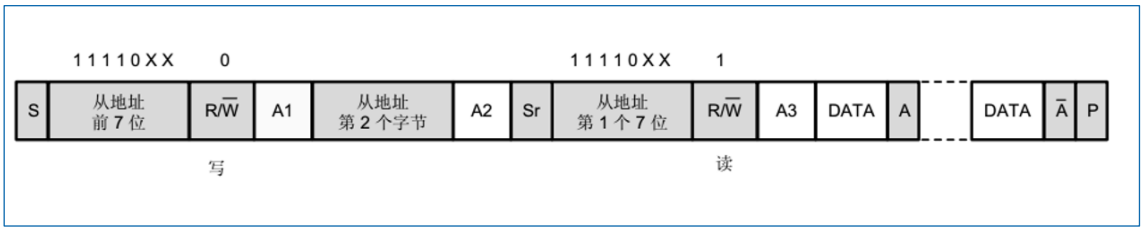


图 18-16 10 位地址读访问（HEAD10R=0）

- 如果主器件对 10 位地址从器件进行寻址、向该从器件发送数据、然后再从该从器件读取数据，则必须首先完成主器件发送过程。然后，重复起始位置 1，10 位从地址配置为 HEAD10R=1。在这种情况下，主器件发送以下序列：重复起始位+从地址 10 位头读取。

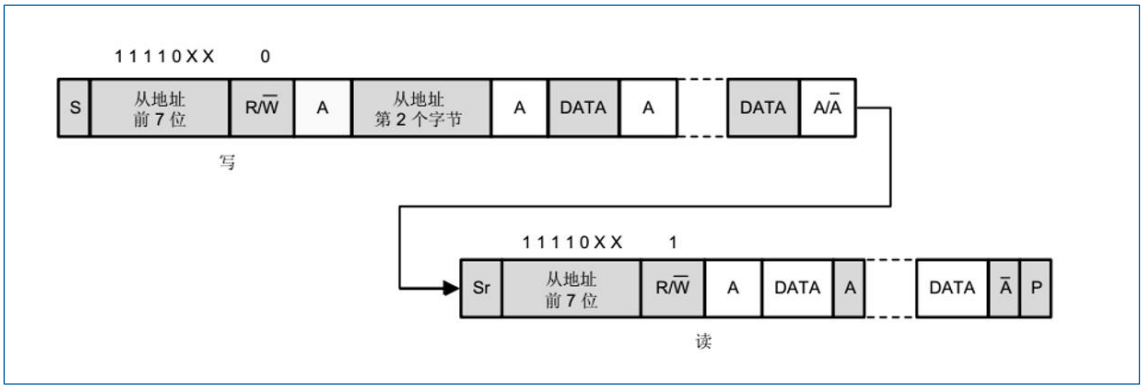


图 18-17 10 位地址读访问（HEAD10R=1）

主发送器

写传输时，在发送完每个字节（即第 9 个 SCL 脉冲（接收到 ACK 时））后，TXIS 标志将置 1。

如果 I2C_CR1 寄存器中的 TXIE 位置 1，TXIS 事件将生成中断。当 I2C_TXDR 寄存器中写入待发送的下一个数据字节时，该标志将被清零。

传输期间的 TXIS 事件的数量对应于 NBYTES[7:0]中编程的值。如果待发送的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES 数据传输完成时，TCR 标志将置 1，并且 SCL 线的低电平将被延展，直到 NBYTES[7:0]被写入非零值。

收到 NACK 时，TXIS 标志不会置 1。

- 当 RELOAD=0 且 NBYTES 数据传输完成时：
  - 在自动结束模式（AUTOEND=1）下，将自动发送停止位。
  - 在软件结束模式（AUTOEND=0）下，TC 标志将置 1 且 SCL 线的低电平将被延展，以便执行以下软件操作：

  可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从地址和待传输字节数来请求发送重复起始位。将 START 位置 1 会将 TC 标志清零，并在总线上发送起始位。

  可通过将 I2C_CR2 寄存器中的 Stop 位置 1 来请求停止位。将 Stop 位置 1 会将 TC 标志清零，并在总线上发送停止位。
- 如果接收到 NACK：TXIS 标志不会置 1，并且接收到 NACK 后会自动发送停止位。I2C_ISR 寄存器中的 NACKF 标志置 1，如果 NACKIE 位置 1，还将生成中断。

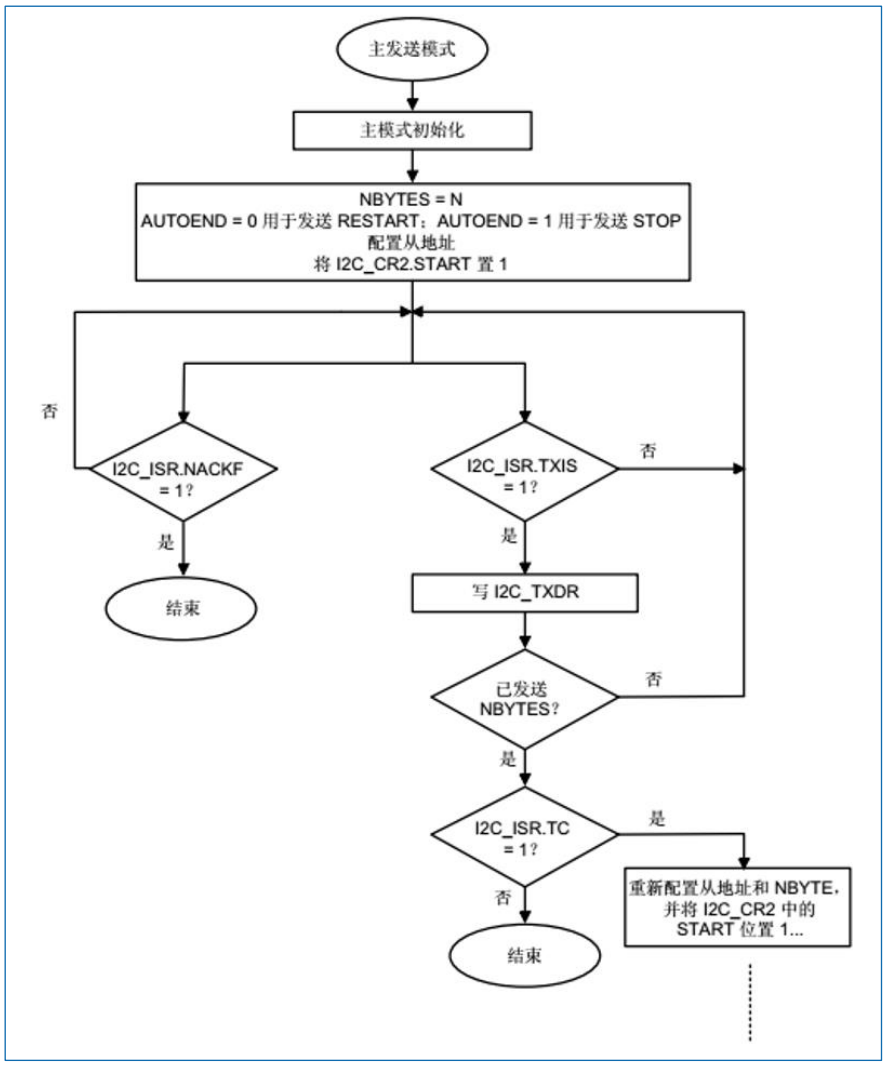


图 18-18 I2C 主发送器的传输序列流程图（N≤255 字节）

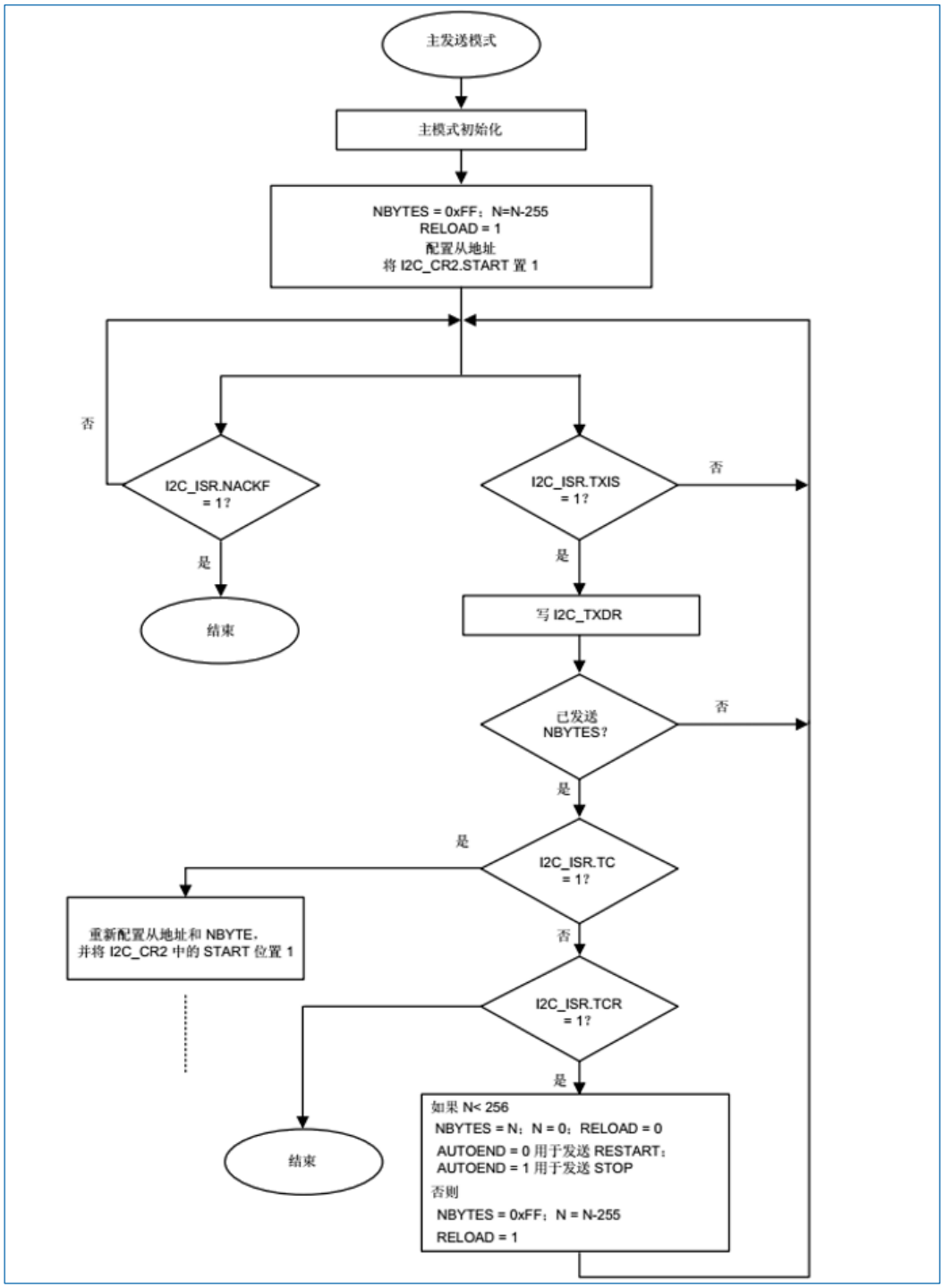


图 18-19 I2C 主发送器的传输序列流程图（N>255 字节）

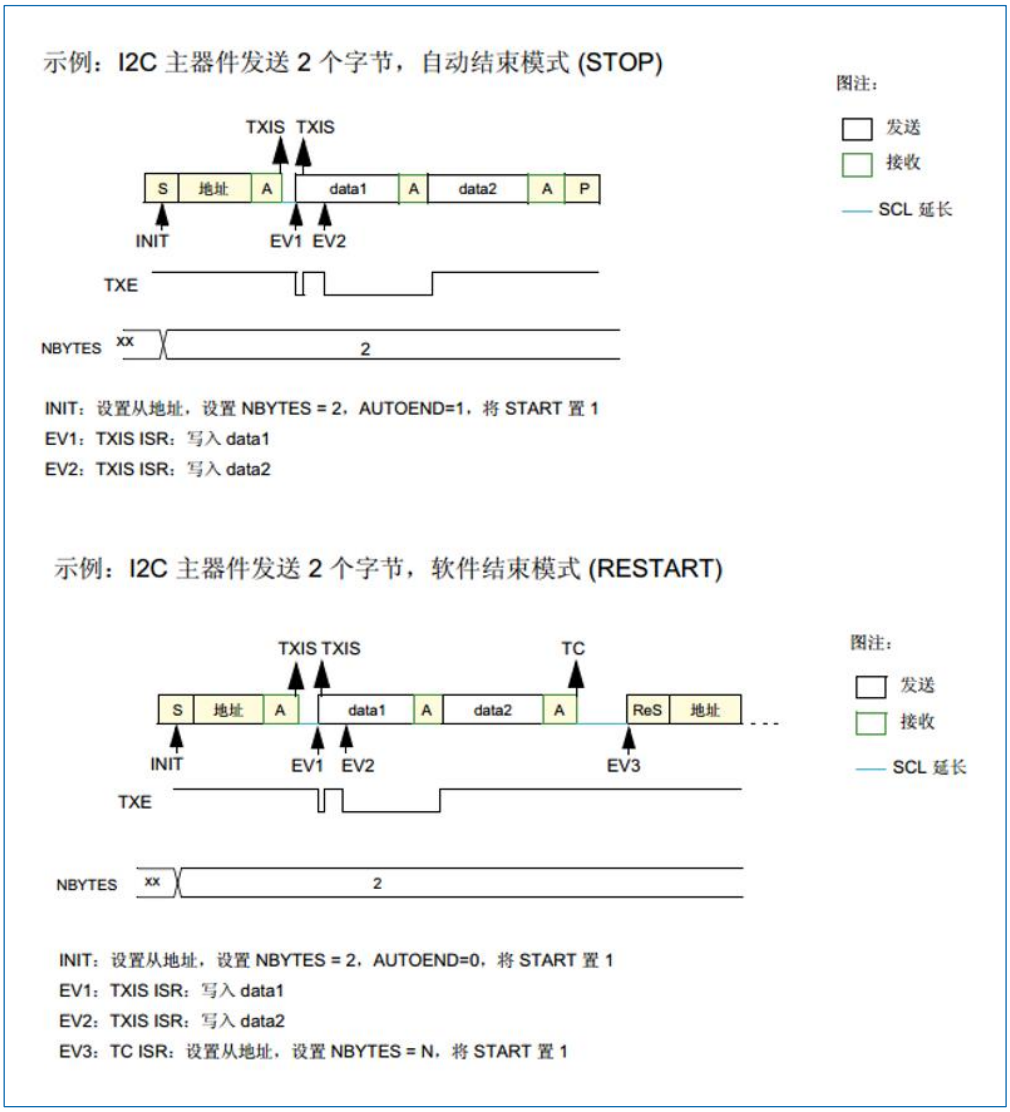


图 18-20 I2C 主发送器的传输总线图

主接收器

读传输时，在接收到每个字节（即第 8 个 SCL 脉冲）后，RXNE 标志将置 1。如果 I2C_CR1 寄存器中的 RXIE 位置 1，RXNE 事件将生成中断。读取 I2C_RXDR 时，将清零该标志。

如果待接收的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES[7:0]数据传输完成时，TCR 标志将置 1，并且 SCL 线的低电平将被延展，直到 NBYTES[7:0]被写入非零值。

- 当 RELOAD=0 且 NBYTES[7:0]数据传输完成时：
  - 在自动结束模式（AUTOEND=1）下，接收到最后一个字节后，将自动发送 NACK 和停止位。
  - 在软件结束模式（AUTOEND=0）下，接收到最后一个字节后，将自动发送 NACK，TC 标志将置 1 且 SCL 线的低电平将被延展，以便执行以下软件操作：

  可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从地址和待传输字节数来请求发送重复起始位。将 START 位置 1 会将 TC 标志清零，并在总线上发送起始位，后跟从地址。

  可通过将 I2C_CR2 寄存器中的 Stop 位置 1 来请求停止位。将 Stop 位置 1 会将 TC 标志清零，并在总线上发送停止位。

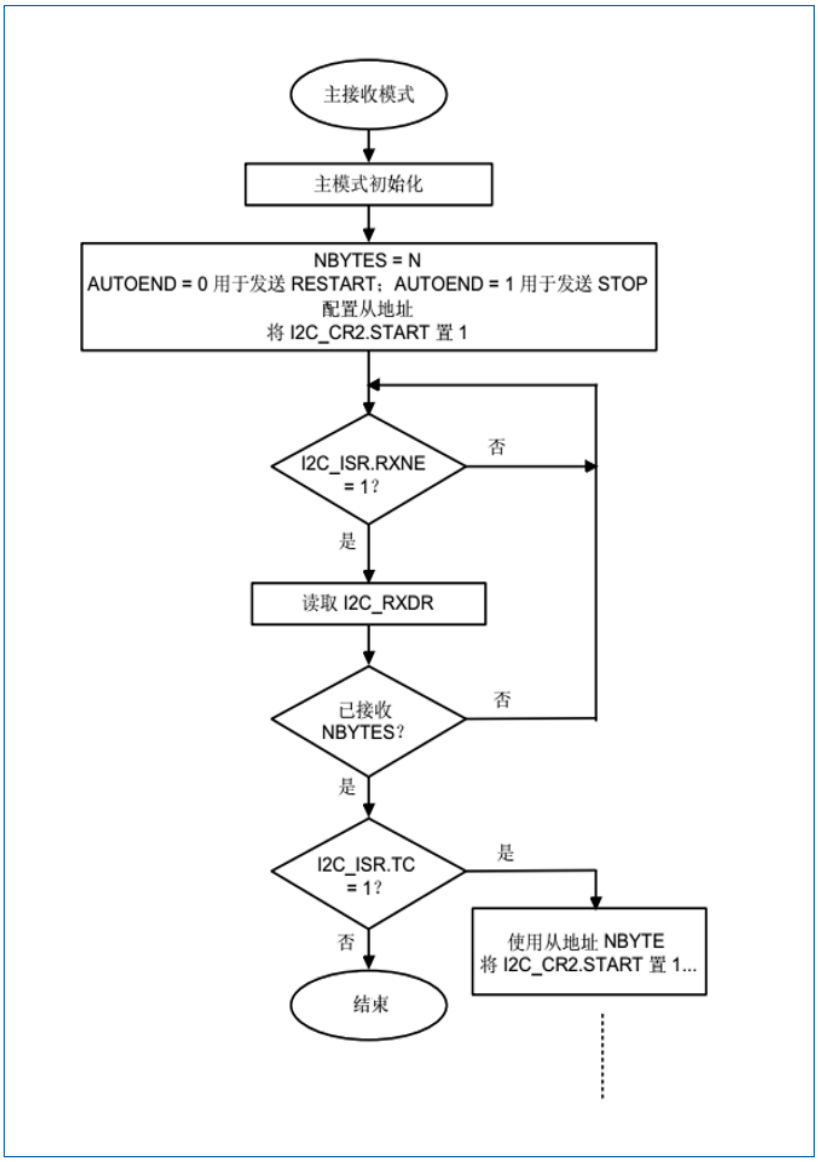


图 18-21 I2C 主接收器的传输序列流程图（N≤255 字节）

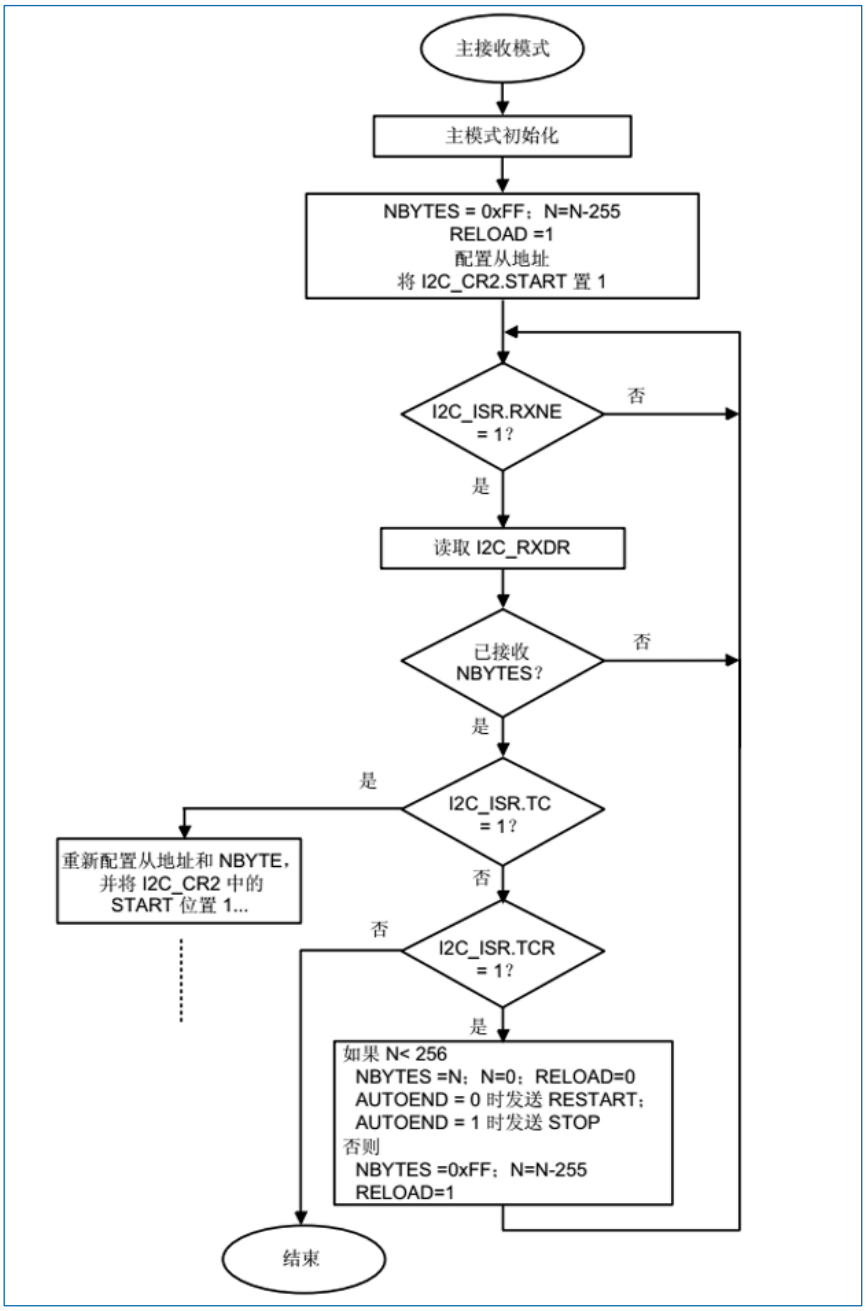


图 18-22 I2C 主接收器的传输序列流程图（N>255 字节）

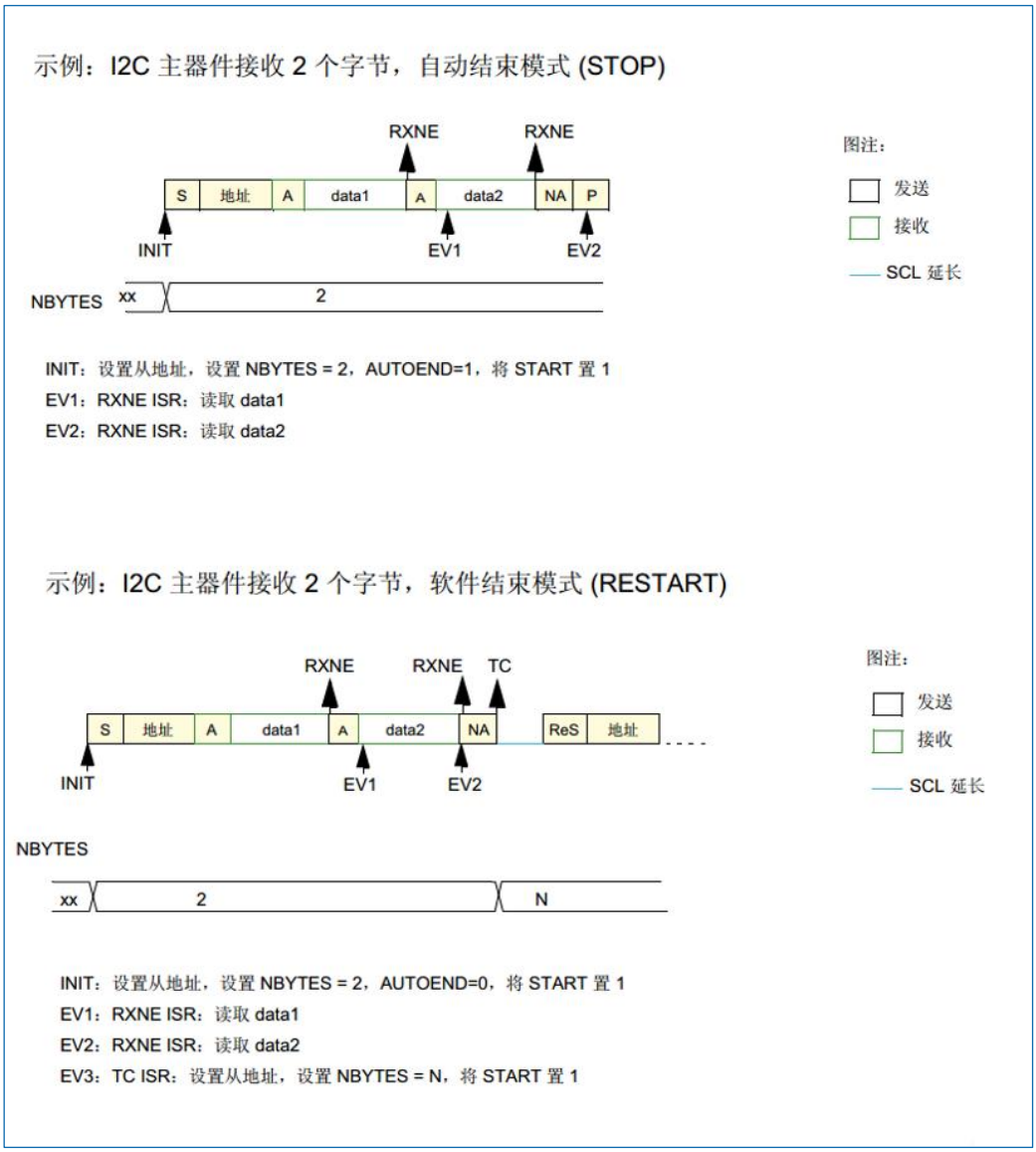


图 18-23 I2C 主接收器的传输总线图

18.2.9 I2C_TIMINGR 寄存器配置示例

下文各表提供了相应示例，以介绍如何编程 I2C_TIMINGR 才能获得符合 I2C 规范的时序。

表 18-5 f_{I2CCLK} =8MHz 时的时序设置示例

参数	标准模式(Sm)		快速模式(Fm)	超快速模式(Fm+)
	10 kHz	100 kHz	400 kHz	500 kHz
PRESC	1	1	0	0
SCLL	0xC7	0x13	0x9	0x6
t_{SCLL}	200x250ns=50µs	20x250ns=5.0µs	10x125ns=1250ns	7x125ns=875ns
SCLH	0xC3	0xF	0x3	0x3
t_{SCLH}	196x250ns=49µs	16x250ns=4.0µs	4x125ns=500ns	4x125ns=500ns
t_{SCL}(1)	约 100µs(2)	约 10µs(2)	约 2500ns(3)	约 2000ns(4)

参数	标准模式(Sm)		快速模式(Fm)	超快速模式(Fm+)
	10 kHz	100 kHz	400 kHz	500 kHz
SDADEL	0x2	0x2	0x1	0x0
t_{SDADEL}	2x250ns=500ns	2x250ns=500ns	1x125ns=125ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	5x250ns=1250ns	5x250ns=1250ns	4x125ns=500ns	2x125ns=250ns

(1). 由于 SCL 内部检测存在延时，SCL 周期 t_{SCL} 大于 t_{SCLL} + t_{SCLH}。为 t_{SCL} 提供的值仅用于举例说明。

(2). t_{SYNC1}+t_{SYNC2} 最小值为 4xTI2CCLK=250ns。t_{SYNC1}+t_{SYNC2}=1000 ns 时的示例。

(3). t_{SYNC1}+ t_{SYNC2} 最小值为 4xTI2CCLK=250ns。t_{SYNC1}+t_{SYNC2}=750 ns 时的示例。

(4). t_{SYNC1}+ t_{SYNC2} 最小值为 4xTI2CCLK=250ns。t_{SYNC1}+t_{SYNC2}=500 ns 时的示例。

表 18-6 f_{I2CCLK}=16MHz 时的时序设置示例

参数	标准模式（Sm）		快速模式（Fm）	超快速模式（Fm+）
	10kHz	100kHz	400kHz	1000kHz
PRESC	3	3	1	0
SCLL	0xC7	0x13	0x9	0x4
t_{SCLL}	200x250ns=50µs	20x250ns=5.0µs	10x125ns=1250ns	5x62.5ns=312.5ns
SCLH	0xC3	0xF	0x3	0x2
t_{SCLH}	196x250ns=49µs	16x250ns=4.0µs	4x125ns=500ns	3x62.5ns=187.5ns
t_{SCL}(1)	约 100µs(2)	约 10µs(2)	约 2500ns(3)	约 1000ns(4)
SDADEL	0x2	0x2	0x2	0x0
t_{SDADEL}	2x250ns=500ns	2x250ns=500ns	2x125ns=250ns	0ns
SCLDEL	0x4	0x4	0x3	0x2
t_{SCLDEL}	5x250ns=1250ns	5x250ns=1250ns	4x125ns=500ns	3x62.5ns=187.5ns

(1). 由于 SCL 内部检测存在延时，SCL 周期 TSCL 大于 t_{SCLL}+t_{SCLH}。为 TSCL 提供的值仅用于举例说明。

(2). t_{SYNC1}+t_{SYNC2} 最小值为 4xTI2CCLK=250ns。t_{SYNC1}+t_{SYNC2}=1000ns 时的示例。

(3). t_{SYNC1}+ t_{SYNC2} 最小值为 4xTI2CCLK=250ns。t_{SYNC1}+t_{SYNC2}=750ns 时的示例。

(4). t_{SYNC1}+ t_{SYNC2} 最小值为 4xTI2CCLK=250ns。t_{SYNC1}+t_{SYNC2}=500ns 时的示例。

18.2.10 SMBus I2C 特性

系统管理总线（SMBus）是一个以 I2C 协议为基础的双线制接口，实现了总线上多设备之间的通信，主要用于系统和电源管理。SMBus 还可以使用 SMBA 信号实现设备之间的通信请求。

该外设与 SMBus 规范第 2.0 版兼容，该版本以 I2C 规范第 2.1 版为基础。SMBus 包括三类器件。

- 从器件，用于接收或响应命令。

- 主器件，用于发出命令、生成时钟和管理传输。
- 主机，专用的主器件，可提供连接系统 CPU 的主接口。主机必须具有主-从器件功能，并且必须支持 SMBus 主机通知协议。SMBus 系统中只允许存在一个主机。

SMBus 可配置为主器件或从器件，也可配置为主机。

总线协议

SMBus 协议包含 11 种可用命令协议。SMBus 器件可以使用任意一种协议进行通信。这 11 种协议分别为快速命令、发送字节、接收字节、写入字节、写入字、读取字节、读取字、过程调用、块读取、块写入以及块写入-块读取过程调用。这些协议由用户通过软件实现。

地址解析协议（ARP）

动态为新器件分配唯一地址可解决 SMBus 从地址冲突的问题。为了提供一种机制来针对地址分配隔离各个器件，各器件必须具有唯一的器件标识符（UDID）。该 128 位数字由软件实现。

该外设支持地址解析协议（ARP）。通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 器件默认地址（0b110 0001）。ARP 命令应通过用户软件实现。

在从模式下，通过仲裁来实现对 ARP 的支持。

有关 SMBus 地址解析协议的详细信息，请参见 SMBus 规范第 2.0 版。

接收的命令和数据应答控制 SMBus 接收器必须能够对接收到的每个命令或数据进行否定应答。要在从模式下实现 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。

主机通知协议

该外设通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来支持主机通知协议。在这种情况下，主机将应答 SMBus 主机地址（0b0001000）。

使用主机通知协议时，连接到总线上的设备作为主器件，而主机作为从器件。

SMBus 报警

器件支持 SMBus ALERT 信号。只具备从功能的器件可通过 SMBALERT#引脚向主机发出信号，指示它想要通信。主机会处理该中断并通过报警响应地址（0b0001100）同时访问所有 SMBALERT#器件。只有那些将 SMBALERT#拉到低电平的器件会应答报警响应地址。

如果配置为从器件（SMBHEN=0），通过将 I2C_CR1 寄存器中的 LERTEN 位置 1 来将 SMBA 引脚拉为低电平。这同时还会使能报警响应地址。

如果配置为主机（SMBHEN=1），当 SMBA 引脚上检测到下降沿且 ALERTEN=1 时，I2C_ISR 寄存器中的 ALERT 标志置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，将生成中断。当 ALERTEN=0 时，即使外部 SMBA 引脚为低电平，也不会产生中断标志。

如果无需 SMBus ALERT 引脚，则当 ALERTEN=0 时，SMBA 引脚可用作标准 GPIO。

数据包错误校验

SMBus 规范中引入了数据包错误校验机制来实现可靠的数据传输。具体的实施方式是在每次数据传输结束时，附加数据包错误校验码（PEC）。PEC 的计算方式是对 START 到 STOP 之间的所有字节（包括地址和读/写位）使用 CRC-8 多项式 C（x）=x^8+x^2+x+1 进行计算。

内置的硬件 PEC 计算器：

在接收端：接收到的最后一个字节数据为 PEC 值，如果与硬件计算的 PEC 不匹配，将自动发送 NACK。

在发送端：发送的最后一个字节数据为 PEC 值。

超时

该外设内置了硬件定时器，以便符合 SMBus 规范第 2.0 版中定义的 3 个超时。

表 18-7 SMBus 超时规范

符号	参数	限值		单位
		最小值	最大值	
$t_{TIMEOUT}$	检测时钟低电平超时	25	35	ms
$t_{LOW:SEXT}$(1)	累积时钟低电平延长时间（从器件）	-	25	ms
$t_{LOW:MEXT}$(2)	累积时钟低电平延长时间（主器件）	-	10	ms

(1). $t_{LOW:SEXT}$ 是一段累积时间，即给定从器件在一条消息的最初起始到停止期间时钟信号可延展的时间。其它从器件或主器件也可能延长时钟，进而导致时钟低电平总延长时间超过 $T_{LOW:SEXT}$。因此，测量该参数时该器件应该是全速主器件寻址的唯一器件。

(2). $t_{LOW:MEXT}$ 是一段累积时间，即主器件在消息的每个字节（定义为 START 到 ACK、ACK 到 ACK 或 ACK 到 STOP）内时钟信号可延展的时间。从器件或其它主器件也可能延长时钟，进而导致时钟低电平总时间超过 $t_{LOW:MEXT}$（针对给定字节）。因此，测量该参数时该全速主器件只寻址一个从器件。

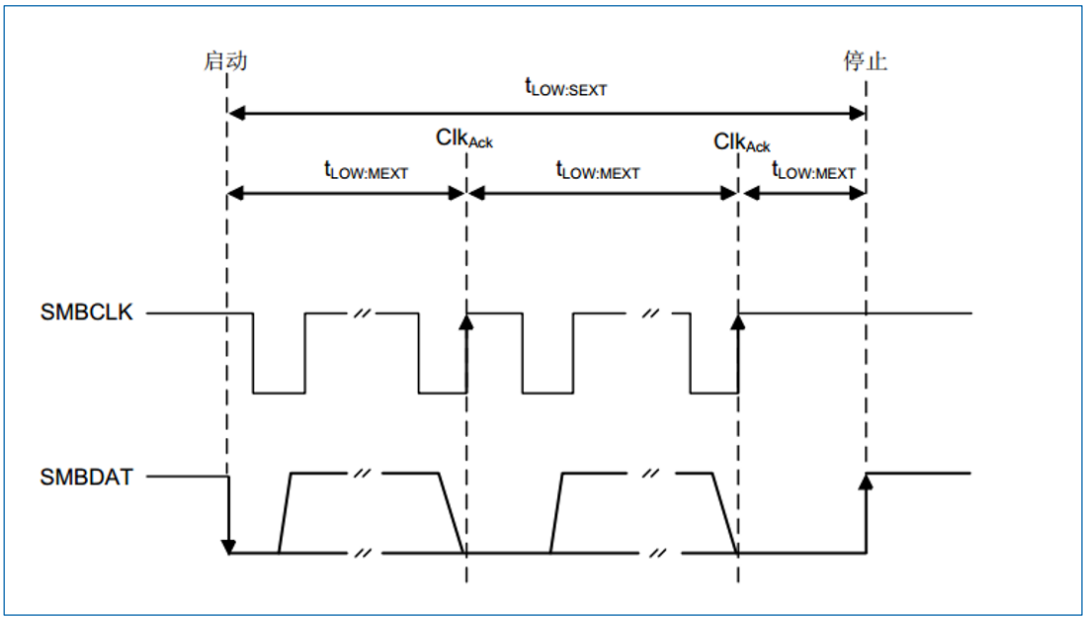


图 18-24 $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的超时间隔

总线空闲检测

如果主器件检测到时钟和数据信号的高电平时间已达 t_{IDLE}（超过 THIGH、MAX），则认为总线空闲。

该时序参数已考虑如下情况：主器件已动态添加至总线，但可能尚未检测到 SMBCLK 或 SMBDAT 线上的状态转换。在这种情况下，主器件必须等待足够长的时间，以确定当前未进行任何传输。外设支持硬件总线空闲检测。

18.2.11 SMBus 初始化

SMBus 的初始化包括 I2C 初始化之外，还必须进行一些其它的特定初始化，以便执行 SMBus 通信。

接收命令和数据应答控制（从模式）

SMBus 接收器必须能够对接收到的每个命令或数据进行否定应答。在从模式下，为了实现 ACK 控制，通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。

特定地址（从模式）

SMBus 作为从设备时，包含以下 3 个特殊地址。这 3 个地址的使能方法如下：

- 通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 器件默认地址（0b110 0001）。
- 通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来使能 SMBus 主机地址（0b000 1000）。
- 通过将 I2C_CR1 寄存器中的 ALERTEN 位置 1 来使能报警响应地址（0b000 1100）。

数据包错误校验

通过将 I2C_CR1 寄存器中的 PECEN 位置 1 来使能 PEC 的计算。然后，借助硬件字节计数器（I2C_CR2 寄存器中的 NBYTES[7:0]）来管理 PEC 传输。使能 I2C 之前，必须配置 PECEN 位。

PEC 传输由硬件字节计数器来管理，因此在从模式下连接 SMBus 时必须将 SBC 位置 1。当 PECBYTE 位置 1 且 RELOAD 位清零时，传输完 NBYTES-1 字节的数据后会传输 PEC。如果 RELOAD 置 1，PECBYTE 将不起作用。

说明：使能 I2C 时，不允许更改 PECEN 配置。

表 18-8 带 PEC 的 SMBus 配置

模式	SBC 位	RELOAD 位	AUTOEND 位	PECBYTE 位
主 Tx/RxNBYTES+PEC+Stop	x	0	1	1
主 Tx/RxNBYTES+PEC+ReSTART	x	0	0	1
从 Tx/Rx+PEC	1	0	x	1

超时检测

将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 和 TEXTEN 位置 1 来使能超时检测。定时器必须按如下方式编程：即在 SMBus 规范第 2.0 版规定的时间最大值之前检测出超时情况。

- $t_{TIMEOUT}$ 检查

  要使能 $t_{TIMEOUT}$ 检查，必须将 12 位 TIMEOUTA[11:0]位编程为定时器重载值，以检查 $t_{TIMEOUT}$ 参数。必须将 TIDLE 位配置为“0”，以检测 SCL 低电平超时。

  然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。

  如果 SCL 的低电平持续时间超过（TIMEOUTA+1）x 2048 x t_{I2CCLK}，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。

说明：TIMEOUTEN 位置 1 时，不允许更改 TIMEOUTA[11:0]位和 TIDLE 位的配置。

- $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 检查

  必须根据外设配置为主器件还是从器件来配置 TIMEOUTB 定时器，以便为从器件校验 $t_{LOW:SEXT}$，为主器件校验 $t_{LOW:MEXT}$。由于标准只规定了最大值，用户可以为这两个参数选择相同的值。

  然后，通过将 I2C_TIMEOUTR 寄存器中的 TEXTEN 位置 1 来使能定时器。

如果 SMBus 外设延展 SCL 的累积时间超过（TIMEOUTB+1）x 2048 x t_{I2CCLK}，并且达到“18.2.10 SMBus I2C 特性”中的“总线空闲检测”一节给出的超时间隔，则 I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。请参见表 18-8。

说明：TEXTEN 位置 1 时，不允许更改 TIMEOUTB 配置。

总线空闲检测

要使能 t_{IDLE} 检查，必须将 12 位 TIMEOUTA[11:0]字段编程为定时器重载值，以获取 t_{IDLE} 参数。必须将 TIDLE 位配置为“1”，以检测 SCL 和 SDA 高电平超时。然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。

如果 SCL 和 SDA 线的高电平持续时间超过（TIMEOUTA+1）x 4 x t_{I2CCLK}，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。

请参见表 18-9。

说明：TIMEOUTEN 置 1 时，不允许更改 TIMEOUTA 和 TIDLE 配置。

18.2.12 SMBus：I2C_TIMEOUTR 寄存器配置示例

- 将 $t_{TIMEOUT}$ 的最大持续时间配置为 25 ms

表 18-9 不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 $t_{TIMEOUT}$=25 ms）

f_{I2CCLK}	TIMEOUTA[11:0] 位	TIDLE 位	TIMEOUTEN 位	$t_{TIMEOUT}$
8 MHz	0x61	0	1	98x2048x125ns=25 ms
16 MHz	0xC3	0	1	196x2048x62.5ns=25 ms
32 MHz	0x186	0	1	391x2048x31.25ns=25 ms

- 将 $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的最大持续时间配置为 8 ms

表 18-10 不同 I2CCLK 频率下的 TIMEOUTB 设置示例

f_{I2CCLK}	TIMEOUTB[11:0] 位	TEXTEN 位	$t_{LOW:EXT}$
8 MHz	0x1F	1	32 x 2048 x 125ns=8 ms
16 MHz	0x3F	1	64 x 2048 x 62.5ns=8 ms
32 MHz	0x7C	1	125 x 2048 x 31.25ns=8 ms

- 将 T_{IDLE} 的最大持续时间配置为 50 µs

表 18-11 不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 t_{IDLE}=50 µs）

f_{I2CCLK}	TIMEOUTA[11:0]位	TIDLE 位	TIMEOUTEN 位	t_{TIDLE}
8 MHz	0x63	1	1	100 x 4 x 125 ns=50 µs
16 MHz	0xC7	1	1	200 x 4 x 62.5 ns=50 µs
32 MHz	0x18F	1	1	400 x 4 x 31.25 ns=50 µs

18.2.13 SMBus 模式

除了 I2C 模式传输管理之外，还提供了一些额外的软件流程图来支持 SMBus。

SMBus 从发送器

在 SMBus 模式下，必须将 SBC 编程为“1”，以便在完成已编程数据字节数的传输后进行 PEC 传输。当 PECBYTE 位置 1 时，NBYTES[7:0]中编程的字节数包含 PEC 传输。在这种情况下，总 TXIS 中断数为 NBYTES-1，如果主器件在完成 NBYTES-1 字节的数据传输后请求传输额外的字节，则将自动发送 I2C_PECR 寄存器的内容。

说明：当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

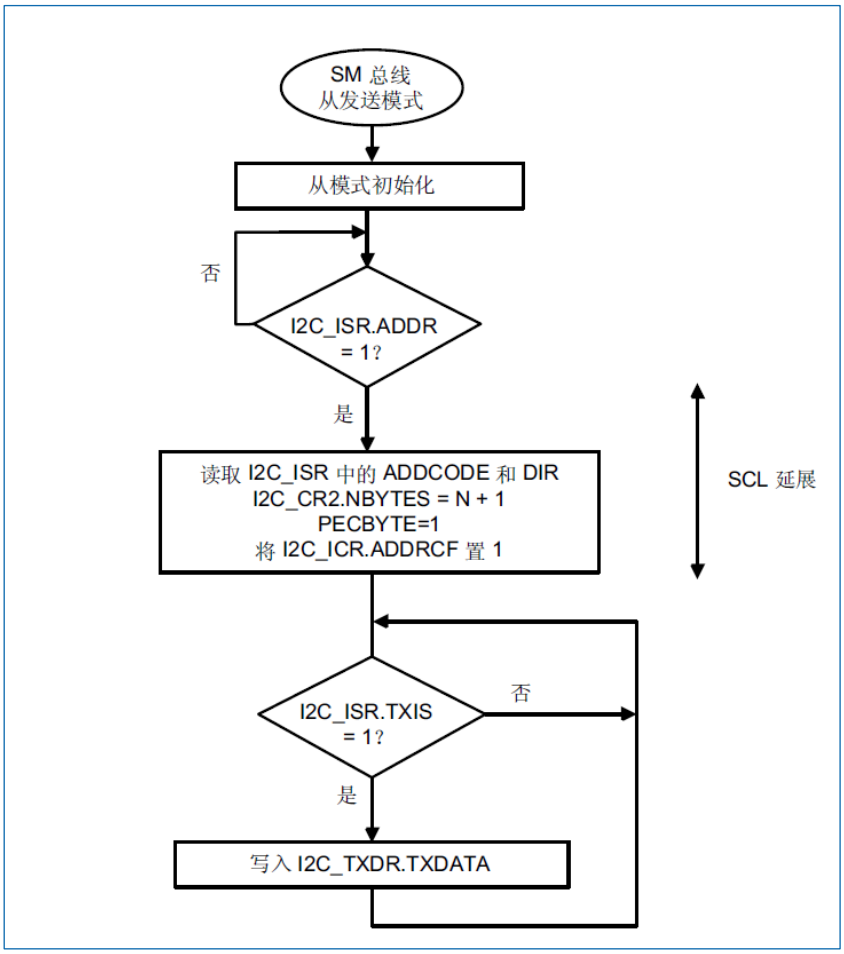


图 18-25 SMBus 从发送器的传输序列流程图（N 字节+PEC）

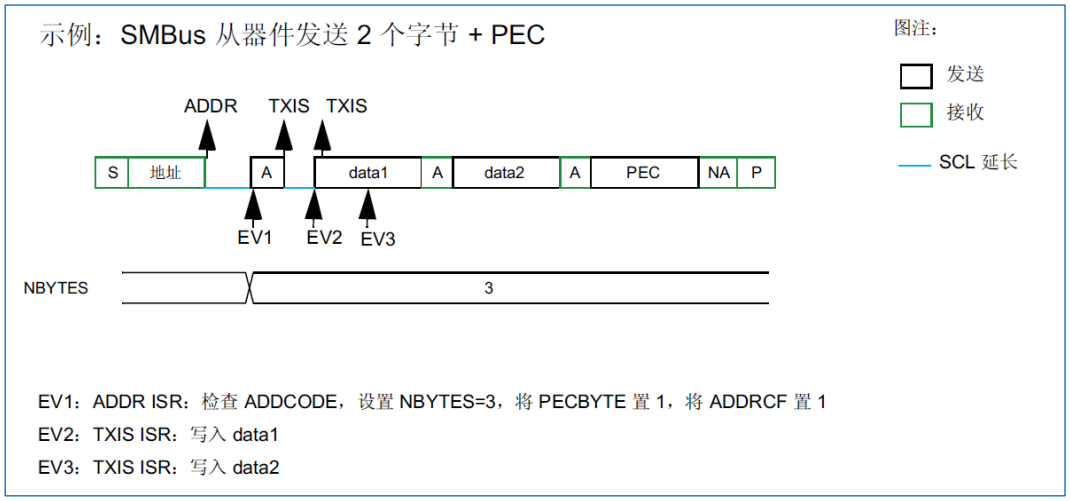


图 18-26 SMBus 从发送器的传输总线图（SBC=1）

SMBus 从接收器

在 SMBus 模式下使用 I2C 时，必须将 SBC 编程为“1”，以便在完成已编程数据字节数的传输后进行 PEC 校验。要对每个字节进行 ACK 控制，必须选择重载模式（RELOAD=1）。更多详细信息，请参见“18.2.7 从模式”中的“从器件字节控制模式”。

