用户手册

User Manual

MM32G0001

基于Arm®Cortex®-M0内核的32位微控制器

版本：V1.0

目录

附图目录

表格目录

1 系统和存储器架构

1.1 系统架构简介

MM32G0001 是基于 ARM Cortex-M0 处理器开发的 32 位微控制器产品，它同时具备了高性能和低功耗的特点。MM32G0001 采用矩阵总线结构，该矩阵包括一个 AHB 主机：CPU，三个从机分别是 SRAM、闪存存储、AHB 总线（含 AHB 到 APB 的总线桥）以及连接在 APB 总线的各种设备。

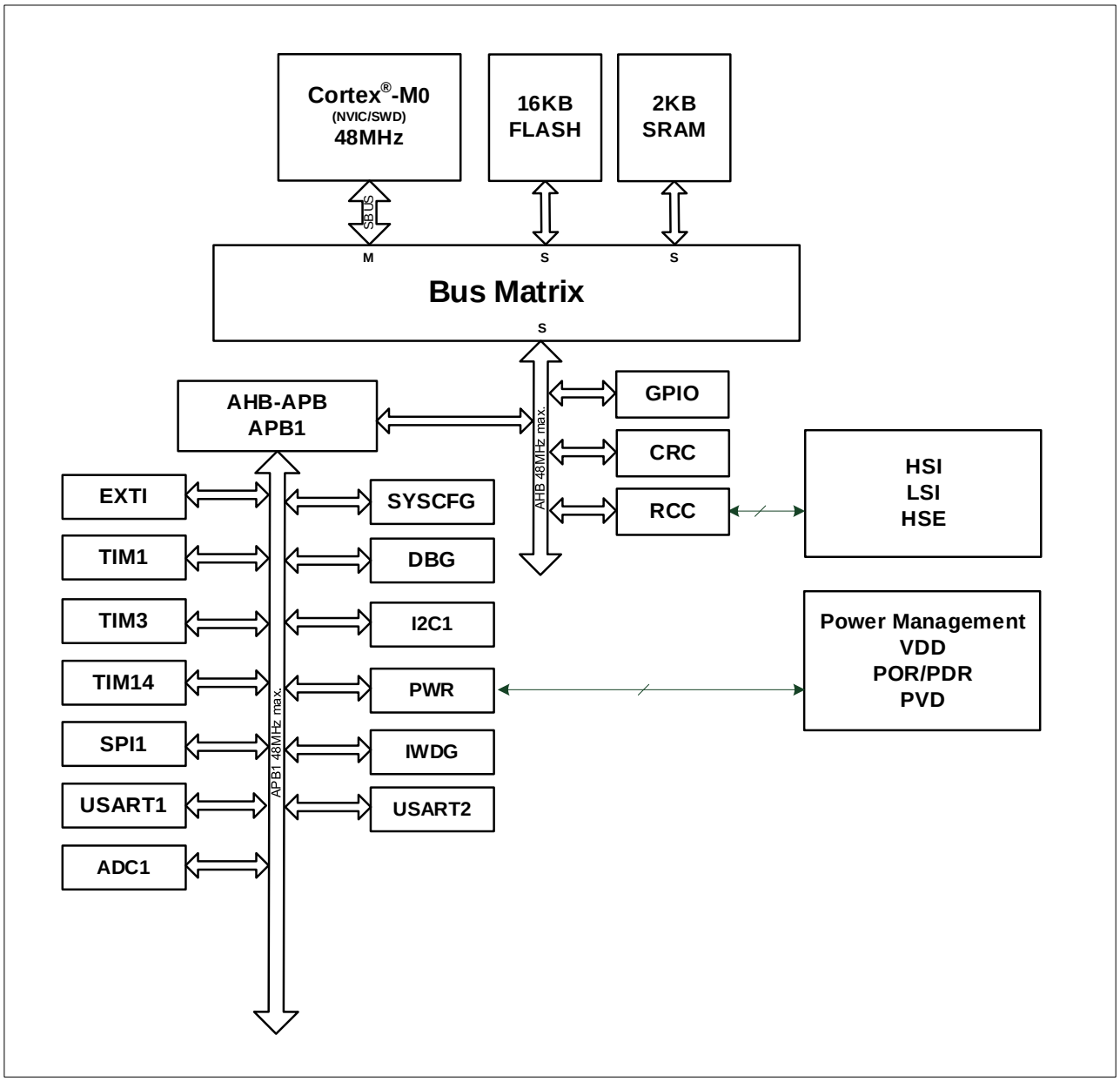

图 1-1 系统架构框图

1.1.1 系统总线

系统总线连接了 CPU 内核和总线矩阵，构造 CPU 与总线矩阵的通路，从而进行数据传输。CPU 可作为主机驱动总线，最终实现对从机如 APB 外设、SRAM、闪存等的读写操作。

1.1.2 总线矩阵

总线矩阵包括一个 AHB 互联矩阵，一个 AHB 总线和一个桥接的 APB 总线。AHB 总线的外设(RCC, GPIO 和 CRC) 通过 AHB 互联矩阵与系统总线连接。在 APB 和 AHB 总线之间连接通过 AHB1APB 桥进行数据交换。当 APB 寄存器进行 8 位 16 位访问，APB 会自动拓宽成 32 位 ，同样的，AHB1APB 桥也具备自动拓宽功能。

1.2 存储器介绍

程序存储器，数据存储器，寄存器和 I/O 接口都位于不同地址范围的存储器地址空间（线性 4GB 的地址空间）。4GB 的地址空间被分为 8 块，每块为 512MB。

1.2.1 存储器映像和寄存器编址

存储器映像请参考各外设对应章节的存储器映像表格。

表 1-1 存储器映像

总线	编址范围	大小	外设
Flash	0x0000 0000 - 0x0000 3FFF	16 KB	可映射为主闪存存储器、系统存储器或 SRAM，有赖于 SYSCFG 寄存器的配置
	0x0000 8000 - 0x07FF FFFF	~127 MB	Reserved
	0x0800 0000 - 0x0800 3FFF	16 KB	主存储区
	0x0800 8000 - 0x1FFD FFFF	~383 MB	Reserved
	0x1FFE 0000 - 0x1FFF F3FF	125KB	Reserved
	0x1FFF F400 - 0x1FFF F7FF	1 KB	系统存储区
	0x1FFF F800 - 0x1FFF F9FF	0.5KB	选项字节
	0x1FFF FA00 - 0x1FFF FFFF	1.5KB	Reserved
SRAM	0x2000 0000 - 0x2000 07FF	2 KB	SRAM
	0x2000 0800 - 0x2FFF FFFF	~255 MB	Reserved
APB1	0x4000 0000 – 0x4000 03FF	1KB	Reserved
	0x4000 0400 – 0x4000 07FF	1KB	TIM3
	0x4000 0800 – 0x4000 2FFF	11KB	Reserved
	0x4000 3000 – 0x4000 33FF	1KB	IWDG
	0x4000 3400 – 0x4000 43FF	4KB	Reserved

总线	编址范围	大小	外设
	0x4000 4400 – 0x4000 47FF	1KB	USART2
	0x4000 4800 – 0x4000 53FF	3KB	Reserved
	0x4000 5400 – 0x4000 57FF	1KB	I2C1
	0x4000 5800 – 0x4000 6FFF	6KB	Reserved
	0x4000 7000 – 0x4000 73FF	1KB	PWR
	0x4000 7400 – 0x4000 FFFF	35KB	Reserved
	0x4001 0000 – 0x4001 03FF	1KB	SYSCFG
	0x4001 0400 – 0x4001 07FF	1KB	EXTI
	0x4001 0800 – 0x4001 23FF	7KB	Reserved
	0x4001 2400 – 0x4001 27FF	1KB	ADC1
	0x4001 2800 – 0x4001 2BFF	1KB	Reserved
	0x4001 2C00 – 0x4001 2FFF	1KB	TIM1
	0x4001 3000 – 0x4001 33FF	1KB	SPI1
	0x4001 3400 – 0x4001 37FF	1KB	DBG
	0x4001 3800 – 0x4001 3BFF	1KB	USART1
	0x4001 3C00 – 0x4001 3FFF	1KB	Reserved
	0x4001 4000 – 0x4001 43FF	1KB	TIM14
	0x4001 4400 – 0x4001 FFFF	47KB	Reserved
AHB	0x4002 0000 – 0x4002 0FFF	4KB	Reserved
	0x4002 1000 – 0x4002 13FF	1KB	RCC
	0x4002 1400 – 0x4002 1FFF	3KB	Reserved
	0x4002 2000 – 0x4002 23FF	1KB	Flash Interface
	0x4002 2400 – 0x4002 2FFF	3KB	Reserved
	0x4002 3000 – 0x4002 33FF	1KB	CRC
	0x4002 3400 – 0x4002 FFFF	51KB	Reserved
	0x4800 0000 – 0x4800 03FF	1KB	PORT A
	0x4800 0400 – 0x4800 07FF	1KB	PORT B

1.2.2 内置的 SRAM

内置最大可到 2K 字节的静态 SRAM。它可以以字节（8 位）、半字（16 位）或字（32 位）进行访问。SRAM 起始地址为 0x2000 0000。

SRAM 可以被 CPU 用最快的系统时钟且不插入任何等待进行访问。

1.2.3 闪存存储器概述

闪存存储器分为两个存储区域：

- 由应用数据和用户数据区组成的主闪存存储块。

- 由选项字节和系统存储器组成的信息块：
  - 选项字节（Option bytes）：包括硬件和存储保护用户配置选项。
  - 系统存储器（System memory）：系统信息。

闪存接口基于 AHB 协议执行指令和数据存取。闪存接口的预取缓冲功能可加速 CPU 执行代码的速度。

2 Embedded FLASH 嵌入式闪存

2.1 闪存主要特性

- 高达 16K 字节闪存存储器
- 存储器结构：
  - 主闪存模块: 最大 4K 字 (4K x 32 位)
  - 信息模块:
    - ◆ 系统存储器: 高达 1K 字节 (1K x 8 位)
    - ◆ 选项字节: 高达 2 x 8 字节

闪存接口的特性为：

- 带预取缓冲器的数据接口 (2 x 32 位)
- 选项字节加载器
- 闪存编程/擦除操作
- 读/写保护
- 低功耗模式

2.2 闪存功能描述

2.2.1 闪存结构

闪存空间由 32 位宽的存储单元组成，既可以存代码又可以存数据。主闪存块按 16 页 (每页 1K 字节) 或 4 扇区 (每扇区 4K 字节)分块，以扇区为单位设置写保护 (参见存储保护相关内容)。

表 2-1 Flash 模块结构

模块	页名称	地址	大小（字节)
主存储块	页 0	0x0800 0000 - 0x0800 03FF	1K
	页 1	0x0800 0400 - 0x0800 07FF	1K
	页 2	0x0800 0800 - 0x0800 0BFF	1K
	页 3	0x0800 0C00 - 0x0800 0FFF	1K
	…	…	…
	…	…	…
	页 15	0x0800 3C00 - 0x0800 3FFF	1K
信息块	系统存储器	0x1FFF F400 - 0x1FFF F7FF	1K
	选项字节	0x1FFF F800 - 0x1FFF F80F	16
闪存存储器接口寄存器	FLASH_ACR	0x4002_2000 - 0x4002 2003	4

模块	页名称	地址	大小（字节)
	FLASH_KEYR	0x4002_2004 - 0x4002 2007	4
	FLASH_OPTKEYR	0x4002_2008 - 0x4002 200B	4
	FLASH_SR	0x4002_200C - 0x4002 200F	4
	FLASH_CR	0x4002_2010 - 0x4002 2013	4
	FLASH_AR	0x4002_2014 - 0x4002 2017	4
	保留	0x4002_2018 - 0x4002 201B	4
	FLASH_OBR	0x4002_201C - 0x4002 201F	4
	FLASH_WRPR	0x4002_2020 - 0x4002 2023	4

2.2.2 FLASH 读操作

嵌入式 Flash 模块可以像普遍存储空间一样直接寻址访问。任何对 Flash 模块内容的读操作都须经过专门的判断过程。

取指令和取数据都是通过 AHB 总线读取访问，能够按照 Flash 访问控制寄存器 (FLASH_ACR) 中得选项所指定的方式执行：

- 取指：预取值缓冲区使能后可提高 CPU 运行速度
- 潜伏期：等待位的个数，保证正确的读取。

取指

CPU 通过 AHB 总线取指。预取指模块的功效在于提高取指效率。

预取缓冲区

预取缓冲区 (2 个 32 位)：在每一次复位以后被自动打开，由于每个缓冲区的大小 (32 位) 与闪存的带宽相同，因此只通过需一次读闪存的操作即可更新整个缓冲区的内容。由于预取缓冲区的存在，CPU 可以工作在更高的主频。CPU 每次取指最多为 32 位的字，取一条指令时，下一条指令已经在缓冲区中等待。

预取控制器

预取控制器会根据预取缓冲区的可用空间来把握访问 Flash 的时机。当预取缓冲区中存在至少一块可用空间时，预取控制器会发起一次读取请求。复位后，预取指缓冲区的默认状态是打开的。只有在 SYSCLK 低于 24MHz，并且 AHB 时钟没有经过任何分频的条件下(SYSCLK 必须等于 HCLK) 才可以开/关预取指缓冲区。通常情况下，预取指缓冲区在初始化过程中就已经决定好开关状态了，而当时 MCU 运行在内部 8MHz 的振荡器下。

注：当 AHB 时钟的预分频器不等于 1 时，预取指缓冲区必须打开访问潜伏期。

访问潜伏期

为了保护对 Flash 的正确读取，必须在 Flash 访问控制寄存器中的 LATENCY[2：0] 中指定预取指控制器的速度比，这个数值等于每次访问 Flash 后到下次访问之间所需插入的等待周期的个数。复位后，这个值默认为零，也就是没有插入等待周期的状态。

2.2.3 Flash 写和擦除操作

嵌入式闪存支持在线编程以及在应用编程。

ICP 是指使用 SWD 在线改变 Flash 的内容，将用户代码烧录到单片机中。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(在应用编程) 能够使用 MCU 支持的任何通信接口 (I/Os，USART，I2C，SPI 等等)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，前提是一部分应用程序必须预先用 ICP 的方法烧写进去。

烧写和擦除操作在整个产品工作电压范围内都可以完成。该操作有下列 7 个寄存器完成：

- 关键字寄存器 (FLASH_KEYR)
- 选项字节关键字寄存器 (FLASH_OPTKEYR)
- Flash 控制寄存器 (FLASH_CR)
- Flash 状态寄存器 (FLASH_SR)
- Flash 地址寄存器 (FLASH_AR)
- 选项字节寄存器 (FLASH_OBR)
- 写保护寄存器 (FLASH_WRPR)

只要 CPU 不去访问 Flash 空间，进行中的 Flash 写操作不会妨碍 CPU 的运行。也就是说， 在对 Flash 进行写/擦除操作的同时，任何对 Flash 的访问都会令总线停顿，直到写/擦除操作完成后才会继续执行，这意味着在写/擦除 Flash 的同时不可以对它取指和访问数据。

在对 Flash 空间做写/擦除操作时，内部振荡器 (HSI) 必须处于开启状态。

对 Flash 空间的解锁

复位后，Flash 存储器默认是受保护状态的，这样可以防范意外的擦除动作。FLASH_CR 寄存器不允许被改写，除非执行一串针对 FLASH_KEYR 寄存器的解锁操作才能开启对 FLASH_CR 的访问权限。这串操作由下面 2 个写操作构成：

- 写关键字 1 = 0x45670123
- 写关键字 2 = 0xCDEF89AB

任何错误的顺序将会锁死 FLASH_CR 直至下次复位。

当发生关键字错误时，会由总线错误引发一次硬件错误中断。KEY1 出错会立即中断，KEY1 正确但 KEY2 错误时会在 KEY2 错的时候引发中断。

主闪存编程

主闪存一次可以编程 16 位。当 FLASH_CR 中的 PG 位为 1 时，直接对相应的地址写一个半字 (16 位)，就是一次编程操作。如果试图写别的长度而不是半字，将引起硬件错误中断。

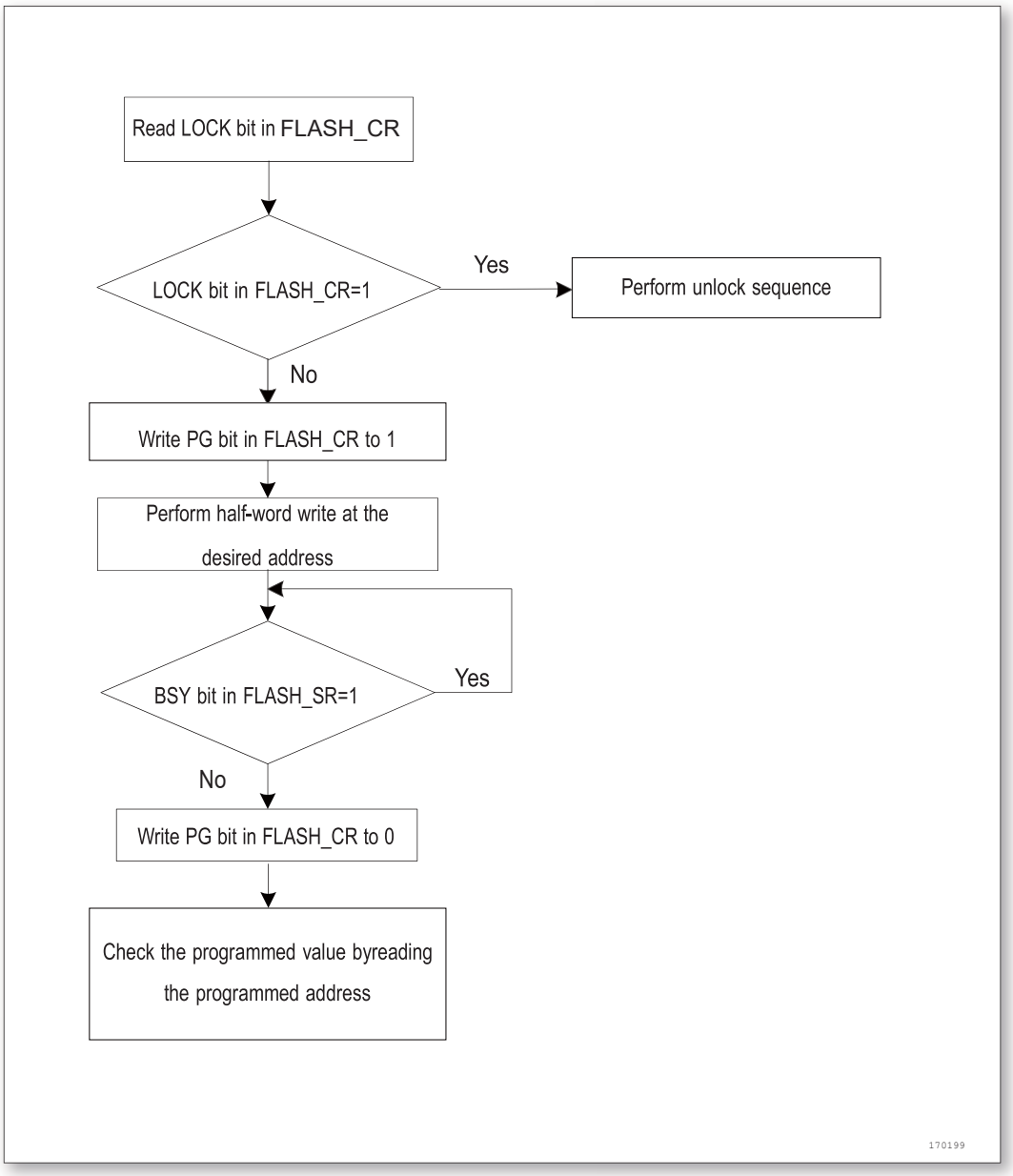

图 2-1 编程流程

Flash 存储器接口会预读待编程字节是否为全 1，如果不是，那么编程操作会自动取消，并且在 FLASH_SR 寄存器的 PGERR 位上提示编程错误警告。

如果待编程地址所对应的 FLASH_WRPR 中的写保护位有效，同样也不会有编程动作，同样也会产生编程错误警告。编程动作结束后，FLASH_SR 寄存器中得 EOP 位会给出提示。