要校验 PEC 字节，必须将 RELOAD 位清零并将 PECBYTE 位置 1。在这种情况下，当接收到 NBYTES-1 字节的数据后，接收的下一个字节将与内部 I2C_PECR 寄存器的内容作比较。如果比较不匹配，则将自动生成 NACK 信号；如果比较匹配，则将自动生成 ACK 信号，而与 ACK 位的值无关。PEC 字节一经接收，便会像任何其它数据一样复制到 I2C_RXDR 寄存器中，并且 RXNE 标志将置 1。

当 PEC 不匹配时，PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

如果无需 ACK 软件控制，用户可编程 PECBYTE=1，在同一写操作下，将 NBYTES 编程为连续接收的字节数。接收到 NBYTES-1 字节的数据后，会将接收的下一个字节视为 PEC 进行校验。

说明：当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

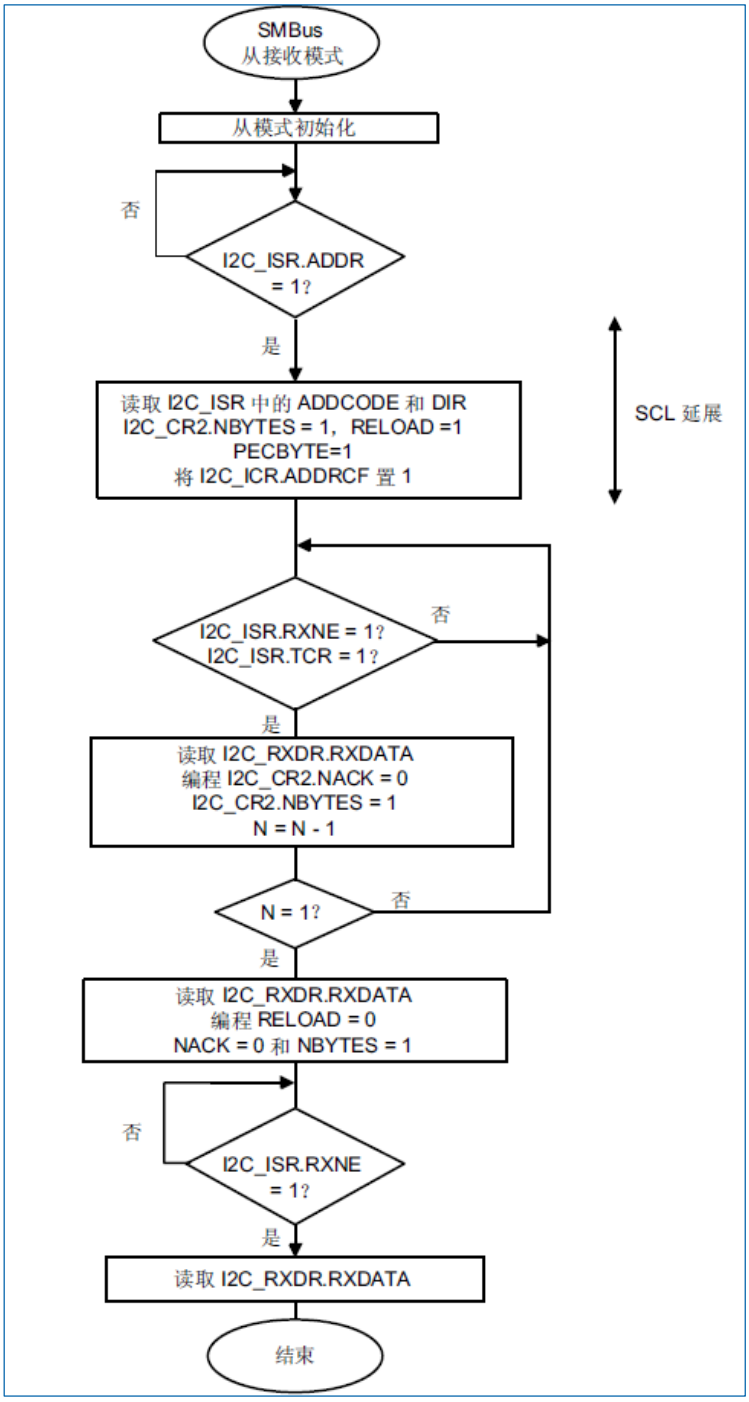


图 18-27 SMBus 从接收器的传输序列流程图（N 字节+PEC）

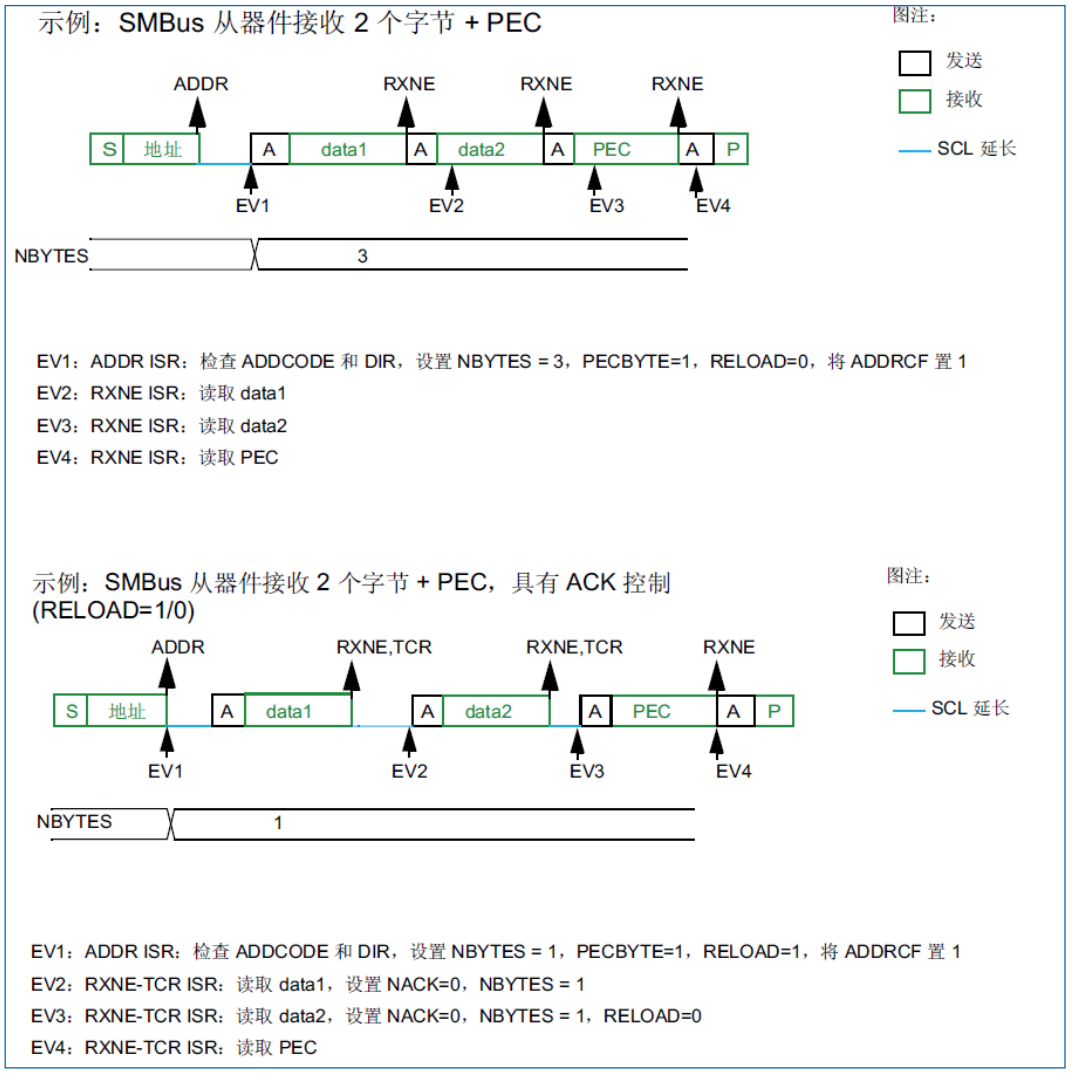


图 18-28 从接收器的总线传输图（SBC=1）

SMBus 主发送器

当 SMBus 主器件想要发送 PEC 时，必须在 START 位置 1 前，将 PECBYTE 位置 1 并在 NBYTES[7:0]字段中设置字节数。在这种情况下，总 TXIS 中断数为 NBYTES-1。因此，如果 PECBYTE 位在 NBYTES=0x1 时置 1，则将自动发送 I2C_PECR 寄存器的内容。

如果 SMBus 主器件想要在 PEC 后发送停止位，则应选择自动结束模式（AUTOEND=1）。在这种情况下，传输 PEC 后将自动发送停止位。

如果 SMBus 主器件想要在 PEC 后发送重复起始位，则必须选择软件模式（AUTOEND=0）。

在这种情况下，发送 NBYTES-1 字节的数据后，将发送 I2C_PECR 寄存器的内容，TC 标志将在传输完 PEC 之后置 1，SCL 线的低电平时间将延长。必须在 TC 中断子程序中设置重复起始位。

说明：当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

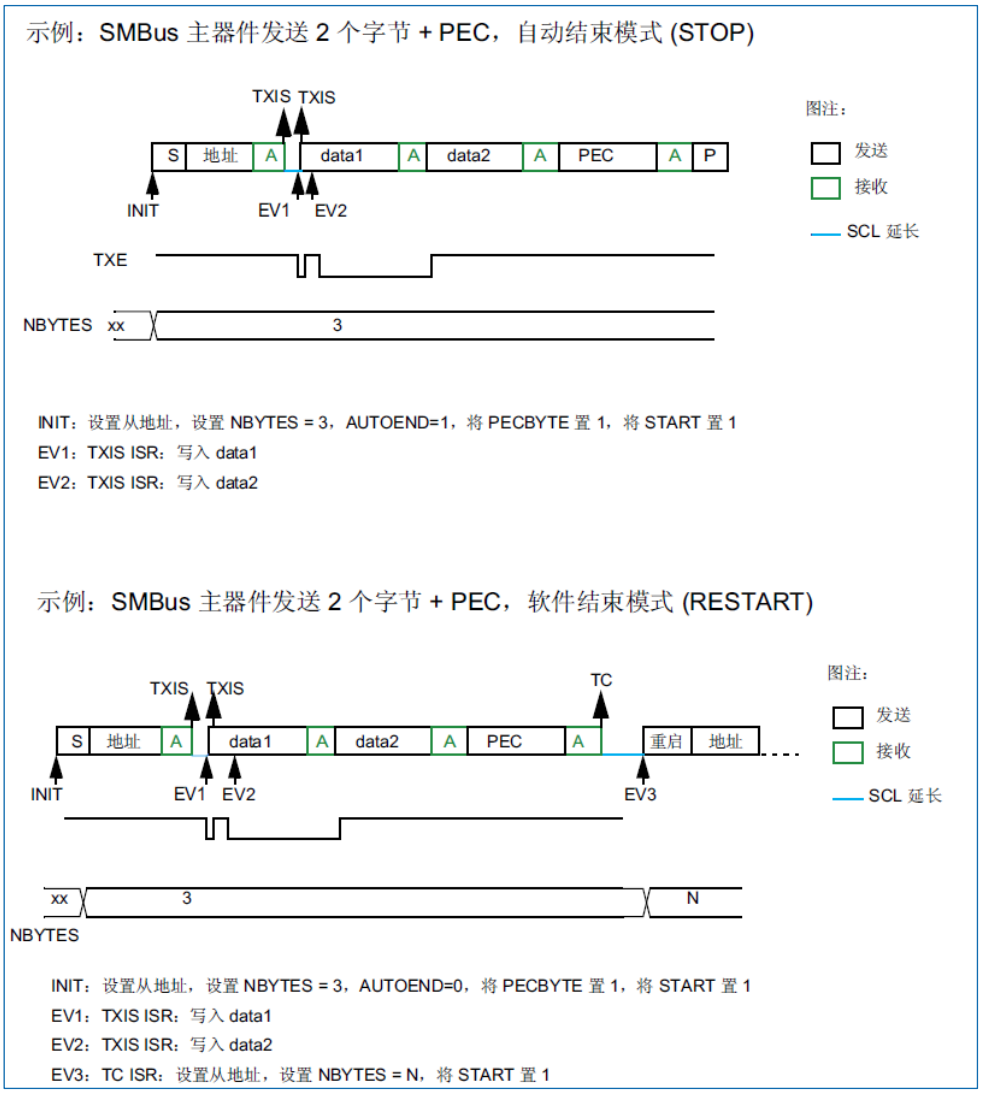


图 18-29 SMBus 主发送器的总线传输图

SMBus 主接收器

当 SMBus 主器件想要接收 PEC，并在传输结束后接收 Stop 时，可选择自动结束模式（AUTOEND=1）。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从地址。在这种情况下，当接收到 NBYTES-1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。PEC 字节（其后跟有停止位）将得到 NACK 响应。

当 SMBus 主接收器想要接收 PEC 字节，并且在传输结束后接收重复起始位时，必须选择软件模式（AUTOEND=0）。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从地址。

在这种情况下，当接收到 NBYTES-1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。接收到 PEC 字节后，TC 标志将置 1，SCL 线的低电平时间将延长。可以在 TC 中断子程序中设置重复起始位。

说明：当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

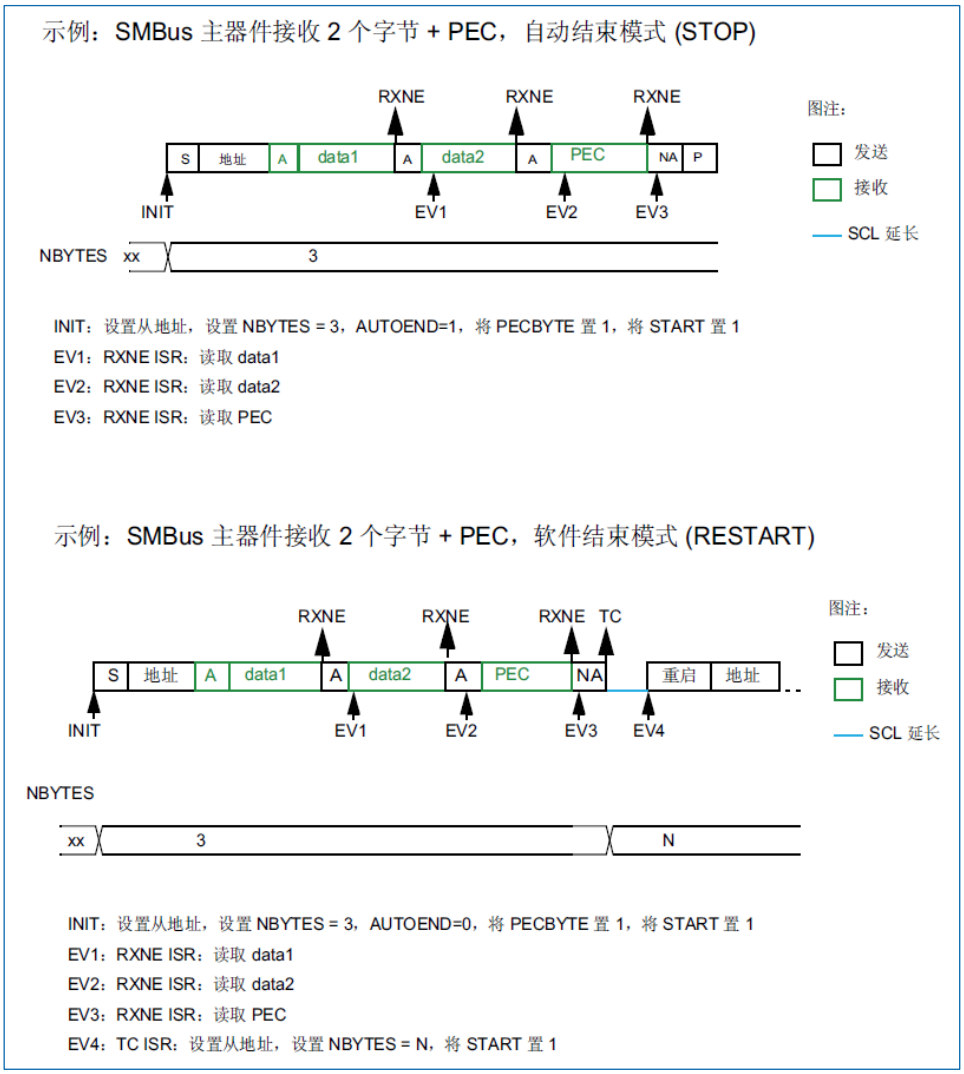


图 18-30 SMBus 主接收器的总线传输图

18.2.14 地址匹配时从停机模式唤醒

作为从设备被正确寻址时，I2C 能够将 MCU 从 Stop 模式唤醒（APB 时钟关断）。从 Stop 模式唤醒支持所有寻址模式。

要实现从 Stop 模式唤醒 MCU，需要注意以下几点：

- 如果 I2C 时钟是系统时钟，或者 WUPEN=0，则接收到 START 后，HSI 振荡器不会接通。
- 数字滤波器与从停机模式唤醒功能不兼容。如果 DNF 位不等于 0，则将 WUPEN 位置 1 将不起任何作用。
- 只有当 I2C 时钟源为 HSI 振荡器时，该功能才可用。
- 必须使能时钟延长（NOSTRETCH=0）才能确保从停机模式唤醒功能正常工作。
- 如果禁止从停机模式唤醒（WUPEN=0），则在进入停机模式前必须禁止 I2C 外设（PE=0）。

将 I2C_CR1 寄存器中的 WUPEN 位置 1，可以使能从停机模式唤醒功能。对于 I2CCLK，必须选择 HSI 振荡器作为时钟源，以便从停机模式唤醒。

在停机模式中，HSI 关闭。当检测到 START 时，I2C 接口将 HSI 接通，并延长 SCL 低电平持续时间，直到 HSI 唤醒。HSI 随后用来接收地址。地址匹配的情况下，MCU 唤醒时间内，I2C 延长 SCL 的低电平持续时间。通过软件清除 ADDR 标志后，此延长操作被释放，传输正常进行。如果地址不匹配，HSI 再次关断，MCU 不被唤醒。

只有 ADDR 中断能够唤醒 MCU。因此，当 I2C 以主器件身份或在 ADDR 标志置 1 后以被寻址从器件

身份执行传输时，不要进入停机模式。将 ADDR 中断程序中的 SLEEPDEEP 位清零，然后仅在 StopF 标志置 1 后将其置 1，可对此进行管理。

18.2.15 错误条件

以下错误条件可能导致通信失败。

总线错误（BERR）

当 SDA 边沿出现且 SCL 为高电平时，会检测到起始或停止位。当检测到起始位或停止位但不位于第 9N 个 SCL 时钟脉冲之后时，则认为出现了总线错误。

只有当 I2C 在传输过程中用作主器件或被寻址为从器件时（即未处于从模式下的地址阶段），才会将总线错误标志置 1。

在从模式下误检测到起始位或重复起始位时，I2C 会像接收到正确的起始位一样进入地址识别状态。

检测到总线错误时，I2C_ISR 寄存器中的 BERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

仲裁丢失（ARLO）

当 SDA 线上发送高电平但在 SCL 上升沿却采样到低电平时，会检测到仲裁丢失。

- 在主模式下，将在地址阶段、数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，SDA 线和 SCL 线被释放，起始控制位由硬件清零，主器件自动切换为从模式。
- 在从模式下，将在数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，传输停止，SCL 和 SDA 线被释放。
- 检测到仲裁丢失时，I2C_ISR 寄存器中的 ARLO 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

上溢/下溢错误（OVR）

当满足 NOSTRETCH=1 和以下条件时，将在从模式下检测到上溢或下溢错误：

- 在接收过程中接收到一个新的字节，但 RXDR 寄存器的值还未被读取。接收的新字节丢失，自动发送 NACK 来响应新字节。
- 在发送过程中：
  - 当 STOPF=1 且应发送第一个数据字节时。TXE=0 时发送 I2C_TXDR 寄存器的内容，否则发送 0xFF。
  - 应发送一个新字节但尚未向 I2C_TXDR 寄存器写入数据时，将发送 0xFF。
  - 检测到上溢或下溢错误时，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

数据包错误校验错误（PECERR）

当接收到的 PEC 字节与 I2C_PECR 寄存器的内容不匹配时，将检测到 PEC 错误。接收到错误的 PEC 后，将自动发送 NACK。

检测到 PEC 错误时，I2C_ISR 寄存器中的 PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

超时错误（TIMEOUT）

满足以下任何条件均会出现超时错误：

- TIDLE=0 且 SCL 的低电平持续时间达到 TIMEOUTA[11:0]位中定义的时间：这用于检测 SMBus 超时。
- TIDLE=1 且 SDA 和 SCL 的高电平持续时间达到 TIMEOUTA[11:0]位中定义的时间：这用于检测总

线空闲情况。

- 主器件的累积时钟低电平延长时间达到了 TIMEOUTB[11:0]位中定义的时间（SMBus tLOW：MEXT 参数）
- 从器件的累积时钟低电平延长时间达到了 TIMEOUTB[11:0]位中定义的时间（SMBus tLOW：SEXT 参数）

当在主模式下检测到超时时，将自动发送停止位。

当在从模式下检测到超时时，将自动释放 SDA 和 SCL 线。

检测到超时错误时，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

报警（ALERT）

当 I2C 接口配置为主机（SMBHEN=1）、使能了报警引脚检测（ALERTEN=1）并且在 SMBA 引脚上检测到下降沿时，ALERT 标志将置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，将生成中断。

18.2.16 调试模式

当 MCU 进入调试模式时（内核停止），SMBus 超时定时器会根据 DBG 模块中的 DBG_I2Cx_SMBus_TIMEOUT 配置位选择继续正常工作还是停止工作。

18.3 I2C 低功耗模式

表 18-12 低功耗模式

模式	说明
睡眠	对 I2C 通信无影响。 I2C 中断可使器件退出睡眠模式。
停机	I2C 模块的寄存器内容仍被保持。

18.4 I2C 中断

下表给出了 I2C 中断请求列表。

表 18-13 I2C 中断请求

中断事件	事件标志	事件标志/中断清除方法	中断使能控制位
接收缓冲区非空	RXNE	读取 I2C_RXDR 寄存器	RXIE
发送缓冲区中断状态	TXIS	写入 I2C_TXDR 寄存器	TXIE
停止位检测中断标志	STOPF	写入 STOPCF=1	STOPIE
传输完成等待重载	TCR	写入 I2C_CR2（NBYTES[7:0]≠ 0）	TCIE
传输完成	TC	写入 START=1 或 Stop=1	
地址匹配	ADDR	写入 ADDRCF=1	ADDRIE
接收到 NACK 应答	NACKF	写入 NACKCF=1	NACKIE
总线错误（Bus error）	BERR	写入 BERRCF=1	ERRIE

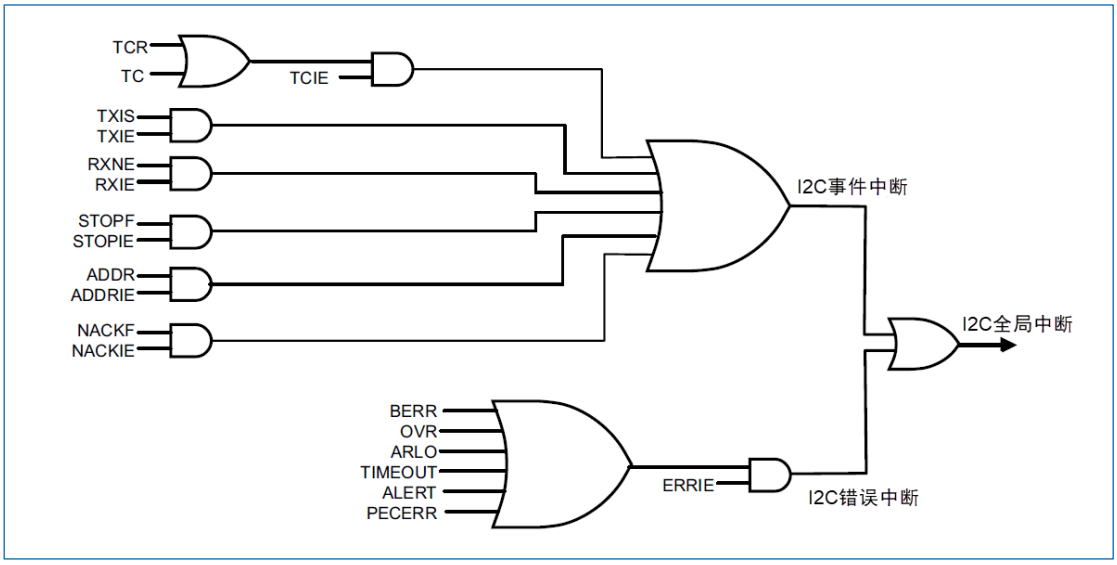
中断事件	事件标志	事件标志/中断清除方法	中断使能控制位
仲裁丢失	ARLO	写入 ARLOCF=1	
上溢/下溢（Overrun/Underrun）	OVR	写入 OVRCF=1	
PEC 错误	PECERR	写入 PECERRCF=1	
超时/t_{LOW} 错误	TIMEOUT	写入 TIMEOUTCF=1	
SMBus 报警	ALERT	写入 ALERTCF=1	

根据产品实现的不同，上述所有中断既可以共享同一个中断向量（I2C 全局中断），也可以分配到 2 个不同的中断向量上（I2C 事件中断和 I2C 错误中断）。

要使能 I2C 中断，需按照以下顺序操作：

1. 配置 NVIC 中的 I2CIRQ 通道并将其使能。
2. 配置 I2C 以生成中断。

I2C 唤醒事件连接到 EXTI 控制器（参见“10.3 EXTI 寄存器”）。



18-31 I2C 中断映射图

18.5 I2C 寄存器

基地址：0x4000 5400

空间大小：0x400

18.5.1 控制寄存器 1（I2C_CR1）

地址偏移：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res								PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	WUPEN	NOSTRETCH	SBC
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res			ANFOFF	DNF[3:0]				ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE
			rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

位 31:24	Res：保留 必须保持复位值。
位 23	PECEN：启用 PEC • 0：禁用 PEC 计算 • 1：启用 PEC 计算

位 22	ALERTEN：SMBus 通知使能
	• 设备模式（SMBHEN = 0）
	  ○ 0：释放 SMBA 引脚为高，并禁用 NACK 之后的通知响应地址头：0001100x。
	  ○ 1：拉低 SMBA 引脚并启用 ACK 之后的通知响应地址头：0001100x。
	• HOST 模式（SMBHEN = 1）
	  ○ 0：SMBus 通知引脚（SMBA）不支持。
	  ○ 1：支持 SMBus 通知引脚（SMBA）。

位 21	SMBDEN：SMBus 器件的默认地址启用 • 0：禁用设备的默认地址。地址 0b1100001x 会被 NACK。 • 1：启用设备的默认地址。地址 0b1100001x 会被 ACK。
位 20	SMBHEN：SMBus HOST 地址启用 • 0：禁用 HOST 地址。地址 0b0001000x 会被 NACK。 • 1：启用 HOST 地址。地址 0b0001000x 会被 ACK。
位 19	GCEN：广播呼叫地址使能 • 0：禁止广播呼叫。地址 0b00000000 会被 NACK。 • 1：启用广播呼叫。地址 0b00000000 会被 ACK。
位 18	WUPEN：从 STOP 模式唤醒 • 0：禁止从 STOP 模式唤醒 • 1：允许从 STOP 模式唤醒
位 17	NOSTRETCH：禁止时钟延长 该位用于在从机模式禁止时钟延长。 • 0：允许时钟延长 • 1：禁止时钟延长
位 16	SBC：从设备模式下的字节控制 该位用于在从机模式使能硬件字节控制。 • 0：从机字节控制模式关闭 • 1：从机字节控制模式使能

	从机字节控制置 1 时，I2C 的 SCL 和 SDA 线都被释放且内部状态机和所有的状态为回到复位值，而控制位的内容被保留。
位 15:13	Res：保留 必须保持复位值。
位 12	ANFOFF：模拟噪声滤波器关闭/开启 • 0：模拟噪声滤波器开启 • 1：模拟噪声滤波器关闭
位 11:8	DNF[3:0]：数字噪声滤波器 该位域用于配置 SDA 和 SCL 输入上的数字噪声滤波器。数字滤波器会用 DNF[3:0] x t_{I2C_CLK} 的长度来工作。 • 0000：数字滤波器禁用 • 0001：数字滤波器启用，并且滤波能力达到 1 个 t_{I2C_CLK}。 …… • 1111：数字滤波器启用，其滤波能力达到 15 个 t_{I2C_CLK}。
位 7	ERRIE：错误中断使能 • 0：错误检测中断禁用 • 1：错误检测中断使能
位 6	TCIE：传输完成中断使能 • 0：传输完成中断禁用 • 1：传输完成中断使能
位 5	STOPIE：STOP 检测中断使能 • 0：停止检测（STOPF）中断禁用 • 1：停止检测（STOPF）中断使能
位 4	NACKIE：收到 NACK 中断使能 • 0：（NACKF）收到中断禁用 • 1：（NACKF）收到中断使能
位 3	ADDRIE：地址匹配中断使能（仅从机） • 0：地址匹配（ADDR）中断禁用 • 1：启用地址匹配（ADDR）中断
位 2	RXIE：接收中断使能 • 0：接收（RXNE）中断禁用 • 1：启用接收（RXNE）中断
位 1	TXIE：TX 中断使能 • 0：发送（TXIS）中断禁用

	• 1：发送（TXIS）中断使能
位 0	PE：外设使能 • 0：外设禁用 • 1：外设使能

18.5.2 控制寄存器 2（I2C_CR2）

地址偏移：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res					PECBYTE	AUTOEND	RELOAD	NBYTES[7:0]							
					rs	rw	rw	rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD[9:8]		SADD[7:1]							SADD[0]
rs	rs	rs	rw	rw	rw	rw		rw							rw

位 31:27	Res：保留 必须保持复位值。
位 26	PECBYTE：包错误检查字节 • 0：没有 PEC 传送。 • 1：要求发送/接收 PEC。 该位由软件置位。 该位在以下任一条件下由硬件清 0： • 在 PEC 传输完后 • 在收到 STOP 条件后 • 在收到地址匹配事件后 • 在 PE=0
位 25	AUTOEND：自动结束模式（主机模式） 该位由软件置 1 和清零。 • 0：软件结束模式：当 NBYTES 个数据传输完毕后，TC 标志被置 1，SCL 被拉低。 • 1：自动结束模式：当 NBYTES 个数据传输完后，会自动发送一个停止条件。
位 24	RELOAD：NBYTES 重装载模式 该位由软件置 1 和清 0。 • 0：在传输完 NBYTES 个字节后，传输结束。 • 1：传输 NBYTES 个字节后，传输并不结束（NBYTES 将被重装载）。当 NBYTES 个数据传输完毕后，TC 标志被置 1，SCL 被拉低。
位 23:16	NBYTES[7:0]：字节数 该位域写入待发送/接收的字节数。从机模式并且 SBC=0 的时候，该位域无效。

位 15	NACK：产生 NACK（从机模式） • 0：在当前字节收到后发送 ACK。 • 1：当前字节接收到后发送一个 NACK。 该位由软件置位。 在满足以下任一条件时，该位由硬件清 0： • NACK 发送后 • 收到 STOP 条件后 • 收到地址匹配事件后 • PE=0
位 14	STOP：产生停止条件（主机模式） 在主机模式下： • 0：不产生 STOP 条件。 • 1：当前字节传输完后产生停止条件。 该位由软件置 1。 在满足以下任一条件时，该位由硬件清 0： • 检测到 STOP 条件后 • PE=0
位 13	START：产生起始条件 • 0：没有起始条件产生 • 1：产生 START/RESTART 条件： 如果 I2C 已经是在主机模式下并且 AUTOEND = 0，RELOAD=0，并且 NBYTES 个字节发送完毕后设置此位会产生重复起始条件；否则只要总线空闲，设置此位将会立即产生一个起始条件。 该位通过软件置 1。 在满足以下任一条件时，该位由硬件清 0： • 在发送完一个起始条件和地址序列之后 • 仲裁丢失 • 超时错误 • PE=0 • 软件向 I2C_ICR 寄存器的 ADDRCF 位写 1
位 12	HEAD10R：10 位地址头只读方向（主接收器模式） • 0：主机发送完整的 10 位从机地址读序列：起始 + 2 字节 10 位地址（写方向） + 重新起始 + 10 位地址中的前 7 位（读方向）。 • 1：主机只发送 10 位地址的前 7 位，接着是读方向。
位 11	ADD10：10 位地址模式（主机模式） • 0：主机按 7 位地址模式操作。 • 1：主机按 10 位地址模式操作。

位 10	RD_WRN：传输方向（主机模式） • 0：主机请求一个写传输。 • 1：主机请求一个读传输。
位 9:8	SADD[9:8]：从机地址位 9:8（主机模式） • 在 7 位地址模式下（ADD10 = 0）：该位域不起作用。 • 在 10 位地址模式下（ADD10 = 1）：该位域应写入待发送的从机地址位的 9:8 位。
位 7:1	SADD[7:1]：从机地址位 7:1（主机模式） • 在 7 位地址模式下（ADD10 = 0）：该位域应写入待发送的 7 位从机地址位。 • 在 10 位地址模式下（ADD10 = 1）：该位域应写入待发送的从机地址位的 7:1 位。
位 0	SADD[0]：从机地址的 0 位（主模式） • 在 7 位地址模式下（ADD10 = 0）：该位域不起作用。 • 在 10 位地址模式下（ADD10=1）：该位域应写入待发送的从机地址位的 0 位。

18.5.3 本机地址 1 寄存器（I2C_OAR1）

地址偏移：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA1EN	Res				OA1MODE	OA1[9:8]		OA1[7:1]							OA1[0]
rw					rw	rw		rw							rw

位 31:16	Res：保留 必须保持复位值。
位 15	OA1EN：本机地址 1 启用 • 0：禁用本机地址 1 接收到从机地址 OA1 后会用 NACK 回应。 • 1：本机地址 1 启用 接收到从机地址 OA1 后会用 ACK 回应。
位 14:11	Res：保留 必须保持复位值。
位 10	OA1MODE：本机地址 1 的 10 位模式 • 0：本机地址 1 是一个 7 位地址。 • 1：本机地址 1 是一个 10 位的地址。
位 9:8	OA1[9:8]：接口地址的 9:8 位

	• 7 位地址模式：不生效 • 10 位地址模式：地址的位 9:8
位 7:1	OA1[7:1]：接口地址的 7:1 位
位 0	OA1[0]：接口地址的 0 位 • 7 位地址模式：不生效 • 10 位地址模式：地址的 0 位

18.5.4 本机地址 2 寄存器（I2C_OAR2）

地址偏移：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA2EN	Res				OA2MSK[2:0]			OA2[7:1]							Res
rw					rw			rw							

位 31:16	Res：保留 必须保持复位值。
位 15	OA2EN：本机地址 2 启用 • 0：禁用本机地址 2 接收到从机地址 OA2 后会用 NACK 回应。 • 1：本机地址 2 启用 接收到从机地址 OA2 后会用 ACK 回应。
位 14:11	Res：保留 必须保持复位值。
位 10:8	OA2MSK [2:0]：本机地址 2 屏蔽 • 000：没有屏蔽 • 001：OA2 [1]被屏蔽掉，可忽略。只有 OA2 [7:2]参与比较。 • 010：OA2 [2:1]被屏蔽掉，可忽略。只有 OA2 [7:3]参与比较。 • 011：OA2 [3:1]被屏蔽掉，可忽略。只有 OA2 [7:4]参与比较。 • 100：OA2 [4:1]被屏蔽掉，可忽略。只有 OA2 [7:5]参与比较。 • 101：OA2 [5:1]被屏蔽掉，可忽略。只有 OA2 [7:6]参与比较。 • 110：OA2 [6:1]被屏蔽掉，可忽略。只有 OA2 [7]参与比较。 • 111：OA2 [7:1]被屏蔽掉，可忽略。没有比较，所有的（除保留地址）收到的 7 位地址都会用 ACK 回应。
位 7:1	OA2 [7:1]：接口地址 7:1 位

位 0	Res：保留 必须保持复位值。

18.5.5 时序寄存器（I2C_TIMINGR）

地址偏移：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESC[3:0]				Res				SCLDEL[3:0]				SDADEL[3:0]			
rw								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH[7:0]								SCLL[7:0]							
rw								rw							

位 31:28	PRESC[3:0]：时序预分频器
位 27:24	Res：保留 必须保持复位值。
位 23:20	SCLDEL[3:0]：数据建立时间 该位域用于生成发送模式中 SDA 的沿和 SCL 上升沿之间的延迟 t_{SCLDEL}。 t_{SCLDEL} =（SCLDEL+1） x t_{PRESC}
位 19:16	SDADEL[3:0]：数据保持时间 该位域用于生成发送模式中 SCL 的下降沿和 SDA 的沿之间的延迟 t_{SDADEL}。 t_{SDADEL}= SDADEL x t_{PRESC}
位 15:8	SCLH[7:0]：SCL 高电平时间（主机模式） 该位域用于在主机模式下产生 SCL 的高电平时间。 t_{SCLH} =（SCLH+1） x t_{PRESC}
位 7:0	SCLL[7:0]：SCL 低电平时间（主机模式） 该位域用于在主机模式下产生 SCL 的低电平时间。 t_{SCLL} =（SCLL+1） x t_{PRESC}

18.5.6 超时寄存器（I2C_TIMEOUTR）

地址偏移：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEXTEN	Res			TIMEOUTB[11:0]											
rw				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUTEN	Res		TIDLE	TIMEOUTA[11:0]											
rw			rw	rw											

位 31	TEXTEN：外部时钟超时启用 • 0：外部时钟超时检测被禁用。 • 1：外部时钟超时检测启用。
位 30:29	Res：保留 必须保持复位值。
位 27:16	TIMEOUTB[11:0]：总线超时 B 此字段用于配置累计时钟延长超时： • 在主机模式下，待检测的累计主机时钟低延长时间（$t_{LOW:MEXT}$）。 • 在从机模式下，待检测的累积从机时钟低延长时间（$t_{LOW:SEXT}$）。 $t_{LOW:EXT}$=（TIMEOUTB+1） x 2048 x t_{I2C_CLK}
位 15	TIMOUTEN：时钟超时检测使能 • 0：禁用 SCL 超时检测 • 1：启用 SCL 超时检测：当 SCL 保持低的时间超过 $t_{TIMEOUT}$（TIDLE=0）或保持高的时间超过 t_{IDLE}（TIDLE=1），会检测到一个超时错误（TIMEOUT=1）。
位 14:13	Res：保留 必须保持复位值。
位 12	TIDLE：空闲时钟超时检测 • 0：TIMEOUTA 用于检测 SCL 低电平超时。 • 1：TIMEOUTA 用于同时检测 SCL 和 SDA 高电平超时（总线空闲条件）。
位 11:0	TIMEOUTA[11:0]：总线超时 A 此字段用于配置： • 当 TIDLE=0 时，SCL 低超时条件 $t_{TIMEOUT}$。 $t_{TIMEOUT}$=(TIMEOUTA+1) x 2048 x t_{I2C_CLK} • 当 TIDLE=1 时，总线空闲条件（SCL 和 SDA 同时为高）。 t_{IDLE}=(TIMEOUTA+1) x 4 x t_{I2C_CLK}

18.5.7 中断和状态寄存器（I2C_ISR）

地址偏移：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res								ADDCODE[6:0]							DIR
								r							r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res	ALERT	TIMEOUT	PECERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
r		r	r	r	r	r	r	r	r	r	r	r	r	r_w1	r_w1

位 31:24	Res：保留 必须保持复位值。
位 23:17	ADDCODE[6:0]：匹配的地址码（从机模式） 该位域由地址匹配事件发生时所接收的地址（ADDR= 1）来更新。 在 10 位地址的情况下，ADDCODE 提供 10 位地址的头 2 位以后的地址。
位 16	DIR：传输方向（从机模式） 该位在地址匹配事件发生时（ADDR= 1）更新。 • 0：写传输，从机进入接收模式。 • 1：读传输，从机进入发送模式。
位 15	BUSY：总线忙
位 14	Res：保留 必须保持复位值。
位 13	ALERT：SMBus 通知 该位在 SMBHEN=1（SMBus HOST 配置）及 ALERTEN=1 的条件下，当 SMBA 脚上检测到 SMBALERT 事件（下降沿）时，由硬件置 1。 通过设置 ALERTCF 位，该位由软件清零。
位 12	TIMEOUT：超时或 tLOW 检测标志 在超时或外部时钟超时发生时，该位被硬件置 1。通过设置 TIMEOUTCF 位，该位由软件清零。
位 11	PECERR：接收中的 PEC 错误 该位在收到的 PEC 值和 PEC 寄存器的内容不匹配时，由硬件置 1。收到错误的 PEC 后，会自动发送一个 NACK。该位可以通过将 PECCF 位置 1，由软件清零。
位 10	OVR：溢出/欠载（从机模式） 该位在从机模式下 NOSTRETCH = 1 时，发生溢出/欠载错误的时候，由硬件置 1。 通过设置 OVRCF 位，该位由软件清零。
位 9	ARLO：仲裁丢失 该位在总线仲裁丢失的情况下由硬件置 1。通过设置 ARLOCF 位，该位由软件清零。
位 8	BERR：总线错误 该位在检测到错位的起始或者停止条件时，由硬件置 1。 通过设置 BERRCF 位，由软件清零。
位 7	TCR：传输完成重加载 该标志在 RELOAD=1 且 NBYTES 个数据发送完毕后，由硬件置 1。

	向 NBYTES 写入了一个非零的值时，TCR 位由软件清除。
位 6	TC：发送完毕（主机模式） 该位在 RELOAD=0、AUTOEND=0 且 NBYTES 个数据发送完毕后，由硬件置 1。 该位在软件将 START 或 STOP 位置 1 的时清零。
位 5	STOPF：停止检测标志 在外设参与传输的以下情况，当总线上检测到一个停止条件时，该位由硬件置 1： • 作为主机，如果由外设生成一个停止条件。 • 作为从机，如果外设在这次传输之前被正确地寻址到。 该位可以通过将 STOPCF 位置 1，由软件清零。
位 4	NACKF：收到 NACK 标志 该位在一个字节传输后收到一个 NACK 时，由硬件设置。 该位可以通过将 NACKCF 位置 1，由软件清零。
位 3	ADDR：地址匹配（从机模式） 当收到的从机地址与其中一个有效的从机地址匹配时，该位由硬件置 1。 该位通过设置 ADDRCF 位，由软件清零。
位 2	RXNE：接收数据寄存器非空（接收） 当接收的数据被复制到 I2C_RXDR 寄存器并准备好被软件读取时，该位由硬件置位。在读取 I2C_RXDR 时，RXNE 会被清除。
位 1	TXIS：发送中断状态（发送） 当 I2C_TXDR 寄存器为空时，该位由硬件置 1，这时必须将待发送的数据写到 I2C_TXDR 寄存器。下一个待发送的数据被写入 I2C_TXDR 寄存器时，该位会被清除。仅当 NOSTRETCH = 1 时，该位由软件写 1，以生成一个 TXIS 事件（如果 TXIE =1 就产生中断）。
位 0	TXE：发送数据寄存器空（发送） 当 I2C_TXDR 寄存器为空时，该位由硬件置 1。下一个待发送的数据被写到 I2C_TXDR 寄存器时，该位会被清 0。 该位可通过软件写 1，以清空发送数据寄存器 I2C_TXDR。

18.5.8 中断清除寄存器（I2C_ICR）

地址偏移：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res		ALERTCF	TIMOUTCF	PECCF	OVRCF	ARLOCF	BERRCF	Res		STOPCF	NACKCF	ADDRCF	Res		
		w	w	w	w	w	w			w	w	w			

位 31:14	Res：保留 必须保持复位值。
位 13	ALERTCF：通知标志清零 对该位写 1，会清除 I2C_ISR 寄存器中的 ALERT 标志位。
位 12	TIMOUTCF：超时检测标志清除 对该位写 1，会清除 I2C_ISR 存器中的 TIMEOUT 标志位。
位 11	PECCF：PEC 错误标志清除 对该位写 1，会清除 I2C_ISR 寄存器中的 PECERR 标志位。
位 10	OVRCF：溢出/欠载标志清除 对该位写 1，会清除 I2C_ISR 寄存器中的 OVR 标志位。
位 9	ARLOCF：仲裁丢失标志清除 对该位写 1，会清除 I2C_ISR 寄存器中的 ARLO 标志位。
位 8	BERRCF：总线错误标志清除 对该位写 1，会清除 I2C_ISR 寄存器中的 BERRF 标志位。
位 7:6	Res：保留 必须保持复位值。
位 5	STOPCF：停止检测标志清除 对该位写 1，会清除 I2C_ISR 寄存器中的 STOPF 标志位。
位 4	NACKCF：收到 NACK 标志清除 对该位写 1，会清除 I2C_ISR 寄存器中的 NACKF 标志位。
位 3	ADDRCF：地址匹配标志清除 对该位写 1，会清除 I2C_ISR 寄存器中 ADDR 标志位，还会清除 I2C_CR2 寄存器中的 START 位。
位 2:0	Res：保留 必须保持复位值。

18.5.9 PEC 寄存器（I2C_PECR）

地址偏移：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Res	PEC[7:0]
	r

位 31:8	Res：保留 必须保持复位值。
位 7:0	PEC [7:0]：包错误检查寄存器 当PECEN = 1 时，此字段包含内部PEC 结果。PEC 在PE = 0 或SWRST 被置1 时，该位域由硬件清零。

18.5.10 接收数据寄存器（I2C_RXDR）

地址偏移：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								RXDATA[7:0]							
								r							

位 31:8	Res：保留 必须保持复位值。
位 7:0	RXDATA [7:0]：8 位接收数据（从I2C 总线接收的数据字节）

18.5.11 发送数据寄存器（I2C_TXDR）

地址偏移：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res								TXDATA[7:0]							
								rw							

位 31:8	Res：保留 必须保持复位值。
位 7:0	TXDATA[7:0]：8 位发送数据（待发送到I2C 总线上的数据字节）

19 通用同步异步收发器（USART）

通用同步异步收发器（USART）支持标准的不归零码（Not return to Zero，NRZ）异步串行数据格式，并且能够进行全双工数据通信。USART 可以通过自带的波特率发生器产生不同的波特率，并且还支持同步通信、单线半双工通信、多处理器通信、硬件流控制、智能卡协议等功能。

19.1 USART 主要特性

- 全双工异步通信
- NRZ 标准格式（标记/空格）
- 可配置为 16 倍过采样或 8 倍过采样（最高收发波特率=USART 时钟频率/8）
- 双时钟域允许：
  - USART 功能和从停机模式唤醒
  - 方便的波特率编程，独立于 PCLK 重新编程
- 自动波特率检测
- 数据字长可编程（7 位、8 位或 9 位）
- 可编程的数据顺序（MSB 或 LSB）
- 停止位可配置（支持 1 个、1.5 个或 2 个停止位）
- 用于同步通信的同步模式和时钟输出
- 单线半双工通信
- 发射器和接收器可单独使能
- 发送和接收的信号极性可单独控制
- 可配置发送（TX）和接收（RX）引脚交换
- RS485 驱动器使能
- 通信控制/错误检测标志
- 奇偶校验控制：
  - 发送奇偶校验位
  - 检查接收数据帧的奇偶性
- 具有多种中断状态标志位
- 多处理器通信
  - 如果地址不匹配，USART 将进入静默模式。
  - 从静默模式唤醒（通过空闲线检测或地址标记检测）

19.2 USART 扩展特性

- LIN 主模式断路发送功能和 LIN 从模式断路检测功能
- 支持红外数据协议（IrDA）
- 支持智能卡协议
- 支持 ModBus 协议通信

19.3 USART 实现

表 19-1 USART 特性

USART 模式/特性	USART1
DMA 传输	不支持
多处理器通信	支持
同步模式	支持
单线半双工通信	支持
双时钟域和从停机模式唤醒	支持
自动波特率检测	支持
Modbus 通信	支持
RS232 硬件流控制	不支持
RS485 驱动器使能	不支持
IrDA SIR ENDEC 模块	支持
LIN 模式	支持
智能卡模式	支持

19.4 USART 功能说明

USART 通信包括四个引脚，分别是接收数据输入引脚（RX）、发送数据输出引脚（TX）、时钟输出引脚（CLK）以及 RS485 驱动器使能引脚（DE）。它们的具体描述如下：

- RX：接收数据输入引脚。该引脚用于接收串行数据。
- TX：发送数据输出引脚。若禁用发送器，则该输出引脚的模式由其 I/O 端口配置决定。若使能发送器，则该引脚在空闲状态下处于高电平。在单线和智能卡模式下，该引脚用于发送和接收数据。
- CK：时钟输出引脚。该引脚用于输出发送器的数据时钟，以便按照 SPI 主模式进行同步发送（起始位和结束位上无时钟脉冲，可通过软件向最后一个数据位发送时钟脉冲）。RX 上可同步接收数据。时钟相位和极性可编程。在智能卡模式下，CK 输出可向智能卡提供时钟。
- DE：该引脚在 RS485 驱动器使能下使用。它用于使能外部收发的发送模式。

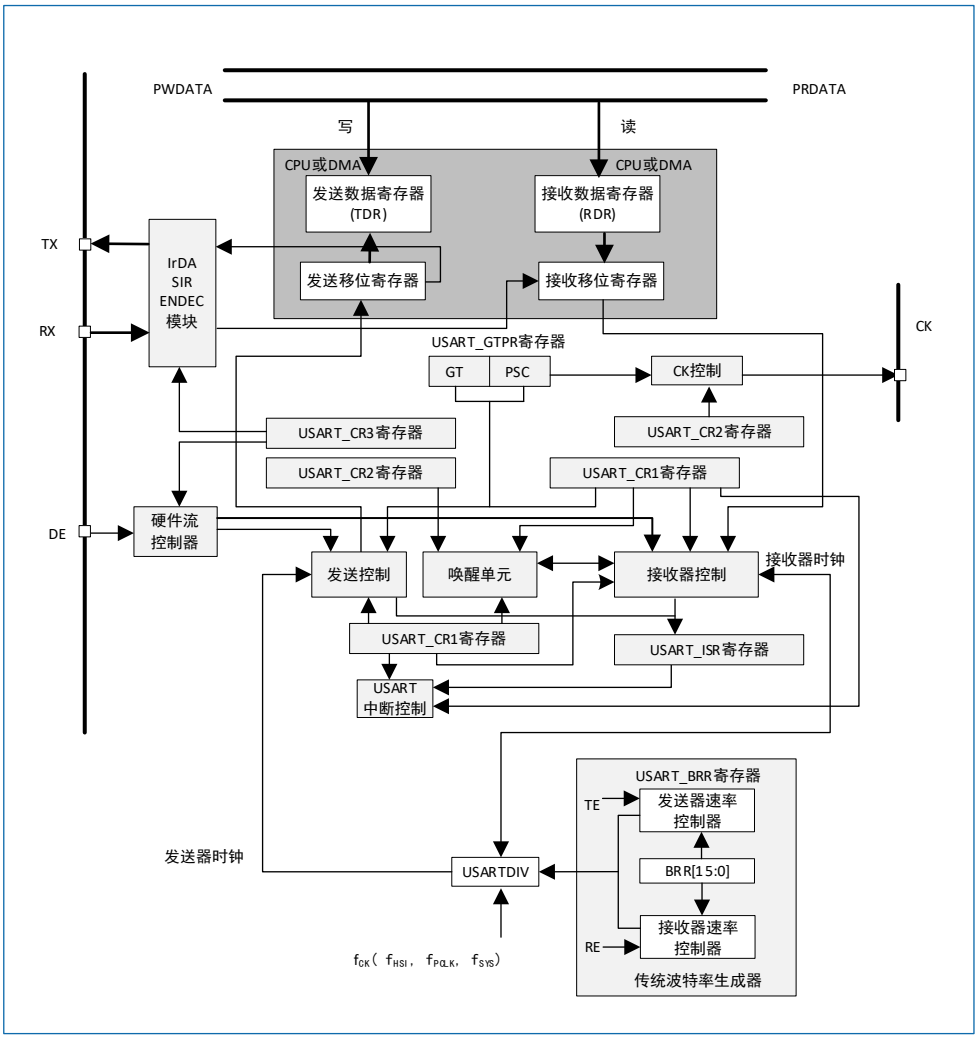


图 19-1 USART 框图

在图 19-1 中，关于在 USART_BRR 寄存器中编码 USARTDIV 的详细信息，请参见“19.4.4 USART 波特率生成”。f_{CK} 可以是 f_{HSI}、f_{PCLK} 或 f_{SYS}。

19.4.1 USART 字符说明

通过配置 USART_CR1 寄存器中的 M[1:0]位来选择 7 位、8 位或 9 位的字长。

- 7 位字符长度：M[1:0]=10
- 8 位字符长度：M[1:0]=00
- 9 位字符长度：M[1:0]=01

说明：在 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率检测（0x7F 帧和 0x55 帧的检测）。

默认情况下，数据引脚（TX 或 RX）在起始位时处于低电平状态，在停止位时处于高电平状态。

通过极性配置（USART_CR2 寄存器中的 TXINV/RXINV 位）使 TX/RX 的有效电平反向。

空闲帧是指一帧数据的平值均为“1”。

中断帧是指一帧数据的电平值均为“0”。

下图给出了不同字长模式下的编程。

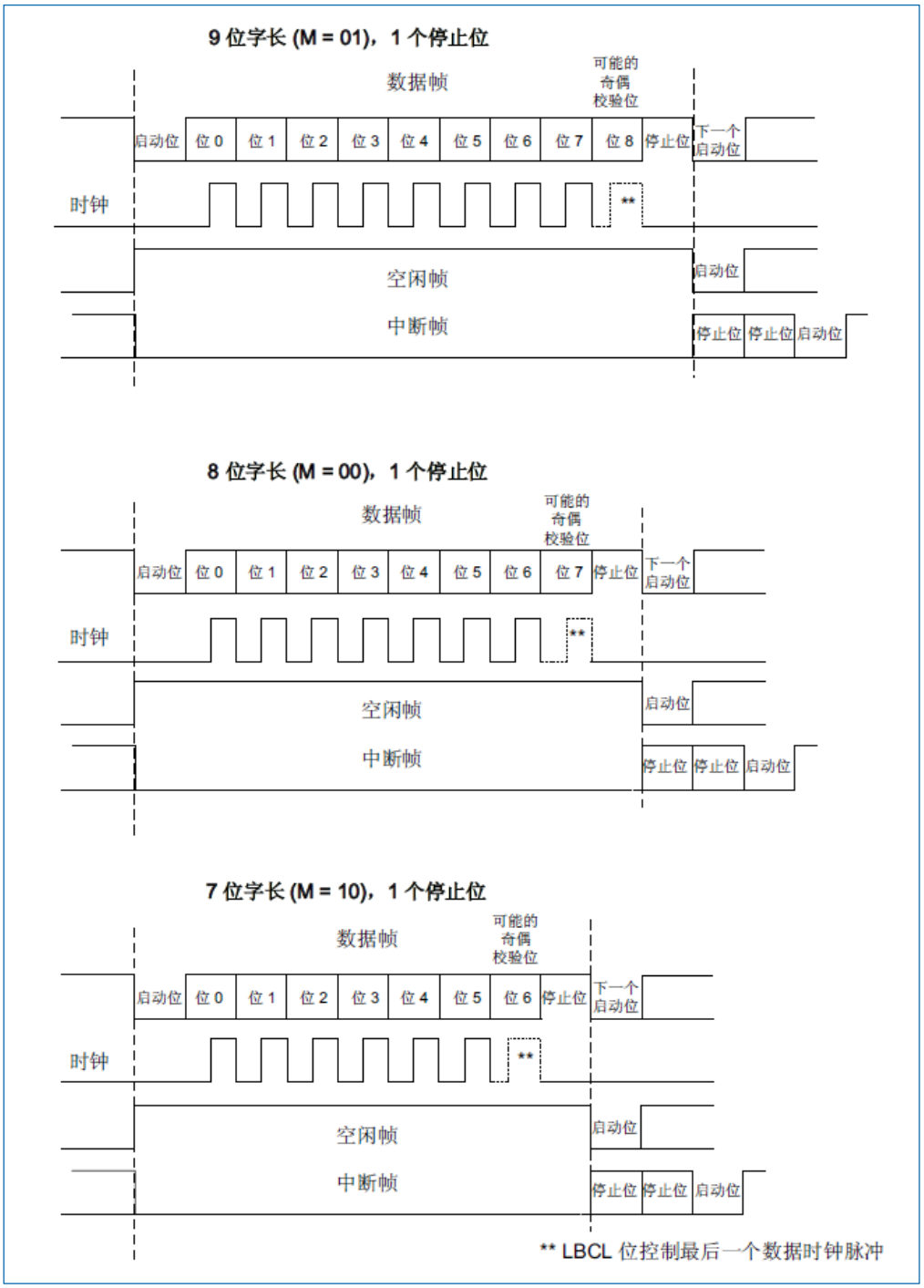


图 19-2 字长编程

19.4.2 USART 发送器

通过配置 USART_CR1 寄存器中的 M[1:0]位来选择发送器的数据字长（7 位、8 位或 9 位）。

通过配置 USART_CR1 寄存器中的发送使能位（TE），以使能发送器。发送移位寄存器中的数据通过 TX 引脚输出，相应的时钟脉冲从 CK 引脚输出。

可配置的停止位

通过配置 USART_CR2 寄存器中的 STOP[1:0]位来选择停止位的个数。

- 1 个停止位：这是停止位数量的默认值。
- 2 个停止位：正常 USART 模式、单线模式和调制解调器模式支持该值。
- 1.5 个停止位：用于智能卡模式。

空闲帧发送将包括停止位。

中断发送是 10 个低电平位（M[1:0]=00 时）、11 个低电平位（M[1:0]=01 时）或 9 个低电平位（M[1:0]=10 时），然后是 2 个停止位。无法传送长中断（中断长度大于 9/10/11 个低电平位）。

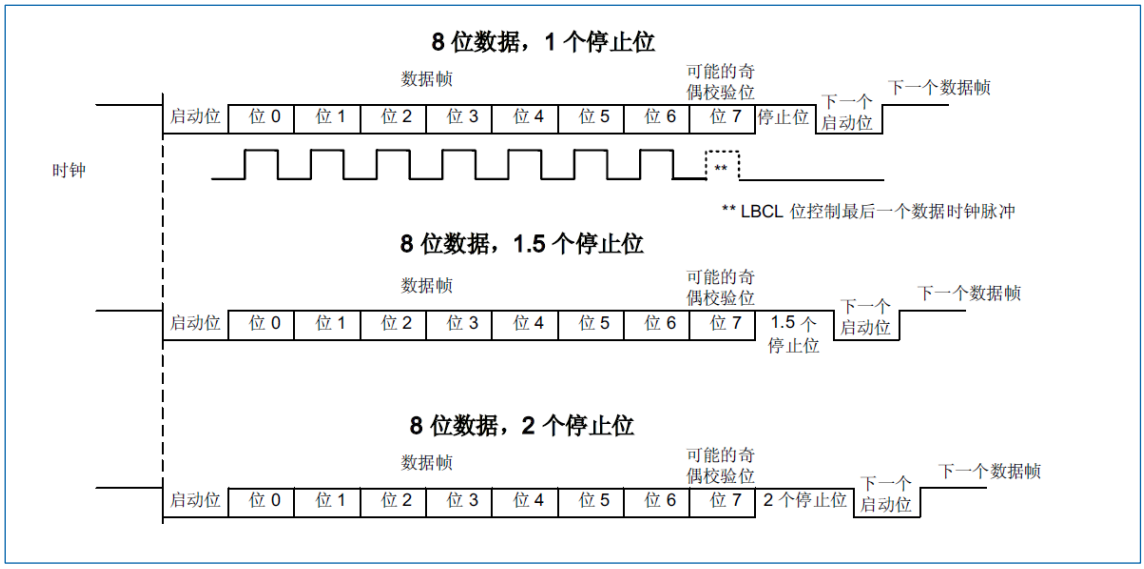


图 19-3 可配置的停止位

字符发送

USART 在发送字符的时候，默认配置（LSB）下 TX 引脚会先发送数据的最低有效位。每个字符都包含数据位、起始位以及停止位。

通过向发送数据寄存器（USART_TDR）写入数据可将 TXE 位清零。TXE 位由硬件置 1，它表示：

- 数据已从 USART_TDR 寄存器移到移位寄存器中并且数据开始发送。
- USART_TDR 寄存器为空。
- USART_TDR 寄存器中可写入下一个数据。

如果数据帧发送完成（停止位后）且 TXE 位置 1，这时 TC 位将变为高电平。若配置 USART_CR1 寄存器中的 TCIE 位，则会产生 TC 中断。向 USART_TDR 寄存器中写入最后一个数据后，必须等待至 TC=1，才可以保证数据准确无误地发送出去。

USART 字符发送时序如图 19-4 所示。

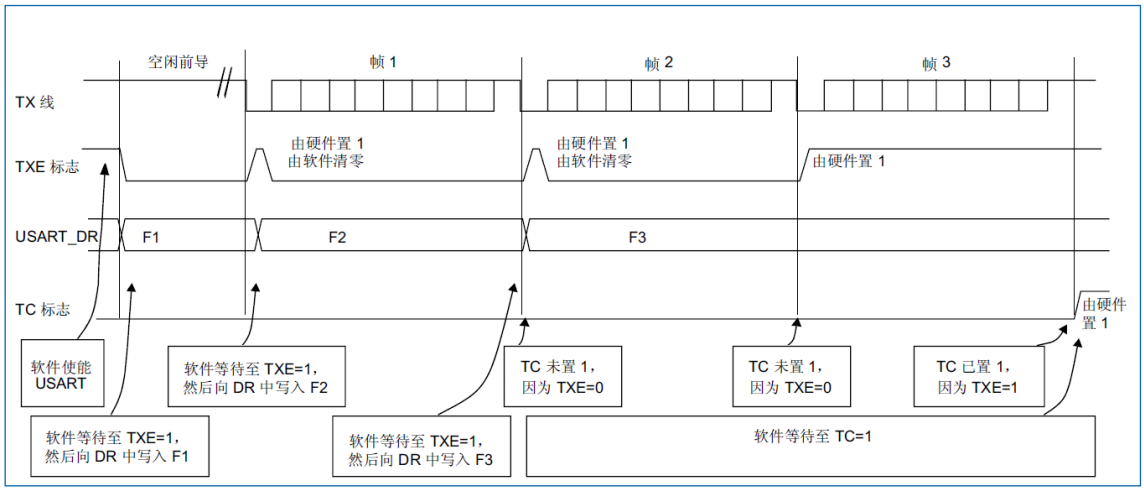


图 19-4 发送时的 TC/TXE 行为

字符发送配置步骤：

1. 配置 USART_CR1 中的 M 位来定义字长。
2. 配置 USART_BRR 寄存器来选择所需波特率。
3. 配置 USART_CR2 中的停止位（STOP[1:0]）个数。
4. 配置 USART_CR1 寄存器中的 UE 位来使能 USART 通信。
5. 配置 USART_CR1 中的 TE 位以便在首次发送时发送一个空闲帧。
6. 在 USART_TDR 寄存器中写入要发送的数据（该操作将清零 TXE 位）。重复这一步骤可满足连续发送数据要求。
7. USART_TDR 寄存器写入最后一个数据后，等待 TC=1，表明数据发送完成。

中断帧（BREAK）发送

若置位 USART_RQR 寄存器中的 SBKRQ 位，将发送一个中断帧。中断帧在整个字符周期内，电平始终为“0”。通过将 USART_ISR 中的 SBKF 位置 1 来发送中断字符，并且在中断字符发送完成后，SBKF 位清零。USART 在中断帧末尾插入 2 个停止位。

空闲帧发送

在初始情况下，将 USART_CR1 寄存器中的 TE 位置 1 会驱动 USART 在第一个数据帧之前发送一个空闲帧。空闲帧在整个字符周期内，电平始终为“1”。发送的空闲帧包含了停止位。

19.4.3 USART 接收器

通过配置 USART_CR1 寄存器中的 M 位来选择接收器接收的数据字长（7 位、8 位或 9 位），接收器支持 0.5、1、1.5 和 2 个停止位。

起始位检测

USART 支持 8/16 倍过采样，起始位检测序列相同。

在 USART 中，识别出特定序列的采样时会检测起始位。例如，此序列为：1 1 1 0 X 0 X 0X 0X 0 X 0X 0。

在过采样中，当进行第一次采样时，检测到第 3 位、第 5 位和第 7 位均为 0；同时进行第二次采样时，检测到第 8 位、第 9 位和第 10 位均为 0 时，则检测到起始位。当配置了 RXNE 和 RXNEIE 时，可以产生中断。

在检测到起始位（RXNE 标志位置 1）的情况下，以下描述均会导致 NF 噪声标志置位：3 个采样位中有 2 位为 0（第 3 位、第 5 位和第 7 位进行采样或第 8 位、第 9 位和第 10 位采样）。

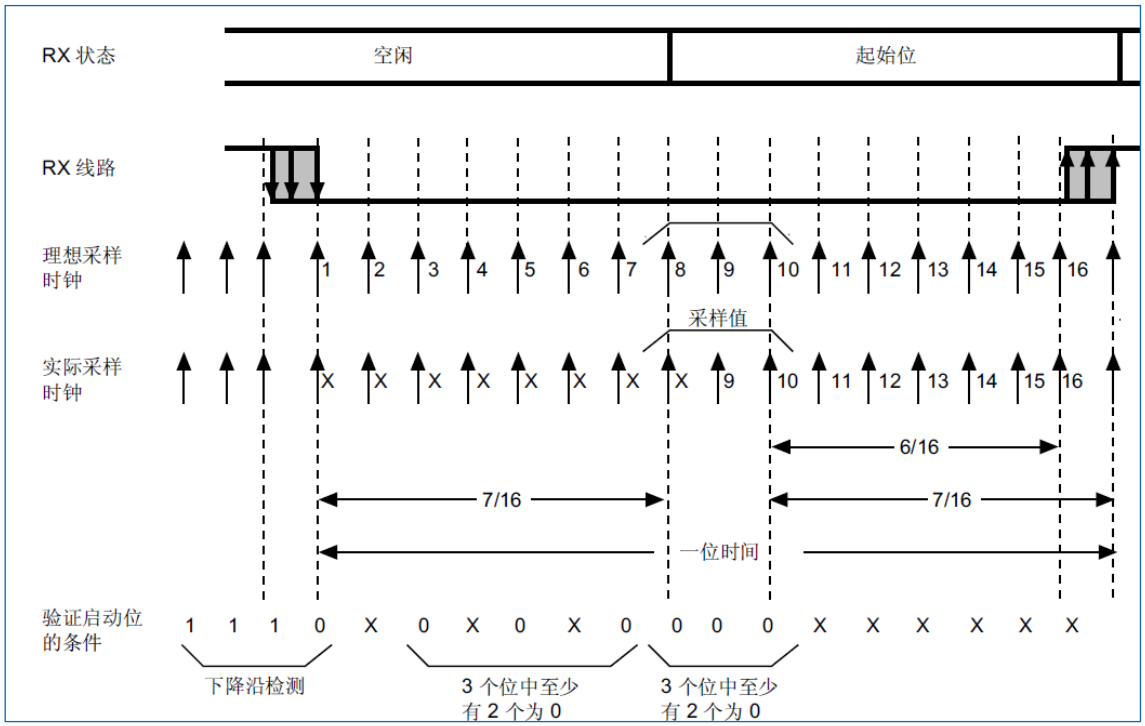


图 19-5 16 倍或 8 倍过采样时的起始位检测

说明：如果序列不完整，起始位检测将中止，接收器将返回空闲状态（无标志位置 1）等待下降沿。

字符接收

USART 接收期间，首先通过 RX 引脚移入数据的最低有效位（默认配置）。

配置字符接收的步骤如下：

1. 配置 USART_CR1 中的 M 位来定义字长。
2. 配置 USART_BRR 寄存器来选择所需波特率。
3. 配置 USART_CR2 中的停止位（STOP[1:0]）个数。
4. 配置 USART_CR1 寄存器中的 UE 位来使能 USART 通信。
5. 将 USART_CR1 寄存器中的 RE 位置 1，以便接收器搜索起始位。

接收到字符时：

- RXNE 位置 1。这表明移位寄存器的内容已传送到 RDR，且可进行读取操作。
- 如果 RXNEIE 位置 1，则会生成中断。
- 如果接收过程中检测到帧错误、噪声错误、上溢错误或校验位错误，错误标志位（FE、NF、ORE 或 PE）会被置 1。
- 在普通模式下，通过软件对 USART_RDR 寄存器执行读操作将 RXNE 位清零。也可以通过将 USART_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE 标志位清零。RXNE 位必须在结束接收下一个字符前清零，否则会发生上溢错误。

中断帧接收

接收到中断字符时，USART_ISR 寄存器中的帧错误标志 FE 会置 1。

空闲帧接收

检测到空闲帧时，USART_ISR 寄存器中的 IDLE 位会被置 1；若 IDLEIE 位被置 1，则会产生中断。同时可以通过置位 USART_ICR 寄存器中的 IDLECF 位来对 IDLE 位进行清零。

上溢错误

当接收到一个字符且 RXNE 位未被复位时，将发生上溢错误。除非 RXNE 位被清零，否则移位寄存

器中的数据不能传送到 RDR 寄存器。

每接收到一个字节后，RXNE 标志位都会被置 1。若在 RXNE 标志位未被清零时接收到字符，则会发生上溢错误。

发生上溢错误时：

- ORE 位被置 1。
- RDR 中的内容不会丢失。
- 移位寄存器中的数据将被覆盖，并且发生上溢错误时接收到的任何数据都将直接被丢弃。
- 如果 RXNEIE 位置 1 或 EIE 位置 1，则会生成中断。
- 通过设置 USART_ICR 寄存器中的 ORECF 位可将 ORE 位清零。

说明：ORE 位置 1 时表示至少 1 个数据丢失。存在两种可能：

- *如果 RXNE=1，则最后一个有效数据存储于接收寄存器 RDR 中并且可进行读取。*
- *如果 RXNE=0，则表示最后一个有效数据已被读取，因此 RDR 中没有要读取的数据。接收到新（和丢失）数据的同时已读取 RDR 中的最后一个有效数据时，会发生该情况。*

选择时钟源和合适的过采样方法

USART 在停机模式唤醒时支持双时钟域，时钟源可以是：PCLK（默认值）、HSI 或 SYSCLK。在低功耗模式下，USART 可以选择 HSI 作为时钟源来接收数据，从低功耗模式下唤醒的方式可以通过接收数据和唤醒模式的配置来实现。

接收器通过配置过采样方式，可从噪声中提取有效数据。这可在最大通信速度与抗噪声/时钟误差性能之间实现平衡。

可通过编程 USART_CR1 寄存器中的 OVER8 位来选择采样方式。采样时钟可以是波特率时钟的 16 倍或 8 倍。

- 选择 8 倍过采样（OVER8=1）以获得更高的传输速率（高达 $f_{CK}/8$），其中 f_{CK} 为时钟源频率。这种情况下，接收器对时钟偏差的最大容差将会降低（请参见“19.4.5 USART 接收器对时钟偏差的容差”）。
- 选择 16 倍过采样（OVER8=0）以增加接收器对时钟偏差的容差。这种情况下，最大传输速率限制为最高 $f_{CK}/16$。

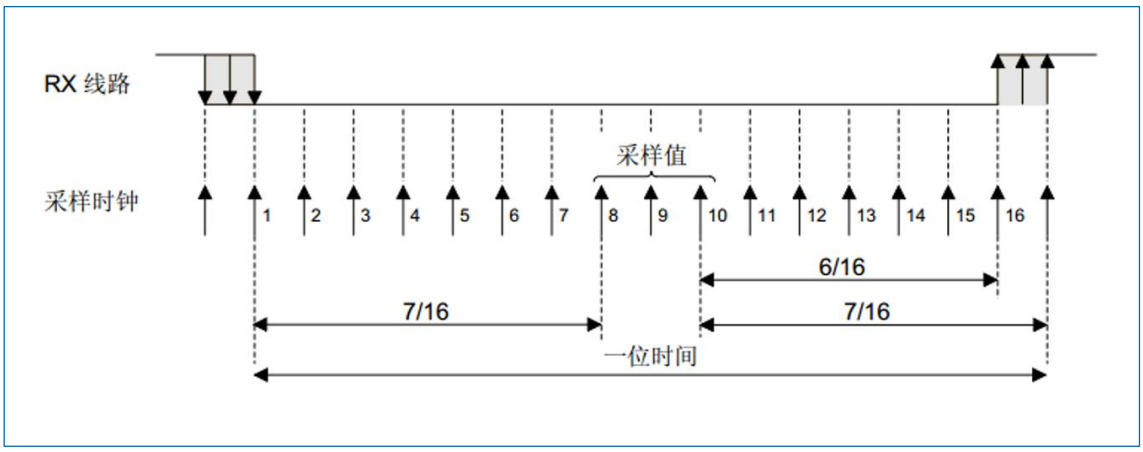


图 19-6 16 倍过采样时的数据采样

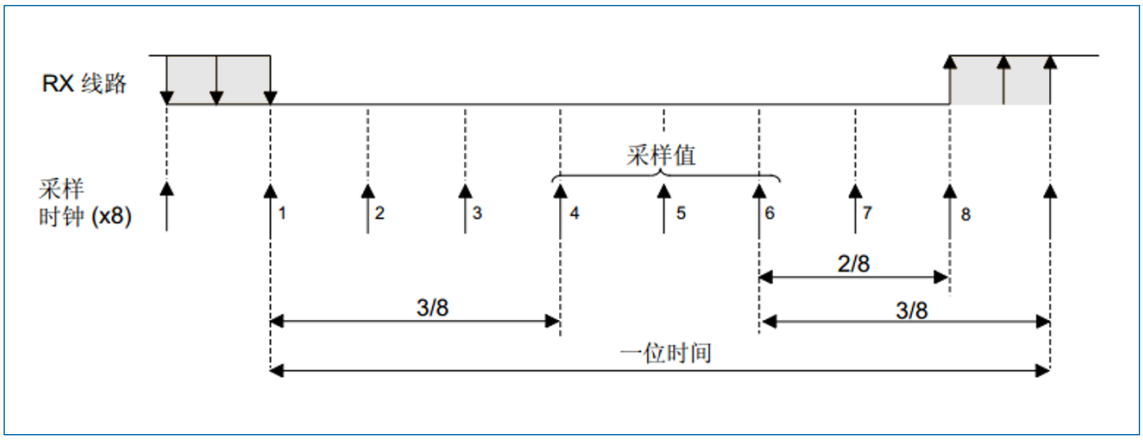


图 19-7 8 倍过采样时的数据采样

通过编程 USART_CR3 寄存器中的 ONEBIT 位选择逻辑电平的采样方式：

- 配置 ONEBIT=0 时，在已接收位的中心进行三次采样。在这种情况下，如果用于多数表决的 3 次采样结果不相等，则 NF 位被置 1。在噪声环境下工作时，选择三次采样的多数表决法。因为在检测到噪声时拒绝接收数据（表 19-2）可提高扛干扰能力。
- 配置 ONEBIT=1 时，在已接收位的中心进行单次采样。线路无噪声时请选择单次采样法（ONEBIT=1）以增加接收器对时钟偏差的容差（请参见“19.4.5 USART 接收器对时钟偏差的容差”）。

表 19-2 通过采样数据进行噪声检测

采样值	NE 状态	接收的位值
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

帧错误

如果接收数据时未在预期时间内识别出停止位，帧错误标志位会被置 1。

检测到帧错误时：

- FE 位由硬件置 1。
- 无效数据从移位寄存器传送到 USART_RDR 寄存器。
- 单字节通信时无中断产生。然而，在 RXNE 位产生中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时将发出中断。通过将 1 写入 USART_ICR 寄存器中的 FECF 位来复位 FE 位。

19.4.4 USART 波特率生成

配置 USART_BRR 寄存器的 BRR[15:0]位（其中 BRR[3:0]：USARTDIV 的小数部分，BRR[15:4]：USARTDIV 的整数部分）来生成接收器和发送器（RX 和 TX）的波特率。

通过配置 USART_CR1 中的 OVER8 位，来选择 16 倍过采样 8 倍过采样。

16 倍过采样（OVER8=0）时的波特率生成公式为：

$$\frac{Tx}{Rx}baud=\frac{f_{ck}}{UARTDIV} \quad (26\text{-}1)$$

8 倍过采样（OVER8=1）时的波特率生成公式为：

$$\frac{Tx}{Rx}baud=\frac{2*f_{ck}}{UARTDIV} \quad (26\text{-}2)$$

其中 USARTDIV 的值由 BRR 位获得。USART 的普通模式和同步模式均支持 16 倍过采样和 8 倍过采样，但 LIN 模式、IrDA 模式仅支持 16 倍过采样。

- 当 OVER8=0 时，BRR=USARTDIV。
- 当 OVER8=1 时：
  - BRR[2:0]=（USARTDIV[3:0]>>1）。
  - BRR[3]必须保持清零。
  - BRR[15:4]=USARTDIV[15:4]

说明：切勿在通信过程中修改波特率寄存器的值，16 倍或 8 倍过采样时，USARTDIV 必须大于或等于 0x10。

USART_BRR 寄存器中获取 USARTDIV 的示例如下：

- 根据公式 26-1，当采用 16 倍过采样时：
  - USARTDIV = $fck/(Tx/Rxbaud)$ = 8000 000/9600
  - BRR=USARTDIV=833D=0341h
- 根据公式 26-2，当采用 8 倍过采样时：
  - USARTDIV= $2*fck/(Tx/Rxbaud)$ = 2*8000 000/9600
  - USARTDIV=1666,66（1667D=683h）
  - BRR[3:0]=3h>>1=1h
  - BRR[15:0]=0x681

  示例 2：通过 f_{CK}=32MHz 获得 921.6K 波特率
- 根据公式 26-1，当采用 16 倍过采样时：
  - USARTDIV= $fck/(Tx/Rxbaud)$ =32000 000/921 600
  - BRR=USARTDIV=35D=23h
- 根据公式 26-2，当采用 8 倍过采样时：
  - USARTDIV=$2*fck/(Tx/Rxbaud)$ =2*32000 000/921 600
  - USARTDIV=70D=46h
  - BRR[3:0]=USARTDIV[3:0]>>1=6h>>1=3h
  - BRR[15:0]=0x43

表 19-3 采用 16 倍或 8 倍过采样时，在 f_{CK}=32MHz 下编程波特率的误差计算 (1)

波特率		16 倍过采样（OVER8=0）			8 倍过采样（OVER8=1）		
序号	所需值	实际值	BRR	误差%=（计算值-所需值）/所需波特率	实际值	BRR	误差%
1	2.4KBps	2.4KBps	0x3415	0	2.4KBps	0x6825	0

波特率		16 倍过采样（OVER8=0）			8 倍过采样（OVER8=1）		
2	9.6KBps	9.6KBps	0xD05	0	9.6KBps	0x1A05	0
3	19.2KBps	19.19KBps	0x683	0.02	19.2KBps	0xD02	0
4	38.4KBps	38.41KBps	0x341	0.04	38.39KBps	0x681	0.02
5	57.6KBps	57.55KBps	0x22C	0.08	57.6KBps	0x453	0
6	115.2KBps	115.1KBps	0x116	0.08	115.11KBps	0x226	0.08
7	230.4KBps	230.21KBps	0x8B	0.08	230.21KBps	0x113	0.08
8	460.8KBps	463.76KBps	0x045	0.64	460.06KBps	0x85	0.08
9	921.6KBps	914.28KBps	0x23	0.79	927.5KBps	0x42	0.79
10	2MBps	2MBps	0x10	0	2MBps	0x20	0
12	4MBps	4MBps	NA	NA	4MBps	0x10	0

(1). CPU 时钟越低，特定波特率的精度就越低。这些数据可用于确定可达到的波特率上限。

19.4.5 USART 接收器对时钟偏差的容差

当总时钟系统偏差小于 USART 接收器的容差时，USART 异步接收器才能正常工作。影响总偏差的因素包括：

- DTRA：发送器误差引起的偏差（其中还包括发送器本地振荡器的偏差）。
- DQUANT：接收器的波特率量化引起的误差。
- DREC：接收器本地振荡器的偏差。
- DTCL：传输线路引起的偏差（通常是由于收发器所引起，它可能会在低电平到高电平转换时序与高电平到低电平转换时序之间引入不对称）。

DTRA＋DQUANT＋DREC＋DTCL＋DWU < USART 接收器的容差

其中，DWU 是从停机模式唤醒时因采样点偏差产生的误差。当 M 位取值不同时，对应的 DWU 值为：

- M[1:0]=00 时：DWU= $\frac{t_{WUUSART}}{10*Tbit}$
- M[1:0]=01 时：DWU= $\frac{t_{WUUSART}}{11*Tbit}$
- M[1:0]=10 时：DWU= $\frac{t_{WUUSART}}{9*Tbit}$

$t_{WUUSART}$ 是从检测到唤醒事件到时钟（由外设请求）与调压器均就绪的时间。

USART 接收器在表 19-4 中指定的最大容许偏差下可正确接收数据，取决于以下选择：

- 由 USART_CR1 寄存器中的 M 位定义的 9 位、10 位或 11 位字符长度。
- 由 USART_CR1 寄存器中的 OVER8 位定义的 8 倍或 16 倍过采样。
- USART_BRR 寄存器的 BRR[3:0]位等于或不等于 0000。
- 使用 1 位或 3 位对数据进行采样，取决于 USART_CR3 寄存器中 ONEBIT 位的值。

表 19-4 BRR[3:0]=0000 时的 USART 接收器容差

M[1:0]位	OVER8 位=0		OVER8 位=1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
00	3.75%	4.375%	2.50%	3.75%
01	3.41%	3.97%	2.27%	3.41%
10	4.16%	4.86%	2.77%	4.16%

表 19-5 BRR[3:0]！=0000 时的 USART 接收器容差

M[1:0]位	OVER8 位=0		OVER8 位=1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
00	3.33%	3.88%	2%	3%
01	3.03%	3.53%	1.82%	2.73%
10	3.7%	4.31%	2.22%	3.33%

说明：当接收的帧恰好包含 10 个（M 位=00）、11 个（M 位=01）或 9 个（M 位=10）位持续时间的空闲帧时，表 19-4 和表 19-5 中指定的数据可能与特例中的数据略微不同。

19.4.6 USART 自动波特率检测

USART 可根据接收的字符检测对端设备的波特率并自动设置 USART_BRR 寄存器的值。

自动波特率检测适用于以下场景：

- 系统的通信速率未知。
- 系统使用精确度相对较低的时钟源。

时钟源频率必须与预期通信速度匹配（16 倍过采样时，波特率介于 $f_{CK}/65535$ 与 $f_{CK}/16$ 之间；8 倍过采样时，波特率介于 $f_{CK}/65535$ 与 $f_{CK}/8$ 之间）。

可通过配置 USART_CR2 寄存器中的 ABRMOD[1:0]字段选择自动波特率检测模式。

这些模式包括：

- 模式 0：以 1 位开头的任意字符。这种情况下，USART 会测量起始位的持续时间（下降沿到上升沿）。
- 模式 1：以 10xx 位模式开头的任意字符。这种情况下，USART 会测量起始位和第一个数据位的持续时间（下降沿到下降沿）。

激活自动波特率检测前，必须先通过向 USART_BRR 寄存器写入非零的波特率值来初始化该寄存器。

配置 USART_CR2 寄存器中的 ABREN 位来使能自动波特率检测功能。在使能该功能后，USART 将等待 RX 线路上的第一个字符。

自动波特率检测操作完成后，USART_ISR 寄存器中的 ABRF 标志位被置 1。

以下两种情况无法保证正确的波特率检测，可能导致 BRR 值损坏，即发生自动波特率错误（ABRE 错误标志位将置 1）：

- 线路受到噪声干扰。
- 通信速率不在自动波特率检测范围内（即 16 倍过采样时持续时间不在 16 个和 65536 个时钟周期之间，8 倍过采样时持续时间不在 8 个和 65536 个时钟周期之间）。

通过配置 ABRF 标志位（通过写入 0）重新使能自动波特率检测。

说明：如果在自动波特率操作期间禁止 USART（UE=0），则可能损坏 BRR 值。

19.4.7 使用 USART 进行多处理器通信

在多处理器通信中，以下位需要清零：

- USART_CR2 寄存器中的 LINEN 位。
- USART_CR3 寄存器中的 HDSEL、IREN 和 SCEN 位。

在多处理器通信中，将其中的一个 USART 作为主 USART，其 TX 输出与其它 USART 的 RX 输入相连接。其它的 USART 作为从 USART，其各自的 TX 在逻辑上通过与运算连接在一起，并且与主 USART 的 RX 输入相连接。

在多处理器配置中，一般情况下只有预期的消息接收方主动接收完整的消息内容，以减少由所有未被寻址的接收器造成的冗余 USART 服务开销。该功能可以通过将未被寻址的器件配置为静默模式来实现。将 USART_CR1 寄存器中的 MME 位置 1，可以进入静默模式。

在静默模式下：

- 接收状态位为 0。
- 禁止任何接收中断。
- USART_ISR 寄存器中的 RWU 位置 1。

配置 USART_CR1 寄存器中 WAKE 位，USART 可使用以下两种方法进入或退出静默模式：

- 若 WAKE 位=0，则进行空闲线路检测。
- 若 WAKE 位=1，则进行地址标记检测。

空闲线路检测（WAKE=0）

向 USART_RQR 寄存器的 MMRQ 位写入 1，USART 进入静默模式（RWU 位置 1）。

当检测到空闲帧时，USART 会被唤醒。此时 RWU 位会被硬件清零。空闲线路检测时静默模式如下图所示:

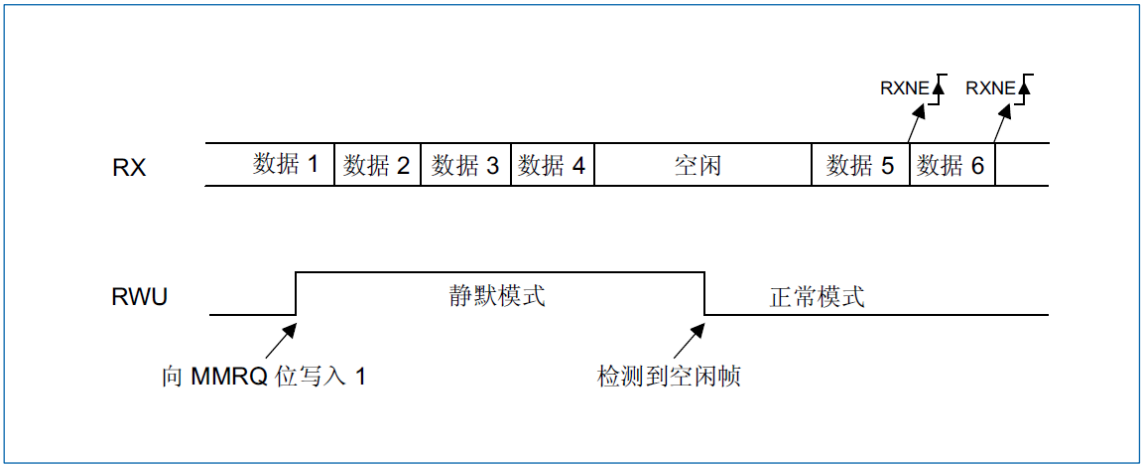


图 19-8 使用空闲线路检测时的静默模式

4 位/7 位地址标记检测（WAKE=1）

在此模式下，如果字节的 MSB 为 1，则将这些字节识别为地址，否则将其识别为数据。

在该模式下，MSB 为 1 的字节被识别为地址，否则被识别为数据。通过配置 USART_CR2 中的 ADDM7 位来选择 4 位/7 位的地址匹配方式，然后接收器会将此 4 位/7 位字与其地址进行比较。

当接收到与其编程地址不匹配的地址字符时，USART 会进入静默模式。此时，RWU 位将由硬件置 1。

USART 进入静默模式后，接收的地址字符不会置位 RXNE 标志。

当接收到与编程地址匹配的地址字符时，USART 会退出静默模式。此时，RWU 位被清零，USART 可开始正常接收后续字节。由于 RWU 位已清零，RXNE 位会针对地址字符置 1。

下图中给出了使用地址标记检测时静默模式行为的示例。

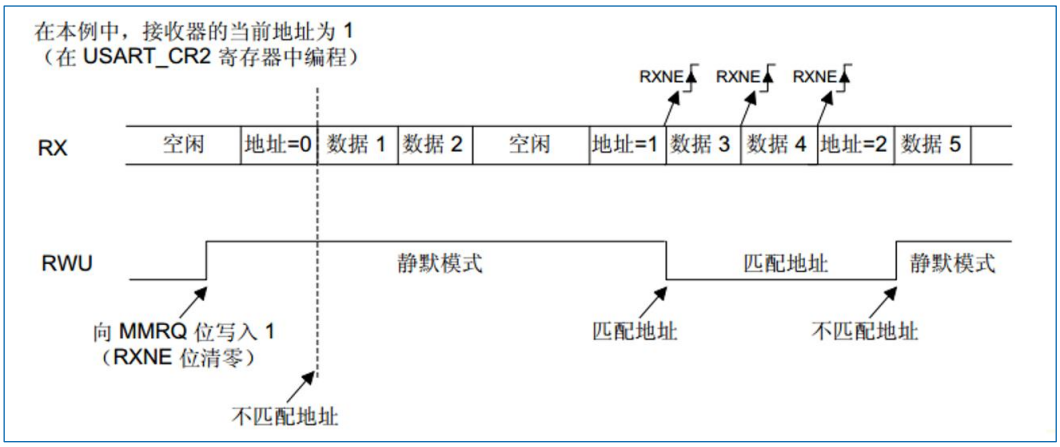


图 19-9 使用地址标记检测时的静默模式

说明：在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0]和 ADD[7:0]）。

19.4.8 使用 USART 进行 Modbus 通信

USART 为 Modbus/RTU 和 Modbus/ASCII 协议的实现提供基本支持。

Modbus/RTU

在此模式下，块结束标志通过超过 2 个字符时间的空闲线路来识别。此功能通过 USART 接收超时功能实现。

通过配置 USART_CR2 寄存器中的 RTOEN 位和 USART_CR1 寄存器中的 RTOIE 位来使能超时功能。在接收线路空闲过程中，完成最后一个停止位接收后，生成中断，并通知软件当前块接收已完成。

Modbus/ASCII

在此模式下，块结束通过特定字符（CR/LF）序列识别。USART 通过字符匹配功能管理此机制。

通过配置 USART_CR2 中的 ADD[7:0]字段中编程特定字符 LF 对应的 ASCII 码以及使能字符匹配中断（CMIE=1），当接收到 LF 字符时，将产生中断，并且通知并检查接收缓存区中的 CR/LF。

19.4.9 USART 奇偶校验

将 USART_CR1 寄存器中的 PCE 位置 1，可以使能奇偶校验控制（发送时生成奇偶校验位，接收时进行奇偶校验检查）。根据 M 位定义的帧长度，下表列出了可能的 USART 帧格式。

表 19-6 帧格式

M[1:0]位	PCE 位	USART 帧 (1)
00	0	|SB|8 位数据|STB|
00	1	|SB|7 位数据|PB|STB|
01	0	|SB|9 位数据|STB|
01	1	|SB|8 位数据|PB|STB|

M[1:0]位	PCE 位	USART 帧 (1)
10	0	|SB|7 位数据|STB|
10	1	|SB|6 位数据|PB|STB|

(1). SB：起始位，STB：停止位，P：奇偶校验位。在数据寄存器中，PB 始终位于 MSB 位置（第 7 位、第 8 位或第 9 位，具体取决于 M 位的值）。

偶校验

计算奇偶校验位，使数据帧和奇偶校验位中“1”的数量为偶数。例如，如果数据=00110101，其中 4 个数据位被置 1，在选择偶校验（USART_CR1 寄存器中的 PS 位=0）时，校验位则为 0。

奇校验

计算奇偶校验位，使数据帧和奇偶校验位中“1”的数量为奇数。例如，如果数据=00110101，其中 4 个数据位被置 1，在选择奇校验（USART_CR1 寄存器中的 PS 位=1）时，校验则位为 1。

接收时进行奇偶校验检查

如果奇偶校验检查失败，则 USART_ISR 寄存器中的 PE 标志被置 1；如果 USART_CR1 寄存器中 PEIE 位被置 1，则会生成中断。通过软件置位 USART_ICR 寄存器中的 PECF 位，可清零 PE 标志位。

发送时的奇偶校验生成

如果 USART_CR1 寄存器中的 PCE 位被置 1，则写入数据寄存器（USART_TDR）中的数据的 MSB 位会被奇偶校验位更改，如果选择偶校验（PS=0），则“1”的数量为偶数；如果选择奇校验（PS=1），则“1”的数量为奇数。

19.4.10 USART LIN（局域互连网络）模式

通过配置 USART_CR2 寄存器中的 LINEN 位来选择 LIN 模式。在 LIN 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 STOP[1:0]和 CLKEN 位。
- USART_CR3 寄存器中的 SCEN、HDSEL 和 IREN 位。

LIN 发送

与正常的 USART 发送相比，在 LIN 主器件中发送时必须采用“19.4.2 USART 发送器”中介绍的步骤，同时还具有以下区别：

- 数据字长度必须为 8 位。
- LINEN 位置 1 以进入 LIN 模式。此时，将 SBKRQ 位置 1，USART 会发送 13 个 bit“0”作为中断帧。再发送 2 个 bit“1”以进行下一次启动检测。

LIN 接收

使能 LIN 模式后，将激活中断帧检测电路。该检测完全独立于正常的 USART 接收器。

使能接收器（USART_CR1 寄存器中 RE=1）后，电路便开始监测启动信号的 RX 输入。检测到起始位后，电路会对接下来的位进行采样，方法与数据采样相同（第 8、第 9 和第 10 次采样）。如果连续 10 个（USART_CR2 中 LBDL=0 时）或 11 个（USART_CR2 中 LBDL=1 时）连续位均检测为“0”，且其后跟随分隔符，则 USART_ISR 寄存器中的 LBDF 标志将置 1。如果 LBDIE 位=1，则会生成中断。在验证中断帧前，会对分隔符进行检查，分隔符表示 RX 线路已恢复到高电平。如果在第 10 或第 11 次采样前已对“1”采样，则中断帧检测电路会取消当前检测，并重新搜索起始位。