主 Flash 存储器标准模式下的编程过程如下：

- 检查 FLASH_SR 中的 BSY 位，以确认上一操作已经结束
- 置 FLASH_CR 寄存器中的 PG 位
- 以半字为单位向目标地址写入数据

- 等待 FLASH_SR 寄存器中的 BSY 归零
- 读数据以校验

注：当 FLASH_SR 中得 BSY 位为 1 的时候，这些寄存器不能写。

Flash 存储器擦除

Flash 存储器可以按页为单位擦除，也可以整片擦除。

页擦除

擦除页的步骤如下：

- 检查 FLASH_SR 中的 BSY 位，以确认上一操作已经结束
- 置 FLASH_CR 寄存器中得 PER 位为 1
- 写 FLASH_AR 寄存器以选择待擦除的页
- 置 FLASH_CR 寄存器中的 STRT 位为 1
- 等待 FLASH_SR 中的 BSY 归零
- 读取已擦除页以校验

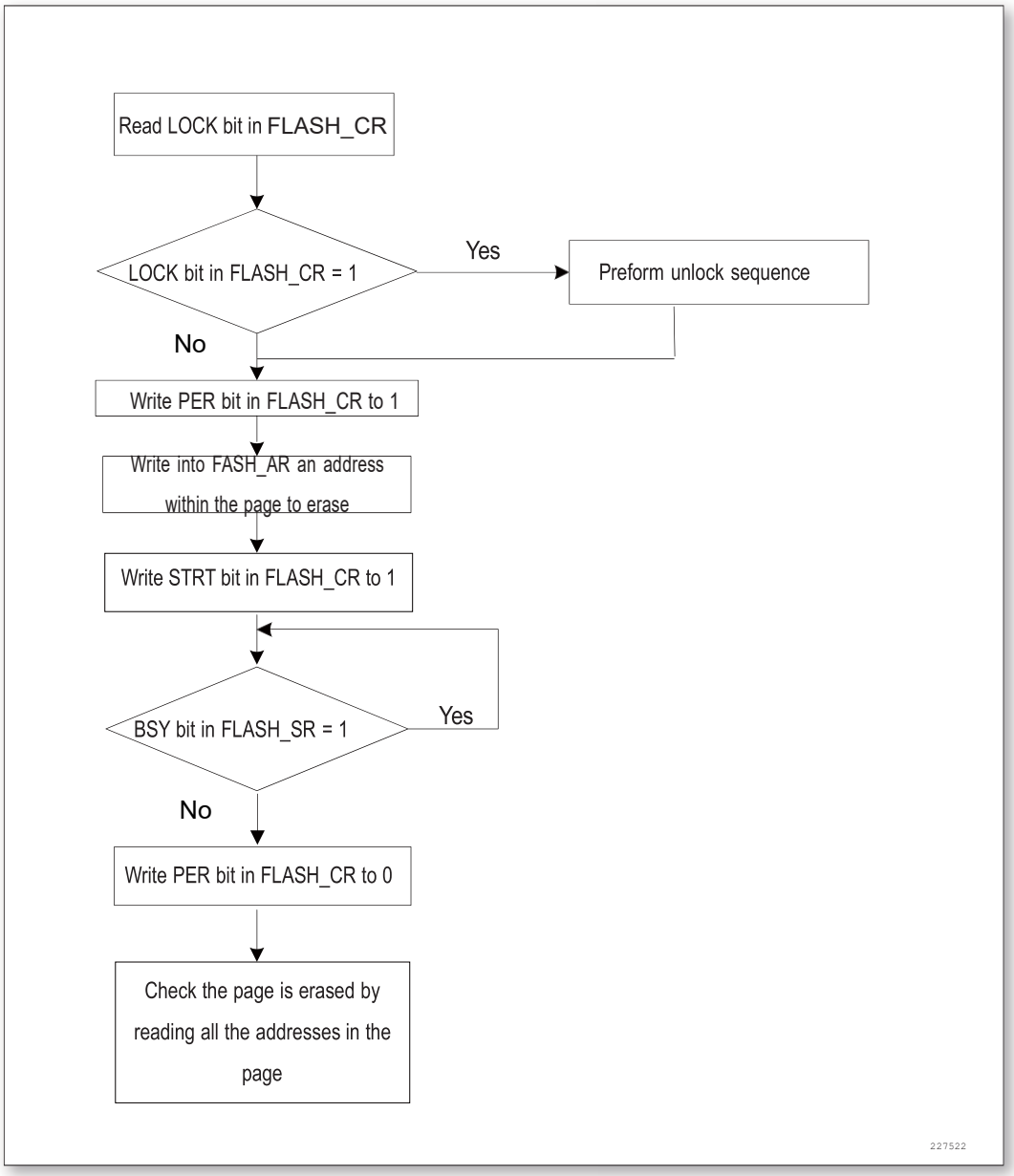

图 2-2 Flash 寄存器页擦除流程

整片擦除

可以用整片擦除命令一次擦除整个 Flash 用户区，但信息块不会受这个命令影响，具体步骤如下：

- 检查 FLASH_SR 中的 BSY 位，以确认上一操作已经结束
- 置 FLASH_CR 寄存器中的 MER 位为 1
- 置 FLASH_CR 寄存器中的 STRT 位为 1
- 等待 BSY 位归零
- 读取全部页并校验

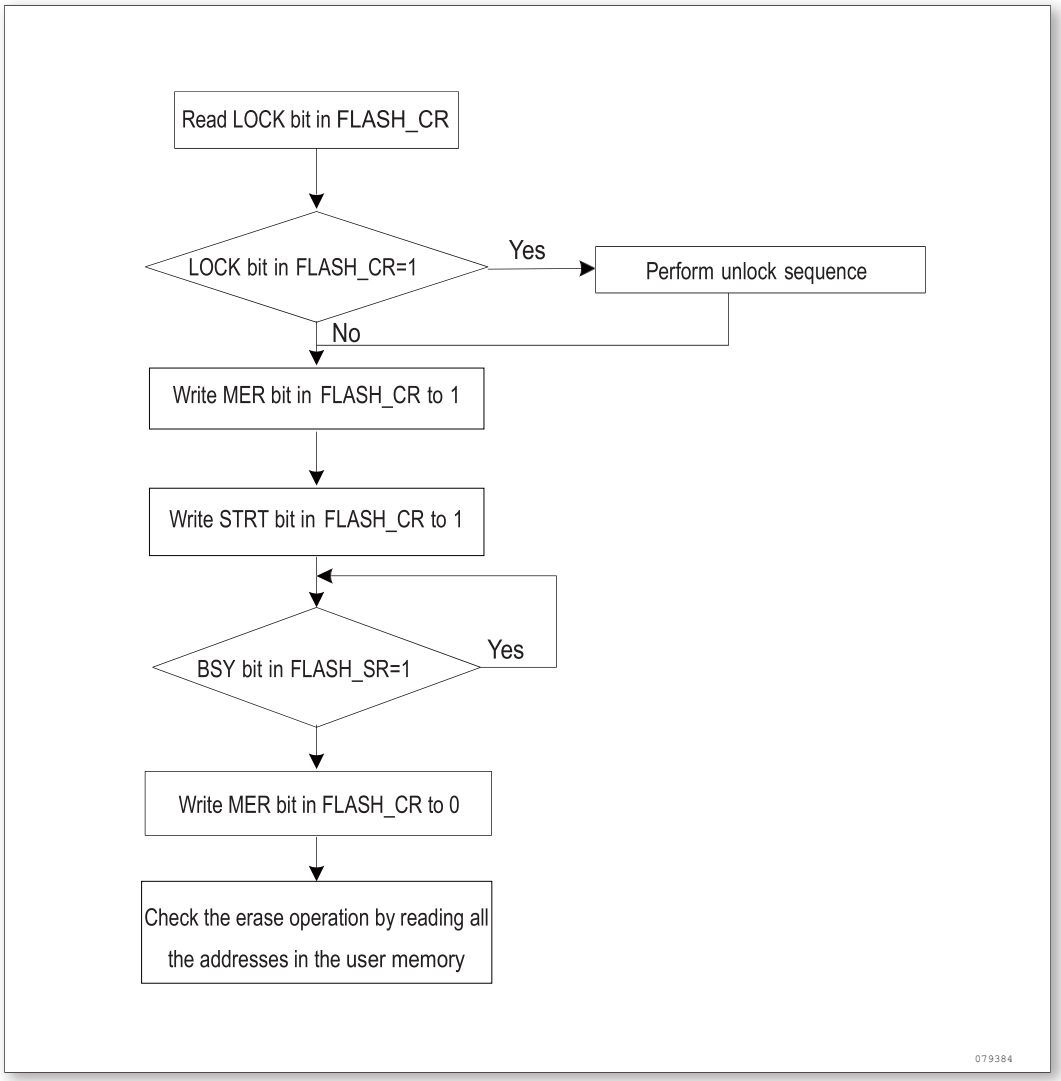

图 2-3 Flash 寄存器整片擦除流程

选项字节编程

选项字节的编程与常规用户地址不同，包括 2 个写保护，1 个硬件配置。解除 Flash 访问限制后，还需要对 FLASH_OPTKEYR 寄存器完成关键字写入操作。完成该操作后，FLASH_CR 寄存器中的 OPTWRE 位会被置‘1’，然后就可以先置位 FLASH_CR 中的 OPTPG 位，再按半字单位写目标地址。同样是会自动检查选项字节是否为 1，否则相关操作会被取消并且在 FLASH_SR 中的 PGERR 位提示错误。编程操作结束后，会由 FLASH_SR 寄存器的 EOP 位给出提示。

选项字节为 16 位数据，有效数据为低 8 位，而高 8 位为低 8 位的反码。在编程过程中，需要先计算出低 8 位的反码作为高 8 位，与低 8 位一起以半字方式写入，保证选项字节的写入值总是高低 8 位互为反码的。步骤如下：

- 检查 FLASH_SR 寄存器中的 BSY 位，以确保上一操作结束

- 解锁 FLASH_CR 寄存器中的 OPTWRE 位
- 置 FLASH_CR 寄存器中 OPTPG 位为 1
- 写数据 (半字) 到目标地址
- 等待 BSY 位归零
- 读取并校验当保护选项字节由保护状态被改成非保护状态时，会自动引发一次整片擦除。 如果用户只想改写其他的字节，则不会引发整片擦除，这个机制用于保护 Flash 的内容。

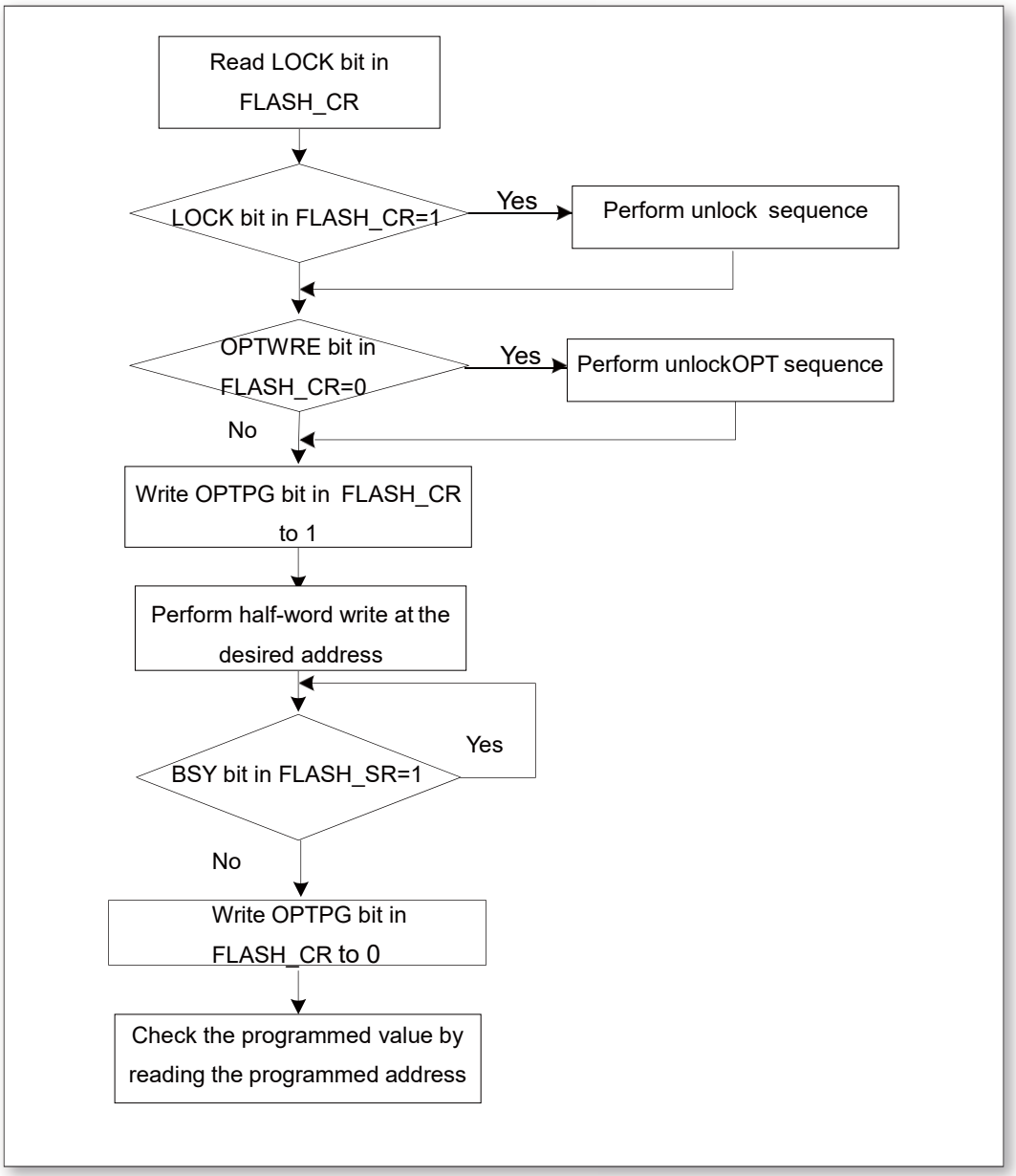

图 2-4 选项字节编程流程

擦除过程

选项字节的擦除过程如下：

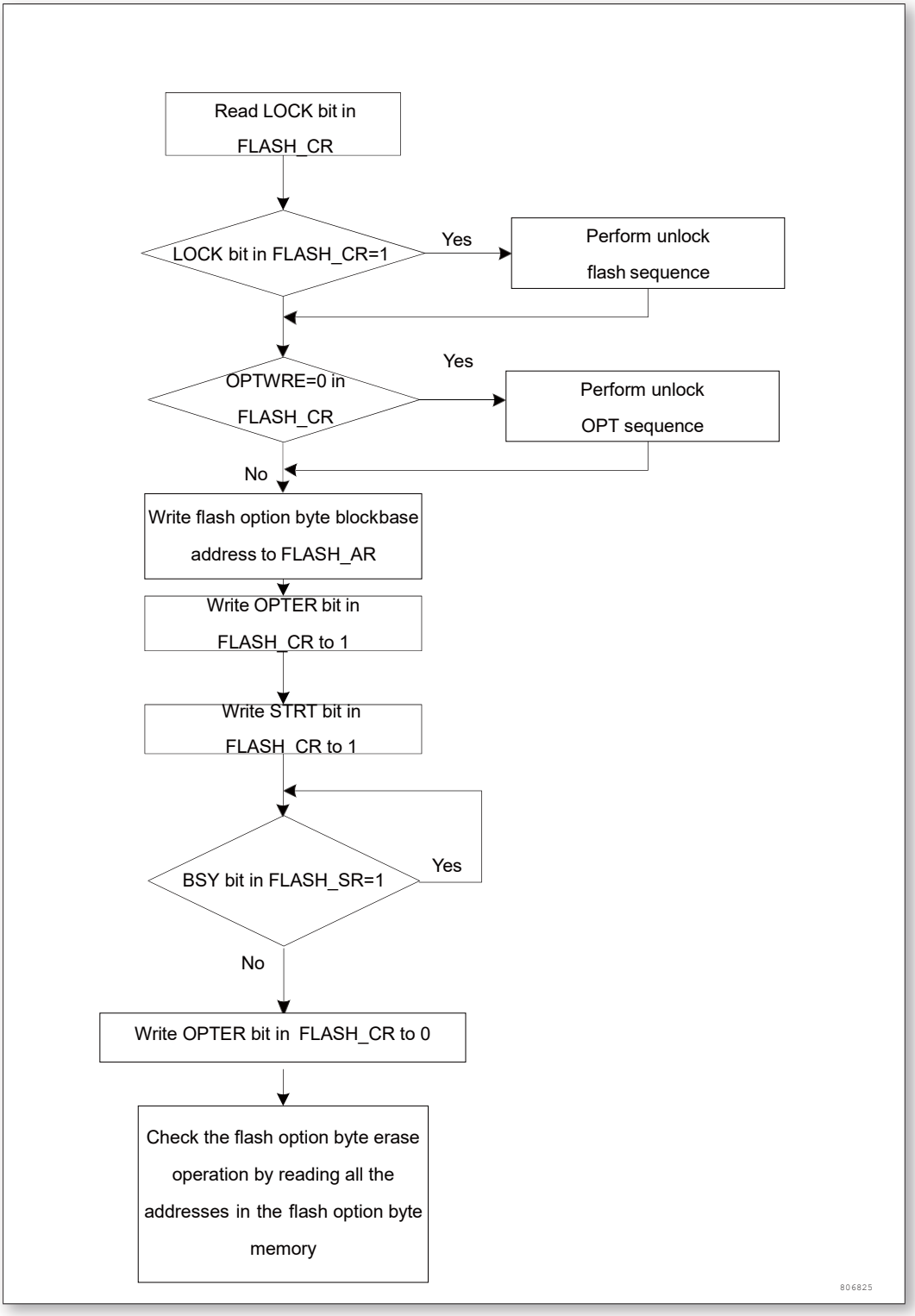

图 2-5 选项字节擦除流程

2.3 存储保护

可以防范用户区 Flash 区的代码被不可信的代码读出，也可以防范在程序跑飞的时候对 Flash 的意外擦除，写保护的最小单位是一个写保护块（4 页）。

2.3.1 读保护

这项保护是通过设置 RDP 一个半字，然后系统重新上电复位，加载了新的 RDPs 后起作用的。

注：如果在设置了读保护时，调试器仍然连接到 SWD 接口，需要执行一次上电复位，而不是（没有调试时的）系统复位。

设置为读保护

按选项字节区块半字编程的操作方式，按顺序写 RDP 半字到对应地址

- 1. 设置 FLASH AR 地址值为 0x1FFFF800，执行区块擦除。
- 2. 写目标值 0x807F 到 0x1FFFF800。
- 3. 软件复位目标系统或者掉电，再上电。

当保护字节被写入相应的值以后：

- 只允许从用户代码执行对主闪存存储器的读操作（以非调试方式从主闪存存储器启动）。
- 读保护后，调试模式下（SRAM boot 和 debug 模式）禁止对 Flash 进行操作，Flash 本身的程序禁止写低 4KB 空间。
- 第 0 ~ 3 页被自动加上了写保护，其它部分的存储器可以通过在主闪存存储器中执行的代码进行编程 (实现 IAP 或数据存储等功能)，但不允许在调试模式下或在从内部 SRAM 启动后执行写或擦除操作 (整片擦除除外)。
- 所有通过 SWD 向内置 SRAM 装载代码并执行代码的功能依然有效，亦可以通过 SWD 从内置 SRAM 启动，这个功能可以用来解除读保护。
- 通过从内置 SRAM 执行代码访问主闪存存储器的操作，通过 SWD(串行线调试)对闪存的访问都将被禁止。

当 RDP 字包含下列数值时，闪存被置于保护状态。

表 2-2 Flash 读保护状态

RDPs 字节的值	读保护状态
RDP=0x807F @0x1FFFF800	保护

注：如选项字节块对应的地址值为非 0xFFFF，需先执行擦除选项字节块的动作，执行擦除选项字节块的动作不会导致自动的整片擦除操作，不会改变读保护状态。

解除读保护

从内置 SRAM 解除读保护的过程是：

按选项字节区域半字编程的操作方式，按顺序写 RDP 一个半字到对应地址:

1. 设置 FLASH_AR 地址值为 0x1FFFF800，执行区块擦除。
2. 写目标值 0x5AA5 到 0x1FFFF800，触发主 Flash 全片擦除，此时状态位为未读保护状态。

进行上电复位以重新加载选项字节，此时读保护被解除。

表 2-3 Flash 解除读保护状态

RDPs 字节的值	读保护状态
对 0x1FFFF800 选项区块擦除 写入 RDP=0x5AA5 @0x1FFFF800 触发对 0x08000000 的主 Flash 全片擦除	解除读保护

2.3.2 主空间写保护

写保护以一个扇区为单位 (4 页) 来控制，配置选项字节中的 WRP 位，随后的系统复位将加载新选项字节就可以使能这个保护。如果试图写入或擦除一个受保护的扇区，会引起 FLASH_SR 中的 WRPRTERR 标志位被置位。

2.3.3 选项字节的写保护

默认状态下，选项字节块始终是可以读且被写保护。要想对选项字节块进行写操作 (编程/擦除) 首先要在 OPTKEYR 中写入正确的键序列 (与上锁时一样)，随后允许对选项字节块的写操作，FLASH_CR 寄存器的 OPTWRE 位标示允许写，清除这位将禁止写操作。

2.4 Flash 中断

表 2-4 Flash 中断请求

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
编程错误	PGERR	ERRIE

2.5 选项字节说明

选项字节由用户根据应用的需要配置；例如：可以选择使用硬件模式的看门狗或软件的看门狗。

在选项字节中每个 32 位的字被划分为下述格式：

表 2-5 选项字节格式

位 31 ～ 24	位 23 ～ 16	位 15 ～ 8	位 7 ～ 0
选项字节 1 的反码	选项字节 1	选项字节 0 的反码	选项字节 0

注：反码必须由用户应用软件实现。

选项字节块中选项字节的组织结构如下表所示。

选项字节可以从下表列出的存储器地址读出，或从选项字节寄存器 (FLASH_OBR) 读出。

注：新写入的选项字节 (用户的或读/写保护的)，在系统复位后才生效。

表 2-6 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			nWRP0	WRP0

表 2-7 选项字节说明

存储器地址	选项字节
0x1FFF F800	位 [31：24] nUSER 位 [23：16] USER：用户选项字节 (保存在 FLASH_OBR[9：2] 中)。这个字节用于配置下列功能： 选择看门狗事件：硬件或软件 注：不使用位 [23: 18] 位 17：nRST_STOP 0：当进入停机 (STOP) 模式时产生复位 1：进入停机 (STOP) 模式时不产生复位 位 16：WDG_SW 0：硬件看门狗 1：软件看门狗位 [15：8]: nRDP 位 [7：0]: RDP：读出保护选项字节 读出保护功能帮助用户保护存在闪存中的代码。该功能由设置 RDP 选项字节启用。当在这个选项字节中写入正确的数值时（RDP = 0x807F），将禁止读出闪存存储器。（RDP 是否开启的结果存储在 FLASH_OBR[1] 中。）
0x1FFF F804	Datax：2 个字节的用户数据 这个地址可以使用选项字节的编程方式编程。 位 [31：24]：nData1 位 [23：16]：Data1(存储在 FLASH_OBR[25：18]) 位 [15：8]：nData0 位 [7：0]：Data0(存储在 FLASH_OBR[17：10])
	WRPx：闪存写保护选项字节 位 [15：8]：nWRP0 位 [7：0]：WRP0(存储在 FLASH_WRPR[7：0])

存储器地址	选项字节
0x1FFF F808	选项字节 WRPx 中的每一个比特位用于保护主存储器中 4 个存储页： 0：实施写保护 1：不实施写保护 一个用户选项字节用于保护总共 32K 字节的主存器。 WRP0：第 0 ~ 31 页的写保护

每次系统复位后，选项字节装载器 (OBL) 读出信息块的数据，并保存在选项字节寄存器 (FLASH_OBR) 中，每个选择位都在信息块中有它的反码位，在装载选择位时反码位用于验证选择位是否正确，如果有任何的差别，将产生一个选项字节错误标志 (OPTERR)。当发生选项字节错误时，对应的选项字节被强置为 0xFF。当选项字节和它的反码均为 0xFF 时(擦除后的状态)，则关闭上述验证功能。

所有的选择位 (不包括它们的反码位) 用于配置该微控制器，CPU 可以读选项字节寄存器。

2.6 寄存器

2.6.1 寄存器总览

表 2-8 FLASH 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	FLASH_ACR	闪存访问控制寄存器	0x00000038
0x04	FLASH_KEYR	FPEC 键寄存器	0x00000000
0x08	FLASH_OPTKEYR	闪存 OPTKEY 寄存器	0x00000000
0x0C	FLASH_SR	闪存状态寄存器	0x00000000
0x10	FLASH_CR	闪存控制寄存器	0x00000080
0x14	FLASH_AR	闪存地址寄存器	0x00000000
0x1C	FLASH_OBR	选项字节寄存器	0x03FFFC1C
0x20	FLASH_WRPR	写保护寄存器	0xFFFFFFFF

2.6.2 FLASH_ACR 闪存访问控制寄存器

偏移地址：0x00

复位值：0x0000 0038

31	30	29	28	27	26	25	24	23	22	21	201	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										Res.	PRFTBE	Res.	LATENCY		
											rw		rw		

Bit	Field	Description
31：6	Res.	保留，必须保持复位值
5	Res.	始终读为 1
4	PRFTBE	预取缓冲区使能（Prefetch buffer enable） 0：关闭预取缓冲区 1：启用预取缓冲区
3	Res.	保留，必须保持复位值
2：0	LATENCY	时延（Latency） 这些位表示 SYSCLK（系统时钟）周期与闪存访问时间的比例。 000：零等待状态，当 0 < SYSCLK ≤ 24MHz 001：一个等待状态，当 24MHz < SYSCLK ≤ 48MHz

2.6.3 FLASH_KEYR FPEC 键寄存器

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

Bit	Field	Description
31：0	FKEYR	FPEC 键（Flash key） 这些位用于输入 FPEC 的解锁键。

注：所有这些位是只写的，读出时返回 0。

2.6.4 FLASH_OPTKEYR 闪存 OPTKEY 寄存器

偏移地址：0x08

复位值：0x0000 0000

31	302	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPTKEYR															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTKEYR															
w															

Bit	Field	Description
31：0	OPTKEYR	选择字节键（Option byte key） 这些位用于输入选项字节的键以解除 OPTWRE。

注：所有这些位是只写的，读出时返回 0。

2.6.5 FLASH_SR 闪存状态寄存器

偏移地址：0x0C

复位值：0x0000 0000

31	302	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	WRPRT ERR	Res.	PGERR	Res.	BSY
										rw	rw		rw		r

Bit	Field	Description
31:6	Res.	保留，必须保持复位值
5	EOP	操作结束（End of operation） 当闪存操作（编程 I 擦除）完成时，硬件设置这位为“1”，写入“1”可以清除这位状态。 注：每次成功的编程或者擦除都会设置 EOP 状态。
4	WRPRTERR	写保护错误（Write protection error） 试图对写保护的闪存地址编程时，硬件设置这位为“1”，写入“1”可以清除这位状态。
3	Res.	保留，必须保持复位值
2	PGERR	编程错误（Programming error） 试图对内容不是“0xFFFF”的地址编程时，硬件设置这位为“1”，写入“1”可以清除这位状态。 注：进行编程操作之前，必须先清除 FLASH_CR 寄存器的 STRT 位。 注：当 PGERR 位为 1，CPU 无法清除，也不能再写入，只能复位。
1	Res.	保留，必须保持复位值
0	BSY	忙（Busy） 该位指示闪存操作正在进行。在闪存操作开始时，该位被置为“1”；在操作结束或发生错误时该位被清除为“0”。

2.6.6 FLASH_CR 闪存控制寄存器

偏移地址：0x10

复位值：0x0000 0080

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

Bit	Field	Description
31:13	Res.	保留，必须保持复位值
12	EOPIE	允许操作完成中断 (End of operation interrupt enable) 该位允许在 FLASH_SR 寄存器中的 EOP 位变为‘1’时产生中断。 0：禁止产生中断 1：允许产生中断
11	Res.	保留，必须保持复位值
10	ERRIE	允许错误状态中断 (Error interrupt enable) 该位允许在发生 FPEC 错误时产生中断(当 FLASH_SR 寄存器中的 PGERR/WRPRTERR 置为‘1’时)。 0：禁止产生中断 1：允许产生中断
9	OPTWRE	允许写选项字节（Option byte write enable） 当该位为“1”时，允许对选项字节进行编程操作。当在 FLASH_OPTKEYR 寄存器写入正确的键序列后，该位被置为“1”。 软件写 0 可清除此位。
8	Res.	保留，必须保持复位值
7	LOCK	锁（Lock） 只能写“1”。当该位为“1”时表示 FPEC 和 FLASH_CR 被锁住。在检测到正确的解锁序列后，硬件自动清除此位为“0”。 在一次不成功的解锁操作后，下次系统复位前，该位不能再被改变。
6	STRT	开始（Start） 当该位为“1”时将触发一次擦除操作。该位只可由软件置为“1”并在 BSY 变为“1”时自动清“0”。
5	OPTER	擦除选项字节（Option byte erase） 擦除选项字节。
4	OPTPG	烧写选项字节（Option byte programming） 对选项字节编程。
3	Res.	保留，必须保持复位值
2	MER	全擦除（Mass erase） 选择擦除所有用户页。
1	PER	页擦除（Page erase） 选择擦除页。
0	PG	编程（Programming） 选择编程操作。

2.6.7 FLASH_AR 闪存地址寄存器

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

Bit	Field	Description
31：0	FAR	用户闪存地址（Flash Address） 当进行页擦除时选择要擦除的页。注意：当 FLASH_SR 中的 BSY 位为“1”时，不能写这个寄存器。

由硬件修改为当前最后使用的地址。页擦除操作中，必须修改这个寄存器以指定要擦除的页。

2.6.8 FLASH_OBR 选项字节寄存器

偏移地址：0x1C

复位值：0x03FF FC1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								Data1						Data0	
								r						r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data0						Res.						nRST_STOP	WDG_SW	RDPRT	OPTERR
r						r						r	r	r	r

Bit	Field	Description
31：26	Res.	保留，必须保持复位值
25：18	Data1	Data1
17：10	Data0	Data0
9：4	Res.	保留，必须保持复位值
3	nRST_STOP	进入停机模式时的复位事件 0：当进入停机（STOP）模式时产生复位 1：进入停机（STOP）模式时不产生复位
2	WDG_SW	选择看门狗事件 0：硬件看门狗 1：软件看门狗

Bit	Field	Description
1	RDPRT	读保护（Read protection level status） 当设置为“1”，表示闪存存储器被读保护。 注：该位为只读。
0	OPTERR	选项字节错误（Option byte error） 当该位为“1”时表示选项字节和它的反码不匹配。 注意：该位为只读。

这个寄存器的复位数值与写入选项字节中的数值相关，OPTERR 位的复位值与加载选项字节时对选项字节和它的反码进行比较的结果相关。

2.6.9 FLASH_WRPR 写保护寄存器

偏移地址：0x20

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
Res.												WRP			
r												r			

Bit	Field	Description
31:4	Res.	保留，必须保持复位值
3:0	WRP	写保护（Write protect） 该寄存器包含由 OBL 加载的写保护选项字节。 0：写保护生效 1：写保护失效

3 CRC 循环冗余校验计算单元

3.1 简介

CRC 计算单元利用固定的多项式来 32 位数据的 CRC 校验值，用于对数据传输或数据存储的完整性进行验证。

3.2 主要特征

- 支持 CRC-32/MPEG-2（以太网）多项式：0x4C11DB7

$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$

- 支持 32 位宽的数据寄存器用于输入/输出
- 硬件计算时间为 3 个 HCLK 周期
- 8 位独立数据寄存器，用于存放临时数据。

3.3 功能描述

3.3.1 功能框图

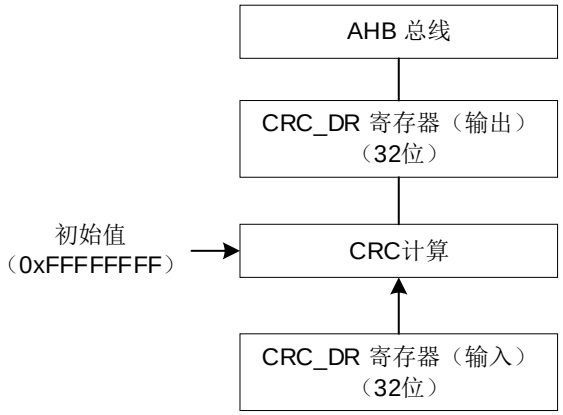

图 3-1 CRC 功能框图

3.3.2 功能概述

- CRC 计算单元含有 1 个 32 位数据寄存器：
- 对该寄存器进行写操作时，作为输入寄存器，可以输入要进行 CRC 计算的新数据。

- 对该寄存器进行读操作时，返回上一次 CRC 计算的结果。
- 每一次写入数据寄存器，其计算结果是前一次 CRC 计算结果和新计算结果的组合 (对整个 32 位字进行 CRC 计算，而不是逐字节地计算)。
- 在 CRC 计算期间会暂停写操作，因此可以对寄存器 CRC_DR 进行背靠背写入或者连续地写-读操作。
- 可以通过设置寄存器 CRC_CR 的 RST 位来重置数据寄存器 CRC_DR 为 0xFFFFFFFF。该操作不影响寄存器 CRC_IDR 内的数据。

3.3.3 使用方法

3.3.3.1 CRC 计算操作步骤

- 使能 CRC 模块时钟；
- 复位 CRC 模块；
- 通过配置 CRC 控制寄存器（CRC_CR）的 RST 位，将 CRC 恢复到初始状态；
- 将数据依次写入 CRC 数据寄存器（CRC_DR）；
- 读取 CRC 数据寄存器（CRC_DR），得到 CRC 计算结果。

3.4 寄存器

3.4.1 寄存器总览

表 3-1 CRC 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	CRC_DR	CRC 数据寄存器	0xFFFFFFFF
0x04	CRC_IDR	CRC 独立数据寄存器	0x00000000
0x08	CRC_CR	CRC 控制寄存器	0x00000000

3.4.2 CRC_DR CRC 数据寄存器

偏移地址：0x00

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DR															
40															

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

Bit	Field	Description
31：0	DR	数据寄存器（Data Register） 写入时，作为输入寄存器，将写入的数据和前一次的结果做 CRC 计算 读取时，返回 CRC 计算的结果

3.4.3 CRC_IDR CRC 独立数据寄存器

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
Reserved															

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

Bit	Field	Description
31：8	Reserved	保留，必须保持复位值
7：0	IDR	8 位通用数据寄存器（General-purpose 8-bit Data Register） 临时存放 1 个字节的数据空间。 不受 CRC 控制寄存器（CRC_CR）的 RST 位影响。

3.4.4 CRC_CR CRC 控制寄存器

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
Reserved															

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

Bit	Field	Description
31：1	Reserved	保留，必须保持复位值
0	RST	复位 CRC 计算单元（CRC Reset） CRC 数据寄存器（CRC_DR）复位为 0xFFFF FFFF 该位只能写“1”，硬件自动清“0”