在使能 LIN 模式（LINEN=1）后，只要发生帧错误，接收器即会立即停止，直到中断帧检测电路接收到“1”（中断帧字不完整时）或接收到分隔符（检测到断路时）。

图 19-10 中显示了中断帧检测器状态机和断路标志的行为。

图 19-11 中列出了中断帧的示例。

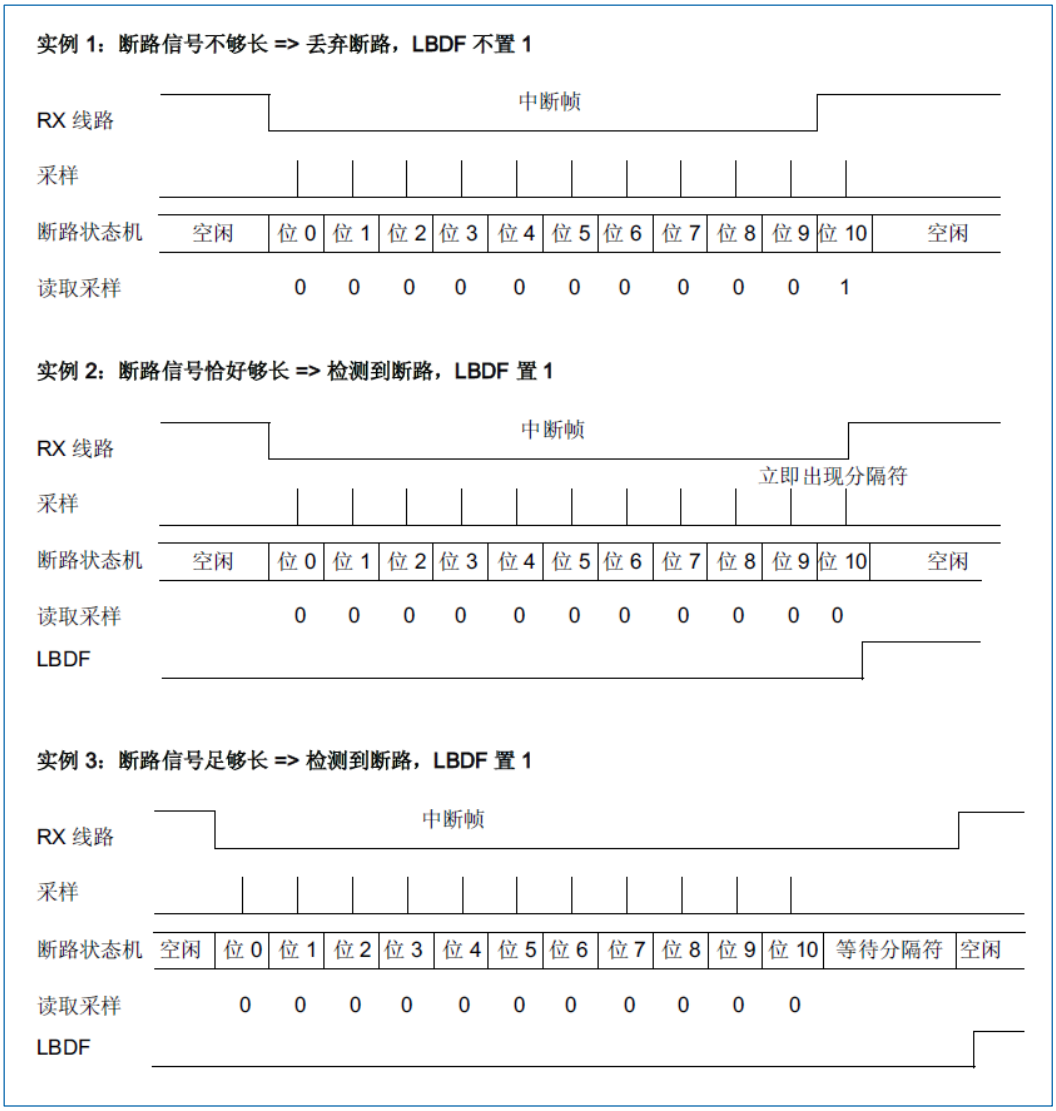


图 19-10 LIN 模式下的断路检测（11 位断路长度——LBDL 位置 1）

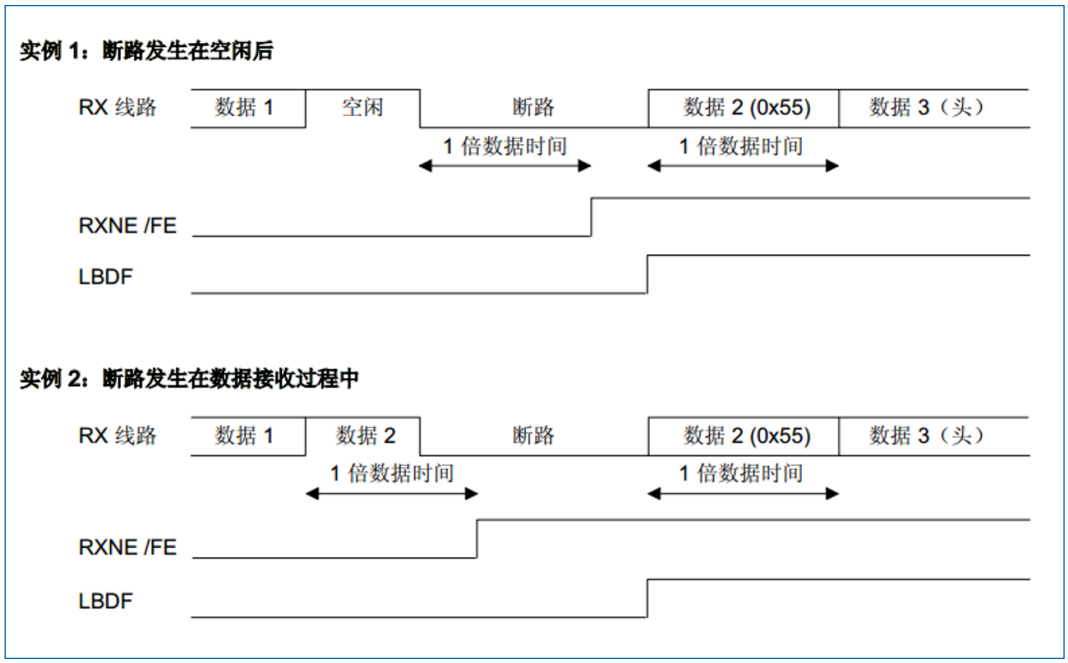


图 19-11 LIN 模式下的断路检测与帧错误检测

19.4.11 USART 同步模式

配置 USART_CR2 寄存器中的 CLKEN 位来选择同步模式。

在同步模式下，需要将以下位清零：

- USART_CR2 寄存器中的 LINEN 位。
- USART_CR3 寄存器中的 SCEN、HDSEL 和 IREN 位。

同步模式包括了 TX 引脚、RX 引脚以及 CK 引脚，可以在主模式下完成双向同步串行通信。CK 引脚是 USART 的时钟输出引脚，只有在数据帧期间发送时钟脉冲。通过配置 USART_CR2 寄存器中 LBCL 位来控制最后一个数据位是否生成时钟脉冲。同时也可以通过配置 USART_CR2 寄存器中的 CPOL 位和 CPHA 位来选择时钟极性和时钟的相位（请参见图 19-12、图 19-13 和图 19-14）。

USART 发送器在同步模式下的工作方式与异步模式下完全相同。但是由于 CK 与 TX 同步（根据 CPOL 和 CPHA），因此 TX 上的数据是同步的。

USART 接收器在同步模式下的工作方式与异步模式下不同。如果 RE=1，则数据在 CK 上采样（上升沿或下降沿，取决于 CPOL 和 CPHA），而不会进行任何过采样。此时需要确保建立时间和保持时间（取决于波特率：1/16 位持续时间）。

说明：CK 引脚需与 TX 引脚结合使用。只有在发送数据的时候才会产生 CK 时钟脉冲。

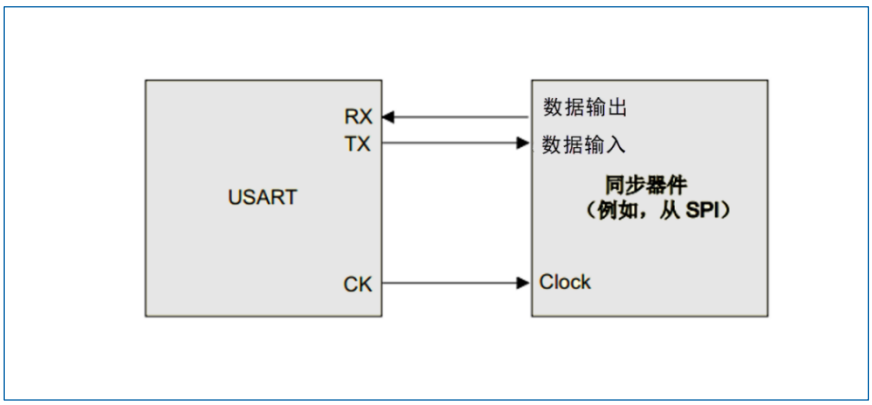


图 19-12 USART 同步发送示例

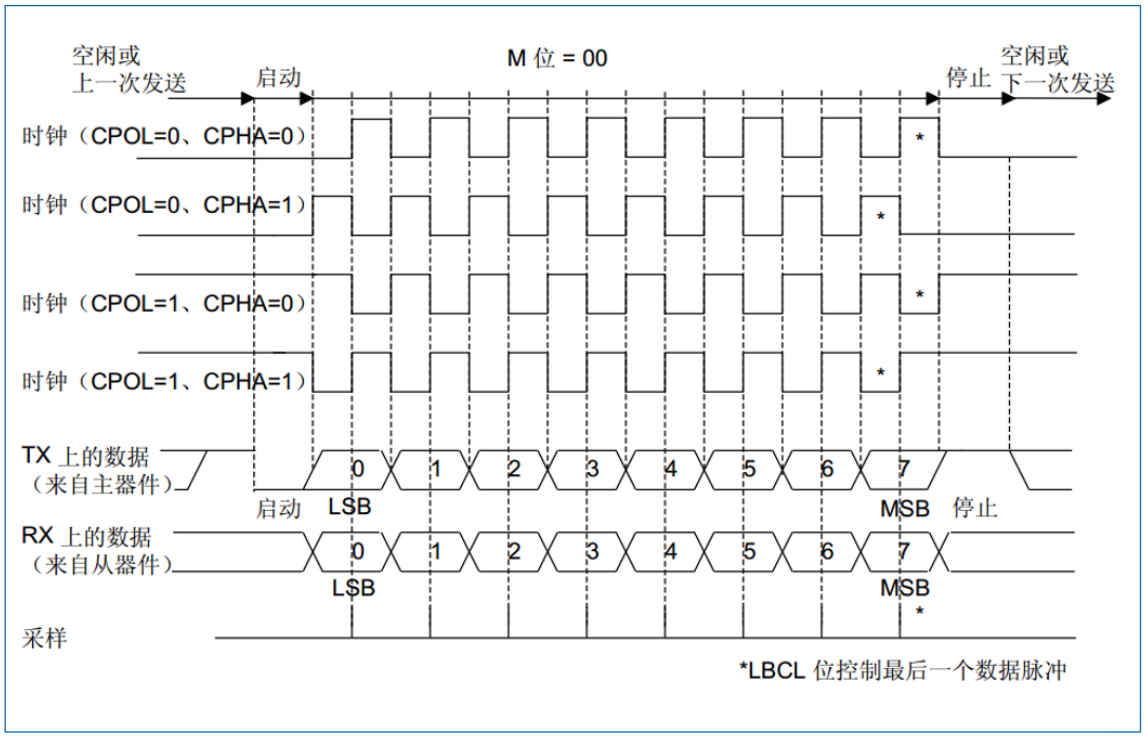


图 19-13 USART 数据时钟时序图（M 位=00）

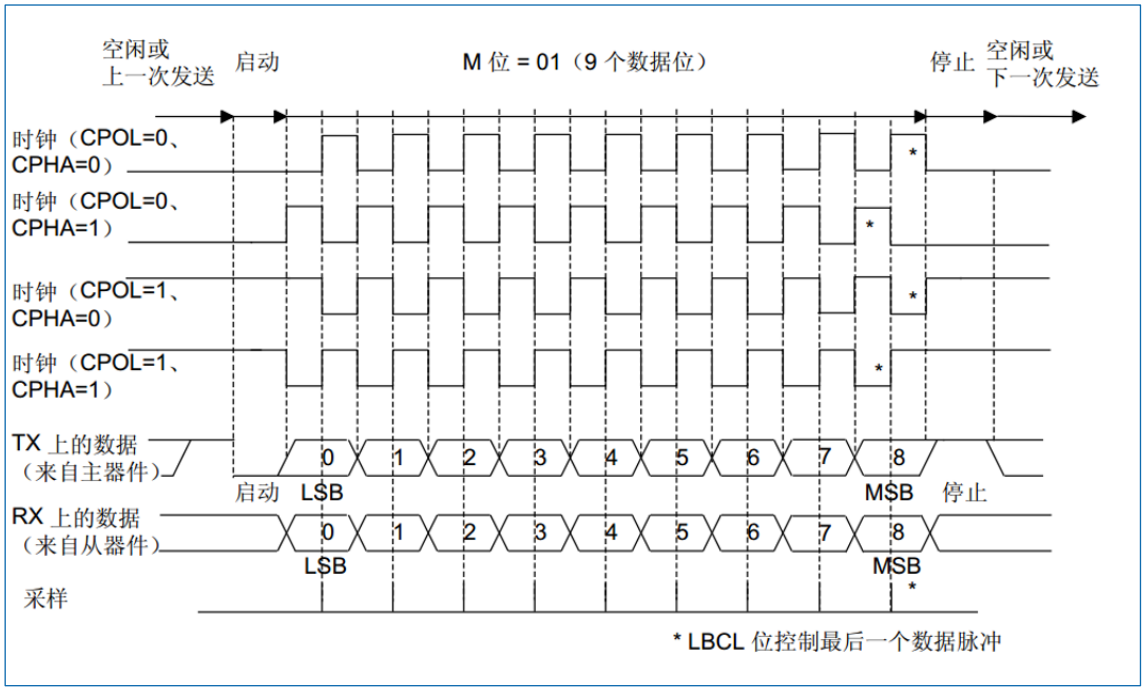


图 19-14 USART 数据时钟时序图（M 位=01）

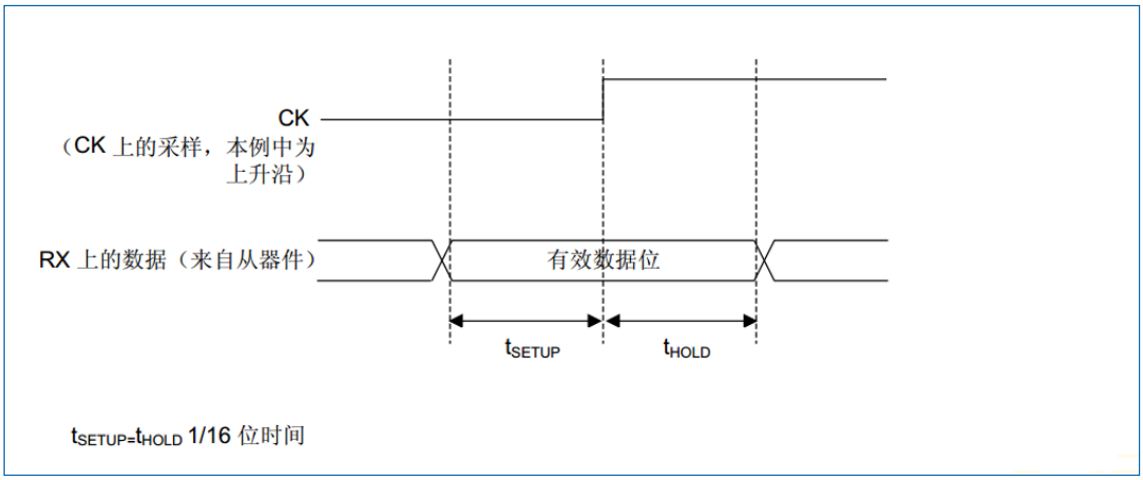


图 19-15 RX 数据建立/保持时间

说明：在智能卡模式下，CK 的功能跟上述描述有所不同。更多详细信息，请参见"19.4.13 USART 智能卡模式"。

19.4.12 USART 单线半双工通信

通过将 USART_CR3 寄存器中的 HDSEL 位置 1 来选择单线半双工模式。在此模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 位和 CLKEN 位。
- USART_CR3 寄存器中的 SCEN 和 IREN 位。

USART 支持单线半双工协议，其中 TX 和 RX 线路从内部相连接。一旦 USART_CR3 寄存器中的 HDSEL 位置 1：

- TX 和 RX 线路从内部相连接。
- RX 引脚被禁用，通信双方通过 TX 连接在一起。
- TX 在空闲状态或接收过程中用作标志 I/O。在单线半双工模式下，TX 需配置为复用功能开漏并外接上拉电阻。除此之外，通信协议与正常 USART 模式下的通信协议相似。此线路上的任何冲突必须由软件管理。

19.4.13 USART 智能卡模式

通过将 USART_CR3 寄存器中的 SCEN 位置 1 选择智能卡模式。

在智能卡模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 位。
- USART_CR3 寄存器中的 HDSEL 和 IREN 位。

此外，需配置 USART_CR2 寄存器中的 CLKEN 位来为智能卡提供时钟。

USART 的智能卡模式支持 ISO 7816-3 标准的异步协议。同时支持 T=0（字符模式）和 T=1（块模式）。

通过配置以下寄存器来实现智能卡模式：

- 8 个位加奇偶校验：USART_CR1 寄存器的 PCE 位=1。
- 1.5 个停止位：USART_CR2 寄存器中 STOP[0:1]=11。在接收时也可以配置成 0.5 个停止位。

在 T=0（字符模式）下，奇偶校验错误会在每个字符结束时指示。

下图为有无奇偶校验错误时，数据线上情况的示例。

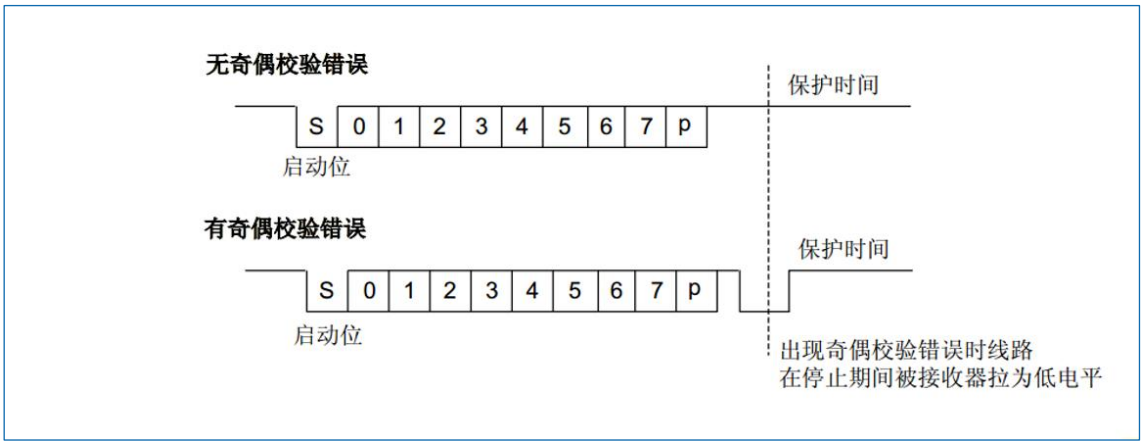


图 19-16 ISO 7816-3 异步协议

在智能卡模式下，TX 引脚需要配置为开漏模式。

智能卡模式采用单线半双工通信协议。

- 从发送移位寄存器传输的数据被延迟至少 1/2 波特率时钟。正常工作时，装有数据的发送移位寄存器会在下一个波特时钟边沿开始移位。在智能卡模式下，这种传输方式更加有效的保证了 1/2 波特率时钟延迟。
- 在智能卡模式下，若发送时检测到奇偶校验错误，则会通过将线路驱动为低电平（NACK）。该信号会导致发送器端出现帧错误，与此同时，USART 会根据协议重新发送数据。若多次发送依然收到 NACK，则 USART 会停止发送。
- 发送时智能卡自动重试：USART 需要检测 NACK 与重复字符的起始位之间插入 2.5 个波特率周期的延迟，若需要重复此操作，需要确保最短 2 个波特率周期。在接收到最后一个重复字符时，TC 位会被置 1。
- 在接收一个 1.5 个停止位的数据帧期间检测到奇偶校验错误，发送线则会在接收数据帧后拉低一个时钟周期（这是为了通知智能卡：发送给 USART 的数据尚未正确接收）。若配置寄存器 USART_CR3 中 NACK 位，则接收器会在奇偶校验错误的情况下发送 NACK 信号（T=1 模式下会使用）。根据智能卡协议规范，若接收到的字符错误，智能卡需重新发送相同的字符，通过配置 SCARCNT 位来指定的最大重试次数后接收到的字符仍然错误，则 USART 停止发送 NACK 信号，并以奇偶校验错误的形式发送出去。
- 接收时智能卡自动重试：若 USART 向智能卡发送 NACK 信号，则智能卡不能重复字符，同时 BUSY 标志位将保持为”1”。
- 在发送时，USART 会在两个连续字符之间插入保护时间（通过配置 USART_GTPR 寄存器中的 GT[7:0]）。通过对保护时间寄存器进行编程，可以延迟 TC 标志的置位。在智能卡模式下，空的发送移位寄存器会触发保护时间计数器，使其向上计数至保护时间寄存器中的值。在此期间，TC 标志被强制为低电平。当保护时间计数器达到设置值时，TC 会被置位。
- TC 标志的释放不受智能卡模式的影响。
- 在接收端，如果检测到奇偶校验错误并发送了 NACK 信号，则接收端不会将 NACK 作为起始位进行检测。

说明：中断字符在智能卡模式下无效。当翻转 TE 位时，不会发送空闲帧。空闲帧（在其它配置中进行了定义）在 ISO 协议中未进行定义。

下图详细介绍了 USART 如何对 NACK 信号采样：

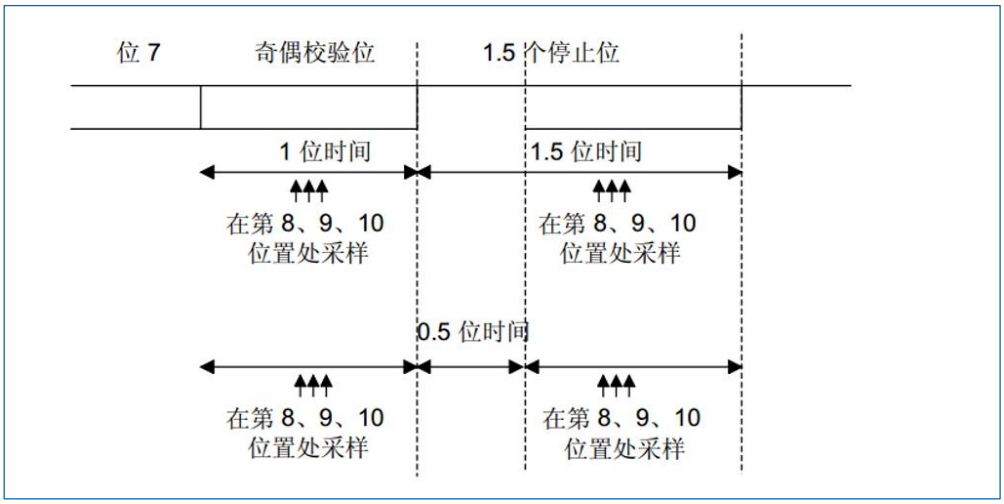


图 19-17 使用 1.5 个停止位检测奇偶校验错误

在智能卡模式下，需要 USART 提供 CK 时钟信号，CK 可以通过一个 5 位预分频器由内部外设输入时钟提供，并且可通过 USART_GTPR 寄存器来配置预分频。CK 频率可在 $f_{CK}/2$ 到 $f_{CK}/62$ 之间进行编程，其中 f_{CK} 为外设输入时钟。

在 T=1（块）模式下，通过将 USART_CR3 寄存器中的 NACK 位清零可停止奇偶校验错误发送。

在块模式下，从智能卡请求读操作时，需要配置 USART_CR2 寄存器中的 RTOEN 位来使能接收器超时，并且将 USART_RTOR 寄存器中的 RTO 设置为块等待时间（BWT）。若未收到智能卡的应答，RTOF 标志位被置 1，若配置了 USART_CR1 寄存器中的 RTOIE 位，则会产生中断。若收到第一个字符，则通过 RXNE 中断发出信号指示。

说明：RTO 计数器遵循以下规则开始计数：

- STOP=00 时，从停止位结束时开始计数。
- STOP=10 时，从第二个停止位结束时开始计数。
- STOP=11 时，从 STOP 位结束后开始计数 1 个位的持续时间。
- STOP=01 时，从 STOP 位开始时开始计数。

正向约定和反向约定

智能卡协议定义了两种约定：正向约定和反向约定。

正向约定：LSB 在前，逻辑位的值 1 对应于信号线路的 H 状态，奇偶校验为偶校验。若使用此约定，必须配置以下控制位：MSBFIRST=0，DATAINV=0（默认值）。

反向约定：MSB 在前，逻辑位的值 1 对应于信号线路的 L 状态，奇偶校验为偶校验。若使用此约定，必须配置以下控制位：MSBFIRST=1，DATAINV=1。

说明：将逻辑数据值取反（0=H，1=L）时，奇偶校验位将同样取反。

为识别智能卡约定，智能卡会将初始字符 TS 作为 ATR（复位应答）的第一个字符发送。TS 支持两种格式：LH HLLL LLLH 和 LH HLHH HLLH。

- （H）LH HLLL LLLH 建立反向约定：状态 L 编码为值 1，时间分量 2 传送最高有效位（MSB 在前）。按反向约定解码时，传送的字节等于“3F”。
- （H）LH HLHH HLLH 建立正向约定：状态 H 编码为值 1，时间分量 2 传送最低有效位（LSB 在前）。按正向约定解码时，传送的字节等于“3B”。

在 2 到 10 的九个时间分量中，如果有偶数个位设置为 1，则字符的奇偶校验正确。

USART 拥有一种可识别任意一种模式的相应操作，同时模式识别通过软件序列完成。

因此，有以下两种方法可用于识别 TS 模式：

方法 1

将 USART 配置为标准智能卡模式/正向约定。这种情况下，TS 模式接收会产生一个奇偶校验错误中断和错误信号到智能卡。

- 奇偶校验错误中断告知软件，智能卡未以正向约定正确应答。
- 为了响应错误信号，智能卡会重试同一 TS 字符。此时重新编程后的 USART 将正确接收该字符。为了应答奇偶校验错误中断，可重新编程 USART，同时向智能卡发送新的复位命令，并且等待 TS。

方法 2

将 USART 编程为 9 位/无奇偶校验模式，无位反向。在此模式下，按如下方式接收两种 TS 模式中的任一种：

- （H）LH HLLL LLLH=0x103->选择反向约定
- （H）LH HLHH HLLH=0x13B->选择正向约定

根据这两种模式检查接收到的字符，如果其中任意一种匹配，则编程相应 USART 以接收下一个字符。

如果两种都未被识别，则可能复位智能卡以重新开始协商。

19.4.14 USART IrDA SIR ENDEC 模块

配置 USART_CR3 寄存器中的 IREN 位来选择 IrDA 模式。

在 IrDA 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN、STOP 和 CLKEN 位。
- USART_CR3 寄存器中的 SCEN 和 HDSEL 位。

在 IrDA 模式下，USART_CR2 寄存器中的 STOP 位必须配置为 1 个停止位。

IrDA SIR 物理层使用反相归零（RZI）调制方案，它以红外光脉冲表示逻辑 0（参见图 19-8）。

串行红外发送编码器将标准 USART 模块输出的非归零码（NRZ）的串口数据调制为 RZI 的红外数据，然后通过 TX 引脚输出到外部。USART 串口红外最高可支持 115.2 Kbps 波特率。串行红外接收解码器将从 RX 引脚接收的 RZI 红外数据解调为 NRZ 串口数据，然后输出给标准 USART 模块。

在正常模式下，USART 所发送的脉冲宽度为一个位周期的 3/16。IrDA 规范要求脉冲宽度需要大于 1.41us。接收器端的干扰检测逻辑会滤除宽度小于 2 个 PSC 周期的脉冲（PSC 是在 USART_GTPR 中编程的预分频器值 PSC[7:0]）。宽度小于 1 个 PSC 周期的脉冲都将被拒绝，但宽度大于 1 个而小于 2 个周期的脉冲可能被接受也可能被拒绝，而宽度大于 2 个周期的脉冲将被接受作为有效脉冲。当 PSC=0 时，IrDA 编码器/解码器不工作。

在低功耗模式下，脉冲宽度不再保持为位周期的 3/16。此时的脉冲宽度为低功耗波特率的 3 倍，最小可为 1.42 MHz。通常此值是 1.8432 MHz（1.42 MHz<PSC<2.12 MHz）。接收时与正常模式下的工作方式一致。

红外是一个半双工通信协议。当发送数据时，即标准 USART 模块发送数据给编码器，解码器会忽略 RX 引脚上的数据。当接收数据时，即标准 USART 模块接收来自解码器的数据，编码器不会编码标准 USART 模块发送的数据。

说明：宽度小于两个但大于一个 PSC 周期的脉冲可能被接受，也可能被拒绝。

接收器的建立时间应由软件进行管理。IrDA 物理层规范规定发送和接收之间至少要经过 10 ms 的延迟（IrDA 是一个半双工协议）。

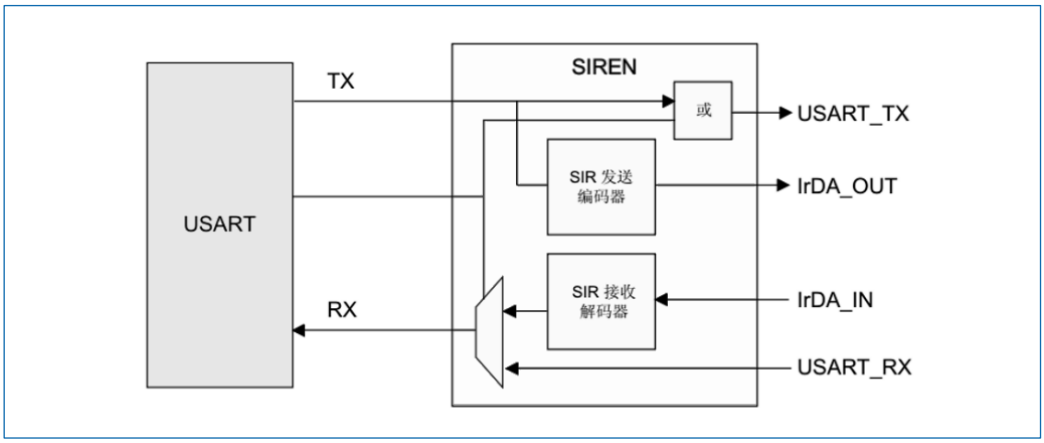


图 19-18 IrDA SIR ENDEC 框图

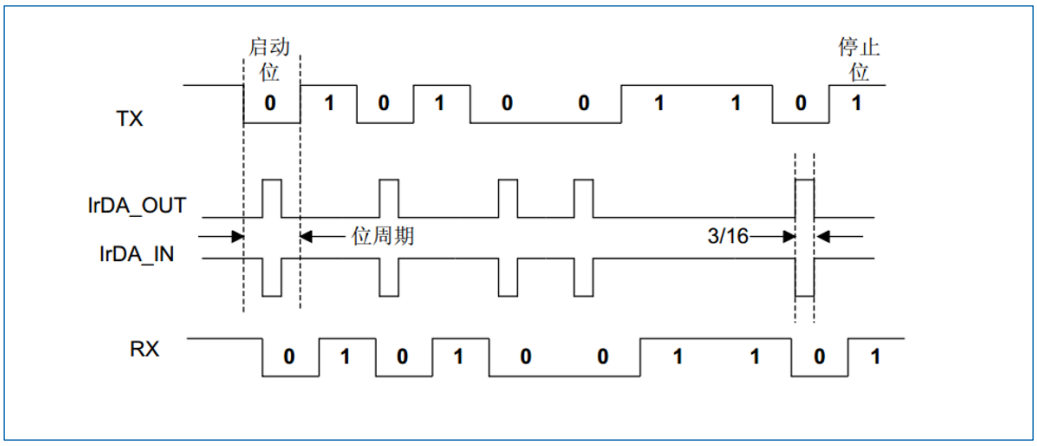


图 19-19 IrDA 数据调制（3/16）正常模式

19.4.15 RS485 驱动器使能

通过将 USART_CR3 控制寄存器中的 DEM 位置 1，可使能 DE 驱动器。可通过 DE（驱动器使能）信号激活外部收发器控制。

有效时间是指从激活 DE 信号与 START 位开始之间的时间，可以通过 USART_CR1 控制寄存器中的 DEAT[4:0]位域来编程。

禁止时间是从完成停止位发送至取消激活 DE 信号之间的时间，可以通过 USART_CR1 控制寄存器中的 DEDT[4:0]位域来编程。

DE 信号的极性可通过 USART_CR3 控制寄存器中的 DEP 位配置。

DEAT 和 DEDT 以采样时间单位表示（1/8 持续时间对应 8 倍过采样，1/16 位持续时间对应 16 倍过采样）。

19.4.16 从停机模式唤醒

当 USART_CR1 寄存器中 UESM 位置 1 且 USART 时钟设置为 HSI 时，USART 能够将 MCU 从停机模式唤醒。

可使用标准 RXNE 中断将 MCU 从停机模式唤醒（必须在进入停机模式前将 RXNEIE 位置 1）。

也可以通过配置 USART_CR3 寄存器中的 WUS[1:0]域来选择唤醒方式，并且在检测到唤醒事件后，WUF 标志会被置 1（与 MCU 处于停机模式还是工作模式无关），若配置了 WUFIE 位，则产生中断。

若在初始化和使能接收器后立即进入停机模式，则必须校验 USART_ISR 寄存器的 REACK 位是否被置位，用于确认 USART 是否已经准备好接收数据。

在进入停机模式前，用户必须确保 USART 未在执行传输。

若 USART 在进入停机模式前处于静默模式：只能使用地址匹配唤醒。

从停机模式唤醒功能并非在所有模式下均可用。例如，该功能在同步模式下不起作用。

允许从停机模式正确唤醒的最大波特率为 1/23.31 μs=42K 波特率。

19.5 USART 低功耗模式

表 19-7 低功耗模式对 USART 的影响

模式	说明
睡眠模式	无影响。USART 中断可使 MCU 退出睡眠模式。
低功耗睡眠模式	无影响。USART 中断可使 MCU 退出低功耗睡眠模式。
停机模式	当 UESM 位置 1 且 USART 时钟设置为 HSI 或 LSE 时，USART 能够将 MCU 从停机模式唤醒。

19.6 USART 中断

表 19-8 USART 中断请求

中断事件	事件标志位	使能控制位
发送数据寄存器为空	TXE	TXEIE
发送完成	TC	TCIE
接收数据寄存器非空	RXNE	RXNEIE
检测到空闲线路	IDLE	IDLEIE
奇偶校验错误	PE	PEIE
LIN 断路	LBDF	LBDIE
噪声错误、上溢错误和帧错误	NF 或 ORE 或 FE	EIE
字符匹配	CMF	CMIE
接收器超时	RTOF	RTOIE
块结束	EOBF	EOBIE
从停机模式唤醒	WUF (1)	WUFIE

(1). WUF 中断仅在停机模式下有效。

USART 中断映射图（请参见图 19-20）。

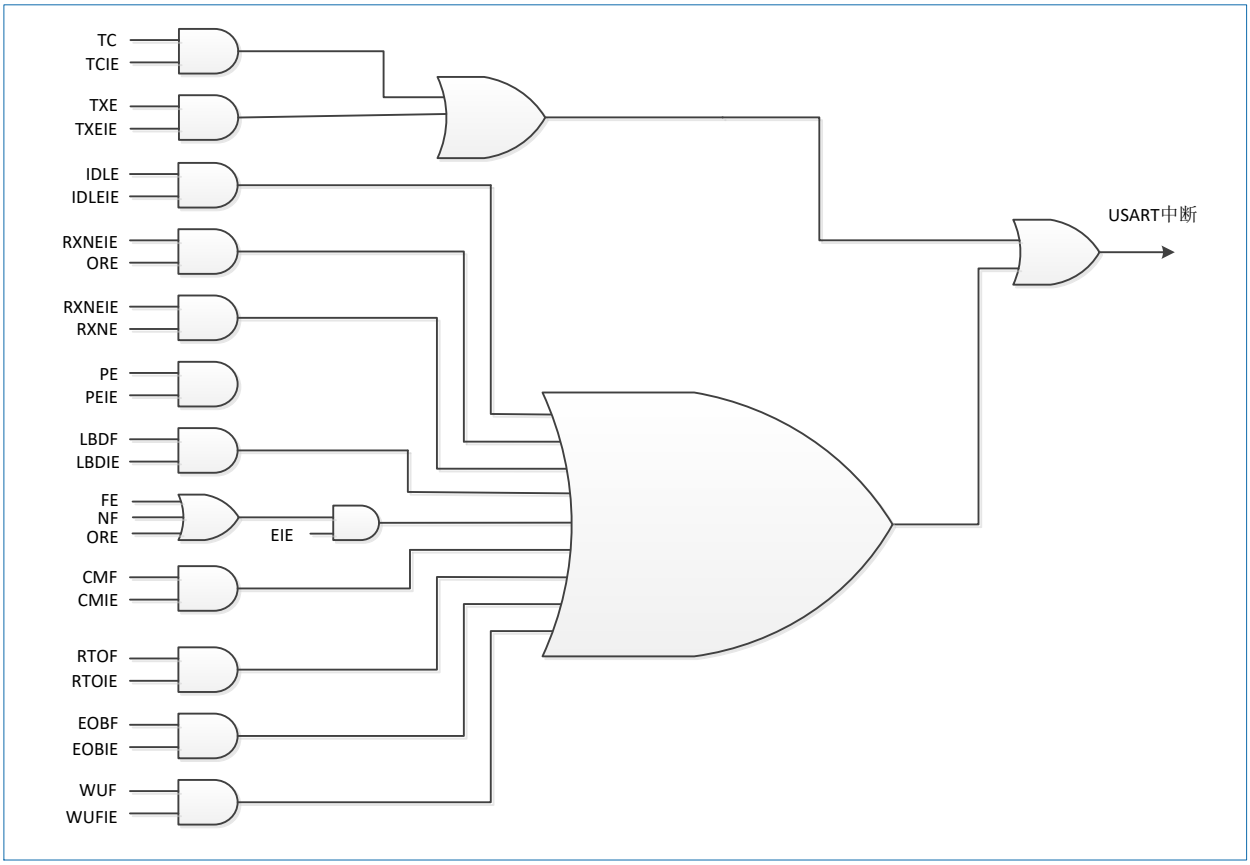


图 19-20 USART 中断映射图

19.7 USART 寄存器

基地址：0x4001 3800

空间大小：0x400

19.7.1 控制寄存器 1（USART_CR1）

地址偏移：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res			M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]				
			rw	rw	rw	rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29	Res：保留 必须保持复位值。
位 28	M1：字长 该位和位 12（M0）配合使用。 • M[1:0]=00：1 个起始位，8 个数据位，n 个停止位 • M[1:0]=01：1 个起始位，9 个数据位，n 个停止位 • M[1:0]=10：1 个起始位，7 个数据位，n 个停止位

	说明：M[1:0]中，位 M[1]=M1，位 M[0]=M0。
位 27	EOBIE：块尾中断使能 该位由软件置 1 和清零。 • 0：中断禁用 • 1：当 USART_ISR 寄存器中的 EOBF 标志被置 1 时，产生 USART 中断。
位 26	RTOIE：收超时中断使能 该位由软件置 1 和清零。 • 0：中断禁用 • 1：当 USART_ISR 寄存器中的 RTOF 标志被置 1 时，产生 USART 中断。
位 25:21	DEAT[4:0]：驱动使能提前时间 该位域定义 DE（驱动器使能）信号激活和第一个发送字节的起始位的时间间隔。它以采样时间为单位（1/8 或者 1/16 位时间，由过采样率决定）。 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 20:16	DEDT[4:0]：驱动使能滞后时间 该位域是一个发送消息的最后一个字节的停止位和释放 DE 信号之间的时间间隔。它以采样时间为单位（1/8 或 1/16 位时间，由过采样率决定）。 如果 USART_TDR 寄存器在 DEDT 时间内被改写，新的数据则会等待至 DEDT 和 DEAT 时间结束后才会被发送。 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 15	OVER8：过采样模式 • 0：16 倍过采样 • 1：8 倍过采样 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 14	CMIE：字符匹配中断使能 该位由软件置 1 和清零。 • 0：中断禁用 • 1：当 USART_ISR 寄存器中的 CMF 标志被置 1 时，引发 USART 中断。
位 13	MME：静默模式使能 该位开启 USART 的静默模式功能。当该位置为 1 时，USART 可以在活动模式和静默模式之间切换。该位由软件置 1 和清零。 • 0：接收器长期处于活动模式。 • 1：接收器可以在活动模式和静默模式间切换。
位 12	M0：字长 该位决定串口字长。 该位由软件置 1 和清零。

	• 0：1 个起始位，8 位数据位，n 个停止位 • 1：1 个起始位，9 个数据位，n 个停止位 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 11	WAKE：接收器唤醒方式 该位决定 USART 从静默模式唤醒的方式。 该位由软件置 1 和清零。 • 0：空闲线 • 1：地址标记 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 10	PCE：校验控制使能 该位选择硬件校验控制（产生和检测）功能。 当校验控制开启，计算完成的校验位被插入到最高位（M=1 时是第九位，M=0 时是第八位），并检测接收数据的校验位。 该位由软件置 1 和清零。一旦该位被置 1，在当前字节之后就激活了校验控制（数据收发时都校验）。 • 0：校验控制禁止 • 1：校验控制使能 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 9	PS：校验选择 该位选择在校验生成和检测功能被打开的时候（PCE=1）使用奇校验还是偶校验。该位由软件置 1 和清零。校验方式会在当前字节结束后生效。 • 0：偶校验 • 1：奇校验 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 8	PEIE：校验错误中断使能 该位由软件置 1 和清零。 • 0：中断禁用 • 1：在 USART_ISR 寄存器中的 PE 被置 1 时，会产生 USART 中断。
位 7	TXEIE：发送寄存器空中断使能 该位由软件置 1 和清零。 • 0：中断禁用 • 1：在 USART_ISR 寄存器中的 TXE 被置 1 的时候会产生 USART 中断。
位 6	TCIE：发送完毕中断使能 该位由软件置 1 和清零。 • 0：中断禁用 • 1：在 USART_ISR 寄存器中的 TC 位被置 1 时，会产生 USART 中断。

位 5	RXNEIE：接收寄存器非空中断使能 该位由软件置 1 和清零。 • 0：中断禁用 • 1：在 USART_ISR 寄存器中的 ORE 或者 RXNE 被置 1 时，会产生 USART 中断。
位 4	IDLEIE：空闲中断使能 该位由软件置 1 和清零。 • 0：中断禁用 • 1：在 USART_ISR 寄存器中的 IDLE 位被置 1 时，会产生 USART 中断。
位 3	TE：发送器使能 这个位打开发送器。 该位由软件置 1 和清零。 • 0：发送器关闭 • 1：发送器打开 *说明：当 TE 被置为 1 后，它和发送开始之间有 1 个位的延迟时间。*
位 2	RE：接收器使能 这个位打开接收器。 该位由软件置 1 和清零。 • 0：接收器被关闭。 • 1：接收器被打开并开始等待起始位。
位 1	UESM：USART 在 Stop 模式下使能 该位由软件置 1 和清零。 • 0：USART 不能从 Stop 模式中唤醒 MCU。 当这个位为零，USART 不能将 MCU 从 Stop 模式下唤醒。当该位为 1 时，USART 能将 MCU 从 Stop 模式唤醒，条件是 USART 的时钟选择位 HSI。 • 1：USART 可以从 Stop 模式中唤醒 MCU。 当这个功能开启时，USART 的时钟源必须为 HSI。
位 0	UE：USART 使能 当这个位被清零，USART 的预分频器和输出都立即停止，并且当前的操作也被取消。对 USART 的设置都不会丢，但 USART_ISR 中所有的状态标志都会被复位。 该位由软件置 1 和清零。 • 0：USART 预分频器和输出关闭（低功耗模式） • 1：USART 开启

19.7.2 控制寄存器 2（USART_CR2）

地址偏移：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

ADD[7:4]	ADD[3:0]	RTOEN	ABRMOD[1:0]	ABREN	MSBFIRST	DATAINV	TXINV	RXINV
rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res	LBDIE	LBDL	ADDM7	Res			
rw	rw	rw		rw	rw	rw	rw		rw	rw	rw				

位 31:28	ADD[7:4]：USART 的节点地址 该位域提供 USART 节点的地址或等待确认的字符码，用于多机通讯并且进入静默状态（或 Stop 模式）时，检测 7 位地址标记。发送器发出的字符的最高位应该为 1。该位域也用于正常接收过程中的字符检测，此时关闭静默状态（例如，在 ModBus 协议下对块尾的检测）。此时，接收的整个 8 位字节与 ADD[7:0]进行比较，如果匹配将会引起 CMF 标志被硬件置 1。 该位域只能在接收器被关闭（RE=0）或者在 USART 被关闭的时候（UE=0）才能改写。
位 27:24	ADD[3:0]：USART 的节点地址 该位域提供 USART 节点的地址或等待确认的字符码，用于多机通讯并且进入静默状态（或者 Stop 模式）时，检测 7 位地址标记。 该位域只能在接收器被关闭（RE=0）或者在 USART 被关闭的时候（UE=0）才能改写。
位 23	RTOEN：接收器超时检测功能使能 该位由软件置 1 和清零。 • 0：接收器超时检测功能关闭 • 1：接收器超时检测功能开启 当这个功能开启，只要 RX 线发现空闲（不是接收）达到 RTOR 寄存器（接收超时寄存器）中设置的时间长度后，USART_ISR 寄存器中的 RTOF 标志会被硬件置 1。
位 22:21	ABRMOD[1:0]：自动波特率检测模式 该位由软件设置或清零。 • 00：对起始位的测量被用来检测波特率。 • 01：下降沿到下降沿的测量（接收数据必须以单个的 1 作为开头，帧格式为 Start 1 0 xxxxxx）。 • 10：保留 • 11：保留 该位域只能在 ABREN=0 或 USART 未被使能的时候（UE=0）改写。
位 20	ABREN：自动波特率检测使能 该位由软件置 1 和清零。 • 0：自动波特率检测被禁止 • 1：自动波特率检测被打开
位 19	MSBFIRST：高位在前 该位由软件置 1 和清零。 • 0：数据在发送和接收的时候，采用起始位在前，后面跟着第 0 位的顺序。 • 1：数据在发送和接收的时候，采用起始位在前，后面跟着最高位（位 7 或者 8）

	的顺序。 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 18	DATAINV：二进制数反向 该位由软件置 1 和清零。 • 0：数据寄存器中的逻辑数据在发送和接收的时候，采用正/直接逻辑。（1=H，0=L） • 1：数据寄存器中的逻辑数据在发送和接收的时候，采用负/反向逻辑。（1=L，0=H）校验位也一样反向。 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 17	TXINV：TX 脚有效电平反向 该位由软件置 1 和清零。 • 0：TX 脚信号工作于标准逻辑电平（VDD =1/idle，Gnd=0/mark）。 • 1：TX 脚信号被反向（VDD =0/mark，Gnd=1/idle），可用于 TX 线上带有外部反相器的情况。 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 16	RXINV：RX 脚有效电平反向 该位由软件置 1 和清零。 • 0：RX 脚信号工作于标准逻辑电平（VDD =1/idle，Gnd=0/mark）。 • 1：RX 脚信号被反向（VDD=0/mark，Gnd=1/idle），可用于 RX 线上带有外部反相器的情况。 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 15	SWAP：交换 TX/RX 引脚 该位由软件置 1 和清零。 • 0：TX/RX 引脚按照标准引脚分配来使用。 • 1：TX 和 RX 的引脚功能交换使用，可用于和其它 UART 口进行交叉互联的情况。 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 14	LINEN：LIN 模式使能 该位由软件置 1 和清零。 • 0：LIN 模式禁止 • 1：LIN 模式使能 LIN 模式打开后，具备发送 LIN 同步断开（13 个低位）的功能，通过 USART_CR1 寄存器的 SBKRQ 位配置，同时还具备 LIN 同步断开信号的检测功能。 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 13:12	STOP[1:0]：停止位 该位域用于定制停止位的个数。 • 00：1 个停止位 • 01：0.5 个停止位

	• 10：2 个停止位 • 11：1.5 个停止位 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 11	CLKEN：时钟使能 该位用来打开 SCLK 引脚的功能。 • 0：SCLK 引脚被禁止。 • 1：SCLK 引脚被使能。 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 10	CPOL：时钟极性 该位允许用户选择同步模式下 SCLK 引脚上的时钟输出的极性。它同 CPHA 位一起决定所需要的时钟/数据时序关系。 • 0：在没有数据传输的时候保持低电平。 • 1：在没有数据传输的时候保持高电平。 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 9	CPHA：时钟相位 该位允许用户选择同步模式下 SCLK 引脚上的时钟输出的相位。它同 CPOL 位一起决定所需要的时钟/数据时序关系。 • 0：第一个时钟沿对准第一位数据。 • 1：第二和时钟沿对准第一位数据。 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 8	LBCL：末位时钟脉冲 该位用来选择在同步模式下，SCLK 脚上传输最后一个位（MSB）的时候是否必须输出时钟脉冲。 • 0：SCLK 脚上在传输末位数据的时候不输出时钟脉冲。 • 1：SCLK 脚上在传输末位数据的时候输出时钟脉冲。 *警告：末位是第 8 位还是 9 位，取决于 USART_CR1 寄存器中的 M 位的设置。* *该位域只能在 USART 未被使能的时候（UE=0）改写。*
位 7	Res：保留 必须保持复位值。
位 6	LBDIE：LIN 断开信号检测中断使能 断开中断屏蔽（利用断开分隔符来检测） • 0：中断禁用 • 1：在 USART_ISR 寄存器中的 LBDF 被置 1 的时候，会产生 USART 中断。
位 5	LBDL：LIN 断开检测长度 该位选择使用 11 位还是 10 位断开检测。 • 0：10 位断开检测

	• 1：11 位断开检测 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 4	ADDM7：7 位地址检测或 4 位地址检测选择 该位选择使用 4 位地址检测还是 7 位地址检测。 • 0：4 位地址检测 • 1：7 位地址检测（8 位数据模式下） 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 3:0	Res：保留 必须保持复位值。