4 PWR 电源控制

4.1 电源

芯片的工作电压（V_{DD}）为 2.0V~5.5V。通过内置的电压调节器提供所需的内核电源。

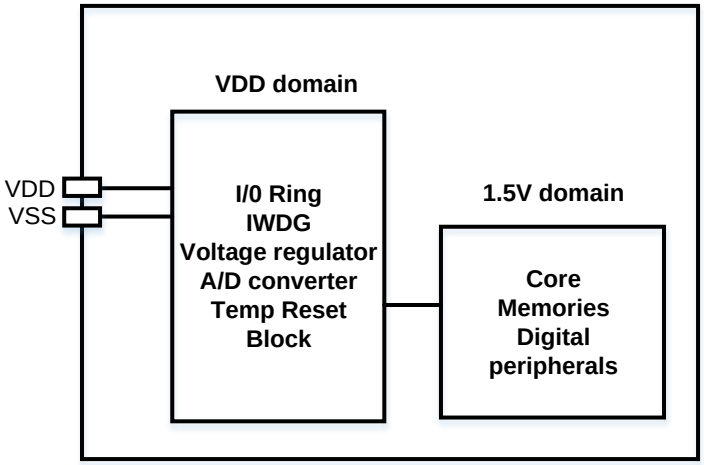

图 4-1 电源控制功能框图

4.1.1 电压调节器

复位后调节器总是使能的。根据应用方式它以 3 种不同的模式工作。

- 运行模式：调节器以正常功耗模式提供 1.5V 电源（内核，内存、外设）。
- 停机模式：调节器以低功耗模式提供 1.5V 电源，以保存寄存器和 SRAM 的内容。
- 深度停机模式：内核以更低功耗模式工作，所有的时钟都已停止，寄存器和 SRAM 的内容依然保存。

4.2 电源管理器

4.2.1 POR 上电复位和 PDR 掉电复位

芯片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 2.0V 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细节请参考数据手册的电气特性部分。

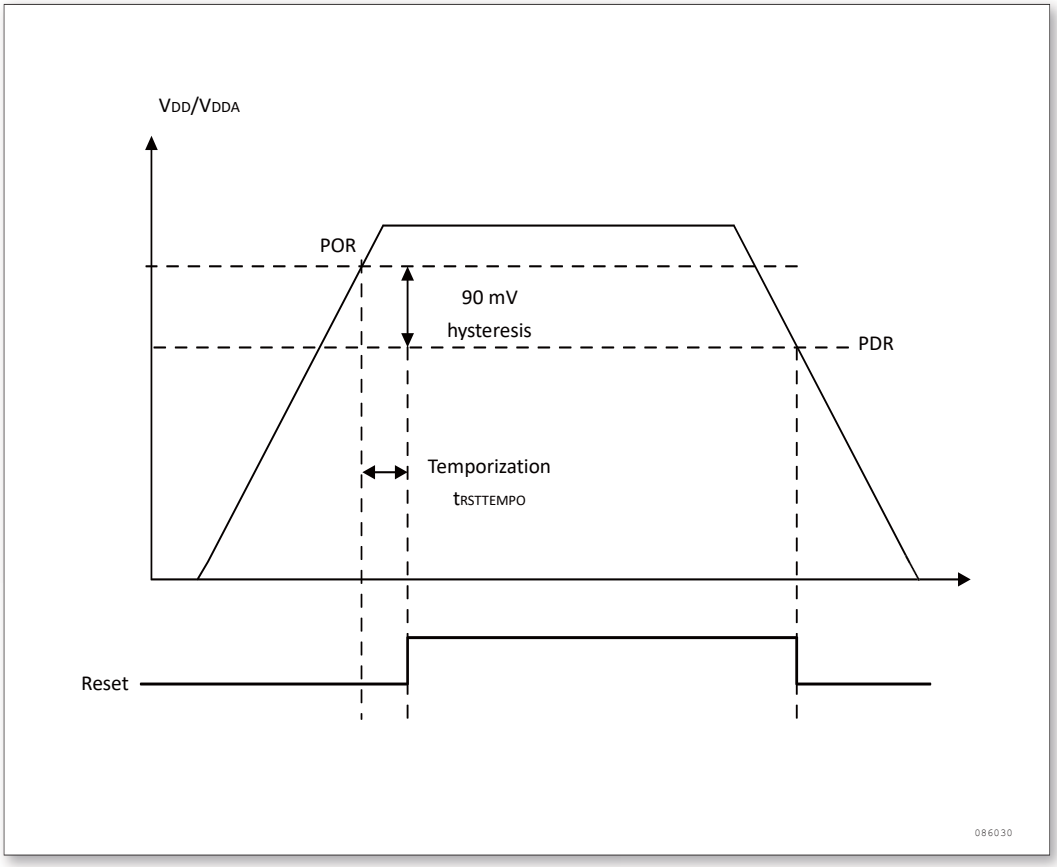

图 4-2 上电复位和掉电复位波形图

4.2.2 PVD 可编程电压监测器

用户可以利用 PVD 对 V_{DD} 电压与电源控制寄存器 PWR_CR 中的 PLS 位进行比较来监控电源，这几位选择监控电压的阈值。

通过设置 PVDE 位来使能 PVD。

电源控制/状态寄存器 PWR_CSR 中的 PVDO 标志用来表明 V_{DD} 是高于还是低于 PWR_CR 中的 PLS 位选择的阈值电压。该事件在内部连接到外部中断的第 16 线，如果该中断在外部中断寄存器中是使能的，该事件就会产生中断。当 V_{DD} 下降到 PVD 阈值以下或当 V_{DD} 升到 PVD 阈值以上时，根据外部中断第 16 线的上升/下降边沿触发设置，就会产生 PVD 中断 （也可以通过软件配置产生 PVD 复位）。例如：这一特性可用于执行紧急关闭任务。

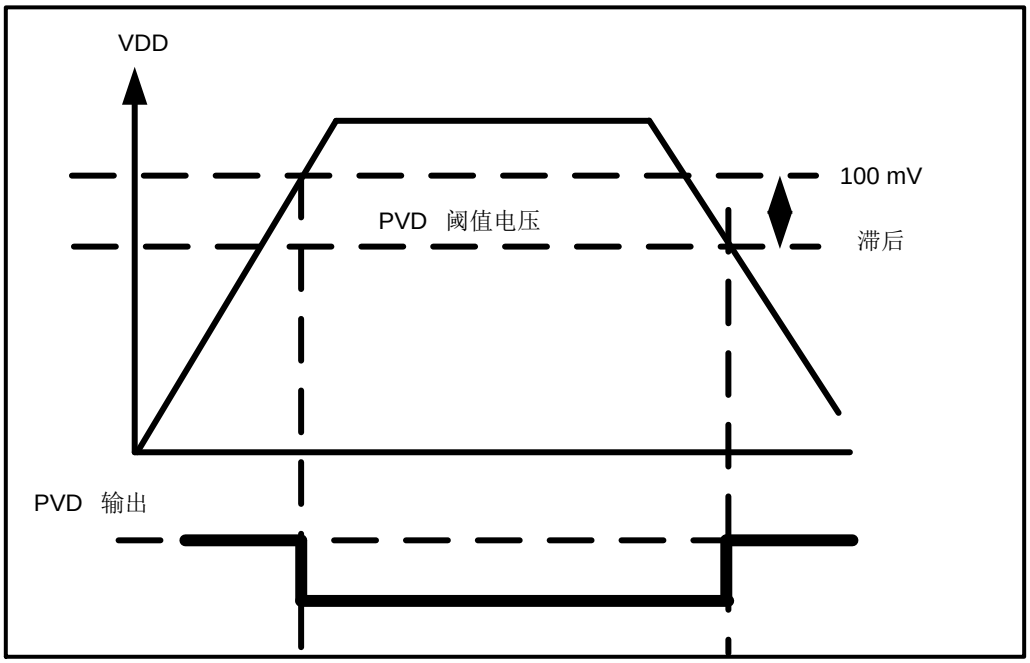

图 4-3 PVD 阈值波形图

4.3 低功耗模式

在系统或电源复位以后，微控制器处于运行状态。当 CPU 不需要继续运行时，可以利用多种低功耗模式来节省功耗，例如等待某个外部事件时。用户需要根据最低电源消耗、最快速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。

芯片有三种低功耗模式：

- 睡眠模式：CPU 停止，所有外设包括 CPU 的外设，如 NVIC、SysTick 等仍在运行。
- 停机模式：所有的时钟都已停止，寄存器和 SRAM 的内容依然保存。
- 深度停机模式：内核以更低功耗模式工作，所有的时钟都已停止，寄存器和 SRAM 的内容依然保存。

此外，在运行模式下，可以通过以下方式中的一种降低功耗：

- 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 4-1 低功耗模式列表

模式	进入方式	唤醒方式	对 1.5V 域时钟的影响	对 VDD 域时钟的影响	电压稳压器
睡眠（SLEEP NOW 或 SLEEP ON EXIT）	WFI（Wait for Interrupt）	任一中断	CPU 时钟关，对其他时钟和	无	开

模式	进入方式	唤醒方式	对 1.5V 域时钟的影响	对 VDD 域时钟的影响	电压稳压器
	WFE（Wait for Event）	唤醒事件	ADC 时钟无影响		
停机	清除 LPDS 位；设置 SLEEPDEEP 位；WFI 或 WFE；	任一外部中断（在外部中断寄存器中设置）	所有使用 1.5V 域的时钟都已关闭	HSI 关闭	开
深度停机	设置 LPDS 位；SLEEPDEEP 位；WFI 或 WFE；	任一外部中断（在外部中断寄存器中设置）			开

4.3.1 运行模式

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、PCLK1）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。

详见：时钟配置寄存器（RCC_CFGR）

外设时钟的控制

在运行模式下，任何时候可以通过停止为外设和内存提供时钟（HCLK 和 PCLKx）来减少功耗。为了在睡眠模式下更多地减少功耗，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（RCC_AHBENR）和 APB1 外设时钟使能寄存器（RCC_APB1ENR）来开关各个外设模块的时钟。

4.3.2 睡眠模式

进入睡眠模式

通过执行 WFI 或 WFE 指令进入睡眠状态。根据 CPU 系统控制寄存器（SCB->SCR）中的 SLEEPONEXIT 位的值，有两种可用于选择进入睡眠模式的机制：

SLEEPNOW：如果 SLEEPONEXIT 位被清除，当 WFI 或 WFE 被执行时，MCU 立即进入睡眠模式。

SLEEPONEXIT：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后 MCU 立即进入睡眠模式。

在睡眠模式下，所有的 I/O 引脚都保持在运行模式时的状态。

退出睡眠模式

中断或事件发生后，睡眠模式立即被唤醒。

关于如何进入/退出睡眠模式，更多的细节参考下表

表 4-2 SLEEPNOW 模式

SLEEP NOW 模式	说明
进入	在以下条件下执行 WFI（Wait for Interrupt） 或 WFE（Wait for Event） 指令： SLEEPDEEP = 0 SLEEPONEXIT = 0
退出	如果执行 WFI 进入睡眠模式：中断（参考中断向量表） 如果执行 WFE 进入睡眠模式：唤醒事件（参考唤醒事件管理）
唤醒延时	立即唤醒

表 4-3 SLEEPONEXIT 模式

SLEEP ON EXIT 模式	说明
进入	在以下条件下执行 WFI（Wait for Interrupt）指令： SLEEPDEEP = 0 SLEEPONEXIT = 1
退出	中断（参考中断向量表）
唤醒延时	立即唤醒

4.3.3 停机模式

停机模式是在 CPU 深度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模式。此时在 1.5V 供电区域的所有时钟都被停止，HSI 的功能被禁止，SRAM 和寄存器内容被保留下来。

在停机模式下，所有的 I/O 引脚都保持在运行模式时的状态。

进入停机模式

通过对独立的控制位进行编程，停机模式根据唤醒方式的不同有两种进入方式：

- 等待外部中断线 WFI 方式进入停机模式：配置电源控制寄存器（PWR_CR）的 LPDS = 0 ；CPU 系统控制寄存器（SCR）的 SLEEPDEEP=1。当 WFI 被执行时，MCU 立即进入停机模式。
- 等待外部事件 WFE 方式进入停机模式：配置电源控制寄存器 （PWR_CR）的 LPDS = 0 ；CPU 系统控制寄存器（SCR）的 SLEEPDEEP=1。当 WFE 被执行时，MCU 立即进入停机模式。

进入停机模式时可选择以下功能：

- 独立看门狗 （IWDG）：可通过写入独立看门狗的键寄存器或硬件选择来启动独立看门狗。

- 内部低速振荡器 （LSI 振荡器）：通过 RCC 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 ADC 仍然消耗电流。通过设置寄存器 ADC_ADCFG 的 ADEN 位为 0 可关闭这个外设。其它没有使用的 GPIO 需要设置模拟输入模式，否则有电流消耗。

退出停机模式

当停机模式被中断或事件唤醒且退出后，系统时钟源硬件自动选择为 HSI 振荡器，时钟频率为 HSI 的 6 分频。

当电压稳压器处于正常功耗模式下，系统从停机模式退出时，将会有一段额外的启动延时。

关于如何进入/退出停机模式，详见下表。

表 4-4 停机模式

停机模式	说明
进入	在以下条件下执行 WFI（Wait for Interrupt） 或 WFE（Wait for Event） 指令： 置位 CPU 系统控制寄存器中的 SLEEPDEEP 位； 复位电源控制寄存器（PWR_CR）中的 LPDS 位； 系统时钟切换至 LSI 或 HSI； 注：为了进入停机模式，所有的外部中断的请求位（中断事件挂起寄存器 EXTI_PR） 标志都必须被清除，否则停机模式的进入流程将会被跳过，程序继续运行。
退出	在以下条件下执行 WFI（Wait for Interrupt） 指令： 任一外部中断线被设置为中断模式 （相应的外部中断向量在 NVIC 中必须使能），参见中断向量表 Wait for Event； 在以下条件下执行 WFE（Wait for Event） 指令： 任一外部中断线被设置为事件模式，例如看门狗中断；
唤醒延时	LSI 或 HSI 的唤醒时间和电压稳压器唤醒产生的额外时间
注意事项	在进入停机模式时需将不使用的 GPIO 设置成模拟输入模式

4.3.4 深度停机模式

深度停机是在 CPU 深度睡眠模式的基础上结合了外设的时钟控制和电压稳压器控制机制的一种低功耗模式。在深度停机模式下，1.5V 域的所有时钟都被停止，HSI 的功能被禁止，SRAM 和寄存器内容被保留下来。

在深度停机模式下，所有的 I/O 引脚都保持在运行模式时的状态。

进入深度停机模式

通过对独立的控制位进行编程，深度停机模式根据唤醒的方式不同有两种进入方式：

- 等待外部中断线 WFI 方式进入深度停机模式：配置电源控制寄存器（PWR_CR）的 LPDS = 1 ；CPU 系统控制寄存器（SCR）的 SLEEPDEEP=1。当 WFI 被执行时，MCU 立即进入深度停机

模式。

- 等待外部事件 WFE 方式进入深度停机模式：配置电源控制寄存器 （PWR_CR）的 LPDS = 1 ；CPU 系统控制寄存器（SCR）的 SLEEPDEEP=1。当 WFE 被执行时，MCU 立即进入深度停机模式。

进入停机模式时可选择以下功能：

- 独立看门狗 （IWDG）：可通过写入独立看门狗的键寄存器或硬件选择来启动独立看门狗。
- 内部低速振荡器（LSI 振荡器）：通过 RCC 控制/状态寄存器 （RCC_CSR） 的 LSION 和 LSI_OEN_LV 位来设置。

退出深度停机模式

当深度停机模式被中断或事件唤醒且退出后，系统时钟为 HSI 振荡器，时钟频率为 HSI 的 6 分频。

当电压稳压器处于运行低功耗模式下，系统从深度停机模式退出时，将会有一段额外的启动延时。

关于如何进入/退出深度停机模式，详见下表。

表 4-5 深度停机模式

深度停机模式	说明
进入	在以下条件下执行 WFI（Wait for Interrupt） 或 WFE（Wait for Event） 指令： 置位 CPU 系统控制寄存器中的 SLEEPDEEP 位； 置位电源控制寄存器（PWR_CR）中的 LPDS 位；
退出	任一外部中断或唤醒事件
唤醒延时	LSI 或 HSI 的唤醒时间和电压稳压器唤醒产生的额外时间

调试模式

默认情况下，如果在进行调试微处理器时，使微处理器进入停止或待机模式，将失去调试连接。这是因为 CPU 内核失去了时钟。

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

4.4 电源控制寄存器

表 4-6 电源控制寄存器概览

Offset	Acronym	RegisterName	Reset
0x00	PWR_CR	电源控制寄存器	0x00000000
0x04	PWR_CSR	电源控制状态寄存器	0x00000000

4.4.1 PWR_CR 电源控制寄存器

地址偏移：0x00

复位值：0x00000000

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		Res	PLS				Res	Res			PVDE	Res			LPDS
			rw								rw				rw

Bit	Field	Description
31:13	Reserved	保留，始终读为 0
12：9	PLS	PVD 电平选择（PVD level selection） 这些位用于选择电源电压监测器的电压阀值。 0000：1.8V　0100：3.0V　1000：4.2V 0001：2.1V　0101：3.3V　1001：4.5V 0010：2.4V　0110：3.6V　1010：4.8V 0011：2.7V　0111：3.9V　其他：保留 注：详细说明参见数据手册中的电气特性部分。
8	Reserved	保留，始终读为 0
7:5	Reserved	保留，始终读为 0
4	PVDE	电源电压监测器（PVD）使能（Power voltage detector enable） 1 = 开启 PVD 0 = 禁止 PVD
3:1	Reserved	保留，始终读为 0
0	LPDS	深睡眠下的低功耗 （Low Power Deepstop） 1：进入停机模式时，电压稳压器处于低功耗模式。 0：进入停机模式时，电压稳压器处于正常功耗模式。 当进入停机模式时，LPDS = 1 时的电流小于 LPDS = 0 时的电流。详见该芯片对应的数据手册。

4.4.2 PWR_CSR 电源控制/状态寄存器

地址偏移：0x04

复位值：0x00000000

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													PVDO	Res	
													r		

Bit	Field	Description
31:3	Reserved	保留，始终读为 0
2	PVDO	PVD 输出（PVD output） 当 PVD 被 PVDE 位使能后该位才有效。 1 = VDD/VDDA 低于由 PLS[3：0]选定的 PVD 阀值 0 = VDD/VDDA 高于由 PLS[3：0]选定的 PVD 阀值 注：在待机模式下 PVD 被停止。因此，待机模式后或复位后，直到设置 PVDE 位之前，该位为 0。
1:0	Reserved	保留，始终读为 0

5 RCC 时钟和复位

5.1 复位单元

5.1.1 简介

系统共有两大类复位：电源复位、系统复位。

5.1.2 功能框图

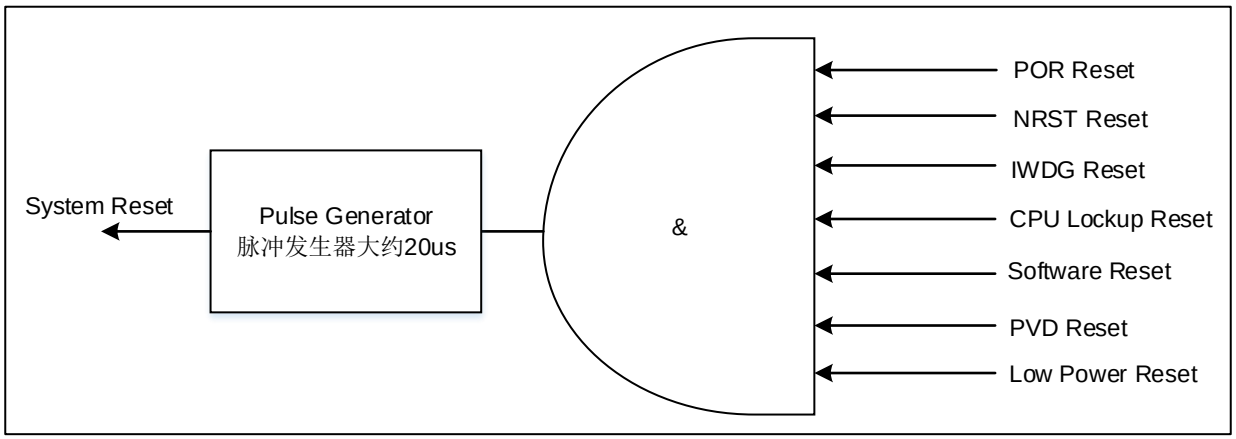

图 5-1 复位功能框图

5.1.3 主要特征

- 复位事件判定：通过控制状态寄存器（RCC_CSR）中的复位标志位来进行判断。
- 电源复位：复位所有寄存器。
- 系统复位：除了时钟控制寄存器（RCC_CSR）中的复位标志以及内部低速振荡器使能标志、电源控制寄存器（PWR_CSR）中的唤醒标志、DBG 控制寄存器（DBG_CR）不受系统复位影响，其余寄存器都将被系统复位。

5.1.4 功能描述

5.1.4.1 电源复位（POR Reset）

电源复位有以下方式：

- 上电复位 (Power-on Reset)
- 掉电复位 (Power-down Reset)

5.1.4.2 系统复位（System Reset）

系统复位有以下几种方式：

- 外部复位（NRST Reset）
- 独立看门狗复位（IWDG Reset）
- 软件复位（Software Reset）
- CPU 死锁复位（CPU Lockup Reset）
- PVD 复位（PVD Reset）

外部复位（NRST Reset）：

- 当通过 NRST Pin 输入低电平时，将会发生外部复位。

独立看门狗复位（IWDG Reset）：

- 计数器开始从其复位值 0x0FFF 开始递减，当递减到达 0x0000 时，将会发生独立看门狗复位。
- 比较/输出如果程序异常，无法正常喂狗，将会发生独立看门狗复位。
- 具体请参考独立看门狗章节

软件复位（Software Reset）：

- 可以通过将 SCB_AIRCR[SYSRESETREQ]置 1，将会发生软件复位

CPU 死锁复位（CPU Lockup Reset）：

- 配置控制状态寄存器（RCC_CSR）的 LOCKUPEN 位为 1，CPU 死锁复位使能；
- 当 CPU 进入锁定状态将会发生 CPU 死锁复位。

PVD 复位（PVD Reset）：

- 配置控制状态寄存器（RCC_CSR）的 PVDRSTEN 位为 1，PVD 复位使能；
- 配置电源控制寄存器（PWR_CR）的 PVDE 位为 1，使能 PVD；
- 配置电源控制寄存器（PWR_CR）的 PLS 位进行 PVD 阈值选择；
- 检测 VDD 电源，当 VDD 电源低于选择的阈值电压时将会发生 PVD 复位。

5.2 时钟单元

5.2.1 简介

四个可配置的的独立系统时钟源：

- 内部高速时钟（HSI）
- 内部高速时钟 6 分频（HSIDIV）
- 外部高速时钟（HSE）
- 内部低速时钟（LSI）

5.2.2 功能框图

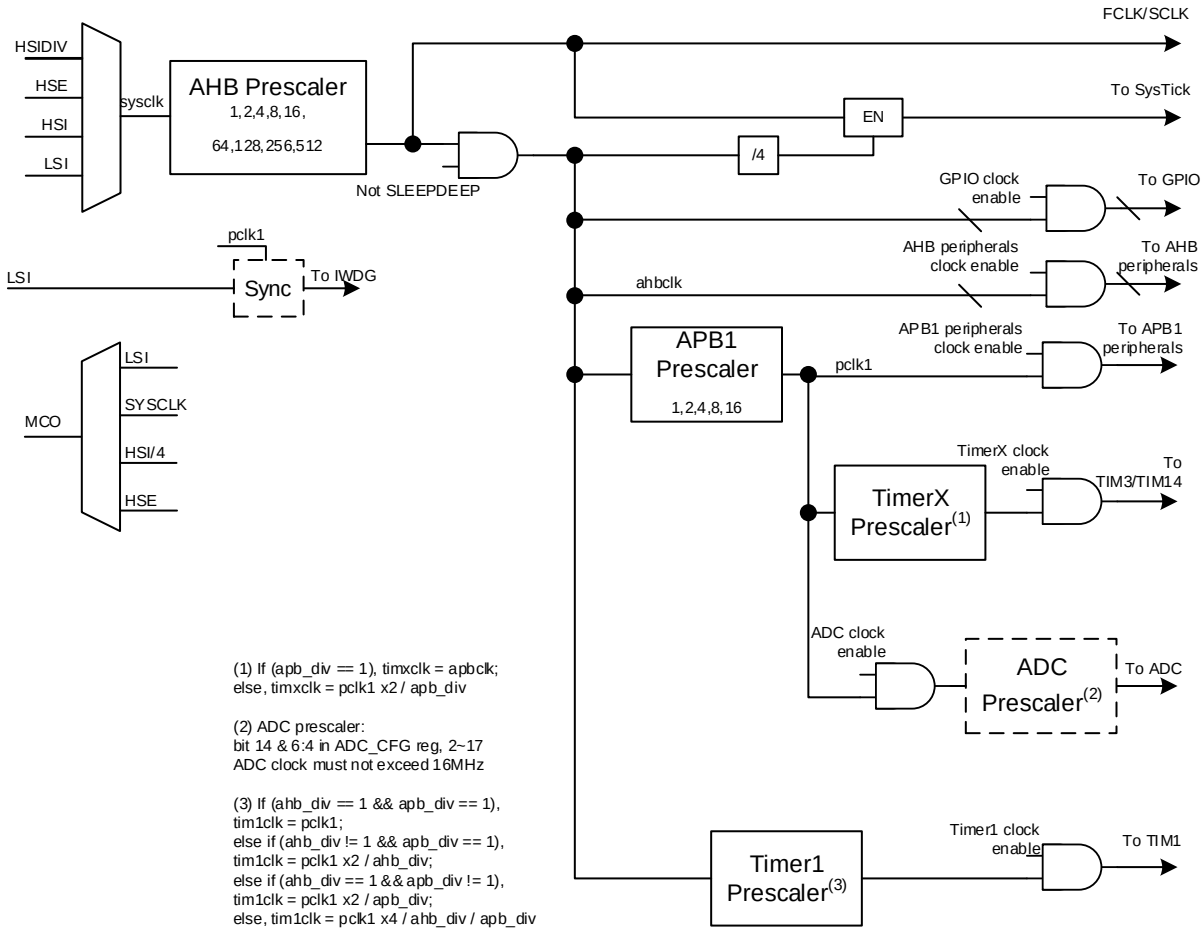

图 5-2 时钟树

5.2.3 主要特征

通过时钟配置寄存器（RCC_CFGR）的预分频控制位来分别配置 AHB、APB1 总线的时钟频率。AHB

和 APB1 总线时钟的最大频率是 48MHz。

5.2.4 功能描述

5.2.4.1 外部高速时钟（HSE）

外部高速时钟的时钟源有以下一种：

- 外部高速输入时钟

通过配置时钟控制寄存器（RCC_CR）的 HSEON 位来选择此时钟输入方式。

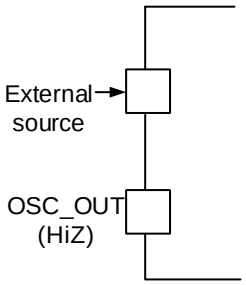

图 5-3 外部高速输入时钟

上图为外部高速输入时钟模块框图

外部高速时钟具有以下特征：

- 必须提供外部高速时钟源或晶体振荡器
- 输入频率范围 4~24MHz
- 推荐外部高速时钟信号为 50%占空比（方波、正弦波），详情请参阅数据手册中有关电气特性部分
- 在使能 HSE 时，必须驱动 OSC_IN 管脚，而 OSC_OUT 引脚在未被使用时应该保持悬空

注意事项：

- 一旦 HSE 启用，HSE 相关配置就不能更改。如需更改配置，必须先禁止 HSE。

启用 HSE 配置流程如下：

- 配置时钟控制寄存器（RCC_CR）中的 HSEON 位为 1，使能 HSE；
- 等待时钟控制寄存器（RCC_CR）中的 HSERDY 位被置位为 1，表示 HSE 稳定，会输出有效时钟信号，此时才可被选择使用作为系统时钟或外设时钟源。

5.2.4.2 内部高速时钟（HSI）

HSI 时钟信号由内部 HSI 振荡器产生，HSI 时钟源在芯片上电后默认启用。

使能 HSI 配置步骤：

- 配置时钟控制寄存器（RCC_CR）中的 HSION 位为 1，使能 HSI；
- 等待时钟控制寄存器（RCC_CR）中的 HSIRDY 位被置位为 1，表示 HSI 稳定，可输出有效时钟，此时才可被选择使用作为系统时钟或外设时钟源。

注意事项：

- 一旦 HSI 启用，HSI 相关配置就不能更改。如需更改配置，必须先禁止 HSI。

5.2.4.3 内部低速时钟（LSI）

LSI 作为一个低功耗时钟源，为独立看门狗提供时钟源。时钟中心频率在 40kHz 左右。详情请参阅数据手册中有关电气特性部分。

使能 LSI 配置步骤：

- 配置控制状态寄存器（RCC_CSR）的 LSION 位置位为 1，使能 LSI；
- 等待控制状态寄存器（RCC_CSR）中的 LSIRDY 位被置位为 1，表示 LSI 稳定， 可输出有效时钟。

注意事项：

- 一旦 LSI 启用，LSI 相关配置就不能更改。如需更改配置，必须先禁止 LSI。

5.2.4.4 中断

表 5-1 RCC 全局中断表

中断事件	事件标志位	使能控制位	标志清除位
RCC_HSERDY	HSERDYF	HSERDYIE	HSERDYC
RCC_HSIRDY	HSIRDYF	HSIRDYIE	HSIRDYC
RCC_LSIRDY	LSIRDYF	LSIRDYIE	LSIRDYC

注：如上标志位/控制位/清除位都可通过时钟中断寄存器（RCC_CIR）进行配置。

5.2.4.5 系统时钟选择（SWS）

四个系统时钟源：

- 内部高速时钟 HSI6 分频（上电后默认）
- 内部高速时钟 HSI

- 外部高速时钟 HSE
- 内部低速时钟 LSI

系统时钟配置步骤：

- 使能需要的系统时钟源（HSI6 分频，HSI，HSE，LSI），每个时钟使能方式不同，具体方式请查看（HSI，HSE，LSI 章节）；
- 等待被选择的时钟源 RDY 信号被置位为 1，表示系统时钟源稳定（当目标时钟源稳定后，系统时钟才可以切换）；
- 通过配置时钟配置寄存器（RCC_CFGR）的 SW 位来选择系统时钟；
- 通过读取时钟配置寄存器（RCC_CFGR）的 SWS 位，判断当前系统时钟的时钟源。

5.2.4.6 系统时钟频率切换

系统时钟频率从低速到高速切换，或者从高速到低速切换，推荐先切换到中速频率过度，高速低速切换间隔至少 1us。

5.2.4.7 外设复位

可以通过 APB1 外设复位寄存器（RCC_APB1RSTR）和 AHB 外设复位寄存器（RCC_AHBRSTR）来实现相应外设的软件复位。

5.2.4.8 微控制器时钟输出（MCO）

微控制器时钟输出（MCO）允许时钟输出到外部 MCO 引脚上。相应 GPIO 端口的配置寄存器必须被配置为复用输出功能。可以选择以下五个时钟信号中的一个作为 MCO 输出时钟：

表 5-2 MCO 与时钟源对应关系

时钟配置寄存器（RCC_CFGR）的 MCO 位	时钟源
00x	没有时钟输出
010	LSI
100	SYSCLK
101	HSI/4
110	HSE

5.2.4.9 独立看门狗时钟

硬件启动独立看门狗，LSI 振荡器将被自动开启，并且不能被关闭；

软件启动独立看门狗，则 LSI 振荡器需通过软件使能开启，在 LSI 振荡器稳定输出后，时钟供应给 IWDG，LSI 可以被软件关闭。

5.3 寄存器

5.3.1 寄存器总览

表 5-3 RCC 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	RCC_CR	时钟控制寄存器	0x00000001
0x04	RCC_CFGR	时钟配置寄存器	0x00000000
0x08	RCC_CIR	时钟中断寄存器	0x00000000
0x10	RCC_APB1RSTR	APB1 外设复位寄存器	0x00000000
0x14	RCC_AHBENR	AHB 外设时钟使能寄存器	0x00000014
0x1C	RCC_APB1ENR	APB1 外设时钟使能寄存器	0x00000000
0x24	RCC_CSR	控制状态寄存器	0x08000000
0x28	RCC_AHBRSTR	AHB 外设复位寄存器	0x00000000
0x40	RCC_SYSCFG	系统配置寄存器	0x00000003

5.3.2 RCC_CR 时钟控制寄存器

偏移地址：0x00

复位值：0x0000 0001

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
Reserved														HSERDY	HSEON
														r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														HSIRDY	HSION
														r	rw

Bit	Field	Description
31：18	Reserved	保留，必须保持复位值

Bit	Field	Description
17	HSERDY	外部高速时钟稳定标志（External High-speed Clock Ready Flag） 由硬件设置。 0：外部高速晶体振荡器未稳定 1：外部高速晶体振荡器已稳定
16	HSEON	外部高速时钟使能（External High-speed Clock Enable） 通过软件置“1”或清“0”。 当进入待机或停机模式时，此位由硬件清“0”。当系统时钟已经或将要使用 HSE 作为时钟源时，则禁止重置此位。 0：禁止外部高速晶体振荡器 1：使能外部高速晶体振荡器
15：2	Reserved	保留，必须保持复位值
1	HSIRDY	内部高速时钟稳定标志（Internal High-speed Clock Ready Flag） 由硬件置“1”，表示内部时钟已经稳定。 在 HSION 位被清除后，HSIRDY 在 3 个 AHB 时钟周期后变“0”。 0：内部高速时钟未稳定 1：内部高速时钟已稳定
0	HSION	内部高速时钟使能（Internal High-speed Clock Enable） 通过软件置“1”或清“0”。 当退出待机或停机模式或外部振荡器用作系统时钟并且发生故障时，此位由硬件置“1”，来迫使内部振荡器使能。当系统时钟已经或将要使用 HSI 作为时钟源时，则禁止重置此位。 0：禁止内部高速时钟 1：使能内部高速时钟

5.3.3 RCC_CFGR 时钟配置寄存器

偏移地址：0x04

复位值：0x0000 0000

访问：无等待状态，字，半字和字节访问

只有当访问发生在时钟切换时，才会插入 1 或 2 个等待周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.					MCO			Res.							
					rw										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					PPRE1			HPRE				SWS		SW	
					rw			rw				r		rw	

Bit	Field	Description
31：27	Reserved	保留，必须保持复位值

Bit	Field	Description
26：24	MCO	微控制器时钟输出（Micro Controller Clock Output） 由软件置“1”或清“0” 010：LSI 时钟输出 100：SYSCLK 时钟 8 分频输出 101：HSI4 分频时钟输出 110：HSE 时钟输出 其他：没有时钟输出 注意： 该时钟输出在启动和切换 MCO 时钟源时可能会被停止。 系统时钟通过 MCO 管脚输出时，请保证输出时钟频率不超过 50MHz。
23：11	Reserved	保留，必须保持复位值
10：8	PPRE1	PPRE1：APB1 预分频系数 通过软件设置来控制 APB1 时钟（PCLK1）预分频系数。 0xx：HCLK 不分频 100：HCLK 2 分频 101：HCLK 4 分频 110：HCLK 8 分频 111：HCLK 16 分频
7：4	HPRE	AHB 预分频系数 通过软件设置来控制 AHB 时钟的预分频系数。 0xxx：SYSCLK 不分频 1000：SYSCLK 2 分频 1001：SYSCLK 4 分频 1010：SYSCLK 8 分频 1011：SYSCLK 16 分频 1100：SYSCLK 64 分频 1101：SYSCLK 128 分频 1110：SYSCLK 256 分频 1111：SYSCLK 512 分频 注： 当 AHB 时钟的预分频系数大于 1 时，必须开启预取缓冲器，详见闪存访问控制寄存器章节。
3：2	SWS	系统时钟选择状态（System Clock Switch Status） 00：HSI 6 分频作为系统时钟 01：HSE 作为系统时钟 10：HSI 作为系统时钟 11：LSI 作为系统时钟

Bit	Field	Description
1：0	SW	系统时钟选择（System Clock Switch） 通过软件配置来选择系统时钟源 当从停止模式中返回时，当直接或间接作为系统时钟的 HSE 出现故障时，硬件都会强制选择 HSI6 分频作为系统时钟。 00：HSI 6 分频作为系统时钟 01：HSE 作为系统时钟 10：HSI 作为系统时钟 11：LSI 作为系统时钟

5.3.4 RCC_CIR 时钟中断寄存器

偏移地址：0x08

复位值：0x0000 0000

访问：无等待周期，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.												HSERDYC	HSIRDYC	Res.	LSIRDYC
												w1c	w1c		w1c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				HSERDYIE	HSIRDYIE	Res.	LSIRDYIE	Res.				HSERDYF	HSIRDYF	Res.	LSIRDYF
				rw	rw		rw					r	r		r

Bit	Field	Description
31：20	Reserved	保留，必须保持复位值
19	HSERDYC	清除 HSE 稳定中断（HSE Ready Interrupt Clear） 通过软件置“1”来清除 HSE 稳定中断标志位 HSERDYF。 0：无效 1：清除 HSE 稳定中断标志位 HSERDYF
18	HSIRDYC	清除 HSI 稳定中断（HSI Ready Interrupt Clear） 通过软件置“1”来清除 HSI 稳定中断标志位 HSIRDYF。 0：无效 1：清除 HSI 稳定中断标志位 HSIRDYF
17	Reserved	保留，必须保持复位值
16	LSIRDYC	清除 LSI 稳定中断（LSI Ready Interrupt Clear） 通过软件置“1”来清除 LSI 稳定中断标志位 LSIRDYF。 0：无效 1：清除 LSI 稳定中断标志位 LSIRDYF