19.7.3 控制寄存器 3（USART_CR3）

地址偏移：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res									WUFIE	WUS[1:0]		SCARCNT[2:0]			Res
									rw	rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	Res	OVRDIS	ONEBIT	CTSIE	CTSE	RTSE	Res		SCEN	NACK	HDSEL	IRLP	IREN	EIE
rw	rw		rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位 31:23	Res：保留 必须保持复位值。
位 22	WUFIE：从 Stop 模式唤醒中断使能 该位由软件置 1 和清零。 • 0：中断禁用 • 1：在 USART_ISR 寄存器中的 WUF 被置 1 时，会产生 USART 中断。
位 21:20	WUS[1:0]：从 Stop 模式唤醒中断标志选择 该位域指定激活 WUF 标志的事件。 • 00：在发生地址匹配事件的时候激活 WUF。 • 01：保留 • 10：WUF 在检测到起始位的时候激活。 • 11：WUF 在得到接受数据寄存器非空事件时激活。 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 19:17	SCARCNT[2:0]：智能卡模式重试计数器 该位域指定智能卡模式中接收和发送的重试次数。 在发送模式下，它指定的是在产生发送错误（FE=1）之前自动重试发送的次数。 在接收模式下，它指定的是在产生接收错误事件前（RXNE 和 PE=1）所作的错误接收的尝试的次数。

	• 0x0：重发功能关闭：在发送模式下不进行自动重发操作。 • 0x1～0x7：自动重传的尝试次数（在提示错误之前） 该位域只能在 USART 未被使能的时候（UE=0）改写。 *注意：当 USART 被使能，该位域只允许写成 0x0，这是为了避免盲目的自动重发数据。*
位 16	Res：保留 必须保持复位值。
位 15	DEP：驱动使能输出脚的极性选择 • 0：DE 信号高有效 • 1：DE 信号低有效 该位只能在 USART 未被使能的时候（UE=0）改写。
位 14	DEM：驱动器使能模式 该位允许用户通过 DE（驱动使能）信号来激活外部收发器的控制端。 • 0：DE 功能被禁止。 • 1：DE 功能被打开。DE 信号在 RTS 脚输出。 该位域只能在 USART 未被使能的时候（UE=0）改写。
位 13	Res：保留 必须保持复位值。
位 12	OVRDIS：溢出检测禁止 该位用于禁止对接收溢出现象的检测。 • 0：当在接收到新数据时上一次接收的数据还未被读取，会引起溢出错误标志 ORE 被硬件置 1。 • 1：溢出检测功能关闭。如果在新的接收数据到来时 RXNE=1 且 ORE≠1，新的数据会覆盖 USART_RDR 中以前的内容。 该位只能在 USART 未被使能的时候（UE=0）改写。
位 11	ONEBIT：单次采样方式使能 该位允许用户选择采样方式。当选择单次采样方式的时候，噪声监测标志（NF）就被禁止了。 • 0：三次采样方式 • 1：单次采样方式 该位只能在 USART 未被使能的时候（UE=0）改写。
位 10	CTSIE：CTS 中断使能 • 0：中断禁用 • 1：在 USART_ISR 寄存器中的 CTSIF 被置 1 时，会产生 USART 中断。
位 9	CTSE：CTS 功能使能

	• 0：关闭 CTS 硬件流控 • 1：打开 CTS 硬件流控，只有在 nCTS 输入上收到有效信号（被拉低）时才会发送数据。如果在数据传送时，nCTS 输入变为无效，会在数据发送完成后再停。如果写数据到发送寄存器的时候，nCTS 处于无效状态，那么直到 nCTS 信号变成有效才发送该数据。 该位只能在 USART 未被使能的时候（UE=0）改写。
位 8	RTSE：RTS 使能 • 0：关闭 RTS 硬件流控 • 1：RTS 输出使能，只会在有接收空间的时候才请求下一个数据。当前数据发送完成后，发送操作就需要暂停下来。如果可以接收数据了，将 nRTS 输出置为有效（拉低）。 该位只能在 USART 未被使能的时候（UE=0）改写。
位 7:6	Res：保留 必须保持复位值。
位 5	SCEN：智能卡模式使能 该位用于打开智能卡模式。 • 0：关闭智能卡模式 • 1：打开智能卡模式 该位只能在 USART 未被使能的时候（UE=0）改写。
位 4	NACK：智能卡 NACK 发送使能 • 0：出现校验错误时，不发送 NACK。 • 1：出现校验错误时，发送 NACK。 该位只能在 USART 未被使能的时候（UE=0）改写。
位 3	HDSEL：半双工模式选择 该位选择单线半双工模式。 • 0：不选择半双工模式。 • 1：选择半双工模式。 该位只能在 USART 未被使能的时候（UE=0）改写。
位 2	IRLP：IrDA 低功耗模式选择 该位选择 IrDA 是普通模式还是低功耗模式。 • 0：正常模式 • 1：低功耗模式 该位只能在 USART 未被使能的时候（UE=0）改写。
位 1	IREN：红外模式使能 该位由软件置 1 和清零。 • 0：不使能红外模式。

	• 1：使能红外模式。 该位只能在 USART 未被使能的时候（UE=0）改写。
位 0	EIE：错误中断使能 在允许帧错误、溢出错误或噪声错误产生中断请求时，需将 EIE 置 1。 • 0：中断禁用 • 1：当 USART_ISR 寄存器中的 FE=1、ORE=1 或 NF=1 时，会产生中断。

19.7.4 波特率寄存器（USART_BRR）

地址偏移：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRR[15:0]															
rw															

位 31:16	Res：保留 必须保持复位值。
位 15:4	BRR[15:4]：USARTDIV 的整数部分 该位域定义 USART 分频器除法因子的整数部分 USARTDIV[15:4]。
位 3:0	BRR[3:0]：USARTDIV 的小数部分 当 OVER8=1，则 BRR[3:0] = USARTDIV[3:0]；当 OVER8=0，则 BRR[3:0] = USARTDIV[3:0]右移 1 位。

19.7.5 保护时间和预分频器寄存器（USART_GTPR）

地址偏移：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GT[7:0]								PSC[7:0]							
rw								rw							

位 31:16	Res：保留 必须保持复位值。
位 15:8	GT[7:0]：保护时间值 该位域设置保护时间长度，以波特时钟为单位。GT 用于智能卡模式。在保护时间结束后，才会设置发送完成标志。

	该位域只能在 USART 未被使能的时候（UE=0）改写。
位 7:0	PSC[7:0]：预分频器值 • 在红外低功耗和正常模式下： PSC[7:0] = IrDA 正常和低功耗模式波特率对 USART 时钟源进行分频，以获得低功耗模式下的频率： 时钟源按寄存器的值来分频（8 位有效）： ○ 00000000：保留（不编程此值） ○ 00000001：1 分频 ○ 00000010：2 分频 ○ … • 智能卡模式：PSC[4:0]：预分频器值 用于设定对 USART 时钟源的分频系数，得到智能卡时钟。将寄存器中的值（低 5 位有效）乘以 2 得到的值作为分频系数进行分频： ○ 00000：保留（不编程此值） ○ 00001：2 分频 ○ 00010：4 分频 ○ 00011：6 分频 ○ … 该位域只能在 USART 未被使能的时候（UE=0）改写。

19.7.6 接收超时寄存器（USART_RTOR）

地址偏移：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BLEN[7:0]								RTO[23:16]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTO[15:0]															
rw															

位 31:24	BLEN[7:0]：块长度 该位域给出了智能卡模式 T=1 接收的块长度。这个值等于信息块字符数 + 结束部分（1-LEC/2-CRC）-1，例如： BLEN = 0 -> 0 个信息字符 + LEC; BLEN = 1 -> 0 个信息字符 + CRC; BLEN = 255 -> 254 个信息字符 + CRC（总共 256 个字符）; 智能卡模式中，当 TXE=0，会导致块长度计数器清零。 该位域也可以在其它模式中使用：块长度计数器在 RE=0 的时候清零和（或）在 EOBCF 位被写 1 的时候清零。
位 23:0	RTO[23:0]：接收超时值

	该位域提供接收超时的设置，单位是波特时钟的时长。 标准模式下，接收完最后一个字节，在整个 RTO 值规定的时长内，再也没有检测到新的起始位的时候，RTOF 标志会被硬件置 1。 在智能卡模式中，这个值用于实现 CWT 和 BWT。 *说明：超时测量是从最后一个字节的起始位开始计算。*

19.7.7 请求寄存器（USART_RQR）

地址偏移：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res											TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ
											w	w	w	w	w

位 31:5	Res：保留 必须保持复位值。
位 4	TXFRQ：发送数据清空请求 向该位写 1 会使 TX 标志被硬件置 1。 这里允许取消数据发送。 该位只能在智能卡模式下使用，当数据由于发生错误导致还未发出，USART_ISR 寄存器的 FE 标志为 1 时使用。 如果 USART 不支持智能卡模式，该位保留并由硬件强制为零。
位 3	RXFRQ：接收数据清空请求 向该位写 1 会使 RXNE 标志被硬件清零。 这里允许直接丢弃还没有读的接收数据，以免提示溢出错误。
位 2	MMRQ：静默模式请求 向该位写 1 将导致 USART 进入静默模式，同时清除 RWU 标志。
位 1	SBKRQ：请求发送断开字符 向该位写 1 会使 SBKF 标志置 1，并在发送状态机可用的时候向线路发出一个断开字符。
位 0	ABRRQ：自动波特率检测请求 向该位写 1 会清除 USART_ISR 中的 ABRF 标志，并在下一次数据接收的时候开始一次自动波特率测量。

19.7.8 中断和状态寄存器（USART_ISR）

地址偏移：0x1C

复位值：0x0000 00C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res									REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY
									r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	Res	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
r	r		r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:23	Res：保留 必须保持复位值。
位 22	REACK：接收使能通知标志 该位由硬件控制，当 USART 读到 REACK 的时候，会发送一个 ACK。该位用于在进入 Stop 模式之前，确认 USART 是否已经准备好接收数据。
位 21	TEACK：发送使能通知标志 该位由硬件控制，当 USART 读到 TEACK 时，将发送一个 ACK。 这使得在写 USART_CR1 寄存器的 TE=0 可以产生一个空闲帧请求，接着将 TE 写成 1 时，用于保证 TE=0 的最小周期的时候用。
位 20	WUF：从 Stop 模式唤醒标志 当检测到唤醒事件时，由硬件置 1。唤醒事件由 WUS 定义。由软件向 USART_ICR 寄存器的 PECF 位写 1，可以清 0。 如果 USART_CR3 寄存器的 WUF=1，会产生中断请求。
位 19	RWU：接收器从静默模式唤醒 该位表示 USART 处于静默模式时，若需要切换成唤醒模式，由硬件清零和置 1。 静默模式控制顺序（地址或空闲模式）用 USART_CR1 寄存器的 WAKE 位来选择。 如果选择由空闲信号唤醒，那该位就只能由软件置 1（USART_RQR 寄存器的 MMRQ 位置 1）。 • 0：接收器处于活动模式。 • 1：接收器处于静默模式。
位 18	SBKF：断开信号发送标志 该位在当要求发送一个断开字符时，由软件置 1（向 USART_CR3 寄存器的 SBKRQ 位写 1）。当断开字符的停止位传出后，该位由硬件自动清零。 • 0：不发送断开字符 • 1：发送断开字符
位 17	CMF：字符匹配标志 当收到一个字符和 ADD[7:0]中设置的内容相同时，该位由硬件置 1。 由软件向 USART_ICR 寄存器的 CMCF 位写 1，可以清除这个标志。 如果 USART_CR1 寄存器中的 CMIE 位为 1，就会产生中断请求。 • 0：没发现字符匹配 • 1：有发现字符匹配

位 16	BUSY：忙标志 该位由硬件置 1 和清零。当 RX 线在通讯时（成功检测到起始位），该位被硬件置 1。数据接收结束后（不管成功与否）会由硬件清零。 • 0：USART 处于空闲（未接收） • 1：正在接收数据
位 15	ABRF：自动波特率检测标志 当自动波特率功能打开时（RXNE 被置 1，若 RXNEIE=1 将产生中断）或者当自动波特率操作失败时（ABRE、RXNE 和 FE 都被置 1），该位被硬件置 1。 开始一轮新的波特率检测（向 USART_RQR 寄存器的 ABRQ 写 1）的时候会被清 0。
位 14	ABRE：自动波特率检测错误 在波特率测量失败的时候（超量程或字符比较失败），该位由硬件置 1。 该位由软件清零，其方法是向 USART_CR3 寄存器的 ABRRQ 位写 1。
位 13	Res：保留 必须保持复位值。
位 12	EOBF：块结束标志 当一个完整的块被接收时，由硬件置 1（例如 T=1 智能卡模式中）。当收到的字节数（从块开始，包括序言部分）大于等于 BLEN+4 时完成检测。如果 USART_CR2 寄存器的 EOBIE=1，会产生中断请求。 由软件向 USART_ICR 寄存器的 EOBCF 位写 1，可以将该位清 0。 • 0：块未结束。 • 1：块结束（足够字节数）。
位 11	RTOF：接收超时标志 如果无通讯的时长达到了 RTOR 寄存器中设定的超时值，该位会被硬件置 1。由软件向 USART_ICR 寄存器的 RTOCF 位写 1，可将该位清 0。如果 USART_CR2 寄存器的 RTOIE=1，会产生中断请求。在智能卡模式中，这个超时相当于 CWT 或 BWT 计时。 • 0：尚未超时。 • 1：已经达到超时。
位 10	CTS：CTS 标志 该位由硬件置 1 和清零。 这是 nCTS 输入脚状态的反向拷贝。 • 0：nCTS 线是高。 • 1：nCTS 线是低。
位 9	CTSIF：CTS 中断标志 如果 CTSE 位为 1，那么当 nCTS 输入状态变化的时候，由硬件置 1。 由软件向 USART_ICR 寄存器的 CTSCF 位写 1，可以清除这个标志。 如果 USART_CR3 寄存器的 CTSIE=1，会产生中断请求。

	• 0：nCTS 线的状态无变化。 • 1：nCTS 线的状态有变化。
位 8	LBDF：LIN 断开检测 当检测到 LIN 断开字符时，由硬件置 1。由软件向 USART_ICR 寄存器的 LBDCF 位写 1，可以清除这个标志。如果 USART_CR2 寄存器的 LBDIE=1，会产生中断请求。 • 0：未检测到 LIN 断开字符。 • 1：检测到 LIN 断开字符。
位 7	TXE：发送数据寄存器空 当 USART_TDR 寄存器中的值被取到移位寄存器时，该位被硬件置 1。 向 USART_TDR 寄存器再写数据就会立即将该位清 0。TXE 标志还可以用其它方式清除，例如向 USART_RQR 寄存器的 TXFRQ 位写 1。这是为了丢弃数据（仅在智能卡模式 T=0，传送失败的情形）。 如果 USART_CR1 寄存器中的 TXEIE 位被置起时，则会产生中断。 • 0：没有数据被传到移位寄存器。 • 1：数据被传到移位寄存器，发送数据寄存器为空。
位 6	TC：发送完成 在 TXE 为 1 的条件下，当数据发送完成的时候，该位由硬件置 1。 向 USART_TDR 寄存器再次写入数据，或者向 USART_ICR 寄存器的 TCCF 位写 1，都可以清除这个标志。 如果 USART_CR1 寄存器中的 TCIE 位是 1，则会产生中断请求。 • 0：发送未完成。 • 1：发送完成。
位 5	RXNE：接收数据寄存器非空 当接收移位寄存器的内容被传递到 USART_RDR 寄存器中时，该位被硬件置 1。 读取 USART_RDR 寄存器的数据就会同时清掉该位。也可通过向 USART_RQR 寄存器中的 RXFRQ 位写 1，来清除该位。 如果 USART_CR1 寄存器中的 RXNEIE 位是 1，就会产生中断请求。 • 0：未收到数据。 • 1：收到的数据已经可读。
位 4	IDLE：空闲线检测 当检测到线路空闲时，由硬件置 1。 如果 USART_CR1 寄存器中的 IDLEIE 位是 1，就产生中断请求。 由软件向 USART_ICR 寄存器的 IDLECF 位写 1，可以清除这个标志。 • 0：未检测到线路空闲。 • 1：检测到线路空闲。
位 3	ORE：溢出错误 在 RXNE=1 的条件下（即上次数据还未读取），串口接收寄存器又接收了一个字节的数

	据并准备送往 RDR 寄存器的时候，该位由硬件置 1。 由软件向 USART_ICR 寄存器的 ORECF 位写 1，可以清除这个标志。 如果 USART_CR1 寄存器中的 RXNEIE 位或 EIE 位为 1，就会产生中断请求。 • 0：无溢出错误。 • 1：检测到溢出错误。
位 2	NF：噪声检测标志 当接收帧的时候检测到噪声，该位由硬件置 1。由软件向 USART_ICR 寄存器的 NFCF 位写 1，可以将该位清 0。 • 0：未检测到噪声。 • 1：检测到噪声。
位 1	FE：帧错误 当检测到一个不同步现象、强噪声或一个断开符号时，该位由硬件置 1。 由软件向 USART_ICR 寄存器的 FECF 位写 1，可以清除这个标志。 在智能卡模式下发送数据时，当重发尝试的次数达到上限，若没有收到成功的回应（卡一直响应 NACK），该位也会被硬件置 1。 如果 USART_CR1 寄存器中的 EIE 位为 1，会产生中断请求。 • 0：未检测到帧错误。 • 1：检测到帧错误或者有收到断开字符。
位 0	PE：校验错误标志 当接收数据的时候发现校验错误，该位由硬件置 1。 由软件向 USART_ICR 寄存器的 PECF 位写 1，可以清除这个标志。 如果 USART_CR1 寄存器中的 PEIE 位为 1，会产生中断请求。 • 0：无校验错误 • 1：校验错误

19.7.9 中断标志清除寄存器（USART_ICR）

地址偏移：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res											WUCF	Res		CMCF	Res
											w			w	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res			EOBCF	RTOCF	Res	CTSCF	LBDCF	Res	TCCF	Res	IDLECF	ORECF	NCF	FECF	PECF
			w	w		w	w		w		w	w	w	w	w

位 31:21	Res：保留 必须保持复位值。
位 20	WUCF：清除从 Stop 模式唤醒标志 向该位写 1，会清除 USART_ISR 寄存器中的 WUF 标志位。

位 19:18	Res：保留 必须保持复位值。
位 17	CMCF：字符匹配标志的清除 向该位写 1，会清除 USART_ISR 寄存器中的 CMF 标志位。
位 16:13	Res：保留 必须保持复位值。
位 12	EOBCF：块结束标志的清除 向该位写 1，会清除 USART_ISR 寄存器中的 EOBF 标志位。
位 11	RTOCF：接收超时标志的清除 向该位写 1，会清除 USART_ISR 寄存器中的 RTOF 标志位。
位 10	Res：保留 必须保持复位值。
位 9	CTSCF：CTS 标志的清除 向该位写 1，会清除 USART_ISR 寄存器中的 CTSIF 标志位。
位 8	LBDCF：LIN 断开检测标志的清除 向该位写 1，会清除 USART_ISR 寄存器中的 LBDF 标志位。
位 7	Res：保留 必须保持复位值。
位 6	TCCF：发送完成标志的清除 向该位写 1，会清除 USART_ISR 寄存器中的 TC 标志位。
位 5	Res：保留 必须保持复位值。
位 4	IDLECF：线路空闲检测标志的清除 向该位写 1，会清除 USART_ISR 寄存器中的 IDLE 标志位。
位 3	ORECF：溢出错误标志的清除 向该位写 1，会清除 USART_ISR 寄存器中的 ORE 标志位。
位 2	NCF：噪声检测标志的清除 向该位写 1，会清除 USART_ISR 寄存器中的 NF 标志位。
位 1	FECF：帧错误标志的清除 向该位写 1，会清除 USART_ISR 寄存器中的 FE 标志位。

位 0	PECF：校验错误标志的清除 向该位写 1，会清除 USART_ISR 寄存器中的 PE 标志位。

19.7.10 数据接收寄存器（USART_RDR）

地址偏移：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res							RDR[8:0]								
							r								

位 31:9	Res：保留 必须保持复位值。
位 8:0	RDR[8:0]：接收的数据值 RDR 寄存器提供输入移位寄存器和内部总线间的并行接口。当接收数据时使能了校验功能，读这个寄存器得到的最高位是校验位。

19.7.11 数据发送寄存器（USART_TDR）

地址偏移：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res							TDR[8:0]								
							rw								

位 31:9	Res：保留 必须保持复位值。
位 8:0	TDR[8:0]：发送数据的值，用于写入待发送的数据字节。 TDR 寄存器提供发送移位寄存器和内部总线间的并行接口。 当发送的时候设置了校验功能（USART_CR1 中的 PCE=1），向最高位（位 7 或位 8，取决于设置的字长）写入的信息无效，因为校验位将替换最高位后再发送。

20 串行外设接口（SPI/I2S）

SPI/I2S 接口可用于使用 SPI 协议或 I2S 音频协议与外部器件进行通信。SPI 或 I2S 模式可通过软件进行选择。器件复位后默认选择 SPI 模式。

串行外设接口（SPI）协议支持与外部器件进行半双工、全双工和单工同步串行通信。该接口可配置为主模式，在这种情况下，它可为外部从器件提供通信时钟（SCK）。该接口还能够在多主模式配置下工作。

集成电路内置音频总线（I2S）也是同步串行通信接口。它能够在从模式或主模式下作为接收器或发送器工作。

它可满足四种不同音频标准的要求，包括 Philips I2S 标准、MSB 和 LSB 对齐标准以及 PCM 标准。

20.1 SPI 和 I2S 主要特征

20.2 SPI 主要特征

- 主模式或从模式操作
- 基于三条线的全双工同步传输
- 基于双线的半双工同步传输，其中一条可作为双向数据线
- 基于双线的单工同步传输，其中一条可作为单向数据线
- 4 位到 16 位传输帧格式选择
- 多主模式功能
- 8 个主模式波特率预分频器，可达 $f_{PCLK}/2$
- 从模式频率可达 $f_{PCLK}/2$
- 对于主模式和从模式都可通过硬件或软件进行 NSS 管理：动态切换主/从操作
- 可编程的时钟极性和相位
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 可触发中断的专用发送和接收标志
- SPI 总线忙状态标志
- 支持 SPI Motorola 模式
- 确保可靠通信的硬件 CRC 功能
  - 在发送模式下可将 CRC 值作为最后一个字节发送
  - 根据收到的最后一个字节自动进行 CRC 错误校验
- 可触发中断的主模式故障和上溢标志
- 可触发中断的 CRC 错误标志
- 支持 SPI TI 模式

20.2.1 I2S 主要特征

- 半双工通信（仅作为发送器或接收器）
- 主模式或从模式操作
- 8 位可编程线性预分频器，可实现精确的音频采样频率（从 8kHz 到 192kHz）
- 数据格式可以是 16 位、24 位或 32 位
- 数据包帧由音频通道固定为 16 位（可容纳 16 位数据帧）或 32 位（可容纳 16 位、24 位、32

位数据帧）

- 可编程的时钟极性（就绪时的电平状态）
- 从发送模式下的下溢标志、接收模式下的上溢标志（主模式和从模式），以及接收和发送模式下的帧错误标志（仅从模式）
- 发送和接收使用同一个 16 位数据寄存器
- 支持的 I2S 协议
  - I2S Philips 标准
  - MSB 对齐标准（左对齐）
  - LSB 对齐标准（右对齐）
  - PCM 标准（在 16 位通道帧或扩展为 32 位通道帧的 16 位数据帧上进行短帧和长帧同步）
- 数据方向始终为 MSB 在前
- 可输出主时钟以驱动外部音频元件。比率固定为 256×F_S（其中 F_S 为音频采样频率）

20.3 SPI/I2S 实现

表 20-1 SPI 实现

SPI 特性(1)	SPI
硬件 CRC 计算	支持
RX/TX FIFO	支持
I2S 模式	支持
TI 模式	支持

20.4 SPI 功能说明

SPI 支持在 MCU 与外部器件之间进行同步串行通信。应用软件可通过轮询状态标志或使用专用 SPI 中断对通信进行管理。SPI 的主要组件及其交互方式如下图所示。

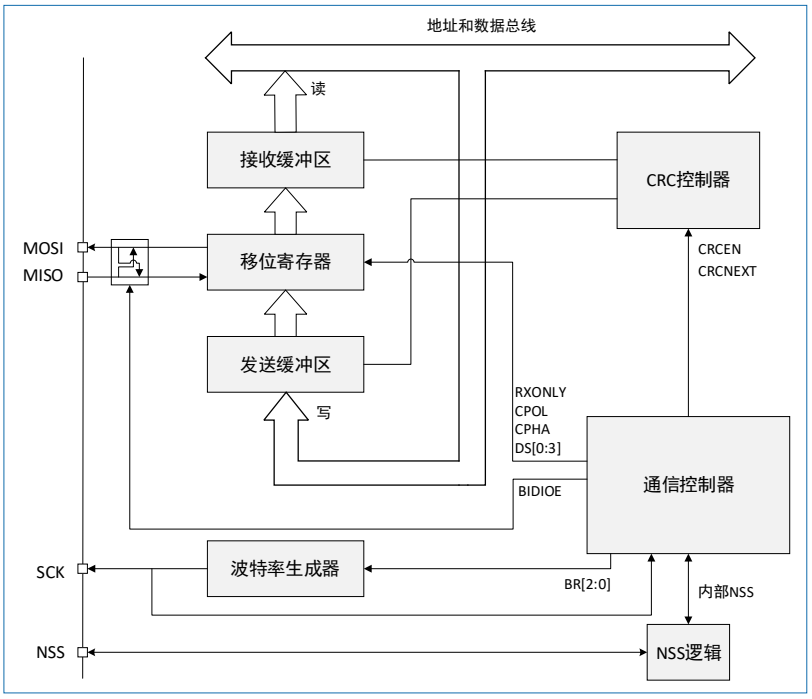


图 20-1 SPI 框图

四个 I/O 引脚专用于与外部器件进行 SPI 通信。

- MISO：主输入/从输出数据。通常情况下，此引脚用于在从模式下发送数据和在主模式下接收数据。
- MOSI：主输出/从输入数据。通常情况下，此引脚用于在主模式下发送数据和在从模式下接收数据。
- SCK：SPI 主器件的串行时钟输出引脚以及 SPI 从器件的串行时钟输入引脚
- NSS：从器件选择引脚。根据 SPI 和 NSS 设置，该引脚可用于：
  - 选择单个从器件以进行通信
  - 同步数据帧
  - 检测多个主器件之间是否存在冲突

详细信息，请参见“20.4.4 从器件选择（NSS）引脚管理”。

SPI 总线支持一个主器件与一个或多个从器件之间进行通信。该总线至少由两条线构成：一条用于时钟信号，另一条用于同步数据传输。其它信号可以根据 SPI 节点间的数据交换及其从器件选择信号管理进行添加。

20.4.1 一个主器件和一个从器件之间的通信

SPI 支持 MCU 基于目标器件和应用要求使用不同的配置进行通信。这些配置使用 2 条或 3 条线（通过软件 NSS 管理），也可以使用 3 条或 4 条线（通过硬件 NSS 管理）。通信始终由主器件发起。

20.4.1.1 全双工通信

默认情况下，SPI 配置为全双工通信。在这种配置下，主器件和从器件的移位寄存器通过 MOSI 和 MISO 引脚之间的两条单向线连接。在 SPI 通信过程中，数据随主器件提供的 SCK 时钟边沿同步移位。主器件通过 MOSI 线将待发送的数据发送给从器件，通过 MISO 线从从器件接收数据。当数据帧传输完成时（所有位均移出），主器件和从器件之间即完成信息交换。

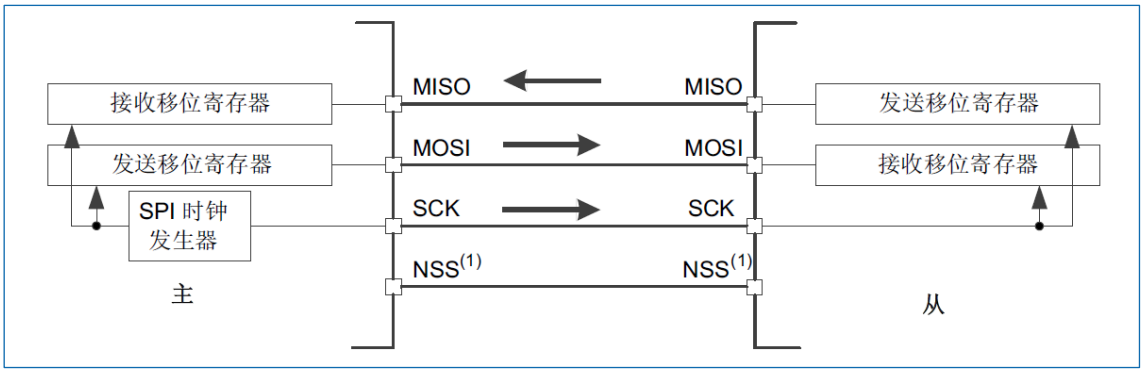


图 20-2 全双工单个主器件/单个从器件应用

(1). NSS 引脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些引脚。之后，必须在内部为主器件和从器件处理硬件控制流。有关更多详细信息，请参见“20.4.4 从器件选择（NSS）引脚管理”。

20.4.1.2 半双工通信

通过将 SPIx_CR1 寄存器的 BIDIMODE 位置 1，SPI 可采用半双工模式进行通信。在这种配置下，使用一条交叉连接线将主器件和从器件的移位寄存器连接起来。在此通信过程中，数据随 SCK 时钟边沿在移位寄存器之间进行移位，传输方向由主器件和从器件通过各自 SPIx_CR1 寄存器中的 BIDIOE 位进行选择。

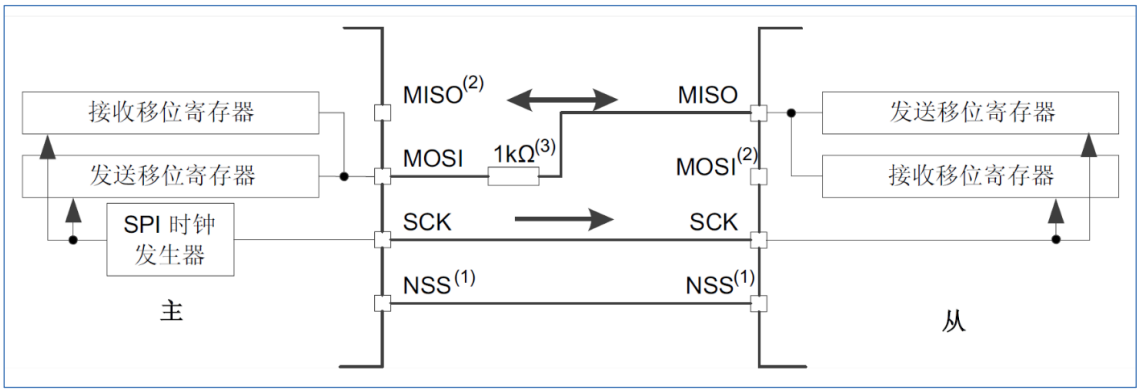


图 20-3 半双工单个主器件/单个从器件应用

(1). NSS 引脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些引脚。之后，必须在内部为主器件和从器件处理硬件控制流。有关更多详细信息，请参见从器件选择（*NSS*）引脚管理。

(2). 在这种配置下，主器件的 MISO 引脚和从器件的 MOSI 引脚可用作 GPIO。

(3). 当以双向模式工作的两个节点间的通信方向不是同步变化时，会出现临界情况，新发送器访问共用数据线，而前一个发送器仍保持线路上的相反值（值取决于 SPI 配置和通信数据）。两个节点会出现冲突，在共用线上短暂提供相反的输出电平，直到下一个节点也相应地改变其方向设置。建议此模式下在 MISO 和 MOSI 引脚之间插入串行电阻以在这种情况下保护输出并限制电流在二者之间流过。

20.4.1.3 单工通信

通过 SPIx_CR1 寄存器中的 RXONLY 位将 SPI 设置为只发送模式或只接收模式，可使 SPI 以单工模式进行通信。在这种配置下，仅使用一条线在主器件和从器件的移位寄存器之间进行传输。其余 MISO 和 MOSI 引脚对不用于通信，可用作标准 GPIO。

- 只发送模式（RXONLY=0）：配置设置与全双工设置相同。应用必须忽略在未使用的输入引脚上捕获的信息。该引脚可以用作标准 GPIO。
- 只接收模式（RXONLY=1）：应用可通过将 RXONLY 位置 1 来禁止 SPI 输出功能。在从器件配置下，MISO 输出被禁止，该引脚可用作 GPIO。当从器件选择信号有效时，从器件继续从 MOSI 引脚接收数据（请参见“20.4.3 多主器件通信”）。基于数据缓冲区的配置产生接收数据事件。在主器件配置下，MOSI 输出被禁止，该引脚可用作 GPIO。只要 SPI 处于使能状态，便不断生成时钟信号。停止时钟的唯一方式是将 RXONLY 位或 SPE 位清零，直至来自 MISO 引脚的传入模式结束，然后基于相应配置填充数据缓冲区结构。

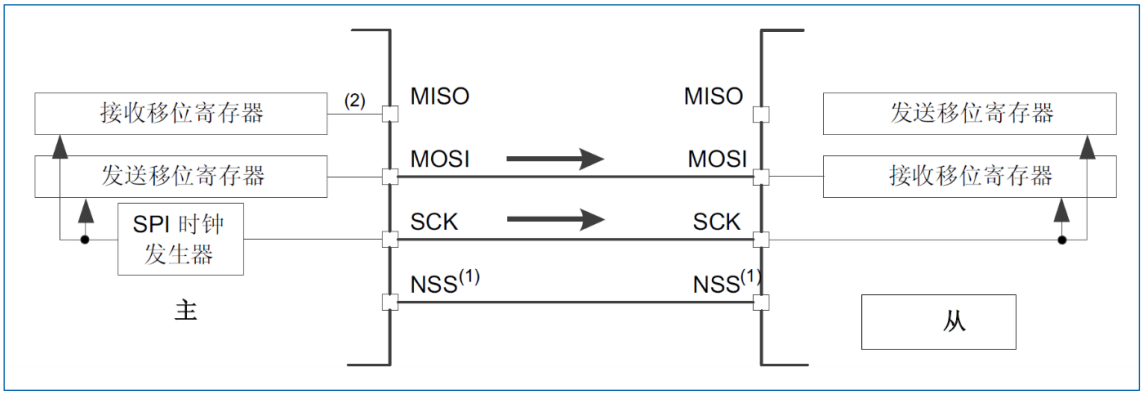


图 20-4 单工单个主器件/单个从器件应用（主器件为只发送模式/从器件为只接收模式）

(1). NSS 引脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些引脚。之后，必须在内部为主器件和从器件处理硬件控制流。有关更多详细信息，请参见“20.4.4 从器件选择（NSS）引脚管理”。

(2). 在发送器 Rx 移位寄存器的输入上捕获意外输入信息。标准只发送模式下必须忽略与发送器接收流相关的所有事件（例如 OVF 标志）。

(3). 在这种配置下，两个 MISO 引脚均可用作 GPIO。

说明：任何单工通信均可以通过半双工通信的一种变型来替换，该变型中设置的数据传输方向不变（在 BIDIOE 位保持不变的同时，双向模式处于使能状态）。

20.4.2 标准多从器件通信

在具有两个或多个独立从器件的配置下，主器件使用 GPIO 引脚来管理每个从器件的片选线（请参见图 20-5）。主器件必须通过拉低与从器件 NSS 输入相连的 GPIO 的电平来单独选择一个从器件。执行该操作后，便建立了标准主器件与专用从器件之间的通信。

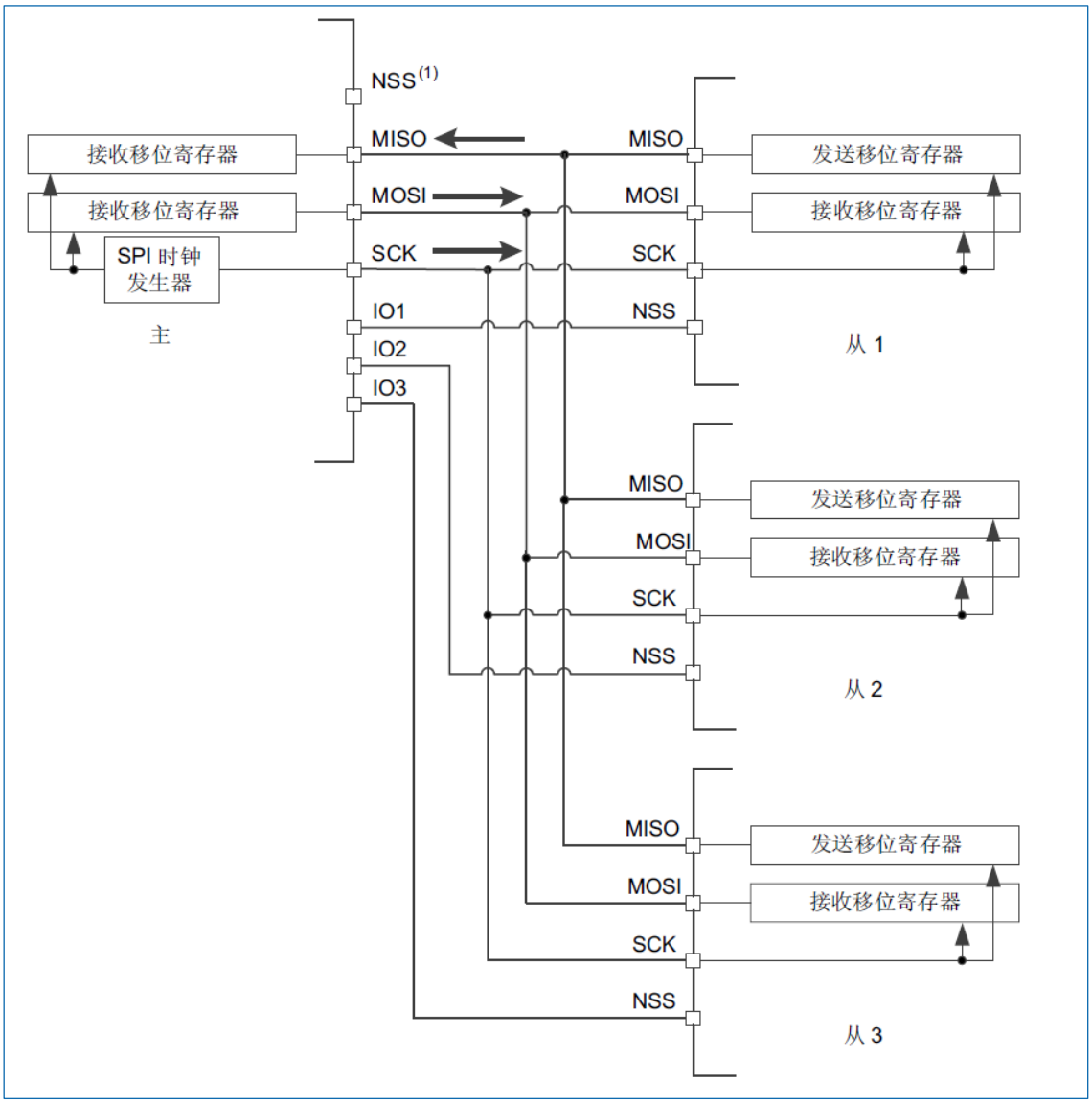


图 20-5 主器件和三个独立的从器件

(1). 此配置的主器件侧不使用 NSS 引脚。该引脚必须在内部管理（SSM=1，SSI=1）以避免任何 MODF 错误。

(2). 由于从器件的 MISO 引脚连在一起，所有从器件 MISO 引脚的 GPIO 配置必须设置为复用功能开漏。

20.4.3 多主器件通信

如果 SPI 总线未用于多主功能，用户可使用内置功能来检测尝试同时控制总线的两个节点间是否存在潜在冲突。对于该检测，NSS 引脚需配置为硬件输入模式。

由于此时只有一个结点可将其输出施加到公用数据线上，因此无法连接超过两个以此模式工作的 SPI 节点。

当节点无效时，默认情况下均保持从模式。一旦一个节点要接管总线的控制，它会将自身切换到主模式，然后通过专用 GPIO 引脚向其它节点的从器件选择输入施加有效电平。会话完成后，有效的从器件选择信号将被释放，控制总线的节点会短暂切换回被动从模式，等待下一个会话开始。

如果两个节点同时发出各自的控制请求，则会出现总线冲突（请参见“20.10.2 SPI 控制寄存器 2（SPI_CR2）”中的模式故障 MODF 事件）。随后，用户可应用某个简单的仲裁过程（例如，在两个节点上施加不同的预定义超时来推迟下一个尝试）。

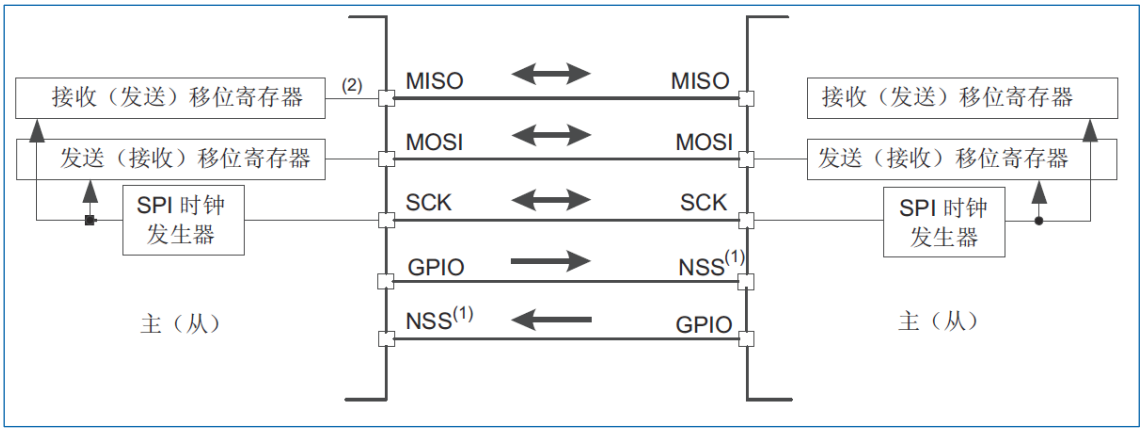


图 20-6 多主器件应用

(1). 在两个节点上，NSS 引脚配置为硬件输入模式。当无效节点配置为从器件时，其有效电平将使能 MISO 线输出控制。

20.4.4 从器件选择（NSS）引脚管理

在从模式下，NSS 用作标准的“片选”输入，使从器件与主器件进行通信。在主模式下，NSS 可用作输出或输入。NSS 用作输入时，可防止多主模式总线冲突；NSS 用作输出时，可驱动单个从器件的从器件选择信号。

可以使用 SPIx_CR1 寄存器中的 SSM 位设置硬件或软件从器件选择管理：

- 软件 NSS 管理（SSM=1）：在这种配置下，由 SPIx_CR1 寄存器中的 SSI 位的值决定内部驱动从器件选择信息。外部 NSS 引脚空闲，可供其它应用使用。
- 硬件 NSS 管理（SSM=0）：在这种情况下，可行的配置有两种：所用配置取决于 NSS 输出配置（SPI_CR2 寄存器中的 SSOE 位）。
  - NSS 输出使能（SSM=0 且 SSOE=1）：仅在将 MCU 设置为主器件时才使用该配置。NSS 引脚由硬件管理。只要在主模式下使能 SPI（SPE=1），NSS 信号便会被驱动为低电平，并且会一直保持低电平状态，直至禁止 SPI（SPE=0）。
  - NSS 输出禁止（SSM=0 且 SSOE=0）：如果 MCU 在总线上用作主器件，此配置可实现多主模式功能。如果在该模式下将 NSS 引脚拉至低电平，SPI 将进入主模式故障状态，器件将在从模式下自动进行重新配置。在从模式下，NSS 引脚用作标准的“片选”输入，当 NSS 线为低电平时将选择从器件。

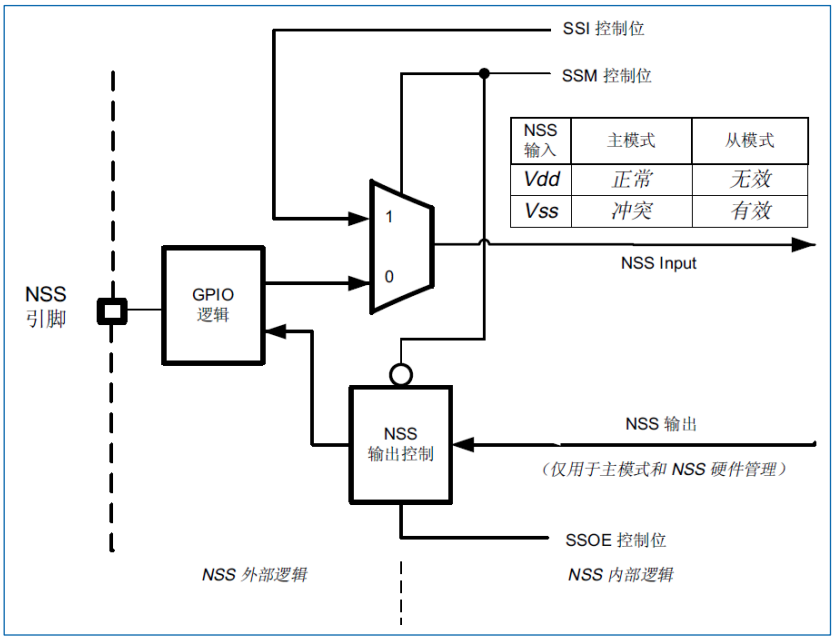


图 20-7 硬件/软件从器件选择管理

20.4.5 通信格式

SPI 通信过程中，将同时执行接收和发送操作。串行时钟（SCK）对数据线上的信息的移位和采样进行同步。通信格式取决于时钟相位、时钟极性和数据帧格式。为了能够在彼此间进行通信，主器件和从器件必须遵循相同的通信格式。

20.4.5.1 时钟相位和极性控制

通过 SPIx_CR1 寄存器中的 CPOL 和 CPHA 位，可以用软件选择四种可能的时序关系。CPOL（时钟极性）位控制不传输任何数据时时钟的空闲状态值。该位对主器件和从器件都有作用。如果将 CPOL 置 0，SCK 引脚在空闲状态处于低电平。如果将 CPOL 置 1，SCK 引脚在空闲状态处于高电平。

如果将 CPHA 位置 1，则会在 SCK 引脚的第二个边沿捕获传输的第一个数据位（如果将 CPOL 位置 0，则为下降沿捕获；如果将 CPOL 位置 1，则为上升沿捕获）。即在每次出现该时钟边沿时锁存数据。如果将 CPHA 位置 0，则会在 SCK 引脚的第一个边沿捕获传输的第一个数据位（如果将 CPOL 位置 1，则为下降沿捕获；如果将 CPOL 位置 0，则为上升沿捕获）。即，在每次出现该时钟边沿时锁存数据。

CPOL（时钟极性）和 CPHA（时钟相位）位的组合用于选择数据捕获时钟边沿。

图 20-8 给出了在 CPHA 和 CPOL 位的四种组合下的 SPI 全双工传输。

说明：在切换 CPOL/CPHA 位之前，必须通过复位 SPE 位来禁止 SPI。

SCK 的空闲状态必须与 SPIx_CR1 寄存器中选择的极性相对应（如果 CPOL=1，则上拉 SCK；如果 CPOL=0，则下拉 SCK）。

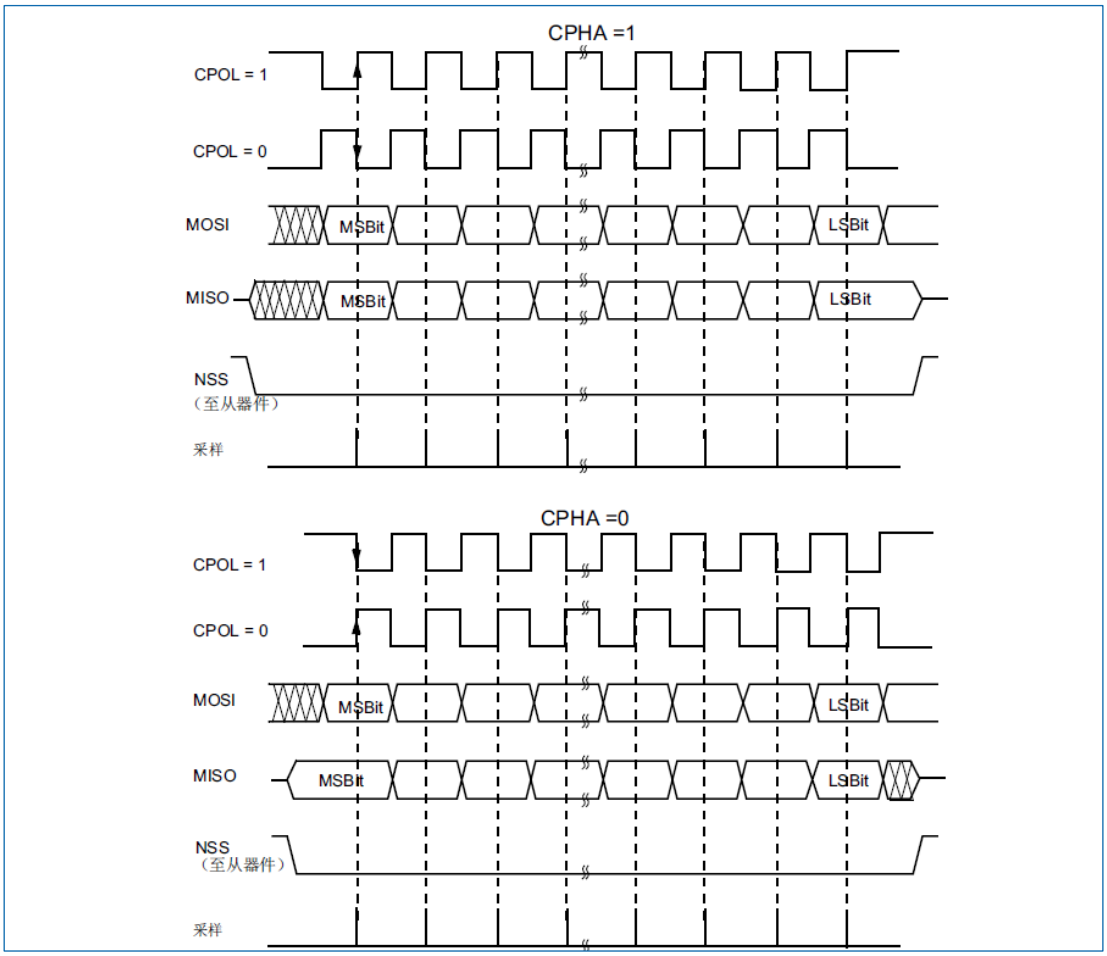


图 20-8 数据时钟时序图

说明：数据位的顺序取决于 LSBFIRST 位的设置。

20.4.5.2 数据帧格式

SPI 移位寄存器可设置为以 MSB 在前或 LSB 在前的方式移出数据，具体取决于 LSBFIRST 位的值。每个数据帧的长度均可配置为 4 位到 16 位，具体取决于使用 SPI_CR2 寄存器中的 DS 位编程的数据长度。所选的数据帧格式适用于发送和接收。无论选定的数据帧大小如何，对 FIFO 的读访问必须与 FRXTH 位的配置对齐。当访问 SPI_DR 寄存器时，数据帧总是右对齐为一个字节（如果数据为一个字节）或半个字。在通信期间，只有数据帧内的位被计时和传输。

20.4.6 SPI 配置

主器件和从器件的配置步骤几乎相同。对于具体的模式设置，请遵从相应章节的内容。若要对标准通信进行初始化，请执行以下步骤：

1. 写相关的 GPIO 寄存器：配置 MOSI、MISO 和 SCK 引脚。
2. 对 SPI_CR1 寄存器执行写操作：
   A. 通过 BR[2:0]位配置串行时钟波特率(1)。
   B. 配置 CPOL 位和 CPHA 位组合，从数据与时钟四组时序定义中选择一种定义(2)。
   C. 通过配置 RXONLY 或 BIDIMODE 和 BIDIOE 来选择单工或半双工模式（RXONLY 和 BIDIMODE 不可同时置 1）。
   D. 配置 LSBFIRST 位以定义帧格式(2)。
   E. 如果需要 CRC，请配置 CRCL 位和 CRCEN 位（SCK 时钟信号处于空闲状态时）。

F. 配置 SSM 和 SSI(2)(3)。

G. 配置 MSTR 位（在多主模式 NSS 配置下，如果主器件配置为防止发生 MODF 错误，则应避免 NSS 上出现状态冲突）。

3. 对 SPI_CR2 寄存器执行写操作：

A. 配置 DS 位来设置数据帧格式（4 位到 16 位）

B. 配置 SSOE (2) (3) (4)。

C. 要使用 TI 模式，请将 FRF 位置 1。（TI 模式下需保持 NSSP 位为 0）

D. 如果两个数据单元之间需要 NSS 脉冲模式，则设置 NSSP 位（在 NSSP 模式下保持 CHPA 和清除 TI 位）。

E. 配置 FRXTH 位。RXFIFO 阈值必须与 SPI_DR 寄存器的读取访问大小对齐。

4. 对 SPI_CRCPR 寄存器执行写操作：需要时配置 CRC 多项式。

注释：
(1). 从模式下无需此步骤，但从器件在 TI 模式下工作时除外。
(2). TI 模式下无需此步骤。
(3). 从模式下无需此步骤。
(4). NSSP 模式下无需此步骤。

20.4.7 使能 SPI 步骤

建议在主器件发送时钟前使能 SPI 从器件。否则，数据传输可能会不正常。从器件的数据寄存器必须包含待发送的数据才能开始与主器件通信（在通信时钟的第一个边沿；如果时钟信号连续，则是在正在进行的通信结束前）。使能 SPI 从器件前，SCK 信号必须稳定为所选极性对应的空闲状态电平。

当 SPI 启用且 TXFIFO 不为空时，或在下一次写入 TXFIFO 时，处于全双工（或任何仅发送模式）的主机开始通信。

在任何主器件只接收模式（RXONLY=1 或 BIDIMODE=1 且 BIDIOE=0）下，使能 SPI 后，主器件立即开始通信且时钟立即开始运行。

从器件接收到来自主器件的正确时钟信号时，它将开始通信。在 SPI 主器件启动传输前，从软件必须写入待发送的数据。

20.4.8 数据发送和接收过程

RXFIFO 和 TXFIFO

SPI 数据传输都通过 32 位嵌入式 FIFO。这使 SPI 能够连续的工作，并且在数据帧较小时防止溢出。每个方向都有自己的 FIFO，称为 TXFIFO 和 RXFIFO。除了启用 CRC 计算的仅接收器模式（从或主）外，这些 FIFO 用于所有 SPI 模式（参见“20.4.14 CRC 计算”）。

FIFO 的处理取决于数据交换模式（双工、单工）、数据帧格式（帧中的位数）、对 FIFO 数据寄存器执行的访问大小（8 位或 16 位），以及访问 FIFO 时是否使用数据打包（参见“20.4.13 TI 模式”）。

对 SPI_DR 寄存器的读取访问返回 RXFIFO 中存储的尚未读取的第一个数值。对 SPI_DR 的写入访问将写入的数据存储在发送队列末尾的 TXFIFO 中。读取访问必须始终与 SPI_CR2 寄存器中的 FRXTH 位配置的 RXFIFO 阈值对齐。FTLVL[1:0]和 FRLVL[1:0]位表示两个 FIFO 的当前占用水平。

对 SPI_DR 寄存器的读取访问必须由 RXNE 事件管理。当数据存储在 RXFIFO 中且达到阈值（由 FRXTH 位定义）时，会触发此事件。清除 RXNE 时，RXFIFO 被视为空。以类似的方式，要传输的数据帧的写访问由 TXE 事件管理。TXFIFO 级别小于或等于其容量的一半时触发此事件。否则，TXE 被清除，TXFIFO 被视为已满。这样，RXFIFO 最多可以存储四个数据帧，而 TXFIFO 在数据帧格式不大于 8 位时最多只能存储三个数据帧。当软件试图以 16 位模式将更多数据写入 TXFIFO 时，此差异可防止 TXFIFO 中已存储的

3x8 位数据帧可能损坏。TXE 和 RXNE 事件都可以通过中断进行轮询或处理。

如果接收到下一个数据时 RXFIFO 的是满的，将导致发生溢出事件（参见“20.4.10 SPI 状态标志”）。溢出事件可以查询处理或中断处理。

当前数据帧传输正在进行时，BSY 位将置 1。当时钟信号连续运行时，BSY 标志在主器件侧的两个数据帧间保持置 1。不过，在从器件侧，它将在每个数据帧传输之间变为 0 并持续至少一个 SPI 时钟周期。

序列处理

多个数据字节可以顺序发送来组成一个消息。启用发送后，在主机 TXFIFO 中的数据会开始发送并连续发完。主机会连续输出时钟信号，直至 TXFIFO 变为空，然后停下来等待新增的数据。

在单接收模式，半双工（BIDIMODE=1，BIDIOE=0），或只发送（BIDIMODE=0，RXONLY=1）模式下，只要 SPI 和只接收模式被使能，主机会立即开始接收序列。主机连续提供时钟信号，直到主机关闭 SPI 或者关闭只接收模式之前都不会停下来。这时主机会连续接收数据。

虽然主机可以以连续模式提供所有交互（SCK 信号是连续的），但它必须尊重从机在任何时候处理数据流及其内容的能力。必要时，主机必须减慢通信速度，并提供较慢的时钟或具有足够延迟的单独帧或数据会话。请注意，在 SPI 模式下，主设备或从设备没有下溢错误信号，并且来自从设备的数据始终由主设备进行处理，

数据帧开始后，从机无法控制或延迟数据序列。出于这个原因，从机必须在开始传输之前准备好数据，并总是一直保持有数据待传（存在 TXFIFO 中）。主机必须在每个序列之间给从机保留足够的时间以准备数据。如果可能的话，序列中的字节的数量应得到限制，以便从机完成自动的数据处理（通过使用 FIFO）。主机必须提供额外的时间给从机处理数据内容。

在多从机并行的系统中，每个序列应该由 NSS 脉冲来分隔，以将每个序列对应到不同的从机。在单从机系统中通过 NSS 控制从机就显得没那么有必要了，但这里有个脉冲还是更好，这可以令从机和每个数据序列完成同步。NSS 既可以由软件管理也可以由硬件管理（参见“20.4.4 从器件选择（NSS）引脚管理”）。

BSY 位被置 1 时，它标志着一个持续的传输。这一点，结合 FTLVL[1:0]位，可用于检查传输是否完成。在系统进入停机模式前这是很有必要的，过早的进入停机模式可能导致数据破坏。判断 BSY 位的另外一个目的是可以用关键来管理 NSS 信号。当 RXNE 标志被置 1，它意味着对当前的传输结束了。即最后一位刚刚采样完毕，以及完整的数据帧存储在 RXFIFO 中。

数据打包

当数据帧的大小适合一个字节（小于或等于 8 位），并且对 SPI_DR 寄存器执行任何 16 位的读写访问时，数据会自动打包在一起。在这种情况下，可以并行处理双数据。SPI 先操作低 8 位，然后操作高 8 位。图 20-9 提供了打包方式顺序处理数据的一个例子。在一次对 SPI_DR 的 16 位写访问后，就会有 2 个字节的数据被发送出去。如果 RXFIFO 的阈值设置为 16 位（FRXTH=0），该序列则只会生成一个接收 RXNE 事件，而不是两个。针对这种单个的 RXNE 事件的响应，接受器必须对 SPI_DR 寄存器作一次 16 位的读访问才能够把数据全都取到。RxFIFO 的阀值和跟进的数据访问的位宽必须保持一致，否则就会丢数据。

字节数为奇数的数据帧传输时，会出现特定的问题。在发送端，可以用 8 位方式写 SPIx_DR 将最后一个字节发出来。在接收端必须改变 Rx_FIFO 门限，以便在接收奇数字节的数据帧的最后字节时能够产生 RXNE 事件。

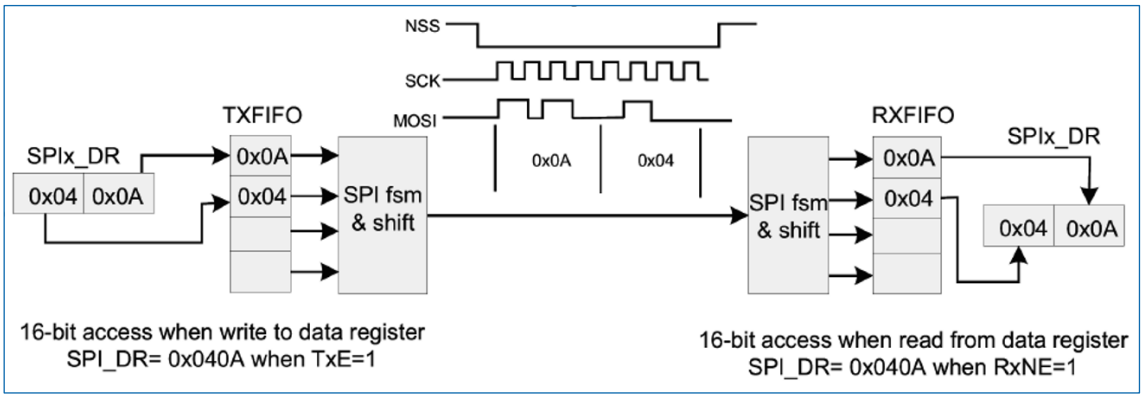


图 20-9 发送和接收 FIFO 中的数据打包

20.4.9 禁用 SPI 步骤

当禁止 SPI 时，必须按照本段中介绍的禁止步骤进行操作。当外设时钟停止时，在系统进入低功耗模式前做到这一点是十分重要的。否则会损坏正在进行的交互。在某些模式下，禁止步骤是停止所进行的连续通信的唯一方式。

当处于全双工或只发送模式下的主器件停止提供待发送的数据时，可结束任何交互。在这种情况下，时钟在最后一个数据传输后停止。

当处理奇数个数据帧时，必须特别注意打包模式，以防止一些虚拟字节交换(参见数据打包部分)，在这些模式下禁用 SPI 之前，用户必须遵循标准的禁用过程。当一个帧事务正在进行或下一个数据帧存储在 TXFIFO 中，SPI 在主发射机被禁用时，SPI 行为是不保证的。

当主机处于任何接收模式时，停止连续时钟的唯一方法是通过 SPE=0 禁用外围设备。这个必须发生在特定的时间窗口内最后一个数据帧采样时间之间的交易，只是第一点和最后一点前转移开始(为了得到一个完整的预期数量数据帧,并防止任何额外的“假”数据后阅读最后一个有效的数据帧)。在该模式下禁用 SPI 必须遵循特定的步骤。

当 SPI 被禁用时，接收但没有读取的数据仍然存储在 RXFIFO 中，并且必须在下一次 SPI 被启用时进行处理，然后开始一个新的序列。为了防止有未读的数据，确保在禁用 SPI 时 RXFIFO 是空的，通过使用正确的禁用过程，或通过软件复位初始化所有 SPI 寄存器，通过控制一个专门用于外设复位的特定寄存器。

标准禁止步骤通过轮询 BSY 状态以及 TXE 标志来检查发送会话是否完全结束。还可以在必须识别正在处理的传输是否结束的特定情况下完成这种检查，例如：当 NSS 信号由任意 GPIO 切换管理且主器件必须为从器件提供 NSS 脉冲结束时。

正确的禁止步骤如下（使用只接收模式时除外）：

1. 等待 RXNE=1，以接收最后的数据。
2. 等待至 TXE=1，然后等待至 BSY=0，再关闭 SPI。
3. 读取接收的数据。

说明：在不连续通信期间，在对 SPI_DR 寄存器执行写操作与 BSY 位置 1 之间有 2 个 APB 时钟周期的延迟。因此，写入最后的数据后，必须先等待 TXE 位置 1，然后等待 BSY 位清零。

某些只接收模式的正确禁止步骤如下：

1. 当最后一个数据帧正在处理时，通过在特定时间窗口内禁止 SPI（SPE=0）来中断接收流。
2. 等待至 BSY=0（最后一个数据帧已处理完）。

3. 读取接收的数据。

说明：要停止连续的接收序列，必须在接收最后一个数据帧时遵循特定的时间窗口。该时间窗口在第一个位采样时开始，在最后一个位的传输开始前结束。

20.4.10 SPI 状态标志

应用可通过三种状态标志监视 SPI 总线的状态。

发送缓冲区为空（TXE）

当传输 TXFIFO 有足够的空间存储要发送的数据时，设置 TXE 标志。

TXE 标志连接到 TXFIFO 深度。标志置 1 并保持直到 TXFIFO 深度是更低或等于 FIFO 深度的 1/2。如果 SPIx_CR2 寄存器中的 TXEIE 位置 1，当 TXE 置 1 时就会生成一个发送中断。当 TXFIFO 电平大于 1/2 时，位自动清除。

接收缓冲区非空（RXNE）

RXNE 标志的设置取决于 SPIx_CR2 寄存器中的 FRXTH 位值:

- 如果 FRXTH 位被置 1，当 RXFIFO 深度达到 1/4，RXNE 位将会置 1 并一直保持。
- 如果 FRXTH 位置 0，当 RXFIFO 深度达到 1/2，RXNE 位将会置 1 并保持。

如果 SPIx_CR2 寄存器中 RXNEIE 位置 1，当 RXNE 置 1 时将会生成接收中断。

忙标志（BSY）

BSY 标志由硬件置 1 和清零（写入此标志没有任何作用）。

当 BSY 置 1 时，表示 SPI 上正在进行数据传输（SPI 总线繁忙）。在主模式下的双向通信接收模式（MSTR=1 且 BIDIMODE=1 且 BIDIOE=0）有一个例外情况，BSY 标志在接收过程中始终保持为 0。

在某些模式下，可使用 BSY 标志来检测传输是否结束，从而避免在进入低功耗模式前禁止 SPI 外设时钟时或通过软件管理 NSS 脉冲结束时破坏最后一次传输。

BSY 标志还可用于避免在多主模式系统中发生写冲突。

在以下任意一种条件下，BSY 标志将清零：

- 正确禁止 SPI 时。
- 在主模式下检测到故障时（MODF 位置 1）。
- 在主模式下，完成了数据发送并且不准备发送任何新数据时。
- 在从模式下，BSY 标志在各传输之间的至少一个 SPI 时钟周期内置为“0”时。

说明：建议始终使用 TXE 和 RXNE 标志（而非 BSY 标志）来处理数据发送或接收操作。

20.4.11 SPI 错误标志

如果以下其中一个错误标志置 1 且已通过将 ERRIE 位置 1 使能了中断，则将生成 SPI 中断。

上溢标志（OVR）

当主机或从机接收到数据时，如果 RXFIFO 没有足够空间存储接收到的数据时就会发生溢出。这种情况通常发生在软件没有足够时间去读出以前接受的数据（存储在 RXFIFO）或数据存储空间有限时，如当启用 CRC 并处于仅接收模式时 RXFIFO 不可用，这种情况下应将接收缓冲区限制到一个数据帧缓冲。

当一个上溢事件产生，新接收到的数据将会被丢弃，不会覆盖 RXFIFO 中以前的值，随后传输的所有数据都将会被丢弃。对 SPI_DR 寄存器的读访问和对 SPI_SR 寄存器的读访问将会清除 OVR 位。

模式故障（MODF）

当主器件的内部 NSS 信号（NSS 硬件模式下为 NSS 引脚，NSS 软件模式下为 SSI 位）被拉低时，将发生模式故障。这会自动将 MODF 位置 1。主模式故障会在以下几方面影响 SPI 接口：

- 如果 ERRIE 位置 1，MODF 位将置 1，并生成 SPI 中断。
- SPE 位清零。这将关闭器件的所有输出，并禁止 SPI 接口。
- MSTR 位清零，从而强制器件进入从模式。

使用以下软件序列将 MODF 位清零：

- 在 MODF 位置 1 时，对 SPIx_SR 寄存器执行读或写访问。
- 对 SPIx_CR1 寄存器执行写操作。

为避免包含多个 MCU 的系统中发生多从模式冲突，必须在 MODF 位清零序列期间将 NSS 引脚拉高。在该清零序列后，可以将 SPE 和 MSTR 位恢复到原始状态。安全起见，硬件不允许在 MODF 位置 1 时将 SPE 和 MSTR 位置 1。在从器件中，MODF 位不可置 1，但由前一次多主模式冲突引起时除外。

CRC 错误（CRCERR）

当 SPIx_CR1 寄存器中的 CRCEN 位置 1 时，此标志用于验证接收数据的有效性。当 CRCEN 位置 1，如果移位寄存器中接收的值与 SPIx_RXCRCR 的值不匹配，SPIx_SR 寄存器中的 CRCERR 标志将置 1。该标志由软件清零。

TI 模式帧格式错误（FRE）

如果 SPI 在从模式下工作，并配置为符合 TI 模式协议，则在通信进行期间出现 NSS 脉冲时，将检测到 TI 模式帧格式错误。出现此错误时，SPIx_SR 寄存器中的 FRE 标志将置 1。

发生错误时不会禁止 SPI，但会忽略 NSS 脉冲，并且 SPI 会等待下一个 NSS 脉冲，然后再开始新的传输。由于错误检测可能导致丢失两个数据字节，因此数据可能会损坏。

读取 SPIx_SR 寄存器时，将清零 FRE 标志。如果 ERRIE 位置 1，则检测到 NSS 错误时将生成中断。在这种情况下，由于无法保证数据的一致性，应禁止 SPI，并在重新使能从 SPI 后，由主器件重新发起通信。

20.4.12 NSS 脉冲模式

该模式由 SPIx_CR2 寄存器中的 NSSP 位激活，仅当 SPI 接口配置为 Motorola SPI master（FRF=0）并在第一个边沿捕获（SPIx_CR1 CPHA =0，CPOL 设置被忽略）时才生效。当激活时，当 NSS 保持高电平至少持续一个时钟周期时，在两个连续的数据帧传输之间产生一个 NSS 脉冲。这种模式允许从机锁存数据。NSSP 脉冲模式是为单主从对应用而设计的。

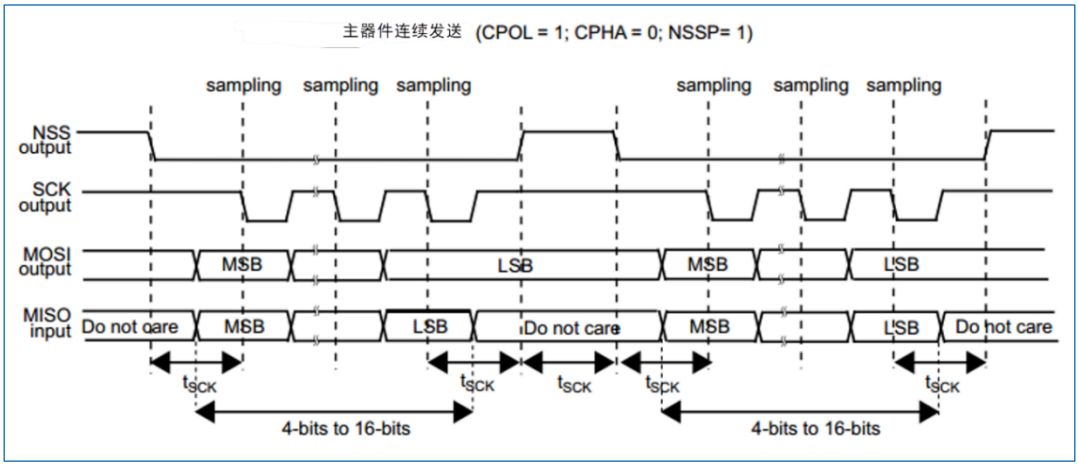


图 20-10 SPI Motorola 主模式 NSS 脉冲生成

说明：当 CPOL=0 时也会遇到类似的行为。在这种情况下，采样边是上升的，NSS 断言和去断言指的就是这个采样边。

20.4.13 TI 模式

主模式下的 TI 协议

SPI 接口与 TI 协议兼容。可以使用 SPIx_CR2 寄存器的 FRF 位来配置 SPI，以兼容此协议。

时钟极性和相位都被强制遵循 TI 协议，和 SPIx_CR1 中的设置无关。NSS 管理也特定于 TI 协议，在这种情况下，无法通过 SPIx_CR1 和 SPIx_CR2 寄存器（SSM、SSI 和 SSOE）来对 NSS 管理进行配置。

在从模式下，SPI 波特率预分频器用于控制在当前传输完成时 MISO 引脚切换为高阻态的时刻（请参见图 19-11）。可以使用任意波特率，因此可以非常灵活地确定此时刻。但是，波特率通常设置为外部主时钟波特率。MISO 信号变为高阻态的延时（trelease）取决于内部重新同步以及通过 SPIx_CR1 寄存器的 BR[2:0]位设置的波特率值。具体公式如下：

$$\frac{t_{baud_{rate}}}{2}+4*t_{pclk}<t_{release}<\frac{t_{baud_{rate}}}{2}+6*t_{pclk}$$

如果从器件在数据帧传输期间检测到错位的 NSS 脉冲，TIFRE 标志将置 1。

如果数据大小为 4 位或 5 位，则在全双工模式或只发送模式下，主机使用的协议在 LSB 之后再增加一个虚拟数据位。TI NSS 脉冲是在这个虚拟位时钟周期以上产生的，而不是在每个周期的 LSB。

此特性不适用于 MotorolaSPI 通信（FRF 位为 0）。

图 19-11 给出了选择 TI 模式时的 SPI 通信波形。

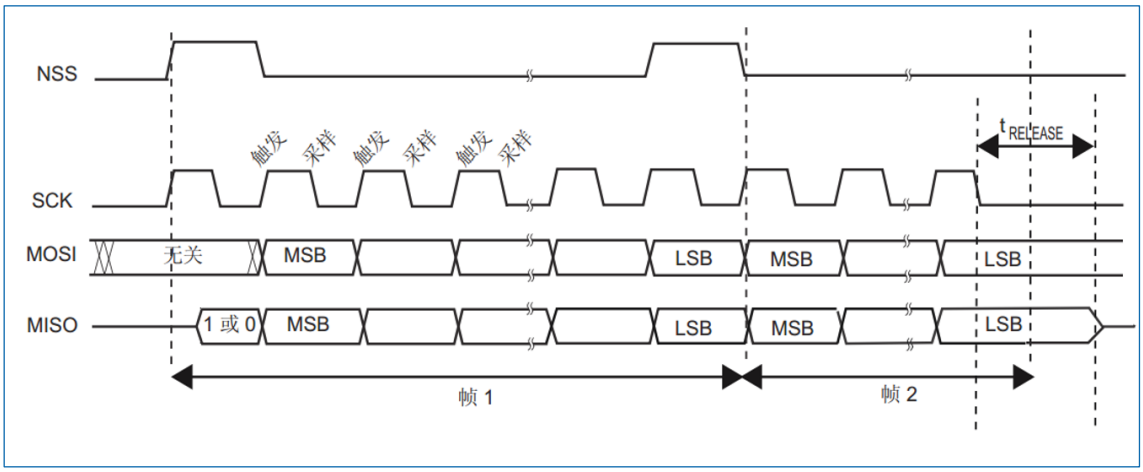


图 20-11 TI 模式传输

20.4.14 CRC 计算

为检查发送数据和接收数据的可靠性，使用两个独立的 CRC 计算器（作用于发送和接收数据流）。SPI 提供 CRC8 或 CRC16 计算，具体取决于通过 CRCL 位选择的数据格式。对于其他长度的数据帧，没有 CRC 可用。

CRC 原理

在使能 SPI（SPE=1）前，通过将 SPIx_CR1 寄存器中的 CRCEN 位置 1 来使能 CRC 计算。使用值为奇数的可编程多项式对每个位计算 CRC 值。在由 SPIx_CR1 寄存器中的 CPHA 位和 CPOL 位定义的采样时钟边沿进行计算。所计算的 CRC 值在数据块末尾自动进行校验，以及针对由 CPU 管理的传输进行校验。当检测到所接收数据内部计算的 CRC 与发送器发送的 CRC 不匹配时，CRCERR 标志将置 1 以指示数据损坏错误。CRC 计算的正确处理步骤取决于 SPI 配置和所选的传输管理。

说明：多项式值只应为奇数。不支持任何偶数值。

CPU 管理的 CRC 传输

通信开始后将一直持续到必须发送或接收 SPIx_DR 寄存器中的最后一个数据帧时。之后，SPIx_CR1 寄存器中的 CRCNEXT 位必须置 1，以指示当前处理的数据帧传输后将处理 CRC 帧传输。CRCNEXT 位必须在最后一个数据帧传输结束前置 1。在 CRC 传输期间，CRC 计算将冻结。

与任何其它数据帧一样，接收的 CRC 存储在 RXFIFO 中，这就是为什么在 CRC 模式下，接收缓冲区必须被认为是一个用于一次只接收一个数据帧的单独的 16 位缓冲区。

CRC 格式的事务在数据序列的末尾通常需要多一个数据帧通信，然而，当设置一个由 16 位 CRC 检查的 8 位数据帧时，需要另外两个帧来发送完整的 CRC 数据。

接收最后一个 CRC 数据后，将执行自动校验，将接收的值与 SPIx_RXCRCR 寄存器中的值进行比较。软件必须校验 SPIx_SR 寄存器中的 CRCERR 标志，以确定数据传输是否损坏。软件通过将 CRCERR 标志写入“0”来将其清零。

接收 CRC 后，CRC 值存储到 RXFIFO 中，且必须在 SPIx_DR 寄存器中进行读取，以将 RXNE 标志清零。

20.5 SPI 中断

在 SPI 通信过程中，中断可由以下事件产生：

- 发送缓冲区准备就绪，可以装载数据。
- 接收缓冲区中接收了数据。
- 主模式故障
- 上溢错误
- TI 帧格式错误
- CRC 校验错误

中断可分别进行使能和禁止。

表 20-2 SPI 中断请求

中断事件	事件标志	使能控制位
发送缓冲区准备就绪，可以装载数据	TXE	TXEIE
接收缓冲区中接收了数据	RXNE	RXNEIE
主模式故障	MODF	ERRIE
上溢错误	OVR	
CRC 错误	CRCERR	
TI 帧格式错误	FRE	

20.6 SPI 接口特性

表 20-3 中给出的 SPI 参数源自在环境温度下进行的试验。图 20-12、图 20-13 和图 20-14 分别描述了 SPI 在主模式下的时序图、在从模式下的时序图（CPHA=0）和在从模式下的时序图（CPHA=1）。

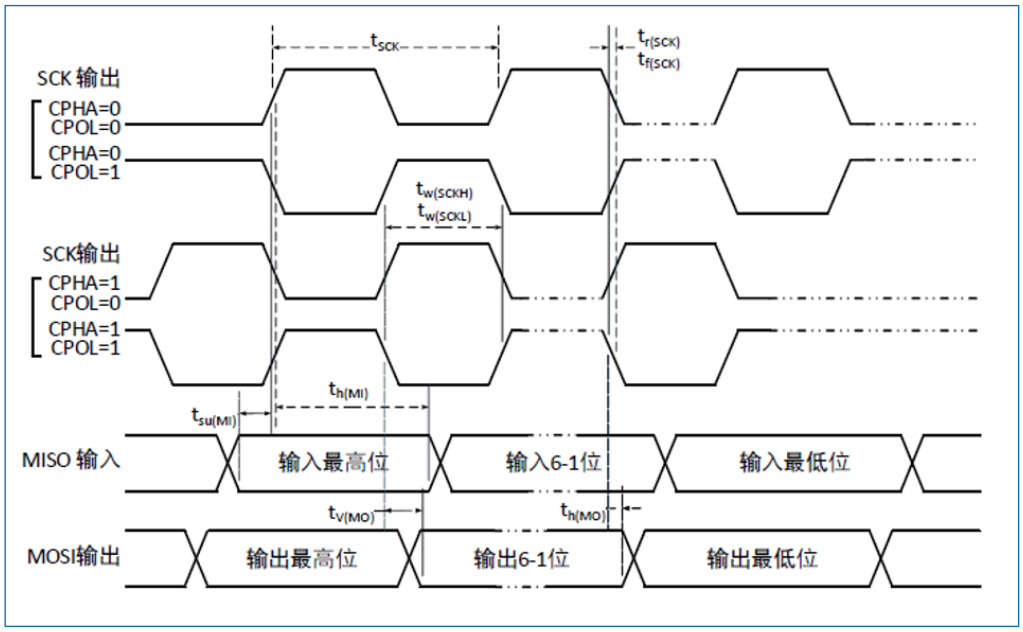


图 20-12 SPI 主模式时序图

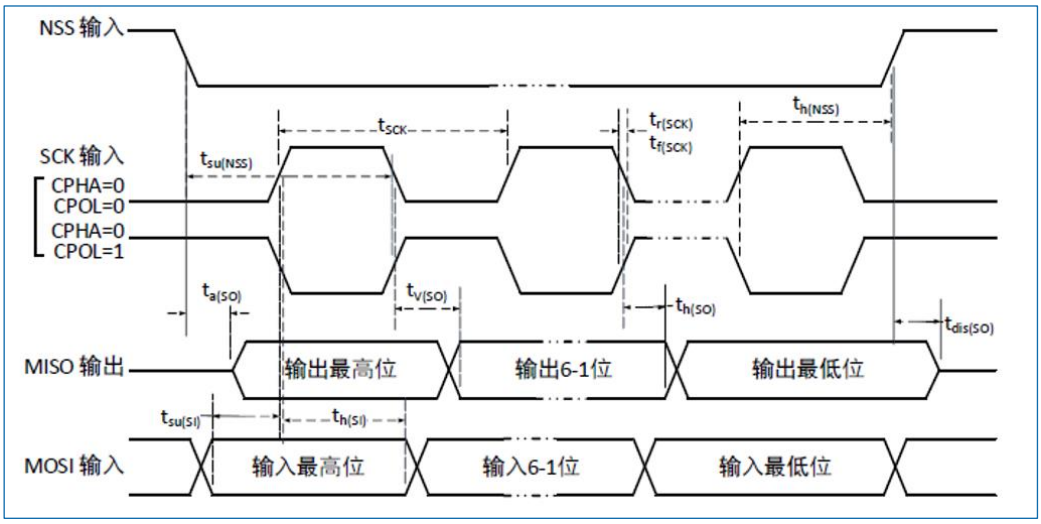


图 20-13 SPI 从模式时序图（CPHA=0）

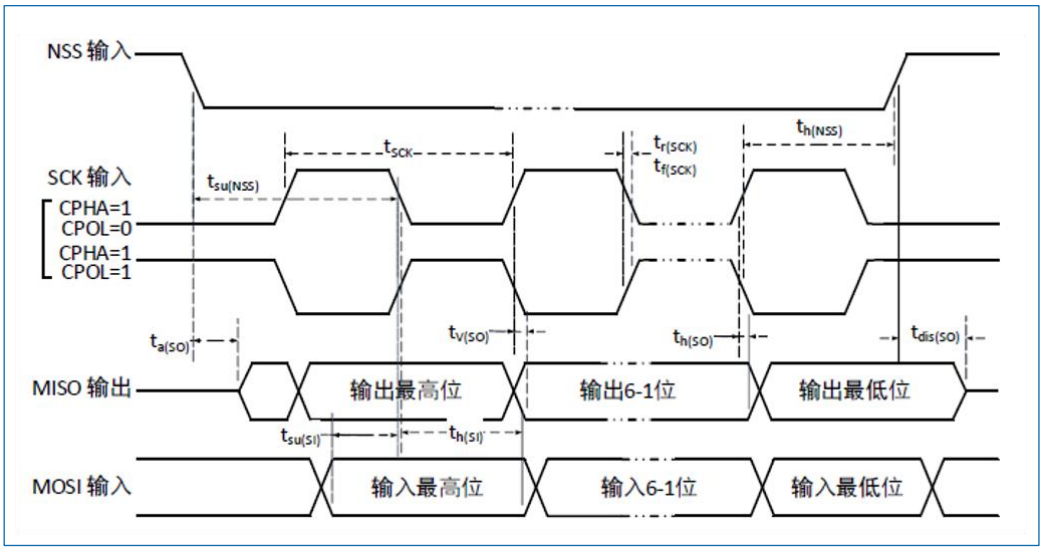


图 20-14 SPI 从模式时序图（CPHA=1）

表 20-3 SPI 接口特性

符号	参数	条件	最小值	最大值	单位
f_{SCK}/t_{SCK}	SPI 时钟频率	主模式	-	16	MHz
		从模式	-	16	MHz
$t_{r(SCK)}/t_{f(SCK)}$	SPI 时钟上升和下降时间	负载电容：C=30pF	-	6	ns
$t_{SU(NSS)}$	NSS 建立时间	从模式	$4t_{PCLK}$	-	ns
$t_{h(NSS)}$	NSS 保持时间	从模式	$2t_{PCLK}+10$	-	ns
$t_{w(SCKH)}/t_{w(SCKL)}$	SCK 高电平和低电平时间	主模式，f_{PCLK}=36MHz，预分配系数=4	$t_{PCLK}/2-2$	$t_{PCLK}/2+1$	ns
$t_{SU(MI)}$	数据输入建立时间	主模式	4	-	ns
$t_{SU(SI)}$		从模式	5	-	ns
$t_{h(MI)}$	数据输入保持时间	主模式	4	-	ns
$t_{h(SI)}$		从模式	5	-	ns
$t_{a(SO)}$	数据输出访问时间	从模式，f_{PCLK}=20 MHz	0	$3t_{PCLK}$	ns
tdis(SO)	数据输出禁止时间	从模式	0	18	ns
tv(SO)	数据输出有效时间	从模式（使能边沿之后）	-	22.5	ns
tv(MO)		主模式（使能边沿之后）	-	6	ns
th(SO)	数据输出保持时间	从模式（使能边沿之后）	11.5	-	ns
th(MO)		主模式（使能边沿之后）	2	-	ns

20.7 I2S 功能说明

20.7.1 I2S 概述

I2S 的框图下图所示。

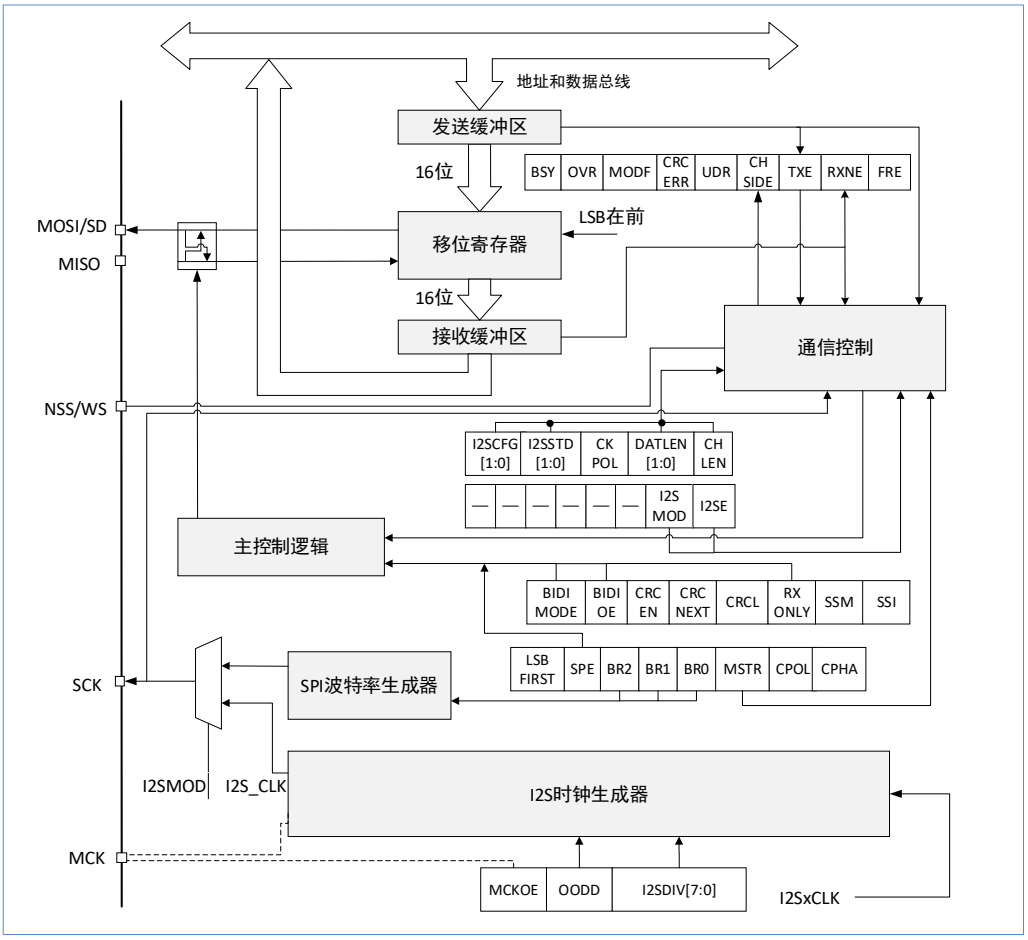


图 20-15 I2S 框图

(1). MCK 映射到 MISO 引脚上。

当使能 I2S 功能（将 SPIx_I2SCFGR 寄存器中的 I2SMOD 位置 1）后，SPI 可用作音频 I2S 接口。此接口使用几乎与 SPI 相同的引脚、标志和中断。

I2S 与 SPI 共用以下三个引脚：

- SD：串行数据（映射到 MOSI 引脚），用于发送或接收两个时分复用的数据通道上的数据（仅半双工模式）。
- WS：字选择（映射到 NSS 引脚），是主模式下的数据控制信号输出以及从模式下的数据控制信号输入。
- CK：串行时钟（映射到 SCK 引脚），是主模式下的串行时钟输出以及从模式下的串行时钟输入。

当某些外部音频设备需要使用主时钟输出时，可以使用其它引脚：

- MCK：当 I2S 配置为主模式（并且 SPIx_I2SPR 寄存器中的 MCKOE 位置 1）时，使用主时钟（单独映射）输出此附加时钟，该时钟以 256×fs 的预配置频率生成，其中 fs 为音频信号采样频率。

I2S 在主模式下使用自身的时钟发生器生成通信时钟。此时钟发生器也是主时钟输出的源。

在 I2S 模式下可以使用两个额外的寄存器。一个是时钟发生器配置寄存器 SPIx_I2SPR，另一个是通用 I2S 配置寄存器 SPIx_I2SCFGR（音频标准、从/主模式、数据格式、数据包帧、时钟极性等）。

在 I2S 模式下不使用 SPIx_CR1 寄存器和所有 CRC 寄存器。同样，也不使用 SPIx_CR2 寄存器中的 SSOE 位以及 SPIx_SR 中的 MODF 和 CRCERR 位。

I2S 使用相同的 SPI 寄存器（SPIx_DR）进行 16 位数据传输。

20.7.2 I2S 全双工

图 20-16 全双工通信显示了如何使用两个 SPI/I2S 接口进行全双工通信。此时，两个 SPI/I2S 的 WS 和 CK 引脚必须连接在一起。

对于主机全双工模式，一个 SPI/I2S 块必须在主模式(I2SCFG='10'或'11')中编程，另一个 SPI/I2S 块必须在从(I2SCFG='00'或'01')中编程。根据应用程序的需要，可以决定是否生成 MCK。。

对于从机全双工模式，两个 SPI/I2S 块必须在从模式编程。其中一个在从接收模式(I2SCFG='01')，另一个在从发送模式(I2SCFG='00')。由外部主设备提供位时钟(CK)和帧同步(WS)。

请注意，全双工模式可用于所有支持的标准:I2S Philips、mbs-justified、LSB-justified 和 PCM。

对于全双工模式，两个 SPI/I2S 接口必须使用相同的标准，具有相同的参数:I2SMOD、I2SSTD、CKPOL、PCMSYNC、DATLEN 和 CHLEN 必须在两个实例上包含相同的值。

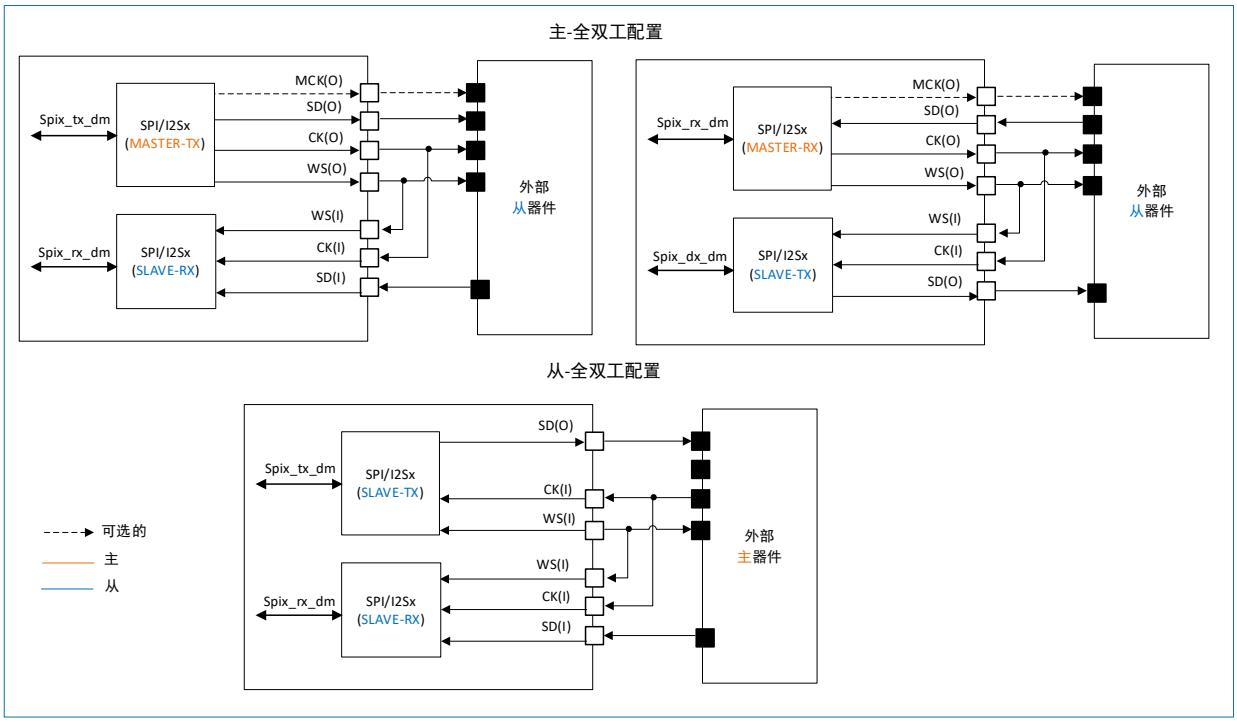


图 20-16 全双工通信

20.7.3 支持的音频协议

三线总线仅需要处理在左右两个通道上时分复用的音频数据。但是，只有一个 16 位寄存器进行发送和接收。所以，需由软件将与每个通道对应的值写入数据寄存器，以及从数据寄存器中读取数据，并通过检查 SPIx_SR 寄存器中的 CHSIDE 位来识别对应的通道。始终先发送左通道数据，而后再发送右通道数据（对于 PCM 协议来说，CHSIDE 没有意义）。

数据和帧格式组合有四种，可采用下列格式发送数据：

- 将 16 位数据封装在 16 位帧中。
- 将 16 位数据封装在 32 位帧中。
- 将 24 位数据封装在 32 位帧中。
- 将 32 位数据封装在 32 位帧中。