Bit	Field	Description
15：12	Reserved	保留，必须保持复位值
11	HSERDYIE	HSE 稳定中断使能（HSE Ready Interrupt Enable） 通过软件置“1”来使能或清“0”来禁止外部振荡器稳定中断。 0：禁止 HSE 稳定中断 1：使能 HSE 稳定中断
10	HSIRDYIE	HSI 稳定中断使能（HSI Ready Interrupt Rnable） 通过软件置“1”来使能或清“0”来禁止内部振荡器稳定中断。 0：禁止 HSI 稳定中断 1：使能 HSI 稳定中断
9	Reserved	保留，必须保持复位值
8	LSIRDYIE	LSI 稳定中断使能（LSI Ready Interrupt Enable） 通过软件置“1”来使能或清“0”来禁止内部 40KHz 振荡器稳定中断。 0：禁止 LSI 稳定中断 1：使能 LSI 稳定中断
7：4	Reserved	保留，必须保持复位值
3	HSERDYF	HSE 稳定中断标志（HSE Ready Interrupt Flag） 在外部高速时钟稳定时，由硬件置“1”。 通过软件将 HSERDYC 位置“1”来清除。 0：无外部振荡器产生的时钟稳定中断 1：外部振荡器导致时钟稳定中断
2	HSIRDYF	HSI 稳定中断标志（HSI Ready Interrupt Flag） 在内部高速时钟稳定时，由硬件置“1”。 通过软件将 HSIRDYC 位置“1”来清除。 0：无内部 HSI 振荡器产生的时钟稳定中断 1：内部 HSI 振荡器导致时钟稳定中断
1	Reserved	保留，必须保持复位值
0	LSIRDYF	LSI 稳定中断标志（LSI Ready Interrupt Flag） 在内部低速时钟稳定时，由硬件置“1”。 通过软件将 LSIRDYC 位置“1”来清除。 0：无内部 40KHz 振荡器产生的时钟稳定中断 1：内部 40KHz 振荡器导致时钟稳定中断

5.3.5 RCC_APB1RSTR APB1 外设复位寄存器

偏移地址：0x10

复位值：0x0000 0000

访问：无等待周期，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.	SYSCFG	DBG	PWR	Res.						I2C1	Res.			USART2	USART1
	rw	rw	rw							rw				rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			SPI1	Res.		ADC1	Res.					TIM14	TIM1	TIM3	Res.
			rw			rw						rw	rw	rw	

Bit	Field	Description
31	Reserved	保留，必须保持复位值
30	SYSCFG	SYSCFG 复位（SYSCFG Reset） 由软件置“1”或清“0” 0：无效 1：复位
29	DBG	DBG 复位（DBG Reset） 由软件置“1”或清“0” 0：无效 1：复位
28	PWR	PWR 复位（Power Interface Reset） 由软件置“1”或清“0” 0：无效 1：复位
27：22	Reserved	保留，必须保持复位值
21	I2C1	I2C1 复位（I2C1 Reset） 由软件置“1”或清“0” 0：无效 1：复位
20：18	Reserved	保留，必须保持复位值
17	USART2	USART2 复位（USART2 Reset） 由软件置“1”或清“0” 0：无效 1：复位
16	USART1	USART1 复位（USART1 Reset） 由软件置“1”或清“0” 0：无效 1：复位
15：13	Reserved	保留，必须保持复位值
12	SPI1	SPI1 复位（SPI1 Reset） 由软件置“1”或清“0” 0：无效 1：复位
11：10	Reserved	保留，必须保持复位值

Bit	Field	Description
9	ADC1	ADC1 复位（ADC1 Reset） 由软件置“1”或清“0” 0：无效 1：复位
8：4	Reserved	保留，必须保持复位值
3	TIM14	TIM14 定时器复位（TIM3 Reset） 由软件置“1”或清“0” 0：无效 1：复位
2	TIM1	TIM1 定时器复位（TIM3 Reset） 由软件置“1”或清“0” 0：无效 1：复位
1	TIM3	TIM3 定时器复位（TIM3 Reset） 由软件置“1”或清“0” 0：无效 1：复位
0	Reserved	保留，必须保持复位值

5.3.6 RCC_AHBENR AHB 外设时钟使能寄存器

偏移地址：0x14

复位值：0x0000 0014

访问：无等待周期，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.													GPIOB rw	GPIOA rw	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									CRC rw	Res.	FLASH rw	Res.	SRAM rw	Res.	

Bit	Field	Description
31：19	Reserved	保留，必须保持复位值
18	GPIOB	GPIOB 时钟使能（GPIOB Clock Enable） 由软件置“1”或清“0” 0：时钟关闭 1：时钟开启

Bit	Field	Description
17	GPIOA	GPIOA 时钟使能（GPIOA Clock Enable） 由软件置“1”或清“0” 0：时钟关闭 1：时钟开启
16：7	Reserved	保留，必须保持复位值
6	CRC	CRC 时钟使能（CRC Clock Enable） 由软件置“1”或清“0” 0：时钟关闭 1：时钟开启
5	Reserved	保留，必须保持复位值
4	Flash	FLASH 时钟使能（FLASH Clock Enable） 由软件置“1”或清“0” 0：时钟关闭 1：时钟开启
3	Reserved	保留，必须保持复位值
2	SRAM	SRAM 时钟使能（SRAM Clock Enable） 由软件置“1”或清“0” 0：时钟关闭 1：时钟开启
1：0	Reserved	保留，必须保持复位值

5.3.7 RCC_APB1ENR APB1 外设时钟使能寄存器

偏移地址：0x1C

复位值：0x0000 0000

访问：无等待周期，字，半字和字节访问

注：当外设时钟没有启动时，软件不能读出外设寄存器的数值

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	SYSCFG	DBG	PWR	Res.						I2C1	Res.			USART2	USART1
	rw	rw	rw							rw				rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			SPI1	Res.		ADC1	Res.					TIM14	TIM1	TIM3	Res.
			rw			rw						rw	rw	rw	

Bit	Field	Description
31	Reserved	保留，必须保持复位值
30	SYSCFG	SYSCFG 时钟使能（SYSCFG Clock Enable） 由软件置“1”或清“0” 0：时钟禁止 1：时钟使能
29	DBG	DBG 时钟使能（DBG Clock Enable） 由软件置“1”或清“0” 0：时钟禁止 1：时钟使能
28	PWR	PWR 时钟使能（Power Clock Enable） 由软件置“1”或清“0” 0：时钟禁止 1：时钟使能
27：22	Reserved	保留，必须保持复位值
21	I2C1	I2C1 时钟使能（I2C1 Clock Enable） 由软件置“1”或清“0” 0：时钟禁止 1：时钟使能
20：18	Reserved	保留，必须保持复位值
17	USART2	USART2 时钟使能（USART2 Clock Enable） 由软件置“1”或清“0” 0：时钟禁止 1：时钟使能
16	USART1	USART1 时钟使能（USART1 Clock Enable） 由软件置“1”或清“0” 0：时钟禁止 1：时钟使能
15：13	Reserved	保留，必须保持复位值
12	SPI1	SPI1 时钟使能（SPI1 Clock Enable） 由软件置“1”或清“0” 0：时钟禁止 1：时钟使能
11：10	Reserved	保留，必须保持复位值
9	ADC1	ADC1 时钟使能（ADC1 Clock Enable） 由软件置“1”或清“0” 0：时钟禁止 1：时钟使能
8：4	Reserved	保留，必须保持复位值

Bit	Field	Description
3	TIM14	TIM14 定时器时钟使能（TIM14 Clock Enable） 由软件置“1”或清“0” 0：时钟禁止 1：时钟使能
2	TIM1	TIM1 定时器时钟使能（TIM1 Clock Enable） 由软件置“1”或清“0” 0：时钟禁止 1：时钟使能
1	TIM3	TIM3 定时器时钟使能（TIM3 Clock Enable） 由软件置“1”或清“0” 0：时钟禁止 1：时钟使能
0	Reserved	保留，必须保持复位值

5.3.8 RCC_CSR 控制状态寄存器

偏移地址：0x24

复位值：0x0800 0000

访问：0-3 等待周期，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
Res.		IWDGRSTF	SFTRSTF	PORRSTF	PINRSTF	Res.	RMVF	LOCKUPF	PVDRSTF	Res.					
		r	r	r	r		w1c	r	r						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								LOCKUPEN	PVDRSTEN	Res.				LSIRDY	LSION
								rw	rw					r	rw

Bit	Field	Description
31：30	Reserved	保留，必须保持复位值
29	IWDGRSTF	独立看门狗复位标志（Independent Watchdog Reset Flag） 在独立看门狗复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0：无独立看门狗复位发生 1：发生独立看门狗复位

Bit	Field	Description
28	SFTRSTF	软件复位标志（Software Reset Flag） 在软件复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0：无软件复位发生 1：发生软件复位
27	PORRSTF	上电/掉电复位标志（POR/PDR Reset Flag） 在上电/掉电复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0：无上电/掉电复位发生 1：发生上电/掉电复位
26	PINRSTF	NRST 管脚复位标志（PIN Reset Flag） 在 NRST 管脚复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0：无 NRST 管脚复位发生 1：发生 NRST 管脚复位
25	Reserved	保留，必须保持复位值
24	RMVF	清除复位标志（Remove Reset Flag） 由软件置“1”来清除复位标志。 0：无效 1：清除复位标志
23	LOCKUPF	CPU 死锁复位标志（CPU Lockup Reset Flag） 在 CPU 发生死锁复位时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0：无 CPU 死锁复位发生 1：发生 CPU 死锁复位
22	PVDRSTF	PVD 复位标志（PVD Reset Flag） 在 PVD 复位发生时由硬件置“1”，且只能由电源复位清除或由软件通过写 RMVF 位清除。 0：无 PVD 复位发生 1：发生 PVD 复位
21：8	Reserved	保留，必须保持复位值
7	LOCKUPEN	CPU 死锁复位使能（CPU Lockup Reset Enable） 0：禁止 CPU 死锁复位 1：使能 CPU 死锁复位
6	PVDRSTEN	PVD 复位使能（PVD Reset Enable） 0：禁止 PVD 产生复位 1：使能 PVD 产生复位
5：2	Reserved	保留，必须保持复位值

Bit	Field	Description
1	LSIRDY	内部低速时钟稳定（Internal Low-speed Oscillator Ready） 由硬件置“1”或清“0”来指示内部 40KHz 振荡器是否稳定。 在 LSION 清“0”后，3 个 AHB 时钟后 LSIRDY 被清“0”。 0：内部 40KHz 振荡器时钟未稳定 1：内部 40KHz 振荡器时钟稳定
0	LSION	内部低速振荡器使能（Internal Low-speed Oscillator Enable） 通过软件置“1”或清“0”，或由电源复位清除。 0：禁止内部 40KHz 振荡器 1：使能内部 40KHz 振荡器

5.3.9 RCC_AHBRSTR AHB 外设复位寄存器

偏移地址：0x28

复位值：0x0000 0000

访问：无等待周期，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.													GPIOB	GPIOA	Res.
													rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															

Bit	Field	Description
31：19	Reserved	保留，必须保持复位值
18	GPIOB	GPIOB 复位（GPIOB Reset） 由软件置“1”或清“0” 0：无效 1：复位
17	GPIOA	GPIOA 复位（GPIOA Reset） 由软件置“1”或清“0” 0：无效 1：复位
16：0	Reserved	保留，必须保持复位值

5.3.10 RCC_SYSCFG 系统配置寄存器

偏移地址：0x40

复位值：0x0000 0003

访问：0-3 等待周期，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.													SFT_NRST_RMP	SECTOR1KCFG	PROG_CHECK_EN
													rw	rw	r

Bit	Field	Description
31：3	Reserved	保留，必须保持复位值
2	SFT_NRST_RMP	SFT_NRST_RMP：软件映射 nRST。 该位只能由电源复位所清除或由软件置 0 清除。 1：将 PA14 映射为 nRST 0：无效果 注：当 RCC_SYSCFG 的 SFT_NRST_RMP 位设为 1 时，将 PA14 映射为 nRST 外部复位，且要求低电平至少 4us。
1	SECTOR1KCFG	SECTOR_1K_CFG：Flash 页擦除时擦除的大小。 1：1K 字节 0：512 字节
0	PROG_CHECK_EN	写 Flash 时是否检查 Flash 内的数据是否是 0xFF 1：检查（硬件固定为 1） 0：不检查

6 GPIO 通用端口

6.1 简介

每个通用 I/O 端口都可以通过两个 32 位的控制寄存器（GPIOx_CRL/GPIOx_CRH）和两个 32 位的复用控制寄存器（GPIOx_AFRL/GPIOx_AFRH）配置为 8 种模式：模拟输入、浮空输入、上拉输入、下拉输入、推挽输出、开漏输出、复用推挽输出和复用开漏输出。

可以自由编程控制每个 I/O 端口，支持字（32 位）、半字（16 位）或字节（8 位）访问所有寄存器。GPIO 寄存器组有 GPIOx_BSRR 和 GPIOx_BRR 位控制寄存器，通过写操作这两个寄存器可以独立的按位控制 GPIOx_ODR 输出 0 或 1。

6.2 主要特征

- 每次 AHB 的写操作，可以更改 GPIOx_ODR 对应的一位或多位
- 所有 I/O 支持编程 EXTI 配置寄存器输出外部触发中断
- 支持配置 GPIO 锁定机制
- 输入支持浮空、上拉、下拉、模拟
- 输出支持推挽与开漏上拉或开漏下拉
- 默认浮空输入，输入输出方向可配
- I/O 输出速度可配

6.3 功能描述

6.3.1 功能框图

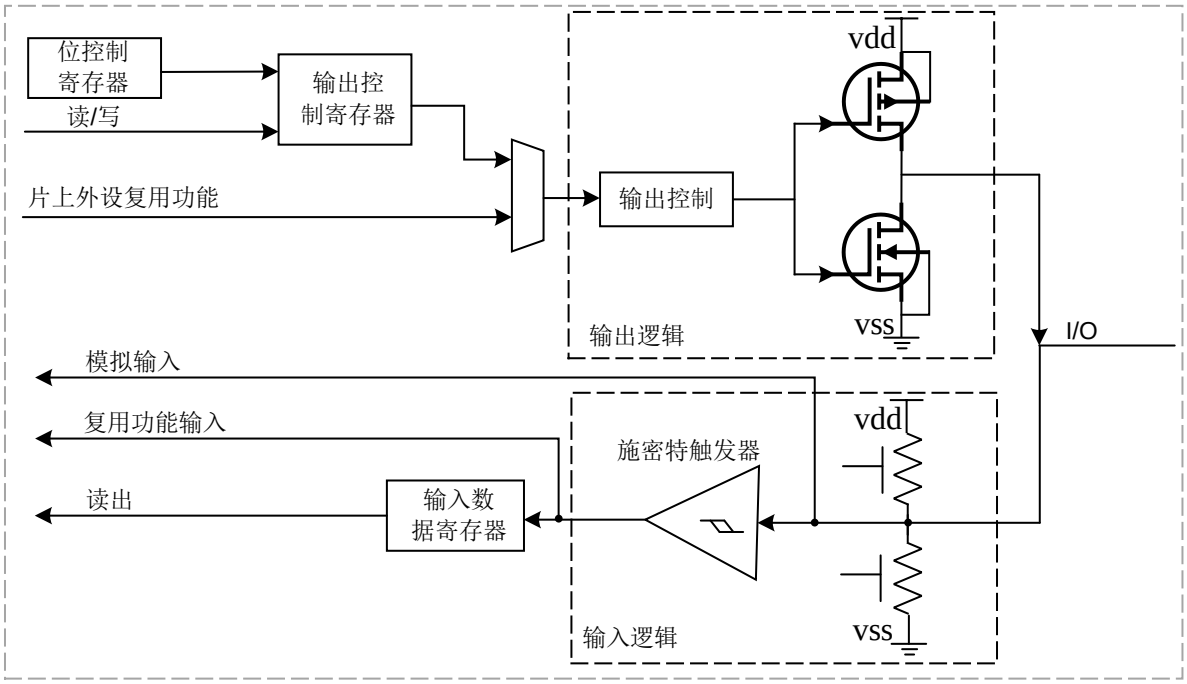

图 6-1 标准 I/O 端口

6.3.2 GPIO 端口配置

表 6-1 端口位配置表（port0 为例）

引脚模式		上下拉	DCR[1:0]		CNF0		MODE0	ODRx
模拟输入		x	x	x	0	0	00	x
通用输入		浮空	x	x	0	1		x
复用输入		上拉	x	x	1	0		1
		下拉	x	x	1	0		0
通用输出	推挽	x	x	x	0	0	01 10 11	0 or 1
	开漏	浮空	x	0	0	1		0 or 1
		上拉	1	1	0	1		0 or 1
		下拉	0	1	0	1		0 or 1
复用输出	推挽	x	x	x	1	0		x
	开漏	浮空	x	0	1	1		x
		上拉	1	1	1	1		x
		下拉	0	1	1	1		x

注：x 表示 I/O 在对应的模式下不用关心，ODR0 代表输出数据寄存器第 0 位。

输入输出参考配置如下：

- 通用输入：

用户只需配置 GPIOx_CRL 中的 CNF0 选择输入模式

- 通用输出：

推挽输出：用户配置 MODE0 选择输出速度，配置 CNF0=00；

开漏输出：用户配置 MODE0 选择输出速度，配置 CNF0=01，如果对 pin 上下拉有要求，需要单独配置 GPIOx_DCR 寄存器，非开漏输出模式，上下拉失效。

- 复用功能：

配置 AFRLx[3:0]与 AFRHx[3:0]寄存器选择复用功能：

推挽复用输出：用户配置 MODE0 选择输出速度，配置 CNF0=10；

推挽开漏输出：用户配置 MODE0 选择输出速度，配置 CNF0=11。

如果输出模式下对 IO 上下拉有要求，需要单独配置 GPIOx_DCR 寄存器，非开漏输出模式，上下拉失效。

在复位期间或复位之后，GPIO 端口被配置成浮空输入模式，串行线调试端口（Serial-Wired Debug pins）默认为为输入 PU/PD 模式。

配置为通用输出模式后，输出数据寄存器（GPIOx_ODR）的值会输出到相应的 I/O 引脚。在每个 AHB 时钟周期，输入数据寄存器（GPIOX_IDR）捕捉 I/O 引脚上的数据。

注：并不是所有芯片都包括 JTAG 和 SWD 调试端口，芯片具体配置可参考芯片数据手册。

- PA14：SWCLK 置于下拉模式
- PA13：SWDIO 置于上拉模式

6.3.3 复用功能

配置复用功能寄存器打开 IO 对应的复用功能。

- 配置 IO 为复用输入功能时，端口选择上拉、下拉或浮空输入。
- 配置 IO 为复用输出功能时，端口选择推挽或开漏输出模式。
- IO 配置为双向复用功能时，端口选择推挽或开漏输出模式，输入变为浮空输入，开漏模式下可配置 GPIOx_DCR 寄存器选择弱上拉或下拉电阻。

当配置端口为复用输出功能时，端口与片上外设输出信号连接。如果仅仅通过软件方式配置 GPIO 引脚为复用输出功能，外设没有被激活，此时输出不确定。

6.3.4 GPIO 锁定机制

GPIO 存在锁定机制，能够保持设定 IO 配置不被改变。当对某一端口执行锁定机制后，在下一次复位之前，不能改变端口对应的配置。锁定键写序列为：

- GPIOx_LCKR[16]='1'+LCKR[15:0]。
- GPIOx_LCKR[16]='0'+LCKR[15:0]。
- GPIOx_LCKR[16]='1'+LCKR[15:0]。

使能 GPIOA 的 PA[0]端口锁定参考配置如下：

- GPIOA->GPIOA_LCKR=0x10001。
- GPIOA->GPIOA_LCKR=0x00001。
- GPIOA->GPIOA_LCKR=0x10001。

当执行完上述三个步骤后，GPIOA_LCKR 寄存器的第 16 位置 1，在下一次软件复位之前，写 GPIOA_LCKR 寄存器无效，GPIOA_LCKR 寄存器的第 16 保持为 1，不会被更改，PA[0]会一直保持锁定之前的配置不变。

当端口被锁定后，只能在软件复位之后才能再次更改端口位的配置，GPIOx_LCKR 寄存器的一个锁定位锁定端口配置寄存器（GPIOx_CRL）与（GPIOx_CRH）中的 4 个位。

注意事项：

以上配置只是锁定了 PA[0]的配置，对于 PA[15:1]以及其它 GPIO 控制寄存器的配置操作依然有效。

6.3.5 输入配置

当 I/O 端口配置为输入时：

- 施密特触发输入使能。
- 输出缓冲被禁用。
- 可以选择浮空、上拉或下拉输入模式。
- I/O 脚上的数据在每个 AHB 时钟被采样到输入数据寄存器。
- 读访问输入数据寄存器可得到 I/O 状态。

下图给出了 I/O 端口的输入配置：

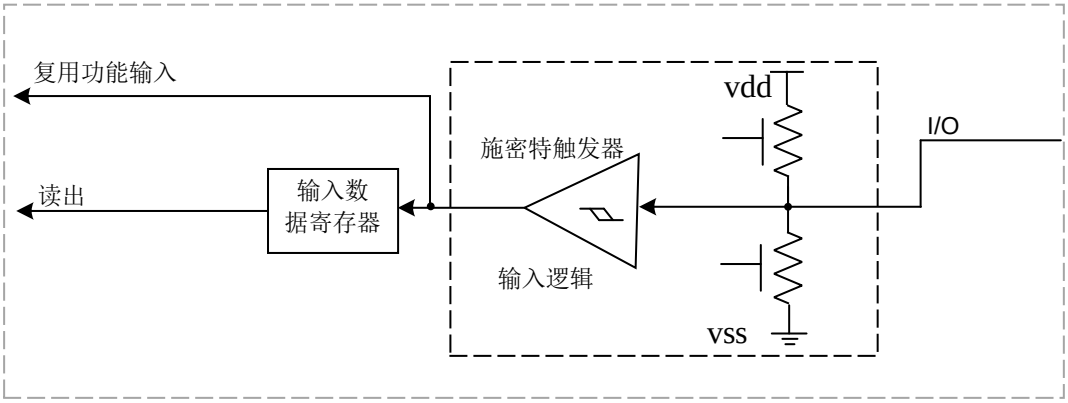

图 6-2 输入浮空/上拉/下拉配置

GPIOA 的 PA[0]端口输入上拉参考配置如下：

- GPIOA->GPIOA_ODR=0x0001。
- GPIOA->GPIOA_CRL=0x00000008。

GPIOA 的 PA[0]端口输入下拉参考配置如下：

- GPIOA->GPIOA_ODR=0x0000。
- GPIOA->GPIOA_CRL=0x00000008。

注意事项：

当端口配置上拉输入时，需要首先配置对应端口的 GPIO_ODR 寄存器对应位输出 1。

当端口配置下拉输入时，需要首先配置对应端口的 GPIO_ODR 寄存器对应位输出 0。

6.3.6 输出配置

当 GPIO 配置为输出时：

- 施密特触发输入使能。
- 输出缓冲使能。
- 通用输出模式下，弱上拉和弱下拉电阻被禁用。
- 开漏模式：端口输出数据寄存器配置为 0 时，对应的引脚输出低电平，端口输出数据寄存器配置为 1 时，对应的管脚处于高阻态。
- 推挽模式：输出寄存器配置为 0 时，对应的引脚输出低电平，输出寄存器配置为 1 时，对应的管脚输出高电平。
- 对端口输出数据寄存器读操作，返回上次写入值。
- 对端口输入数据寄存器进行读操作，获得当前 I/O 的状态。

下图为 I/O 端口的输出配置：

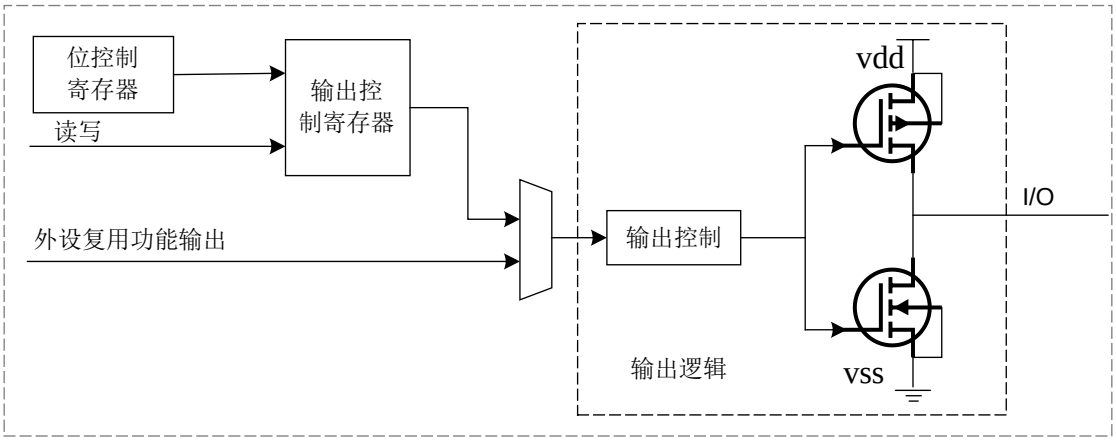

图 6-3 输出配置

6.3.7 复用功能配置

当配置引脚为复用功能时：

- 施密特触发输入使能。
- 输出缓冲器可以配置为开漏或推挽。
- 在开漏输出模式下，通过配置 GPIOx_DCR 寄存器选择弱上拉或下拉电阻。
- 当配置为输入时，可选弱上拉或弱下拉电阻。
- I/O 脚上数据在每个 AHB 时钟周期被采样到输入数据寄存器。

下图为 I/O 端口复用功能的配置，具体见 AFRL 与 AFRH 寄存器与数据手册部分。

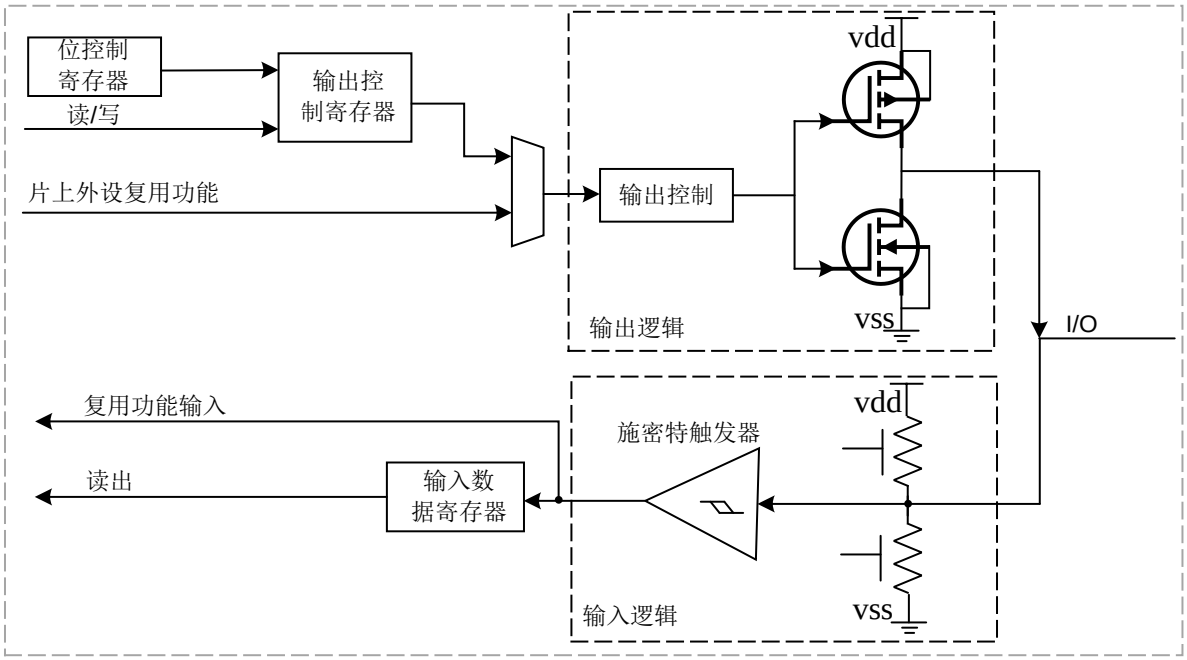

图 6-4 复用功能配置

6.3.8 模拟输入配置

当 I/O 端口被配置成模拟输入配置时：

- 输出缓冲器禁用。
- 施密特触发输入禁用。
- 弱上拉与弱下拉电阻禁用。
- 端口输入数据寄存器保持为 0。

下图为 I/O 端口的模拟输入配置

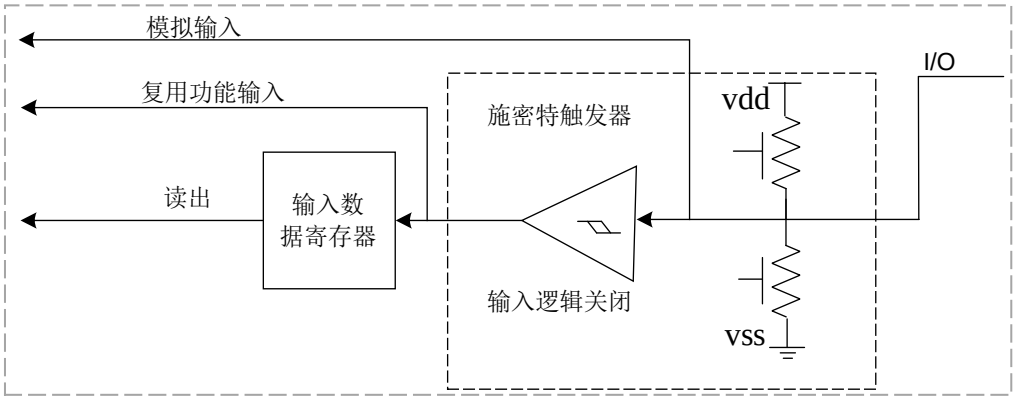

图 6-5 模拟输入

6.3.9 SWD 复用功能重映射

SWD 调试接口信号被映射到 GPIO 端口上，如下表所示：

表 6-2 SWD 复用功能重映射

复用功能	GPIO 端口
SWDIO	PA13
SWCLK	PA14

6.4 寄存器

6.4.1 寄存器总览

表 6-3 GPIO 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	GPIOx_CRL	端口配置低寄存器	见下方描述
0x04	GPIOx_CRH	端口配置高寄存器	见下方描述
0x08	GPIOx_IDR	端口输入数据寄存器	0x0000XXXX
0x0C	GPIOx_ODR	端口输出数据寄存器	0x00000000
0x10	GPIOx_BSRR	端口设置/清除寄存器	0x00000000
0x14	GPIOx_BRR	端口位清除寄存器	0x00000000
0x18	GPIOx_LCKR	端口配置锁定寄存器	0x00000000
0x1C	GPIOx_DCR	端口输出开漏控制寄存器	0x00000000
0x20	GPIOx_AFRL	端口复用功能低位寄存器	见下方描述
0x24	GPIOx_AFRH	端口复用功能高位寄存器	见下方描述

6.4.2 GPIOx_CRL 端口配置低寄存器

偏移地址：0x00

复位值：GPIOA_CRL：0x4444 4444，GPIOB_CRL：0x0000 0044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7		MODE7		CNF6		MODE6		CNF5		MODE5		CNF4		MODE4	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3		MODE3		CNF2		MODE2		CNF1		MODE1		CNF0		MODE0	
rw		rw		rw		rw		rw		rw		rw		rw	

Bit	Field	Description
31:30	CNF7	端口配置位（y=7..0）
27:26	CNF6	配置 MODEy 等于 0，端口为输入模式，此时配置 CNFy 位选择输入模式：
23:22	CNF5	00：模拟输入模式
19:18	CNF4	01：浮空输入模式
15:14	CNF3	10：上拉/下拉输入模式
11:10	CNF2	11：保留
7:6	CNF1	配置 MODEy 不等于 0，端口为输出模式，此时配置 CNFy 位选择输出模式：
3:2	CNF0	00：通用推挽输出模式
29:28	MODE7	01：通用开漏输出模式
25:24	MODE6	10：复用功能推挽输出模式
21:20	MODE5	11：复用功能开漏输出模式
17:16	MODE4	端口输入输出配置（MODEy）（y = 0..7）
13:12	MODE3	软件配置相应的 I/O 端口；参考端口位配置表
9:8	MODE2	配置 MODEy 不等于 0 时，不同配置输出速度不同：
5:4	MODE1	00：输入模式；
1:0	MODE0	其它配置输出速度参考芯片数据手册部分。

6.4.3 GPIOx_CRH 端口配置高寄存器

偏移地址：0x04

复位值：GPIOA_CRH：0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF15		MODE15		CNF14		MODE14		CNF13		MODE13		CNF12		MODE12	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF11		MODE11		CNF10		MODE10		CNF9		MODE9		CNF8		MODE8	
rw		rw		rw		rw		rw		rw		rw		rw	

Bit	Field	Description
31:30	CNF15	端口配置位（y=15..8） 配置 MODEy 等于 0，端口为输入模式，此时配置 CNFy 位选择输入模式： 00：模拟输入模式 01：浮空输入模式 10：上拉/下拉输入模式 11：保留 配置 MODEy 不等于 0，端口为输出模式，此时配置 CNFy 位选择输出模式： 00：通用推挽输出模式 01：通用开漏输出模式 10：复用功能推挽输出模式 11：复用功能开漏输出模式 端口输入输出配置（MODEy）（y = 15..8） 软件配置相应的 I/O 端口；参考端口位配置表 配置 MODEy 不等于 0 时，不同配置输出速度不同： 00：输入模式； 其它配置输出速度参考芯片数据手册部分。
27:26	CNF14	
23:22	CNF13	
19:18	CNF12	
15:14	CNF11	
11:10	CNF10	
7:6	CNF9	
3:2	CNF8	
29:28	MODE15	
25:24	MODE14	
21:20	MODE13	
17:16	MODE12	
13:12	MODE11	
9:8	MODE10	
5:4	MODE9	
1:0	MODE8	

6.4.4 GPIOx_IDR 端口输入数据寄存器

偏移地址：0x08

复位值：0x0000 XXXX

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
IDRy(y=15~0)															
r															

Bit	Field	Description
31:16	Reserved	始终读为 0
15:0	IDRy	端口输入数据（y=15..0） 读出的值代表对应的 I/O 状态

6.4.5 GPIOx_ODR 端口输出数据寄存器

偏移地址：0xC

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
ODRy(y=15~0)															
rw															

Bit	Field	Description
31:16	Reserved	始终读为 0
15:0	ODRy	端口输出数据（y=15..0） 配置为通用输出模式时，写入值输出到对应的 IO 注：操作 GPIOx_BSRR（x=A..F）寄存器可以分别独立的对各个 ODR 位置 1 或清 0。

6.4.6 GPIOx_BSRR 端口设置/清除寄存器

偏移地址：0x10

复位值：0x0000 000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRy(y=15~0)															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BSy(y=15~0)															
w															

Bit	Field	Description
31:16	BRy	端口清除位 y（y=15..0） 写 0 相应的 ODRY 位保持不变 写 1 清除对应的 ODRY 位为 0
15:0	BSy	端口置位 y（y=15..0） 写 0 相应的 ODRY 位保持不变 写 1 置位对应的 ODRY 位为 1 注：同时写 BSy 位与 BRy 位为 1 时，BSy 的优先级高于 BRy

6.4.7 GPIOx_BRR 端口位清除寄存器

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
BRy(y=15~0)															
w															

Bit	Field	Description
31:16	Reserved	始终读为 0
15:0	BRy	端口清除位 y（y=15..0） 写 0 相应的 ODRY 位保持不变 写 1 清除对应的 ODRY 位为 0

6.4.8 GPIOx_LCKR 端口配置锁定寄存器

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.															LCKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCKy(y=15~0)															
rw															

Bit	Field	Description
31:17	Reserved	始终读为 0
16	LCKK	锁键（Lock key） 该位可随时读出,它只可通过锁键写入序列修改。 0：端口配置锁键位未被激活 1：端口配置锁键位被激活,下次软件复位前 GPIOx_LCKR 寄存器被锁住 锁键序列：写 1->写 0->写 1
15:0	LCKy	端口 x 的锁位 y（y = 15..0） 这些位可读可写但只能在 LCKK 位为 0 时写入。 0：不锁定端口的配置 1：锁定端口的配置

6.4.9 GPIOx_DCR 端口输出开漏控制寄存器

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
PX15		PX14		PX13		PX12		PX11		PX10		PX9		PX8	
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PX7		PX6		PX5		PX4		PX3		PX2		PX1		PX0	
rw															

Bit	Field	Description
31:2	PX15-PX1	见 PX0
1:0	PX0	PX0[1：0]： 11：开漏输出模式下，端口上拉 01：开漏输出模式下，端口下拉 x0：开漏输出模式下，端口无上下拉

6.4.10 GPIOx_AFRL 端口复用功能低位寄存器

偏移地址：0x20

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR7				AFR6				AFR5				AFR4			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR3				AFR2				AFR1				AFR0			
rw				rw				rw				rw			

Bit	Field	Description
31:0	AFRy	端口 x 的位 y（y = 0..7）的复用功能选择位，软件写配置。 0000：AF0 0001：AF1 0010：AF2 0011：AF3 0100：AF4 0101：AF5 0110：AF6 0111：AF7 1000：AF8 1001：AF9 1010：AF10 1011：AF11 1100：AF12 1101：AF13 1110：AF14 1111：AF15