当使用 32 位数据包中的 16 位数据时，前 16 位（MSB）为有效位，16 位 LSB 被强制清零，无需任何软件操作（只需一个读/写操作）。

对于所有数据格式和通信标准而言，始终会先发送最高有效位（MSB 在前）。

I2S 接口支持四种音频标准，可使用 SPIx_I2SCFGR 寄存器中的 I2SSTD[1:0]和 PCMSYNC 位对其进行配置。

I2S Philips 标准

使用 WS 信号来指示当前正在发送的数据所属的通道。该信号从当前通道数据的第一个位（MSB）之前的一个时钟开始有效。

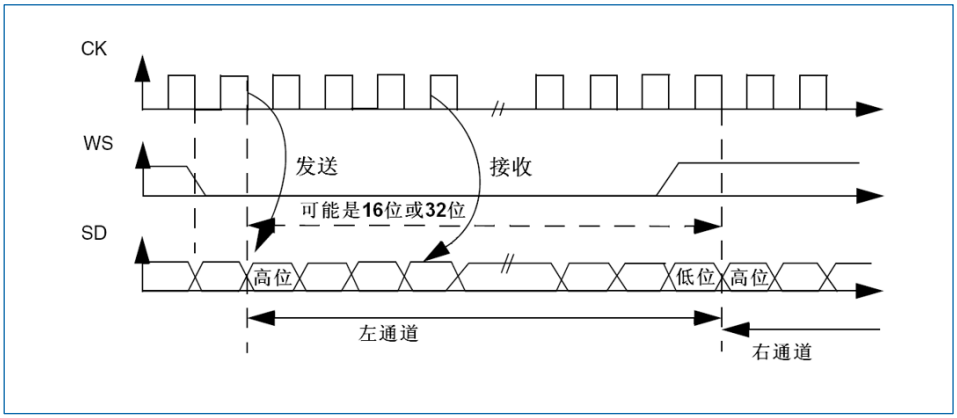


图 20-17 I2S Philips 协议波形（16/32 位全精度，CPOL=0）

发送方在时钟信号（CK）的下降沿改变数据，接收方在上升沿读取数据。WS 信号也在 CK 的下降沿变化。

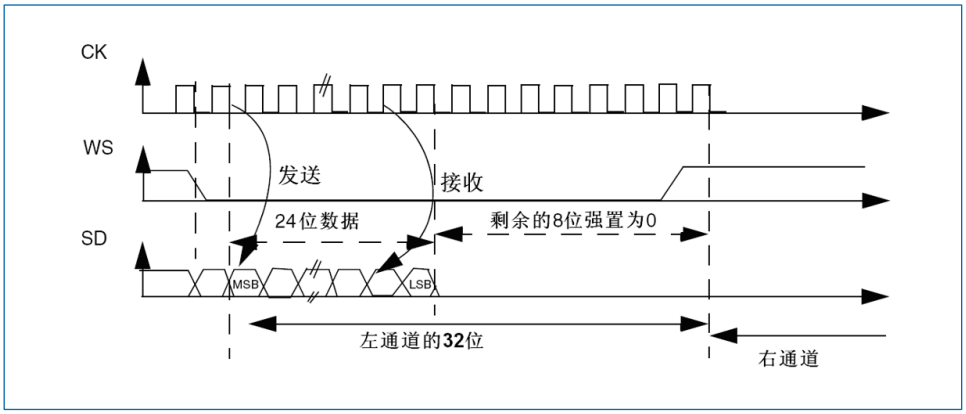


图 20-18 I2S Philips 标准波形（24 位帧，CPOL=0）

该模式需要对 SPIx_DR 寄存器执行两次写入或读取操作。

- 在发送模式下

如果需要发送 0x8EAA33（24 位）：

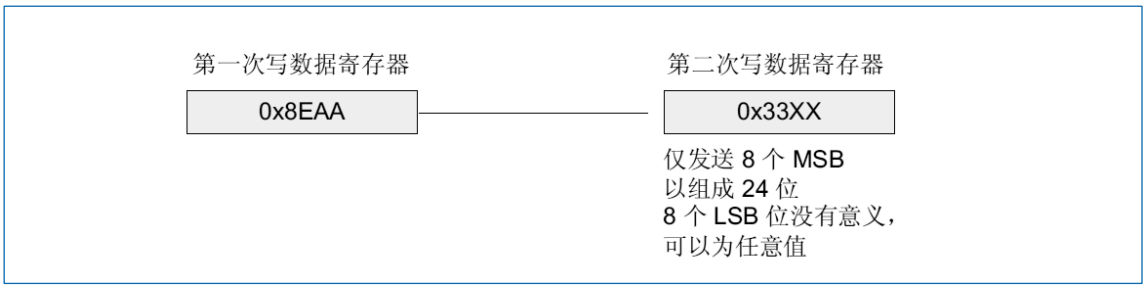


图 20-19 发送 0x8EAA33

- 在接收模式下

如果接收数据 0x8EAA33：

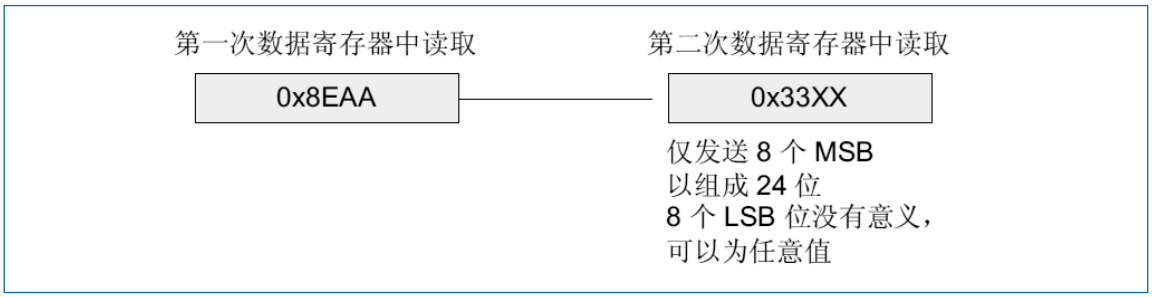


图 20-20 接收 0x8EAA33

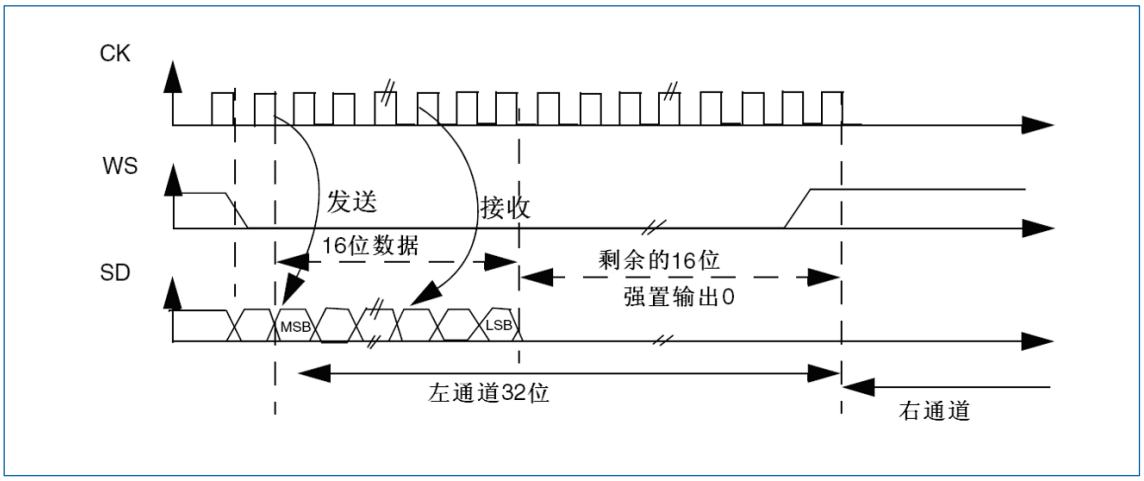


图 20-21 I2S Philips 标准（16 位扩展为 32 位数据包帧，CPOL=0）

如果在 I2S 配置阶段选择将 16 位数据帧扩展到 32 位通道帧，则只需要访问一次 SPIx_DR 寄存器。扩展到 32 位中高半字（16 位 MSB）被硬件置为 0x0000。

如果要发送的数据或已接收的数据为 0x76A3（0x76A30000 扩展为 32 位），则需要执行图 20-22 中显示的操作。

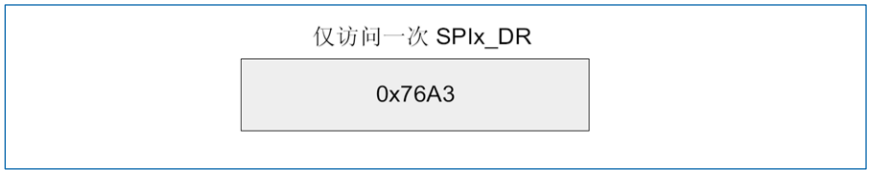


图 20-22 16 位数据帧扩展到 32 位通道帧的示例

发送时，每次将 MSB 写入 SPIx_DR，TXE 标志就会置 1，并在中断使能的情况下触发中断，以将要发送的新数据加载到 SPIx_DR 寄存器。即使硬件填充的低 16 位 0x0000 还未发送，也会如此，因为低 16 位是由硬件发送。

接收时，接收到第一个半字（高 16 位），则硬件将 RXNE 标志置 1，并在中断使能的情况下触发中断。

这样，就延长了两个写入或读取操作之间的时间间隔，从而可防止出现下溢或上溢情况（具体取决于数据传输方向）。

MSB 对齐标准

此标准同时生成 WS 信号和第一个数据位（即 MSB）。

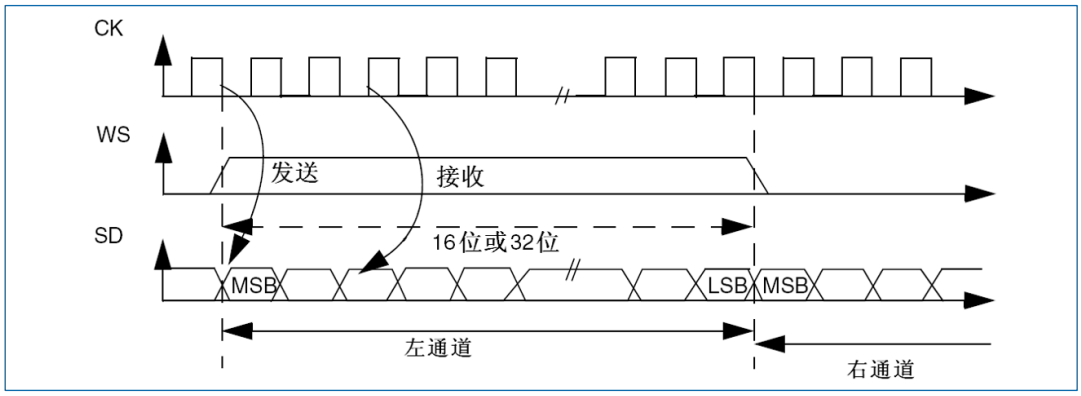


图 20-23 MSB 对齐的 16 位或 32 位全精度长度，CPOL=0

发送方在时钟信号的下降沿改变数据；接收方在上升沿读取数据。

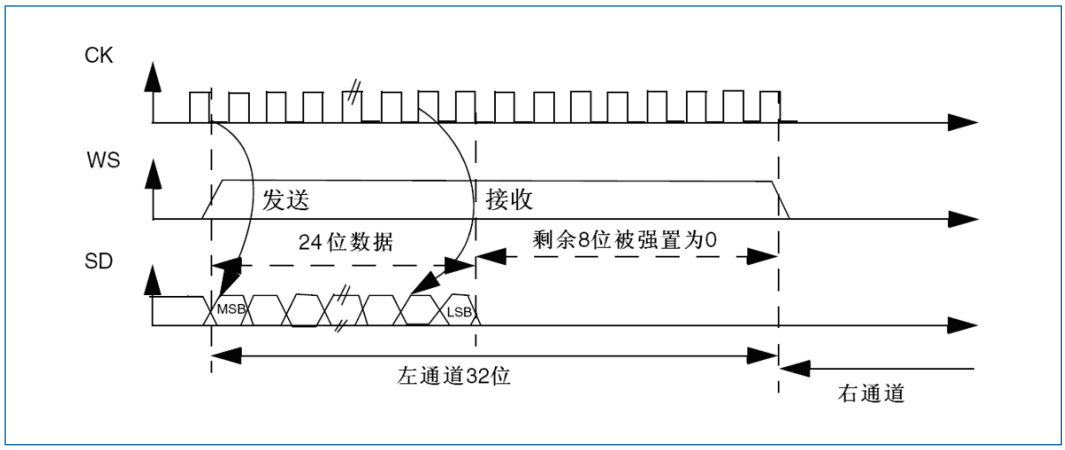


图 20-24 MSB 对齐的 24 位帧长度，CPOL=0

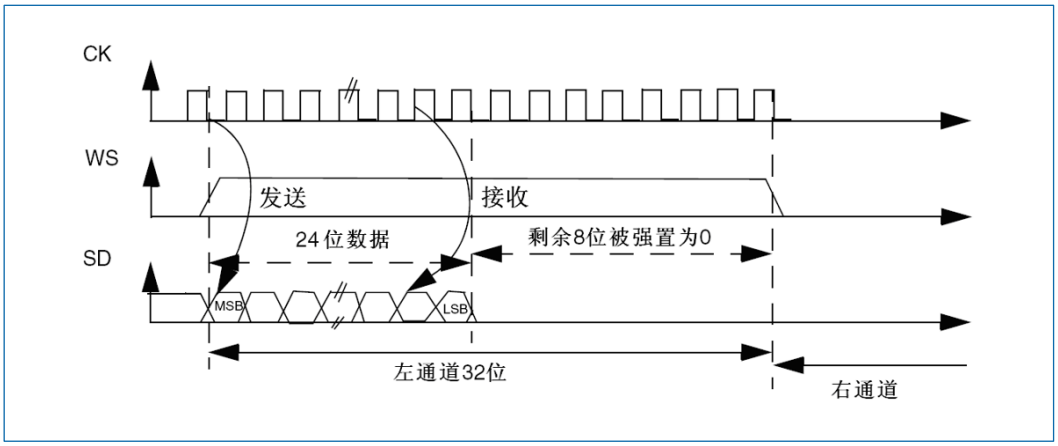


图 20-25 MSB 对齐 16 位数据扩展到 32 位数据包帧，CPOL=0

LSB 对齐标准

该标准与 MSB 对齐标准类似（对于 16 位和 32 位全精度帧格式，没有任何不同）。

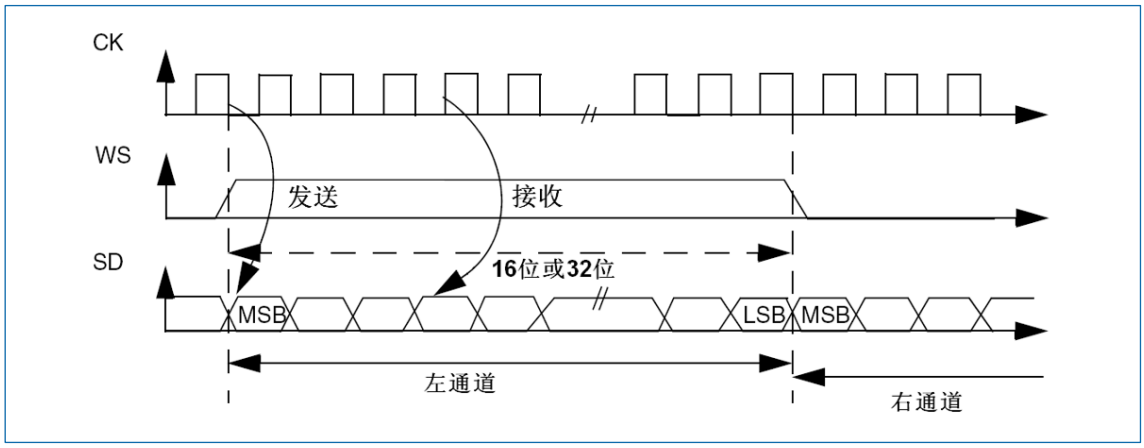


图 20-26 LSB 对齐的 16 位或 32 位全精度，CPOL=0

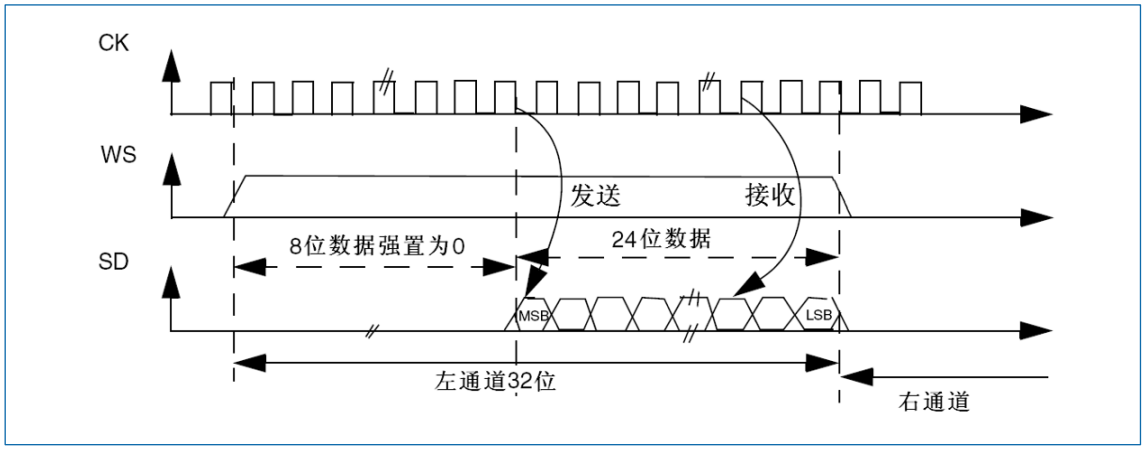


图 20-27 LSB 对齐的 24 位帧长度，CPOL=0

- 在发送模式下

如果需要发送数据 0x3478AE，则需要通过软件对 SPIx_DR 寄存器执行两次写入操作。下面给出了这些操作。

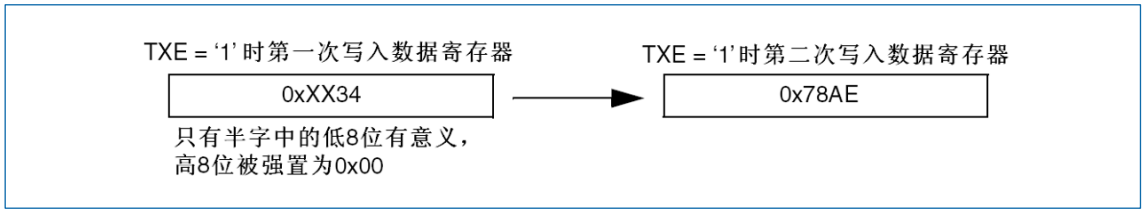


图 20-28 发送 0x3478AE 所需的操作

- 在接收模式下

如果接收到数据 0x3478AE，则在每个 RXNE 事件时需要对 SPIx_DR 寄存器执行两次连续的读取操作。

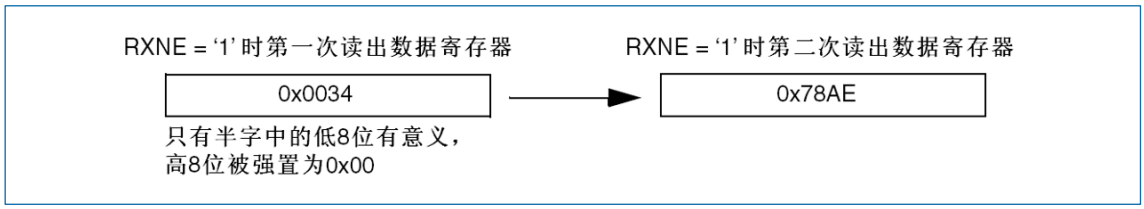


图 20-29 接收 0x3478AE 时所需的操作

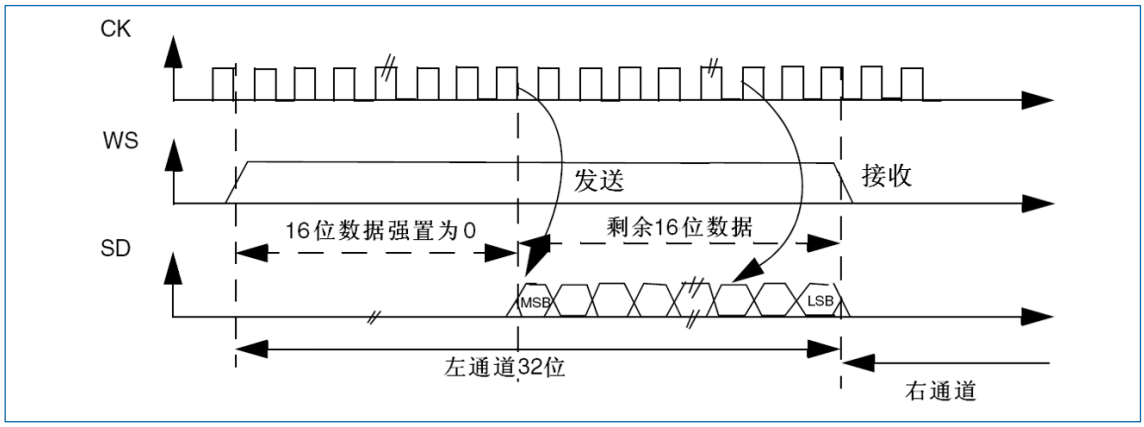


图 20-30 扩展为 32 位数据包帧的 LSB 对齐的 16 位，CPOL=0

如果在 I2S 配置阶段选择将 16 位数据帧扩展到 32 位通道帧，则只需要访问一次 SPIx_DR 寄存器。扩展到 32 位中高半字（16 位 MSB）被硬件置为 0x0000。在这种情况下，其对应于半字 MSB。

如果要发送的数据或已接收的数据为 0x76A3（0x000076A3 扩展为 32 位），则需要执行图 20-31 中显示的操作。

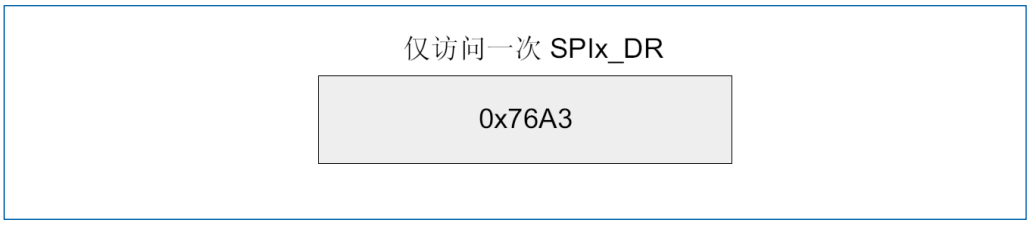


图 20-31 16 位数据帧扩展到 32 位通道帧的示例

在发送模式下，发生 TXE 事件时，应用程序需要写入要发送的数据（此例中为 0x76A3）。

首先发送 0x000 字段（扩展到 32 位）。有效数据（0x76A3）发送到 SD 后，TXE 标志会被再次置 1。

在接收模式下，当接收到有效半字后（而非 0x0000 字段），即会置位 RXNE。

这样，就延长了两个写入或读取操作之间的时间间隔，以防止出现下溢或上溢情况。

PCM 标准

对于 PCM 标准，无需使用通道信息。可使用两种 PCM 模式（短帧和长帧），并且可使用 SPIx_I2SCFGR 寄存器中的 PCMSYNC 位来配置。

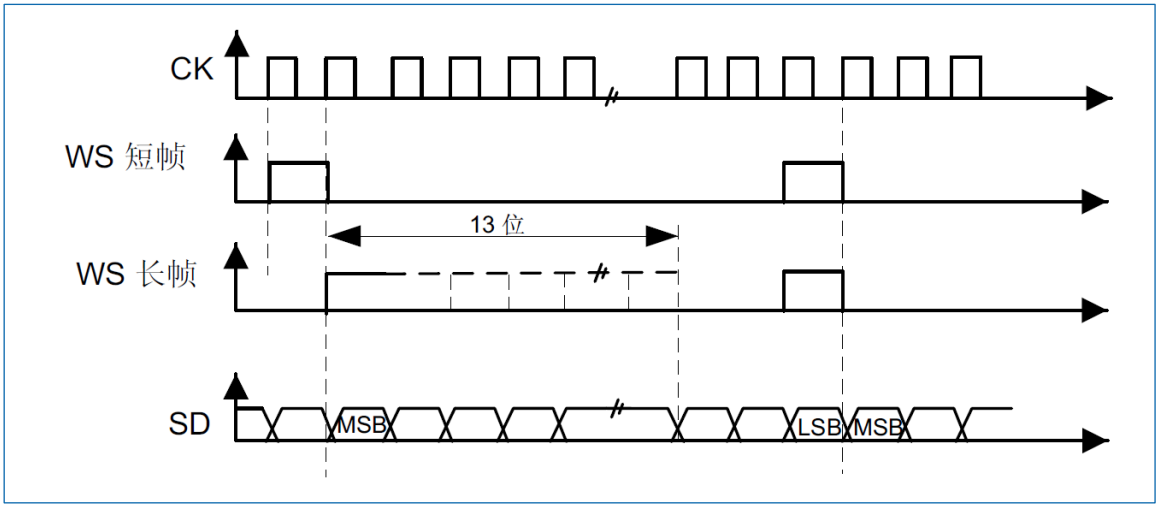


图 20-32 PCM 标准波形（16 位）

对于长帧同步，在主模式下会将 WS 信号持续 13 个周期。

对于短帧同步，WS 同步信号的持续时间仅为一个周期。

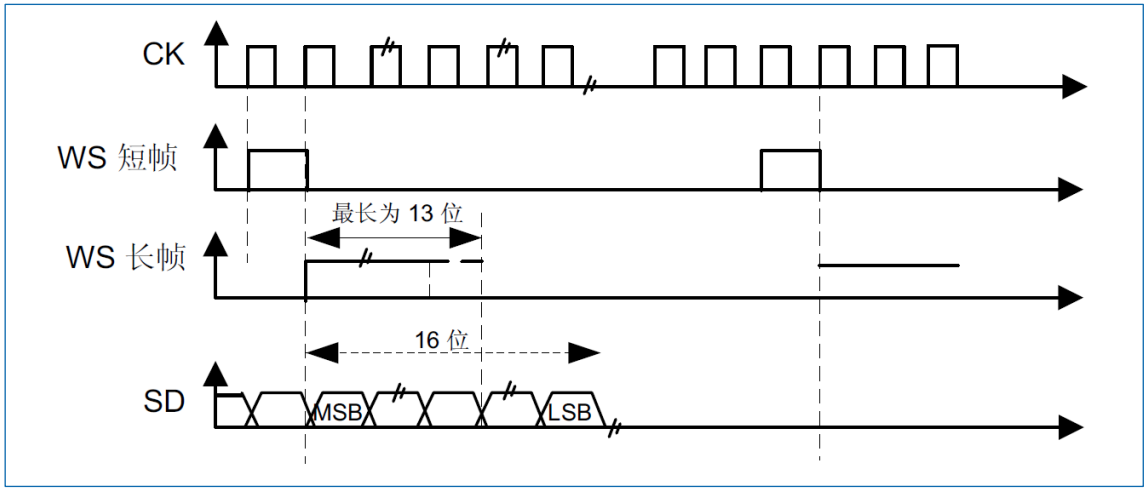


图 20-33 PCM 标准波形（16 位扩展到 32 位数据包帧）

说明：对于两种模式（主/从模式）和两种同步（短/长同步），即使在从模式下，也需要指定两组连续数据（以及两个同步信号）之间位的个数（SPIx_I2SCFGR 寄存器中的 DATLEN 位和 CHLEN 位）。

20.7.4 启动描述

图 20-34 显示了在 MASTER 模式下如何处理串行接口，当 I2S 被启用时(I2SE=1)。同时也显示了 CKPOL 对生成信号的影响。

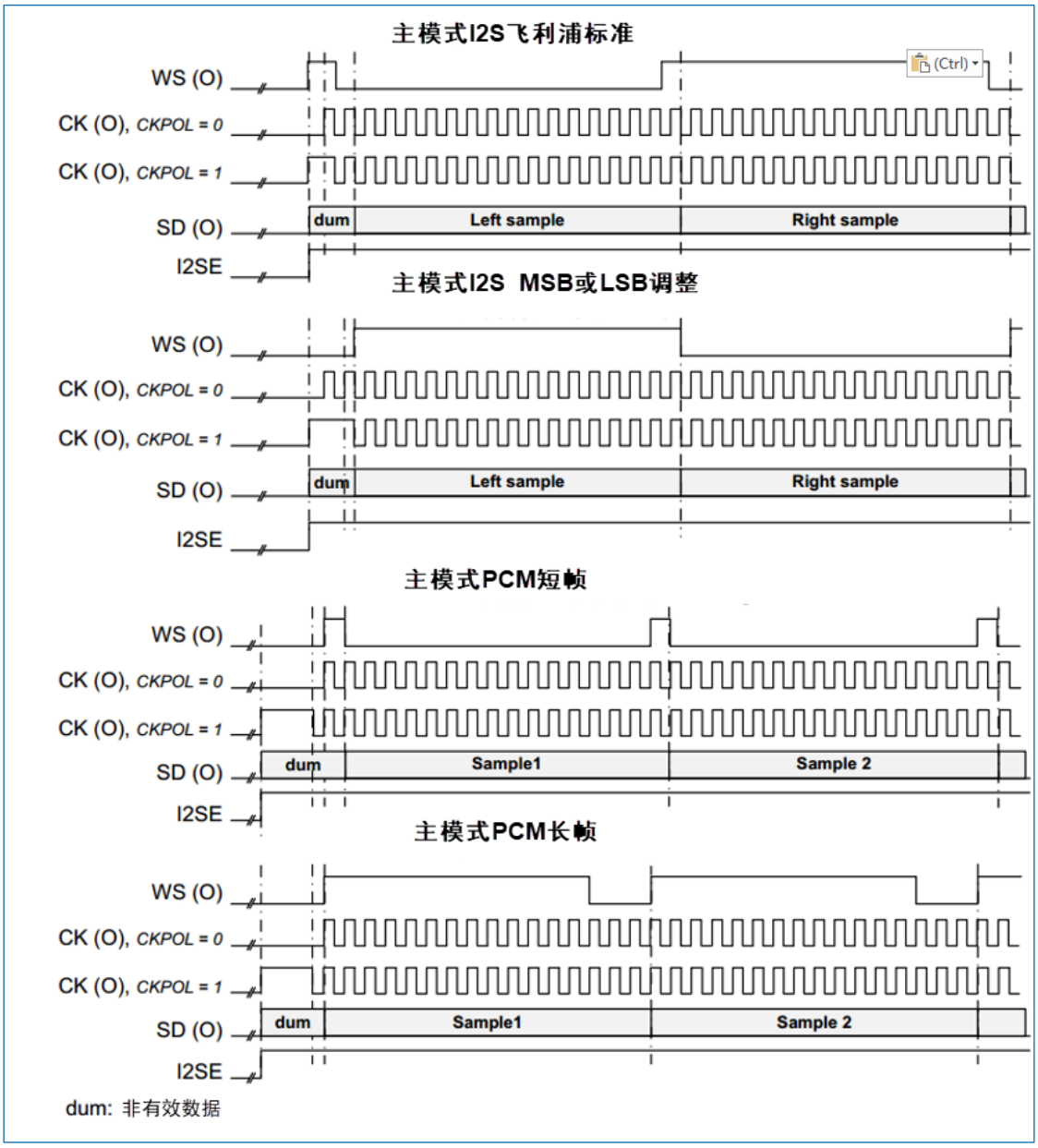


图 20-34 主模式下启动序列

在从模式下，用户必须在 WS 变得活跃之前启用音频接口。

这意味着当飞利浦标准 I2S 的 WS = 1 时，I2SE 位必须设置为 1，或者当其他标准的 WS = 0。

20.7.5 时钟发生器

I2S 比特率用来确定 I2S 数据线上的数据流和 I2S 时钟信号频率。

I2S 比特率=每个通道的位数×通道数×音频采样频率

- 对于 16 位双通道音频：I2S 比特率=16 × 2 × fs
- 如果数据包为 32 位宽：I2S 比特率=32 x 2 x fs

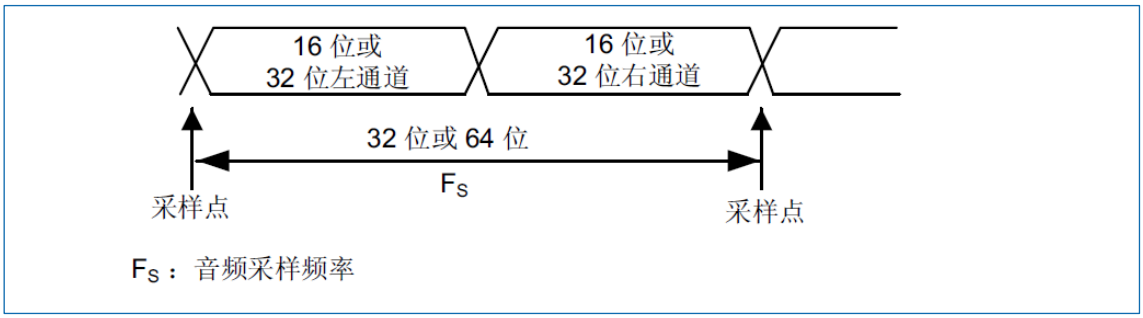


图 20-35 音频采样频率定义

配置主模式时，需要正确地对线性分频器进行设置，以便采用所需的音频频率进行通信。

图 20-36 展示了通信时钟架构。I2Sx 时钟始终为系统时钟。

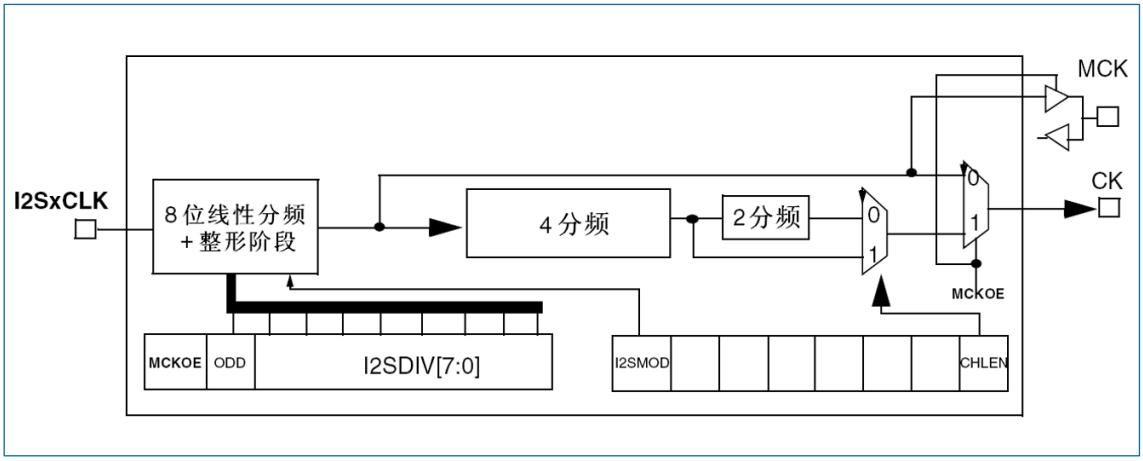


图 20-36 I2S 时钟发生器架构

上图说明：图中 x=2。

音频采样频率可以是 192kHz、96kHz、48kHz、44.1kHz、32kHz、22.05kHz、16kHz、11.025kHz 或 8kHz（或此范围内的任何其它值）。为达到所需频率，需要根据以下公式对线性分频器进行编程：

- 输出主时钟（SPIx_I2SPR 寄存器中的 MCKOE 置 1）时：

fs=I2SxCLK/[（16*2）*（（2*I2SDIV）+ODD）*8）]（通道帧宽度为 16 位时）

fs=I2SxCLK/[（32*2）*（（2*I2SDIV）+ODD）*4）]（通道帧宽度为 32 位时）

- 禁止主时钟输出（MCKOE 位清零）时：

fs=I2SxCLK/[（16*2）*（（2*I2SDIV）+ODD）]（通道帧宽度为 16 位时）

fs=I2SxCLK/[（32*2）*（（2*I2SDIV）+ODD）]（通道帧宽度为 32 位时）

表 20-4 提供了针对不同时钟配置的示例精度值。

说明：还可以采用其它配置以达到更好的时钟精度。

表 20-4 使用标准 8MHz GPIO 外部输入时钟时的音频频率精度

SYSCLK（MHz）	数据长度	I2SDIV	I2SODD	MCLK	目标 fs（Hz）	实际 fs（kHz）	误差
32	16	5	0	无	96000	100	4.1667%
32	32	2	0	无	96000	100	4.1667%
32	16	10	1	无	48000	47.619	0.7937%

SYSCLK（MHz）	数据长度	I2SDIV	I2SODD	MCLK	目标 fs（Hz）	实际 fs（kHz）	误差
32	32	5	0	无	48000	50	4.1667%
32	16	11	1	无	44100	43.478	1.4098%
32	32	5	1	无	44100	45.454	3.0715%
32	16	15	1	无	32000	32.258	0.8065%
32	32	8	0	无	32000	31.25	2.3430%
32	16	22	1	无	22050	22.222	0.7811%
32	32	11	1	无	22050	21.739	1.4098%
32	16	31	1	无	16000	15.873	0.7937%
32	32	15	1	无	16000	16.129	0.8065%
32	16	45	1	无	11025	10.989	0.3264%
32	32	22	1	无	11025	11.111	0.7811%
32	16	62	1	无	8000	8	0.0000%
32	32	31	1	无	8000	7.936	0.7937%
32	16	2	0	有	32000	31.25	2.3430%
32	32	2	0	有	32000	31.25	2.3430%
32	16	3	0	有	22050	20.833	5.5170%
32	32	3	0	有	22050	20.833	5.5170%
32	16	4	0	有	16000	15.625	2.3428%
32	32	4	0	有	16000	15.625	2.3428%
32	16	5	1	有	11025	11.363	3.0715%
32	32	5	1	有	11025	11.363	3.0715%
32	16	8	0	有	8000	7.812	2.3428%
32	32	8	0	有	8000	7.812	2.3428%

20.7.6 I2S 主模式

I2S 可配置为主模式。这意味着将在 CK 引脚输出串行时钟，在 WS 引脚生成字选信号。主时钟（MCK）可以输出，也可以不输出，具体由 SPIx_I2SPR 寄存器中的 MCKOE 位控制。

主模式配置步骤

1. 设置 SPIx_I2SPR 寄存器的 I2SDIV[7:0]位，以定义串行时钟波特率，从而达到相应的音频采样频率。SPIx_I2SPR 寄存器的 ODD 位也需要设置。
2. 设置 CKPOL 位，定义时钟在空闲时的电平状态。如果需要为外部 DAC/ADC 音频组件提供主时

钟 MCK，则将 SPIx_I2SPR 寄存器的 MCKOE 位置 1（I2SDIV 和 ODD 值应根据 MCK 输出的状态进行计算。有关详细信息，请参见“20.7.5 时钟发生器”）。

3. 将 SPIx_I2SCFGR 寄存器中的 I2SMOD 位置 1 以激活 I2S 功能，通过 I2SSTD[1:0]和 PCMSYNC 位选择 I2S 标准，通过 DATLEN[1:0]位选择数据长度并通过配置 CHLEN 位选择每个通道的位数。此外，通过 SPIx_I2SCFGR 寄存器的 I2SCFG[1:0]位选择 I2S 主模式和方向（发送器或接收器）。
4. 如果需要，通过对 SPIx_CR2 寄存器执行写操作来选择所有可能的中断源。
5. SPIx_I2SCFGR 寄存器的 I2SE 位必须置 1。

WS 和 CK 配置为输出模式。如果 SPIx_I2SPR 的 MCKOE 位置 1，则 MCK 也是输出。

发送序列

将半字写入发送缓冲区后，发送序列随即开始。

假设写入发送缓冲区的第一个数据对应于左通道数据。数据从发送缓冲区传输到移位寄存器时，TXE 置 1，并且必须将对应于右通道的数据写入发送缓冲区。CHSIDE 标志指示将发送的数据对应的通道。TXE 标志置 1 时，CHSIDE 标志有意义，因为该标志在 TXE 变为高电平时进行更新。

一个完整帧表示先进行左通道数据发送，再进行右通道数据发送。不存在仅发送左通道的部分帧。

首位发送期间，数据按半字并行加载到16位移位寄存器中，然后以串行方式移位并输出到MOSI/SD引脚（MSB 在前）。每次数据从发送缓冲区传输到移位寄存器后，TXE 标志都将置 1，如果 SPIx_CR2 寄存器的 TXEIE 位置 1，将产生中断。

有关各种 I2S 标准模式的写操作的更多详细信息，请参见“20.7.3 支持的音频协议”。

为确保连续进行音频数据发送，必须在当前数据发送结束前将下一个要发送的数据写入 SPIx_DR 寄存器。

要通过将 I2SE 清零来关闭 I2S，必须等待 TXE=1 且 BSY=0。

接收序列

此工作模式与发送模式相同，只有第 3 点存在不同（请参见“20.7.6 I2S 主模式”所述的步骤），即通过 I2SCFG[1:0]位设置主器件接收模式。

无论数据或通道长度如何，音频数据始终按 16 位数据包进行接收。这意味着，每当接收缓冲区满时，RXNE 标志即置 1，并且如果 SPIx_CR2 寄存器的 RXNEIE 位置 1，还将产生中断。所接收的右通道或左通道的音频值可通过一次或两次接收操作进入接收缓冲区，具体取决于数据和通道长度配置。

读取 SPIx_DR 寄存器即会使 RXNE 位清零。

CHSIDE 在每次接收后进行更新。它由 I2S 单元所产生的 WS 信号触发。

有关各种 I2S 标准模式中读操作的更多详细信息，请参见“20.7.3 支持的音频协议”。

如果在先前收到的数据尚未读取时又接收到新数据，将产生上溢错误并将 OVR 标志置 1。

如果 SPIx_CR2 寄存器的 ERRIE 位置 1，将产生中断以指示该错误。

要关闭 I2S，需要执行特定操作来确保 I2S 正确完成传输周期而不启动新的数据传输。该序列取决于数据和通道长度的配置，以及所选的音频协议模式。在以下情况下：

- 32 位通道长度上扩展的 16 位数据长度（DATLEN=00 且 CHLEN=1），使用 LSB 对齐模式（I2SSTD=10）。
  A. 等待倒数第二个 RXNE=1（n-1）
  B. 然后等待 17 个 I2S 时钟周期（使用软件循环）
  C. 禁止 I2S（I2SE=0）
- 32 位通道长度上扩展的 16 位数据长度（DATLEN=00 且 CHLEN=1），使用 MSB 对齐、I2S 或

PCM 模式（分别为 I2SSTD=00、I2SSTD=01 或 I2SSTD=11）。

A. 等待最后一个 RXNE

B. 然后等待 1 个 I2S 时钟周期（使用软件循环）

C. 禁止 I2S（I2SE=0）

- 对于 DATLEN 和 CHLEN 的所有其它组合，无论通过 I2SSTD 位选择何种音频模式，都将执行以下序列来关闭 I2S：

A. 等待倒数第二个 RXNE=1（n-1）

B. 然后等待一个 I2S 时钟周期（使用软件循环）

C. 禁止 I2S（I2SE=0）

说明：传输期间，BSY 标志保持低电平。

20.7.7 I2S 从模式

对于从配置而言，I2S 可配置为发送模式或接收模式。

此工作模式所遵循的规则与 I2S 主模式配置基本相同。在从模式下，I2S 接口不产生时钟。

时钟和 WS 信号从 I2S 接口所连接的外部主器件输入。这样，用户便不需要配置时钟。

应遵循如下配置步骤：

1. 将 SPIx_I2SCFGR 寄存器的 I2SMOD 位置 1 以选择 I2S 模式，通过 I2SSTD[1:0]位选择 I2S 标准，通过 DATLEN[1:0]位选择数据长度并通过配置 CHLEN 位选择帧中每个通道的位数。此外，通过 SPIx_I2SCFGR 寄存器的 I2SCFG[1:0]位选择从器件的模式（发送或接收）。
2. 如果需要，通过对 SPIx_CR2 寄存器执行写操作来选择所有可能的中断源。
3. SPIx_I2SCFGR 寄存器的 I2SE 位必须置 1。

发送序列

当外部主器件发送时钟并且通过 NSS_WS 信号请求传输数据时，发送序列开始。必须首先使能从器件，然后外部主器件才能开始通信。主器件开始通信前，还必须加载 I2S 数据寄存器。

对于 I2S、MSB 对齐和 LSB 对齐模式，要写入数据寄存器的第一个数据项对应于左通道的数据。通信开始时，数据从发送缓冲区传输到移位寄存器。TXE 标志随即置 1，以请求将右通道的数据写入 I2S 数据寄存器。

CHSIDE 标志指示将发送的数据对应的通道。与主发送模式相比，在从模式下，CHSIDE 由来自外部主器件的 WS 信号触发。这意味着，从器件需要首先为发送第一个数据做好准备，然后主器件才能产生时钟。WS 置位意味着首先发送左通道数据。

说明：必须要在主器件发出的第一个时钟出现在 CK 线上至少 2 个 PCLK 周期之前置位 I2SE。

首位发送期间，数据按半字从内部总线并行加载到 16 位移位寄存器中，然后以串行方式移位并输出到 MOSI/SD 引脚（MSB 在前）。每次数据从发送缓冲区传输到移位寄存器后，TXE 标志都将置 1，如果 SPIx_CR2 寄存器的 TXEIE 位置 1，将产生中断。

说明：仅当 TXE 标志为 1 时，才可以尝试向发送缓冲区写入数据。

有关各种 I2S 标准模式中写操作的更多详细信息，请参见“20.7.3 支持的音频协议”。

为确保连续进行音频数据发送，必须在当前数据发送结束前将下一个要发送数据写入 SPIx_DR 寄存器。如果在数据尚未写入 SPIx_DR 寄存器时下一个数据通信的首个时钟边沿到来，下溢标志将置 1 并可能产生中断。通过这种方式，软件可以获知所传输的数据不正确。如果 SPIx_CR2 寄存器的 ERRIE 位置 1，则当 SPIx_SR 寄存器中的 UDR 标志变为 1 时，将产生中断。这种情况下，必须关闭 I2S 并从左通道开始重新启动数据传输。

要通过将 I2SE 位清零来关闭 I2S，必须等待 TXE=1 且 BSY=0。

接收序列

此工作模式与发送模式相同，只有第 1 点存在不同（请参见“20.7.7 I2S 从模式”所述的步骤），即通过 SPIx_I2SCFGR 寄存器的 I2SCFG[1:0]位设置从器件接收模式。

无论数据长度或通道长度如何，音频数据始终按 16 位数据包进行接收。这意味着，每当接收缓冲区填满时，SPIx_SR 寄存器中的 RXNE 标志即置 1，并且如果 SPIx_CR2 寄存器的 RXNEIE 位置 1，还将产生中断。所接收的右通道或左通道的音频值可能通过一次或两次接收操作进入接收缓冲区，具体取决于数据长度和通道长度配置。

每次接收要从 SPIx_DR 寄存器读取的数据时，CHSIDE 标志都将更新。该标志由外部主器件所管理的外部 WS 线路触发。

读取 SPIx_DR 寄存器即会使 RXNE 位清零。

有关各种 I2S 标准模式的读操作的更多详细信息，请参见“20.7.3 支持的音频协议”。

如果在先前收到的数据尚未读取时又接收到新数据，将产生上溢错误，OVR 标志将置 1。如果 SPIx_CR2 寄存器的 ERRIE 位置 1，将产生中断以指示该错误。

要在接收模式下关闭 I2S，必须在接收到最后一个 RXNE=1 后立即将 I2SE 清零。

说明：外部主器件应能够通过音频通道以 16 位或 32 位数据包发送/接收数据。

20.7.8 I2S 状态标志

应用程序可通过四个状态标志来全面监视 I2S 总线的状态。

忙标志（BSY）

BSY 标志由硬件置 1 和清零（写入此标志没有任何作用）。该标志表示 I2S 通信层的状态。

BSY 置 1 时，表示 I2S 正在忙于通信。在主接收模式（I2SCFG=11）中，BSY 标志的情况例外，该标志在接收期间仍保持低电平。

如果软件需要禁止 I2S，可使用 BSY 标志检测传输是否结束。这可以避免损坏最后传输的数据。为此，必须严格遵循下述步骤。

在传输开始时（I2S 处于主接收器模式时除外），将 BSY 标志置 1。

出现以下情况时，BSY 标志被硬件清零：

- 传输完成时（主发送模式除外，在该模式下通信是连续的）
- 禁止 I2S 时

当通信连续时：

- 在主发送模式下，BSY 标志在所有传输期间均保持高电平。
- 在从模式下，BSY 标志在每次传输之间变为低电平并持续一个 I2S 时钟周期。

说明：请勿使用 BSY 标志处理每次数据发送或接收，最好改用 TXE 标志和 RXNE 标志。

发送缓冲区为空（TXE）

如果此标志置 1，表示发送缓冲区为空，可将要发送的下一个数据加载到其中。发送缓冲区已包含要发送的数据时，TXE 标志复位。禁止 I2S（I2SE 位复位）时，该标志也会复位。

接收缓冲区非空（RXNE）

此标志置 1 时，表示接收缓冲区中存在有效的已接收数据。读取 SPIx_DR 寄存器时，该标志复位。

通道方向（CHSIDE）

在发送模式下，此标志将在 TXE 变为高电平时进行刷新。此标志指示 SD 上要传输的数据所属的通道。如果在从发送模式下发生下溢错误事件，此标志将不可靠，在恢复通信前需要关闭并重新开启 I2S。

在接收模式下，该标志将在 SPIx_DR 中接收数据时进行刷新。它表示接收的数据所属的通道。请注意，如果发生错误（例如 OVR），此标志将失去意义，应通过禁止并重新使能 I2S（根据需要更改配置）来复位。

此标志在 PCM 标准中没有意义（短帧和长帧模式）。

当 SPIx_SR 中的 OVR 或 UDR 标志置 1，并且 SPIx_CR2 中的 ERRIE 位也置 1 时，将产生中断。中断源被清除后，可通过读取 SPIx_SR 状态寄存器来将此中断清除。

20.7.9 I2S 错误标志

I2S 单元共有三个错误标志。

下溢标志（UDR）

在从发送模式下，如果在软件尚未将任何值加载到 SPIx_DR 之前出现第一个数据发送时钟，此标志将置 1。SPIx_I2SCFGR 寄存器中的 I2SMOD 位置 1 时，可以使用此标志。如果 SPIx_CR2 寄存器中的 ERRIE 位置 1，可产生中断。

UDR 位通过 SPIx_SR 寄存器上的读操作进行清零。

上溢标志（OVR）

如果在尚未从 SPIx_DR 寄存器读取上一个数据时又接收到新数据，此标志将置 1。因此，传入的数据将丢失。如果 SPIx_CR2 寄存器中的 ERRIE 位置 1，可产生中断。

这种情况下，将不会用接收到的新数据更新接收缓冲区的内容。对 SPIx_DR 寄存器执行的读操作将返回先前正确接收的数据。主器件后续发送的所有其它半字都将丢失。

要将 OVR 位清零，应首先对 SPIx_DR 寄存器执行读操作，然后再对 SPIx_SR 寄存器进行读访问。

帧错误标志（FRE）

仅当 I2S 配置为从模式时，此标志才可由硬件置 1。如果外部主器件没有按照从器件期望的那样改变 WS 线，则此标志将置 1。如果失去同步，要从此状态中恢复并将外部主器件与 I2S 从器件重新同步，需执行下列步骤：

1. 禁止 I2S。
2. 在 WS 线上检测到正确的电平时将其重新使能（WS 线在 I2S 模式下为高电平，在 MSB 对齐、LSB 对齐或 PCM 模式下为低电平）。

主器件与从器件之间的同步失效可能是由于 SCK 通信时钟或 WS 帧同步信号线上存在噪音干扰。如果 ERRIE 位置 1，可产生错误中断。读取状态寄存器时，同步失效标志（FRE）由软件清零。

20.8 I2S 中断

表 20-5 为 I2S 中断的列表。

表 20-5 I2S 中断请求

中断事件	事件标志	使能控制位
发送缓冲区为空	TXE	TXEIE
接收缓冲区非空	RXNE	RXNEIE
上溢错误	OVR	ERRIE

中断事件	事件标志	使能控制位
下溢错误	UDR	
帧错误	FRE	

20.9 I2S 接口特性

表 20-6 中给出的 I2S 参数源自在环境温度下进行的试验。图 20-37 和图 20-38 分别描述了 I2S 基于飞利浦协议在主模式下的时序图和在从模式下的时序图。

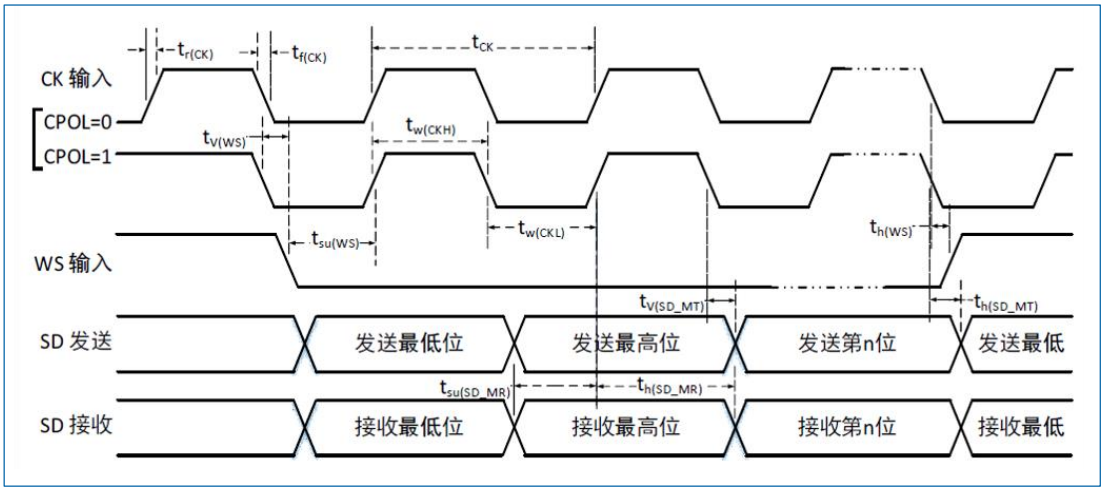


图 20-37 I2S 总线主模式时序图（飞利浦协议）

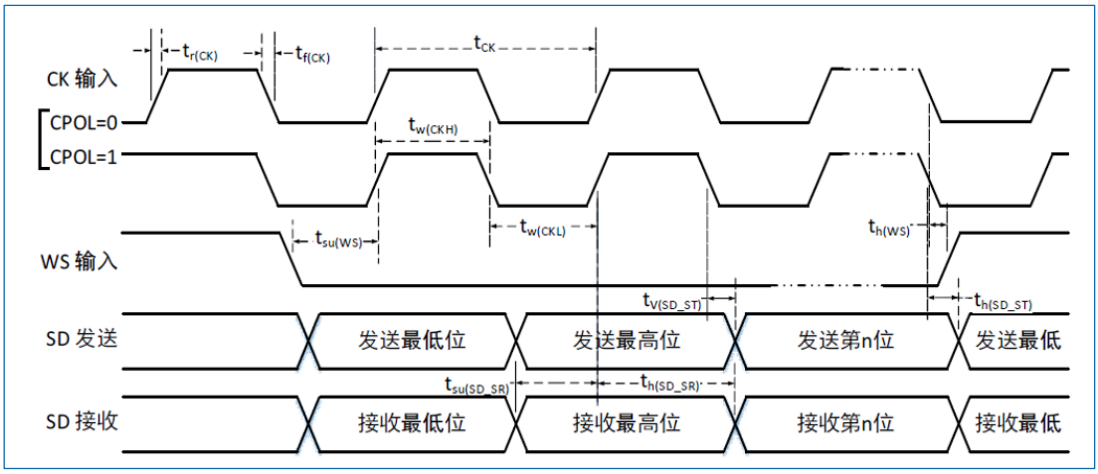


图 20-38 I2S 总线从模式时序图（飞利浦协议）

表 20-6 I2S 接口特性

符号	参数	条件	最小值	最大值	单位
f_{SCK}/t_{SCK}	I2S 时钟频率	主模式	-	8	MHz
		从模式	-	8	MHz
$t_{r(SCK)}/t_{f(SCK)}$	I2S 时钟上升和下降时间	负载电容：C=30pF	-	20	ns
$t_{V(WS)}$	WS 有效时间	主模式	-	5	ns
$t_{SU(WS)}$	WS 建立时间	从模式	10	-	ns
$t_{w(SCKH)}/t_{w(SCKL)}$	SCK 高电平和低电平时	主模式，f_{PCLK}=36MHz，预分配系数=4	40	60	ns

符号	参数	条件	最小值	最大值	单位
	间				
$t_{SU(SD\_MR)}$	数据输入建立时间	主模式	5	-	ns
$t_{SU(SD\_SR)}$		从模式	5	-	ns
$t_{h(WS)}$	WS 保持时间	主模式	$2t_{PCLK}$	-	ns
		从模式	$2t_{PCLK}$	-	ns
$t_{h(SD\_MR)}$	数据输入保持时间	主模式	5	-	ns
$t_{h(SD\_SR)}$		从模式	4	-	ns
$t_{h(SD\_MT)}$	数据输出保持时间	主模式（使能边沿之后）	-	5	ns
$t_{h(SD\_ST)}$		从模式（使能边沿之后）	-	5	ns
$t_{v(SD\_MT)}$	数据输出有效时间	主模式（使能边沿之后）	-	5	ns
$t_{v(SD\_ST)}$		从模式（使能边沿之后）	-	4	ns

20.10 SPI 寄存器

基地址：0x4001 3000

空间大小：0x400

20.10.1 SPI 控制寄存器 1（SPIx_CR1）

地址偏移：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDIMODE	BIDIOE	CRCEN	CRCNEXT	CRCL	RXONLY	SSM	SSI	LSBFIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw

位 15	BIDIMODE：双向数据模式使能 • 0：选择 2 线的单向数据模式 • 1：选择单线双向数据模式
位 14	BIDIOE：双向模式输出使能 与 BIDIMODE 位共同决定在“单线双向”模式下数据的输出方向。 • 0：输出禁止（仅接收模式）； • 1：输出使能（仅发送模式）。
位 13	CRCEN：硬件 CRC 计算使能 • 0：CRC 计算禁用 • 1：CRC 计算使能

位 12	CRCNEXT：下一个值发送自 CRC • 0：下一个发送的值来自发送缓冲区。 • 1：下一个发送的值来自发送 CRC 寄存器。
位 11	CRCL：CRC 长度 该位由软件设置和清零，用于选择 CRC 长度。 • 0：8 位 CRC 长度 • 1：16 位 CRC 长度
位 10	RXONLY：仅接收 与 BIDIMODE 位共同决定在“2 线单向”模式下数据的输出方向。 在多个从设备的配置中，在未被访问的从设备上该位被置 1。这使得只有被访问的从设备有输出，从而避免造成数据线上数据冲突。 • 0：全双工（发送和接收） • 1：输出禁止（仅接收模式）
位 9	SSM：软件从机管理 当 SSM 被置位时，NSS 引脚上的电平由 SSI 位的值决定。 • 0：禁止软件从设备管理。 • 1：启用软件从设备管理。
位 8	SSI：内部从设备选择 此位只有当 SSM 位为 1 的时候才生效。它决定了 NSS 上的电平，在 NSS 引脚上的 I/O 操作无效。
位 7	LSBFIRST：帧格式 • 0：先发送 MSB • 1：先发送 LSB
位 6	SPE：SPI 使能 • 0：禁止 SPI 设备 • 1：开启 SPI 设备
位 5:3	BR[2:0]：波特率控制 • 000：$f_{PCLK}/2$ • 001：$f_{PCLK}/4$ • 010：$f_{PCLK}/8$ • 011：$f_{PCLK}/16$ • 100：$f_{PCLK}/32$ • 101：$f_{PCLK}/64$ • 110：$f_{PCLK}/128$ • 111：$f_{PCLK}/256$

位 2	MSTR：主设备选择 • 0：配置为从设备 • 1：配置为主设备
位 1	CPOL：时钟极性 • 0：空闲状态时，SCK 保持低电平。 • 1：空闲状态时，SCK 保持高电平。
位 0	CPHA：时钟相位 • 0：第一个时钟沿对准第一位数据。 • 1：第二和时钟沿对准第一位数据。

20.10.2 SPI 控制寄存器 2（SPI_CR2）

地址偏移：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res			FRXTH	DS[3:0]				TXEIE	RXNEIE	ERRIE	FRF	NSSP	SSOE	Res	
			rw	rw				rw	rw	rw	rw	rw	rw		

位 15:13	Res：保留 必须保持复位值。
位 12	FRXTH：FIFO 接收门限 此位用于设置触发 RXNE 事件时 RXFIFO 的阈值。 • 0：如果 FIFO 的存储水平大于或等于 1/2（16 位），产生 RXNE 事件。 • 1：如果 FIFO 的存储水平大于或等于 1/4（8 位），产生 RXNE 事件。
位 11:8	DS[3:0]：数据位宽 该位域可配置 SPI 传输数据的位宽： • 0000：不使用 • 0001：不使用 • 0010：不使用 • 0011：4 位 • 0100：5 位 • 0101：6 位 • 0110：7 位 • 0111：8 位 • 1000：9 位 • 1001：10 位 • 1010：11 位 • 1011：12 位 • 1100：13 位

	• 1101：14 位 • 1110：15 位 • 1111：16 位 如果软件试图写一个“不使用”的值，该位域会被赋值“0111”（8 位）。
位 7	TXEIE：TX 缓冲器空中断使能 • 0：TXE 中断屏蔽。 • 1：TXE 中断没有被屏蔽。 该位用于在 TXE 标志置 1 的时候，产生一个中断请求。
位 6	RXNEIE：RX 缓冲区非空中断使能 • 0：RXNE 中断屏蔽。 • 1：RXNE 中断没有被屏蔽。该位用于在 RXNE 标志置 1 的时候，产生一个中断请求。
位 5	ERRIE：错误中断使能 该位控制在出现错误事件（SPI 模式中的 CRCERR、OVR 和 MODF；TI 模式中的 FRE，I2S 模式中的 OVR、UDR 和 FRE）时是否产生中断。 • 0：错误中断屏蔽 • 1：错误中断使能
位 4	FRF：帧格式 • 0：SPI Motorola 模式 • 1：SPI TI 模式
位 3	NSSP：NSS 脉冲管理 该位仅在主模式下使用。它允许 SPI 在连续传输时，两个数据传输之间产生一个 NSS 脉冲。在单个数据传输的情况下，它会在传输结束后将 NSS 脚强制为高电平。在 CPHA=1 或 FRF=1 的时候，该位无效。 • 0：没有 NSS 脉冲 • 1：产生 NSS 脉冲
位 2	SSOE：SS 输出使能 • 0：在主模式下 SS 输出被禁用，SPI 接口可以工作在多主机的配置下。 • 1：SPI 接口启用的同时，在主模式下启用 SS 输出。SPI 接口不能在多主环境下工作。
位 1:0	Res：保留 必须保持复位值。

20.10.3 SPI 状态寄存器（SPI_SR）

地址偏移：0x08

复位值：0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res			FTLVL[1:0]		FRLVL[2:0]		FRE	BSY	OVR	MODF	CRCERR	UDR	CHSIDE	TXE	RXNE
			r		r		r	r	r	r	rc_w0	r	r	r	r

位 15:13	Res：保留 必须保持复位值。
位 12:11	FTLVL[1:0]：FIFO 发送存储水平 该位由硬件设置或清零。 • 00：FIFO 空 • 01：1/4 FIFO • 10：1/2 FIFO • 11：FIFO 满（当 FIFO 门限大于 1/2 时，认为是满）
位 10:9	FRLVL[1:0]：FIFO 接收存储水平 该位由硬件设置或清零。 • 00：FIFO 空 • 01：1/4 FIFO • 10：1/2 FIFO • 11：FIFO 满
位 8	FRE：TI 帧格式错误 • 0：未发生帧格式错误。 • 1：发生了一个帧格式错误。
位 7	BSY：忙标志 该位由硬件置 1 和清零。 • 0：SPI（或 I2S）不忙。 • 1：（SPI 或 I2S）通信忙或发送缓冲区不为空。
位 6	OVR：溢出标志 该位由硬件置位，由软件序列复位。 • 0：未发生溢出。 • 1：发生溢出。
位 5	MODF：模式故障 该位由硬件置位，由软件序列复位。 • 0：未发生模式故障。 • 1：发生模式故障。
位 4	CRCERR：CRC 错误标志 该位由硬件置位，由软件清零。 • 0：收到的 CRC 值和 SPI_RXCRCR 的值是匹配的。

	• 1：收到的 CRC 值和 SPI_RXCRCR 值不匹配。
位 3	UDR：欠载标志 该位由硬件置位，由软件序列复位。 • 0：未发生欠载。 • 1：发生欠载。
位 2	CHSIDE：通道标志 • 0：需要传输或者接收左声道。 • 1：需要传输或者接收右声道。
位 1	TXE：发送缓冲区为空标志 • 0：TX 缓冲区非空。 • 1：TX 缓冲器空。
位 0	RXNE：接收缓冲区非空标志 • 0：RX 缓冲区空。 • 1：RX 缓冲非空。

20.10.4 SPI 数据寄存器（SPI_DR）

地址偏移：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
rw															

位 15:0	DR[15:0]：数据寄存器 数据寄存器作为 RX 和 TX FIFO 的接口，用于发送或者接收数据。 当读取数据寄存器时，RX FIFO 会被访问，而写入数据寄存器则会访问 TX FIFO。

20.10.5 SPI 的 CRC 多项式寄存器（SPI_CRCPR）

地址偏移：0x10

复位值：0x0007

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCPOLY[15:0]															
rw															

位 15:0	CRCPOLY[15:0]：CRC 多项式寄存器 该寄存器包含 CRC 计算多项式。根据需要，可以配置成另一个多项式。

20.10.6 SPI 接收 CRC 寄存器（SPI_RXCRCR）

地址偏移：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCRC[15:0]															
r															

位 15:0	RXCRC[15:0]：RX CRC 寄存器 当启用 CRC 计算功能，RXCRC 位包含所接收数据计算出来的 CRC 值。 当 SPI_CR1 寄存器的 CRCEN 位写 1 时，这个寄存器被复位。CRC 计算使用 SPI_CRCPR 中的多项式。 • 当数据帧格式被设置为 8 位（SPI_CR1 的 CRCL 位为 0）时，仅低 8 位按照 CRC8 的方法参与计算。 • 当数据帧格式为 16 位（SPI_CR1 的 CRCL 位为 1）时，寄存器中的所有 16 位按照 CRC16 的方法参与计算。

20.10.7 SPI 发送 CRC 寄存器（SPI_TXCRCR）

地址偏移：0x18

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxCRC[15:0]															
rw															

位 15:0	TxCRC[15:0]：TX CRC 寄存器 当启用 CRC 计算功能，TxCRC 包含所接收数据计算出来的 CRC 值。 当 SPI_CR1 寄存器的 CRCEN 位写 1 时，这个寄存器被复位。CRC 计算使用 SPI_CRCPR 中的多项式。 当数据帧格式被设置为 8 位（SPI_CR1 中的 CRCL 位为 0）时，仅低 8 位按照 CRC8 的方法参与计算。 当数据帧格式为 16 位（SPI_CR1 的 CRCL 位为 1）时，寄存器中的所有 16 位按照 CRC16 的方法参与计算。

20.10.8 SPI_I2S 配置寄存器（SPI_I2SCFGR）

地址偏移：0x1C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				I2SMOD	I2SE	I2SCFG[1:0]		PCMSYNC	Res	I2SSTD[1:0]		CKPOL	DATLEN[1:0]		CHLEN
				rw	rw	rw		rw		rw		rw	rw		rw

位 15:12	Res：保留 必须保持复位值。
位 11	I2SMOD：I2S 模式选择 • 0：选中 SPI 模式。 • 1：选中 I2S 模式。
位 10	I2SE：I2S 使能

	• 0：I2S 的外设被禁用。 • 1：I2S 外设被使能。
位 9:8	I2SCFG[1:0]：I2S 配置模式 • 00：从机 - 发送 • 01：从机 - 接收 • 10：主机 - 发送 • 11：主机 - 接收
位 7	PCMSYNC：PCM 帧同步 • 0：短帧同步 • 1：长帧同步
位 6	Res：保留 必须保持复位值。
位 5:4	I2SSTD[1:0]：I2S 标准选择 • 00：I2S 飞利浦标准 • 01：高字节对齐标准（左对齐） • 10：低字节对齐标准（右对齐） • 11：PCM 标准
位 3	CKPOL：静止态时钟极性 • 0：I2S 时钟静止态为低电平 • 1：I2S 时钟静止态为高电平
位 2:1	DATLEN[1:0]：待传输数据长度 • 00：16 位数据长度 • 01：24 位数据长度 • 10：32 位数据长度 • 11：不允许
位 0	CHLEN：声道长度（每个音频通道的数据位数） • 0：16 位宽 • 1：32 位宽 只有在 DATLEN= 00 时，该位的写操作才有意义，否则声道长度都由硬件固定为 32 位。

20.10.9 SPI_I2S 预分频寄存器（SPI_I2SPR）

地址偏移：0x20

复位值：0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res						MCKOE	ODD	I2SDIV[7:0]							

	rw	rw	rw

位 15:10	Res：保留 必须保持复位值。
位 9	MCKOE：主时钟输出使能 • 0：主时钟输出被禁用 • 1：主时钟输出启用
位 8	ODD：奇系数预分频 • 0：实际分频系数 = I2SDIV *2 • 1：实际分频系数 =（I2SDIV *2）+1
位 7:0	I2SDIV[7:0]：I2S 线性预分频器 不允许设置 I2SDIV[7:0] =0 或者 I2SDIV[7:0] =1 的值。

21 蜂鸣器（Beeper）

Beep 蜂鸣器内置超低功耗 7 位定时器，工作时钟可配置为 GPIO 外部输入时钟或 114kHz 片内 LSI 慢速时钟；定时器使用向下计数的方式计数，可输出 1、2、4、8kHz 频率脉冲。

21.1 蜂鸣器主要特性

- 超低功耗 7 位向下计数器
- 时钟源可选
- 1、2、4、8kHz 频率脉冲输出
- 停机（Stop）模式下，定时触发 ADC 采样

21.2 蜂鸣器功能说明

21.2.1 蜂鸣器框图

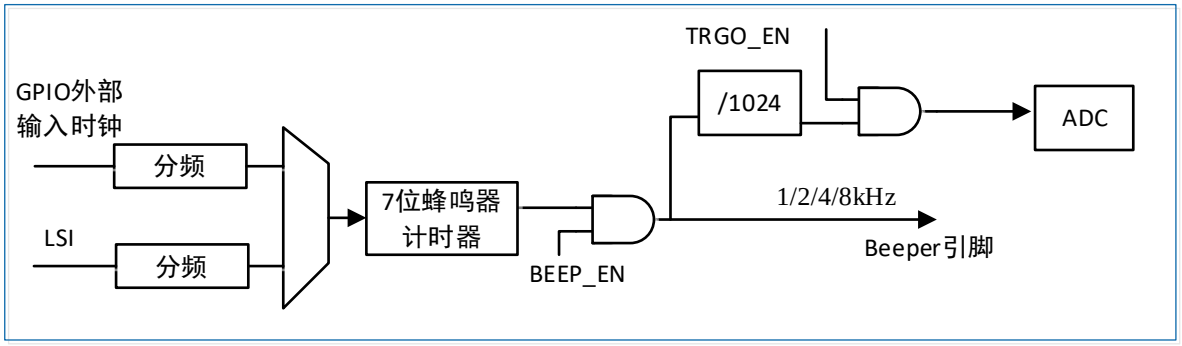


图 21-1 蜂鸣器框图

21.2.2 定时触发

在 MCU 停机（Stop）模式下，蜂鸣器可继续工作并可定时触发（TRGO）ADC 采样；定时触发 ADC 采样的频率为蜂鸣器输出脉冲频率的 1/1024。例如 Beeper 当前输出的蜂鸣脉冲为 1kHz，那么定时触发 ADC 采样的频率为 1kHz/1024 ≈ 0.98Hz（周期约为 1.02 秒）。

21.3 Beeper 寄存器

基地址：0x4000 7C00

空间大小：0x400

21.3.1 配置寄存器（BEEP_CFGR）

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res											TRGO_PRE[1:0]		BEEP_FREQ[1:0]		BEEP_CKSEL
											rw		rw		rw

位 31:5	Res：保留 必须保持复位值。
位 4:3	TRGO_PRE[1:0]：选择输出 TRGO 定时信号的预分频时钟 TRGO 定时信号的频率为预分频时钟信号的 1024 分频。 • 00：1kHz@LSI 时钟源（时钟源 128 分频） • 01：2kHz@LSI 时钟源（时钟源 64 分频） • 10：4kHz@LSI 时钟源（时钟源 32 分频） • 11：8kHz@LSI 时钟源（时钟源 16 分频）
位 2:1	BEEP_FREQ[1:0]：蜂鸣器的分频时钟源 • 00：1kHz@LSI/GPIO 外部输入时钟源（时钟源 128 分频） • 01：2kHz@LSI/GPIO 外部输入时钟源（时钟源 64 分频） • 10：4kHz@LSI/GPIO 外部输入时钟源（时钟源 32 分频） • 11：8kHz@LSI/GPIO 外部输入时钟源（时钟源 16 分频）
位 0	BEEP_CKSEL：选择蜂鸣器的计时时钟 • 0：选择 114 kHz LSI 作为蜂鸣器的计时时钟。 • 1：选择 GPIO 外部输入时钟作为蜂鸣器的计时时钟。

21.3.2 控制寄存器（BEEP_CR）

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CR_WBUSY	Res														
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res														TRGO_EN	BEEP_EN
														rw	rw

位 31	CR_WBUSY：APB 总线是否正在写 CR 寄存器（只读寄存器） • 0：CR 寄存器未被 APB 总线写操作。 • 1：CR 寄存器正在被 APB 总线写操作。 （当 CR_WBUSY=1 时，软件禁止再次对 CR 寄存器进行写操作。）
位 30:2	Res：保留 必须保持复位值。
位 1	TRGO_EN：关闭或开启 TRGO 输出 • 0：关闭 TRGO 的输出（TRGO 信号保持 0 电平输出），并复位 10 位的 TRGO 计数器。 • 1：开启 TRGO 的输出。 *注意：当 CR_WBUSY=1 时，对 TRGO_EN 寄存器的写操作将无效。*

位 0	BEEP_EN：关闭或开启蜂鸣器 • 0：关闭蜂鸣器，并复位 7-bit 的 Beeper counter 计数器和 10-bit 的 TRGO counter 计数器。 • 1：开启蜂鸣器。 *注意：当 CR_WBUSY=1 时，对 BEEP_EN 寄存器的写操作将无效。*

22 设备电子签名（UID）

设备电子签名存储在 Flash 模块的系统存储区中，可以使用 JTAG/SWD 或 CPU 对其进行读取。

UID 包含出厂前编程的标识数据，这些标识数据允许用户固件或其它外部设备将其接口与 HK32F030M 的特性自动匹配。

22.1 唯一设备 ID 寄存器（64 位）

基地址：0x1FFF F838

空间大小：0x08

唯一设备标识符适用于以下应用场景。

- 用作序列号。
- 在对内部 Flash 进行编程前将唯一 ID 与软件加密原语和协议结合使用时，用作安全密钥以提高 Flash 中代码的安全性。
- 激活安全自举过程等。

64 位的唯一设备标识符提供了一个对于任何设备和任何上下文都唯一的参考号码。用户永远不能改变该位域。

64 位的唯一设备标识符也可以以单字节/半字/字等不同方式读取，然后使用自定义算法连接起来。

22.1.1 UID 寄存器 0（U_ID0）

偏移地址：0x00

复位值：0xXXXX XXXX

说明：此处 X 表示出厂前编程设置。

U_ID0 为 UID 的低 32 位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[15:0]															
r															

位 31:0	UID[31:0]：唯一身份标志的 31:0 位（Unique ID bits）

22.1.2 UID 寄存器 1（U_ID1）

偏移地址：0x04

复位值：0xXXXX XXXX

说明：此处 X 表示出厂前编程设置。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[63:48]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[47:32]															

r

位 31:0	UID[64:32]：唯一身份标志的 63:32 位（Unique ID bits）

23 调试支持（DBG）

23.1 概述

HK32F030M 器件的内核是 Cortex®-M0，该内核包含用于高级调试功能的硬件扩展。调试扩展允许内核可以在取指（指令断点）或取访问数据（数据断点）时停止内核。内核停止时，可以查询内核的内部状态和系统的外部状态。查询完成后，将恢复内核和系统并恢复程序执行。

当调试主机与 HK32F030M MCU 相连并进行调试时，将使用调试功能。

提供一个调试接口：

- 串行接口

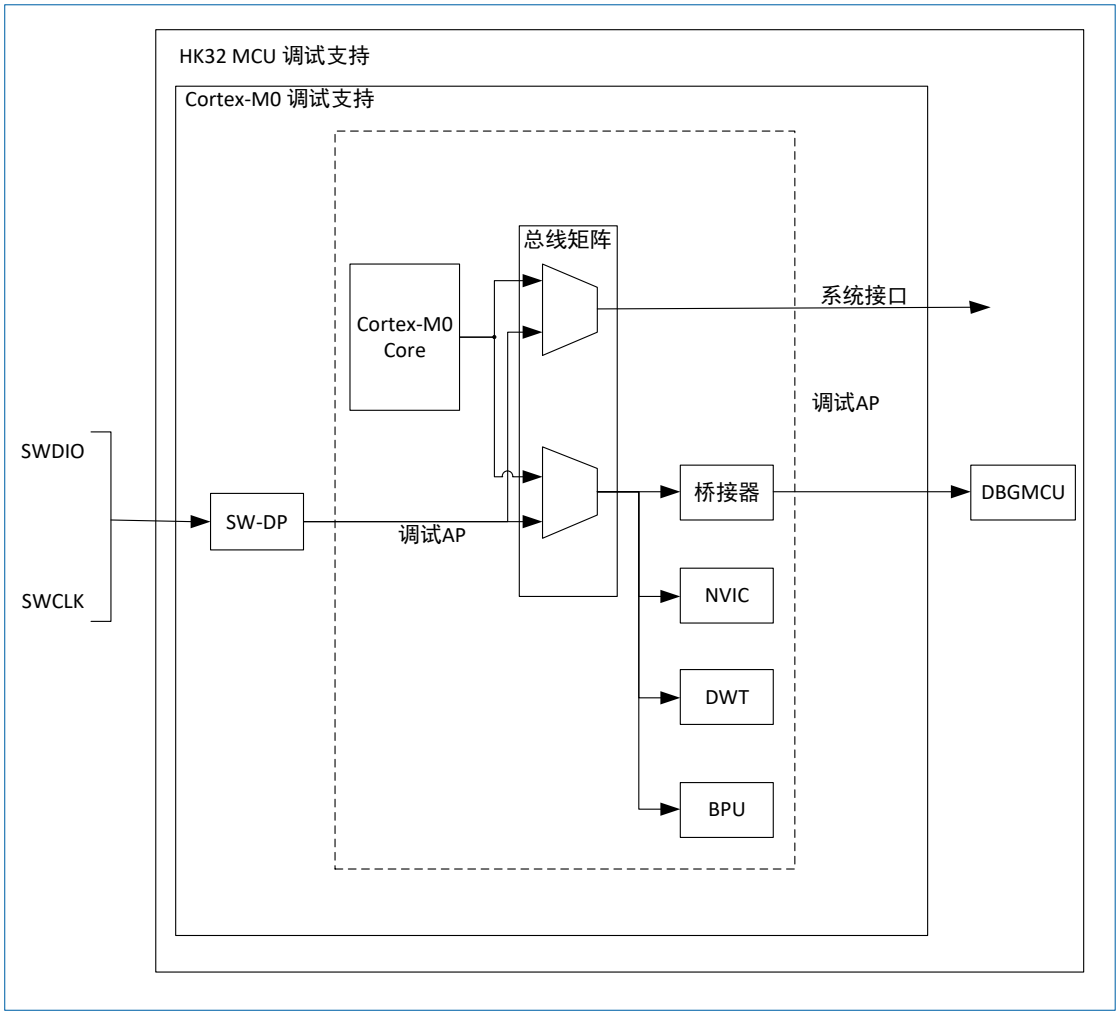


图 23-1 MCU 和 Cortex®-M0 级调试支持框图

Cortex®-M0 内核中内置的调试功能是 ARM® CoreSight 设计套件的一部分。ARM® Cortex®-M0 内核提供集成片上调试支持。它包括：

- SW-DP：串行线
- BPU：断点单元
- DWT：数据观察点触发

它还包括专用于 HK32F030M 的调试功能：

- 灵活调试引脚分配

- MCU 调试盒（支持低功耗模式和对外设时钟的控制等）

说明：有关 ARM® Cortex®-M0 内核支持的调试功能的详细信息，请参见 Cortex®-M0 技术参考手册。

23.2 ARM®参考文档

- Cortex®-M0 技术参考手册（TRM）
- ARM®调试接口 V5
- ARM® CoreSight 设计套件版本 r1p1 技术参考手册

23.3 引脚排列和调试端口引脚

HK32F030M MCU 的不同封装有不同的有效引脚数。

23.3.1 SWD 端口引脚

两个引脚被用作 SW-DP 的输出，作为通用 I/O 的复用功能。所有封装都提供这些引脚。

表 23-1 SW 调试端口引脚

SW-DP 引脚名称	SW 调试端口		引脚分配
	类型	调试分配	
SWDIO	IO	串行线数据输入/输出	PD5
SWCLK	I	串行线时钟	PB5

23.3.2 SW-DP 引脚分配

复位（SYSRESETn 或 PORESETn）后，用于 SW-DP 的引脚将分配为可由调试主机立即使用的专用引脚。

但是 MCU 可以禁止 SWD 端口，进而可释放相关引脚以用作通用 I/O (GPIO)。有关如何禁止 SW-DP 端口引脚的更多详细信息，请参见“8.2.2 I/O 引脚复用功能复用器和映射”。

23.3.3 SWD 引脚上的内部上拉和下拉

用户软件释放 SW I/O 后，GPIO 控制器便会控制这些引脚。GPIO 控制寄存器的复位状态会将 I/O 置于等效的状态：

- SWDIO：输入上拉
- SWCLK：输入下拉

由于内置上拉和下拉电阻，因此无需添加外部电阻。

23.4 SWD 端口

23.4.1 SWD 协议简介

此同步串行协议使用两个引脚：

- SWCLK：从主机到目标的时钟
- SWDIO：双向

利用该协议，可以同时读取和写入两组寄存器组（DPACC 寄存器组和 APACC 寄存器组）。传输数据时，LSB 在前。

对于 SWDIO 双向管理，必须在电路板上对线路进行上拉。这些上拉电阻可在内部配置。无需外部

上拉电阻。

每次在协议中更改 SWDIO 的方向时，都会插入转换时间，此时线路即不受主机驱动也不受目标驱动。默认情况下，此转换时间为一位时间，但也可以通过配置 SWCLK 频率来调整。

23.4.2 SWD 协议序列

每个序列包括三个阶段:

1. 主机发送的数据包请求（8 位）。
2. 目标发送的确认响应（3 位）。
3. 主机或目标发送的数据传输阶段（33 位）。

表 23-2 数据包请求（8 位）

位	名称	说明
0	启动	必须为 1
1	APnDP	● 0：DP 访问 ● 1：AP 访问
2	RnW	● 0：写请求 ● 1：读请求
4:3	A[3:2]	DP 或 AP 寄存器的地址字段
5	奇偶校验	该位表示前面几位（包括启动位）的奇偶校验值
6	停止	0
7	驻留	不受主机驱动。由于存在上拉，因此必须由目标读为 1。

有关 DPACC 和 APACC 寄存器的详细说明，请参见 Cortex®-M0 TRM。

数据包请求后面始终为转换时间（默认 1 位），此时主机和目标都不会驱动线路。

表 23-3 ACK 响应（3 位）

位	名称	说明
0..2	ACK	● 001：FAULT ● 010：WAIT ● 100：OK

仅当发生 READ 事务或者接收到 WAIT 或 FAULT 确认时，ACK 响应后才必须是转换时间。

表 23-4 DATA 传输（33 位）

位	名称	说明
0-31	WDATA 或 RDATA	写入或读取数据
32	奇偶校验	32 个数据位的单奇偶校验

仅当发生 READ 事务时，DATA 传输后才必须是转换时间。

23.4.3 SW-DP 状态机（复位、空闲状态、ID 代码）

SW-DP 的状态机有一个用于标识 SW-DP 的内部 ID 代码。该代码符合 JEP-106 标准。此 ID 代码是默认的 ARM®代码，设置为 0x0BB11477（相当于 Cortex®-M0）。

说明：在目标读取此 ID 代码前，SW-DP 状态机是不工作的。

- 在上电复位后或者线路处于高电平超过 50 个周期后，SW-DP 状态机处于复位状态。
- 如果在复位状态后线路处于低电平至少两个周期，SW-DP 状态机处于空闲状态。
- 复位状态后，该状态机必须首先进入空闲状态，然后对 DP-SW ID CODE 寄存器执行读访问。否则，目标将在另一个事务上发出 FAULT 确认响应。

有关 SW-DP 状态机的更多详细信息，请参见 Cortex®-M0+ TRM 和 CoreSight 设计套件 r1p0 TRM。

23.4.4 DP 和 AP 读/写访问

- 不延迟对 DP 的读访问：可以立即发送目标响应（如果 ACK=OK），也可以延迟发送目标响应（如果 ACK=WAIT）。
- 延迟对 AP 的读访问。这意味着会在下次传输时返回访问结果。如果要执行的下次访问不是 AP 访问，则必须读取 DP-RDBUFF 寄存器来获取结果。每次进行 AP 读访问或 RDBUFF 读请求时都会更新 DP-CTRL/STAT 寄存器的 READOK 标志，以便了解 AP 读访问是否成功。
- SW-DP 有写缓冲区（用于 DP 或 AP 写入），这样即使在其它操作仍未完成时，也可以接受写入操作。如果写缓冲区已满，则目标确认响应为 WAIT。但 IDCODE 读取、CTRL/STAT 读取或 ABORT 写入除外，这几项操作在写缓冲区已满时也会被接受。
- 由于存在异步时钟域 SWCLK 和 HCLK，因此写操作后（奇偶校验位后）还需要两个额外的 SWCLK 周期，以使写入操作在内部生效。应在将线路驱动为低电平时（空闲状态）应用这些周期。在写 CTRL/STAT 寄存器以提出一个上电请求时，这一点特别重要。否则下一个操作（在内核上电后才有效的操作）会立即执行，这将导致失败。

23.4.5 SW-DP 寄存器描述

当 APnDP=0 时能够访问这些寄存器。

表 23-5 SW-DP 寄存器

A[3:2]	R/W	SELECT 寄存器的 CTRLSEL 位	寄存器	说明
00	读取		IDCODE	制造商代码设置为 Cortex®-M0 的默认 ARM®代码。 0x0BB11477（标识 SW-DP）
00	写		ABORT	
01	读/写	0	DP-CTRL/STAT	目的： ● 请求系统或调试上电 ● 配置 AP 访问的传输操作 ● 控制比较和验证操作 ● 读取一些状态标志（上溢和上电确认）
01	读/写	1	WIRE CONTROL	用于配置物理串行端口协议（如转换时间的持续时间）。
10	读取		READ RESEND	允许从已损坏的调试软件传输中恢复读取数据，无需重复执行原始 AP 传输。
10	写		SELECT	用于选择当前访问端口和活动的 4 字寄存器窗口。

A[3:2]	R/W	SELECT 寄存器的 CTRLSEL 位	寄存器	说明
11	读/写		READ BUFFER	由于已发出 AP 访问，因此该读缓冲区非常有用（在执行下个 AP 事务时提供读取 AP 请求的结果）。 此读取缓冲区捕获 AP 中的数据，显示为前一次读取的结果，无需启动新操作。

23.4.6 SW-AP 寄存器描述

当 APnDP=1 时能够访问这些寄存器。

有多个 AP 寄存器，这些寄存器按以下组合进行寻址：

- 移位值 A[3:2]
- DP SELECT 寄存器的当前值

表 23-6 32 位调试端口寄存器，通过移位值 A[3:2]进行寻址

地址	A[3:2]值	说明
0x0	00	保留位，必须保持复位值。
0x4	01	DP CTRL/STAT 寄存器，用于： ● 请求系统或调试上电 ● 配置 AP 访问的传输操作 ● 控制比较和验证操作 读取一些状态标志（上溢和上电确认）
0x8	10	DP SELECT 寄存器：用于选择当前访问端口和有效的 4 字寄存器窗口。 ● 位 31:24：APSEL，用于选择当前 AP（select the current AP） ● 位 23:8：保留 ● 位 7:4：APBANKSEL，用于在当前 AP 上选择有效的 4 字寄存器窗口。 ● 位 3:0：保留
0xC	11	DP RDBUFF 寄存器：用于通过调试器在执行一系列操作后获取最后结果（无需请求新的 JTAG-DP 操作）

23.5 内核调试

通过内核调试寄存器调试内核。通过调试访问端口调试访问这些寄存器。它由四个寄存器组成：

表 23-7 内核调试寄存器

寄存器	说明
DHCSR	32 位调试停止控制和状态寄存器： 此寄存器提供有关处理器状态的信息，能够使内核进入调试停止状态并提供处理器步进功能。
DCRSR	17 位调试内核寄存器选择器寄存器： 此寄存器选择需要进行读写操作的处理器寄存器。
DCRDR	32 位调试内核寄存器数据寄存器： 此寄存器保存在寄存器与 DCRSR（选择器）寄存器选择的处理器之间读取和写入的数据。
DEMCR	32 位调试异常和监视控制寄存器： 此寄存器提供向量捕获和调试监视控制。

这些寄存器在系统复位时不复位。它们只能通过上电复位来复位。有关更多详细信息，请参见 Cortex®-M0 TRM。

为了在复位后立即使内核进入调试停止状态，必须：

- 使能调试和异常监视控制寄存器的位 0（VC_CORRESET）
- 使能调试停止控制和状态寄存器的位 0（C_DEBUGEN）

23.6 BPU（断点单元）

Cortex®-M0 BPU 实现提供四个断点寄存器。BPU 是 ARMv7-M（Cortex®-M3 和 Cortex®-M4）中提供的 Flash 补丁和断点（FPB）模块的一部分。

23.6.1 BPU 功能

处理器断点实现了基于 PC 的断点功能。

有关 BPU CoreSight 标识寄存器及其地址和访问类型的更多信息，请参见 ARMv6-M ARM®和 ARM® CoreSight 组件技术参考手册。

23.7 DWT（数据观察点）

Cortex®-M0 DWT 实现提供了两个观察点寄存器组。

23.7.1 DWT 功能

处理器观察点实现了数据地址和基于 PC 的观察点功能（即 PC 采样寄存器），并支持比较器地址掩码，如 ARMv6-M ARM®中所述。

23.7.2 DWT 程序计数器采样寄存器

实现数据观察点单元的处理器还实现了 ARMv6-M 可选 DWT 程序计数器采样寄存器（DWT_PCSR）。此寄存器允许调试程序定期采样 PC，无需停止处理器。这可提供粗略分析。有关更多信息，请参见 ARMv6-M ARM®。

Cortex®-M0 DWT_PCSR 记录通过条件代码的指令以及未通过条件代码的指令。

23.8 MCU 调试组件（DBG）

MCU 调试组件帮助调试器为以下各项提供支持：

- 低功耗模式
- 断点期间的定时器、看门狗和 I2C 的时钟控制

23.8.1 对低功耗模式的调试支持

要进入低功耗模式，必须执行指令 WFI 或 WFE。

MCU 支持多个低功耗模式，这些模式可以禁止 CPU 时钟或降低 CPU 功耗。

内核不允许在调试会话期间关闭 FCLK 或 HCLK。由于调试期间需要使用它们进行调试连接，因此其必须保持激活状态。MCU 集成了特殊方法，允许用户在低功耗模式下调试软件。

为此，调试主机首先必须设置一些调试配置寄存器，以更改低功耗模式行为：

- 在睡眠模式下，FCLK 和 HCLK 仍有效。因此，此模式对标准调试功能没有任何限制。
- 在停机模式下，调试程序必须事先将 DBG_STOP 位置 1。这样便可使能内部 RC 振荡器时钟，以在停机模式下为 FCLK 和 HCLK 提供时钟。

23.8.2 对定时器、看门狗和 I2C 的调试支持

断点期间，必须选择定时器和看门狗的计数器的行为方式：

- 在产生断点时，计数器继续计数。例如，当 PWM 控制电机时，通常需要这种方式。
- 在产生断点时，计数器停止计数。用于看门狗时需要这种方式。

对于 I2C，用户可以选择在断点期间阻止 SMBUS 超时。

23.9 DBGMCU 寄存器

基地址：0x4001 5800

空间大小：0x400

23.9.1 MCU 器件 ID 代码寄存器（DBGMCU_IDCODE）

偏移地址：0x000

复位值：0x1000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REV_ID[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res				DEV_ID[11:0]											
				r											

位 31:16	REV_ID[15:0]：航顺品牌的序列号
位 15:12	Res：保留位 读该位域的值为 0。
位 11:0	DEV_ID[11:0]：器件 ID

以下以 HK32F030Mx4 MCU 为例，说明其 ID 号。

器件	REV_ID	DEV_ID
HK32F030Mx4	0x1000	0x003

23.9.2 调试 MCU 配置寄存器（DBGMCU_CR）

偏移地址：0x004

POR 复位值：0x0000 0000

注意：POR 为上电复位，不是系统复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res														DBG_STOP	Res
														rw	

位 31:2	Res：保留

	必须保持复位值。
位 1	DBG_STOP：停机模式下调试 ● 0:（FCLK = OFF，HCLK = OFF） 在停机模式下，时钟控制模块会关闭所有的时钟（包括 HCLK 和 FCLK）。 当从停机模式退出时，时钟配置与 RESET 后的时钟配置相同（CPU 时钟由内部 8M HSI 提供）。因此软件必须重新编程时钟控制器，以启用 PLL、Xtal 等。 ● 1:（FCLK = ON，HCLK = ON ） 在这种情况下，当进入停机模式时，FCLK 和 HCLK 由内部 RC 振荡器提供，内部 RC 振荡器在 STOP 模式下保持激活状态。 当从停机模式退出时，软件必须重新编程时钟控制器以启用 PLL、Xtal 等。例如：与 DBG_STOP = 0 的情况处理方式相同。
位 0	Res：保留 必须保持复位值。

23.9.3 调试 MCU APB1 冻结寄存器（DBGMCU_APB1_FZ）

偏移地址：0x008

POR 复位值：0x0000 0000

注意：POR 为上电复位，不是系统复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res										DBG_I2C1_SMBUS_TIMEOUT	Res				
										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res			DBG_IWDG_STOP	DBG_WWDG_STOP	Res						DBG_TIM6_STOP	Res		DBG_TIM2_STOP	DBG_TIM1_STOP
r			r	r							rw	r		rw	rw

位 31:22	Res：保留 必须保持复位值。
位 21	DBG_I2C1_SMBUS_TIMEOUT：当内核停止时，SMBus 超时模式停止。 ● 0：与正常模式相同的行为。 ● 1：SMBus 超时被冻结。
位 20:13	Res：保留 必须保持复位值。
位 12	DBG_IWDG_STOP：当内核停止时，调试的独立看门狗停止。 ● 0：即使内核停止，IWDG 看门狗计数器时钟仍继续工作。 ● 1：当内核停止时，IWDG 看门狗计数器时钟会被停止。
位 11	DBG_WWDG_STOP：当内核停止时，调试的窗口看门狗停止。

	• 0：即使内核停止，WWDG 看门狗计数器时钟仍继续工作。 • 1：当内核停止时，WWDG 看门狗计数器时钟会被停止。
位 10:5	Res：保留 必须保持复位值。
位 4	DBG_TIM6_STOP：当内核停止时，调试的 TIM6 停止。 • 0：即使内核停止，TIM6 计数器时钟仍继续工作。 • 1：当内核停止时，TIM6 看门狗计数器时钟会被停止。
位 3:2	Res：保留 必须保持复位值。
位 1	DBG_TIM2_STOP：当内核停止时,调试的 TIM2 停止。 • 0：即使内核停止，TIM2 计数器时钟仍继续工作。 • 1：当内核停止时，TIM2 看门狗计数器时钟会被停止。
位 0	DBG_TIM1_STOP：当内核停止时，调试的 TIM1 停止。 • 0：即使内核停止，TIM1 计数器时钟仍继续工作。 • 1：当内核停止时，TIM1 看门狗计数器时钟会被停止。

24 缩略语与术语

24.1 寄存器描述中的缩略语

缩写	全称	中文描述
r	read-only	只读
w	write-only	只写
rc_w0	read/clear this field by writing ‘0'	可读；可通过对该位域写0 清除。 对该位域写1，该位域值无变化。
rc_w1	read/clear this field by writing ‘1'	可读；可通过对该位域写1 清除。 对该位域写0，该位域值无变化。
rs	read/set	可读写，与rw 有区别，通常设置该位域为1 时启动某种硬件动作；当完成硬件动作后，该位域会被硬件自动清0。
rw	read/write	可读写该位或指定位。

24.2 缩略语

缩写	全称	中文描述
AHB	Advanced High-Performance Bus	高级高性能总线
APB	Advanced Peripheral Bus	外围总线
BGR	Bandgap reference	带隙参考
GPIO	General Purpose Input Output	通用输入输出
EXTI	Extended interrupts and events controller	外部中断事件控制器
NVIC	Nested vectored interrupt controller	嵌套中断向量列表控制器
HSI	High-Speed Internal（Clock Signal）	高速内部（时钟信号）
IAP	In-Application Programming	在线应用编程
ICP	In Circuit Programing	在电路编程
LSI	Low-Speed Internal（Clock Signal）	低速内部（时钟信号）
MCU	Microcontroller Unit	微控制单元
OBL	Option Byte Loader	选项字节装载器
SWD	Serial Wire Debug	内核集成的调试口，它是基于SWD 协议的 2 线调试接口。

24.3 术语

名称	中文描述
Byte	字节，8 位数据长度。
Half word	半字，16 位的数据或指令长度。
Option byte	选项字节，保存在Flash 中的MCU 配置字节。
Word	字，32 位的数据或指令长度。

25 重要提示