6.4.11 GPIOx_AFRH 端口复用功能高位寄存器

偏移地址：0x24

复位值: GPIOA_AFRH: 0xF00F FFFF

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

AFR15				AFR14				AFR13				AFR12			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR11				AFR10				AFR9				AFR8			
rw				rw				rw				rw			

Bit	Field	Description
31:0	AFRy	端口 x 的位 y（y = 8..15）的复用功能选择位，软件写配置。 0000：AF0 0001：AF1 0010：AF2 0011：AF3 0100：AF4 0101：AF5 0110：AF6 0111：AF7 1000：AF8 1001：AF9 1010：AF10 1011：AF11 1100：AF12 1101：AF13 1110：AF14 1111：AF15

7 EXTI 中断和事件

7.1 简介

嵌套向量中断控制器（NVIC）连接处理器核，管理低延迟的异常和中断处理。NVIC 内部包含 2 位的中断优先级配置位，从而可提供 4 个中断优先级等级。EXTI 模块包括边沿检测电路，能够产生中断请求或者唤醒事件，边沿检测支持上升沿、下降沿、任意边沿配置。每一个边沿检测电路支持独立的使能与屏蔽。

7.2 主要特征

- 独立触发与屏蔽每个中断
- 软件配置中断/事件输出
- 产生唤醒事件唤醒低功耗模式
- 挂起寄存器保存对应每条中断线的状态
- 所有 GPIO 支持配置为 EXTI 的触发源
- 支持上升沿触发，下降沿触发和任意边沿触发

7.3 功能描述

7.3.1 功能框图

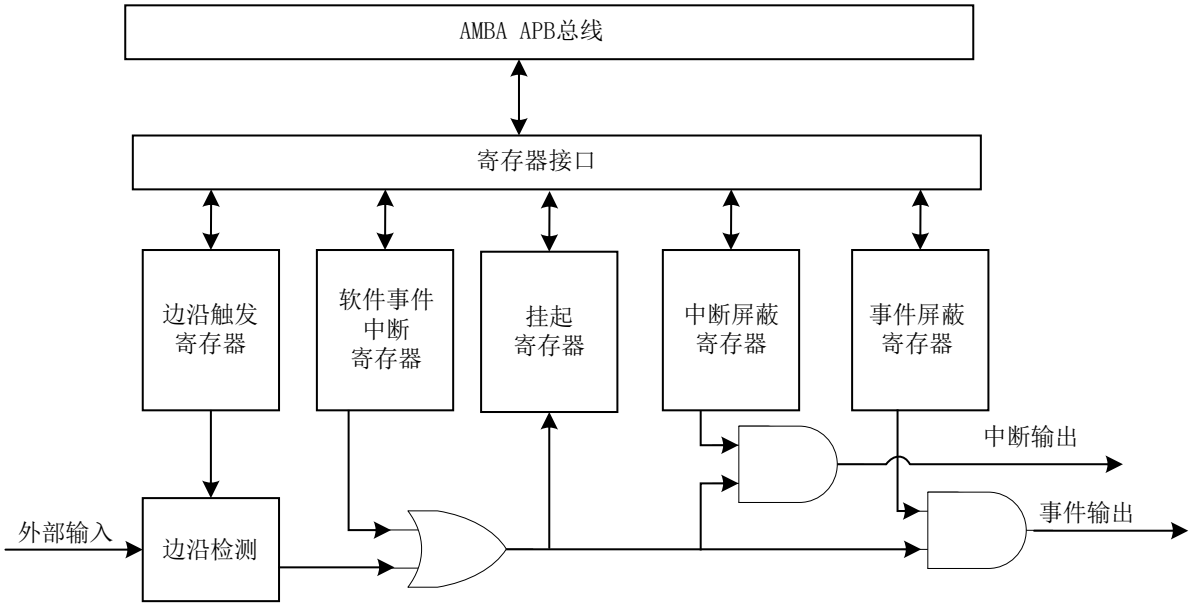

图 7-1 EXTI 结构框图

7.3.2 中断和异常向量

在 Handler 模式下，Cortex-M0 处理器与内嵌中断向量控制（NVIC）对所有的异常进行优先级区分处理。当异常发生时，系统会将当前处理的工作压栈，执行完中断服务程序后出栈。取向量与当前工作的压栈并行进行的，提高了中断的效率，下表分别列出了异常类型与中断向量。

表 7-1 异常向量表

位置	优先级	优先级类型	名称	说明	地址
				保留	0x0000 0000
	-3	固定	Reset	复位	0x0000 0004
	-2	固定	NMI	不可屏蔽中断 RCC 时钟安全系统(CSS)连接到 NMI	0x0000 0008
	-1	固定	硬件失效 (HardFault)	所有类型的失效	0x0000 000C

表 7-2 中断向量表

位置	优先级	优先级类型	名称	说明	地址
	3	可设置	SVCall	通过 SWI 指令的系统服务调用	0x0000 002C
		-		保留	0x0000 0030
				保留	0x0000 0034
	5	可设置	PendSV	可挂起的系统服务	0x0000 0038

位置	优先级	优先级类型	名称	说明	地址
	6	可设置	SysTick	系统嘀嗒定时器	0x0000 003C
0	7	可设置	IWDG	连到 EXTI17 的独立看门狗中断	0x0000_0040
1	8	可设置	PVD	连到 EXTI16 的电源电压检测(PVD)中断	0x0000_0044
2	9	可设置	保留	保留	0x0000_0048
3	10	可设置	Flash	闪存全局中断	0x0000_004C
4	11	可设置	RCC	RCC 全局中断	0x0000_0050
5	12	可设置	EXTI0_1	EXTI 线[1：0]中断	0x0000_0054
6	13	可设置	EXTI2_3	EXTI 线[3：2]中断	0x0000_0058
7	14	可设置	EXTI4_15	EXTI 线[15：4]中断	0x0000_005C
8	15	可设置	保留	保留	0x0000_0060
9	16	可设置	保留	保留	0x0000_0064
10	17	可设置	保留	保留	0x0000_0068
11	18	可设置	保留	保留	0x0000_006C
12	19	可设置	ADC	ADC 全局中断	0x0000_0070
13	20	可设置	TIM1_BRK_UP_TRG_COM	TIM1 刹车、更新、触发、COM 中断	0x0000_0074
14	21	可设置	TIM1_CC	TIM1 捕捉比较中断	0x0000_0078
15	22	可设置	保留	保留	0x0000_007C
16	23	可设置	TIM3	TIM3 全局中断	0x0000_0080
17	24	可设置	保留	保留	0x0000_0084
18	25	可设置	保留	保留	0x0000_0088
19	26	可设置	TIM14	TIM14 全局中断	0x0000_008C
20	27	可设置	保留	保留	0x0000_0090
21	28	可设置	保留	保留	0x0000_0094
22	29	可设置	保留	保留	0x0000_0098
23	30	可设置	I2C1	I2C1 全局中断	0x0000_009C
24	31	可设置	保留	保留	0x0000_00A0
25	32	可设置	SPI1	SPI1 全局中断	0x0000_00A4
26	33	可设置	保留	保留	0x0000_00A8
27	34	可设置	USART1	USART1 全局中断	0x0000_00AC
28	35	可设置	USART2	USART2 全局中断	0x0000_00B0
29	36	可设置	保留	保留	0x0000_00B4
30	37	可设置	保留	保留	0x0000_00B8
31	38	可设置	保留	保留	0x0000_00BC

7.3.3 唤醒事件管理

EXTI 模块支持产生中断或者事件用于将系统从低功耗模式下唤醒，用户执行 WFE 指令进入相应的低功耗模式后，可以通过配置 EXTI 线产生事件输出唤醒系统，用户执行 WFI 进入低功耗模式后，可以通过配置 EXTI 线产生中断输出唤醒系统，具体详细配置参考电源控制章节。

7.3.4 中断功能描述

要使能中断功能，产生中断，首先配置边沿检测触发寄存器为需要的触发类型，打开相应的中断屏蔽寄存器的对应位允许中断请求。在对应的外部中断线检测到配置的触发条件时，产生一个中断请求，挂起寄存器对应位置 1，通过对挂起寄存器对应位写 1，将清除中断。

配置产生事件，首先配置边沿检测触发寄存器为需要的触发类型，打开相应的事件屏蔽寄存器的对应位允许事件请求。在对应的外部中断线检测到配置的触发条件时，产生一个事件请求。

使能软件中断/事件寄存器的对应位，也能够产生中断/事件请求。

7.3.5 硬件中断输出

配置硬件中断源的具体步骤如下：

- 打开对应中断线的屏蔽位（EXTI_IMR），使能中断。
- 配置对应中断线的触发寄存器位（EXTI_RTSR/EXTI_FTSR）。
- 打开对应连接到 NVIC 的中断通道，使得中断请求能够传递到 CPU，被正确的响应。

当配置 EXTIx（x=31~0）线产生中断输出后，EXTI_PR 寄存器的对应位会置 1，需要清除 EXTI_PR 寄存器的对应挂起位才能再次检测 EXTIx（x=31~0）线的翻转并产生中断。

清除 EXTI_PR 寄存器挂起位有以下三种方式：

- EXTI_PR 寄存器的挂起位写 1。
- 如果配置了上升沿触发选择寄存器（EXTI_RTSR），对应位写 0 会清除挂起位。如果配置了下降沿触发选择寄存器（EXTI_FTSR），对应位写 0 会清除挂起位。
- 通过改变 EXTI 线的边沿检测极性清除。

7.3.6 硬件事件输出

配置硬件事件源的具体步骤如下：

- 打开对应事件线的屏蔽位（EXTI_EMR）。
- 配置对应事件线的触发寄存器位（EXTI_RTSR/EXTI_FTSR）。

7.3.7 软件中断与事件输出

支持通过软件的方式配置产生中断与事件，具体步骤如下：

- 使能事件或中断使能位（EXTI_IMR，EXTI_EMR）。
- 配置软件中断事件寄存器对应位为 1（EXTI_SWIER）。

7.3.8 外部中断映射

所有的 GPIO 均可用做 EXTI 的触发源用于产生中断或事件请求，通过配置 SYSCFG 章节的 SYSCFG_EXTICRx 寄存器，同时支持内部模块（包括 PVD 和 IWDG）触发。具体存在的连接关系如下表所示。

表 7-3 EXTI 触发源

外部中断线	IO 映射	控制位
EXTI0	PA0;PB0	SYSCFG_EXTICR1 寄存器中的 EXTI0
EXTI1	PA1;PB1	SYSCFG_EXTICR1 寄存器中的 EXTI1
EXTI2	PA2	SYSCFG_EXTICR1 寄存器中的 EXTI2
EXTI3	PA3	SYSCFG_EXTICR1 寄存器中的 EXTI3
EXTI4	PA4	SYSCFG_EXTICR2 寄存器中的 EXTI4
EXTI5	PA5	SYSCFG_EXTICR2 寄存器中的 EXTI5
EXTI6	PA6	SYSCFG_EXTICR2 寄存器中的 EXTI6
EXTI7	PA7	SYSCFG_EXTICR2 寄存器中的 EXTI7
EXTI8	PA8	SYSCFG_EXTICR3 寄存器中的 EXTI8
EXTI9	PA9	SYSCFG_EXTICR3 寄存器中的 EXTI9
EXTI10	PA10	SYSCFG_EXTICR3 寄存器中的 EXTI10
EXTI11	PA11	SYSCFG_EXTICR3 寄存器中的 EXTI11
EXTI12	PA12	SYSCFG_EXTICR4 寄存器中的 EXTI12
EXTI13	PA13	SYSCFG_EXTICR4 寄存器中的 EXTI13
EXTI14	PA14	SYSCFG_EXTICR4 寄存器中的 EXTI14
EXTI15	PA15	SYSCFG_EXTICR4 寄存器中的 EXTI15

其他的外部中断/事件控制器的连接如下：

EXTI 线 16 连接到 PVD 输出

EXTI 线 17 连接到 IWDG 输出

7.4 寄存器

7.4.1 寄存器总览

表 7-4 EXTI 寄存器总览

Offset	Acronym	Register Name	Reset
0x00	EXTI_IMR	中断屏蔽寄存器	0x00000000
0x04	EXTI_EMR	事件屏蔽寄存器	0x00000000
0x08	EXTI_RTSR	上升沿触发选择寄存器	0x00000000
0x0C	EXTI_FTSR	下降沿触发选择寄存器	0x00000000
0x10	EXTI_SWIER	软件中断事件寄存器	0x00000000
0x14	EXTI_PR	挂起寄存器	0x00000000

7.4.2 中断屏蔽寄存器 (EXTI_IMR)

偏移地址：0x0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.														IMR17	IMR16
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:0	IMRx	线 x 中断使能位 1：配置该位为 1，使能线 x 对应的中断 0：配置该位为 0，禁止线 x 对应的中断

7.4.3 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.														EMR17	EMR16
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMR15	EMR14	EMR13	EMR12	EMR11	EMR10	EMR9	EMR8	EMR7	EMR6	EMR5	EMR4	EMR3	EMR2	EMR1	EMR0

rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:0	IMRx	线 x 事件使能位 1：配置该位为 1，使能线 x 对应的事件 0：配置该位为 0，禁止线 x 对应的事件

7.4.4 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.														TR17	TR16
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:0	TRx	线 x 对应中断或事件的触发极性 1：配置该位为 1，使能线 x 对应的上升沿触发中断或事件 0：配置该位为 0，禁止线 x 对应的上升沿触发中断或事件

7.4.5 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.														TR17	TR16
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:0	TRx	线 x 对应中断或事件的触发极性 1：配置该位为 1，使能线 x 对应的下降沿触发中断或事件 0：配置该位为 0，禁止线 x 对应的下降沿触发中断或事件

7.4.6 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.														SWIER 17	SWIER 16
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIER 15	SWIER 14	SWIER 13	SWIER 12	SWIER 11	SWIER 10	SWIER 9	SWIER 8	SWIER 7	SWIER 6	SWIER 5	SWIER 4	SWIER 3	SWIER 2	SWIER 1	SWIER 0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:0	SWIERx	线 x 上的软件配置中断或事件使能 写 1 将设置 EXTI_PR 寄存器中相应的挂起位，同时配置 EXTI_IMR 或 EXTI_EMR 中对应位为 1，能够产生中断或事件。
		注：向 EXTI PR 寄存器的对应位写 1，可以清除该位

7.4.7 软件中断事件挂起寄存器 (EXTI_PR)

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.														PR17	PR16
														rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bit	Field	Description
31:0	PRx	线 x 触发挂起位 1：发生了选择的触发请求 0：没有发生触发请求 外部中断线上出现选择的边沿事件时，该位被置 1，写 1 清除该位，也可以通过改变边沿检测的极性清除。

8 ADC 模拟/数字转换

8.1 简介

12 位 ADC 是逐次逼近式的模拟数字转换器（SAR A/D 转换器）。

A/D 转换器支持多种工作模式：单次转换和连续转换模式，并且可以选择通道自动扫描。A/D 转换的启动方式有软件设定、外部引脚触发以及各个定时器启动。窗口比较器 (模拟看门狗) 允许应用程序检测输入电压是否超出了用户设定的高/低阈值值。ADC 的输入时钟不得超过 16MHz，它是由 PCLK1 经分频产生。

8.2 主要特征

- 最高 12 位可编程分辨率的 SAR ADC，多达 8 路外部输入通道和 1 路内部通道
- 高达 1Msps 转换速率
- 支持普通工作模式：
  - 单次转换模式： A/D 转换在指定通道完成一次转换
  - 单周期扫描模式： A/D 转换在所有指定通道 (从低序号通道到高序号通道，或从高序号通道到低序号通道) 完成一个周期转换
  - 连续扫描模式： A/D 转换连续执行单周期扫描模式直到软件停止 A/D 转换。若需中途修改转换通道只能停止 A/D 转换，配置完相关寄存器再重新开启转换。
- 支持任意通道工作模式：
  - 单次转换模式：在指定通道完成一次转换
  - 单周期扫描模式：A/D 转换在所有指定通道 (可按照任意顺序) 完成一个周期转换
  - 连续扫描模式：A/D 转换连续执行单周期扫描模式直到软件停止 A/D 转换。若想修改转换通道，不必停止转换，可配置相应寄存器，在下一个扫描周期开始将进行新的通道转换。
- 通道采样时间、分辨率可软件配置
- A/D 转换开始条件
  - 软件启动
  - 外部触发启动，且软件可配置外部触发延时
  - Timer1/3 匹配或 TRGO 信号，外部 EXTI 信号源
- 模拟看门狗功能。转换结果可和指定的值相比较，当转换值和设定值相匹配时，用户可设定

是否产生中断请求

8.3 系统框图

ADC 系统框图如下：

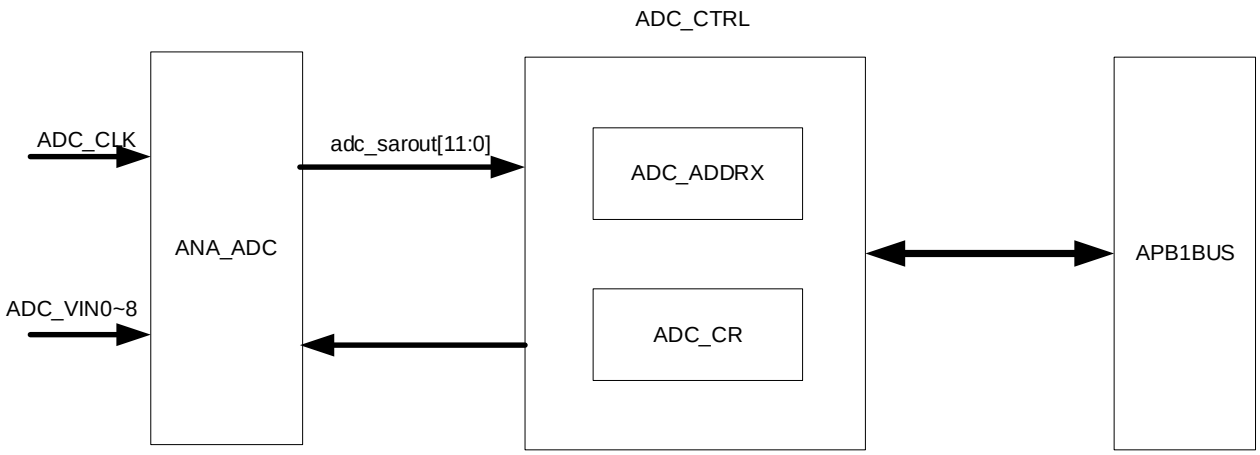

图 8-1 ADC 系统框图

8.4 ADC 功能描述

下图显示了 ADC 框图

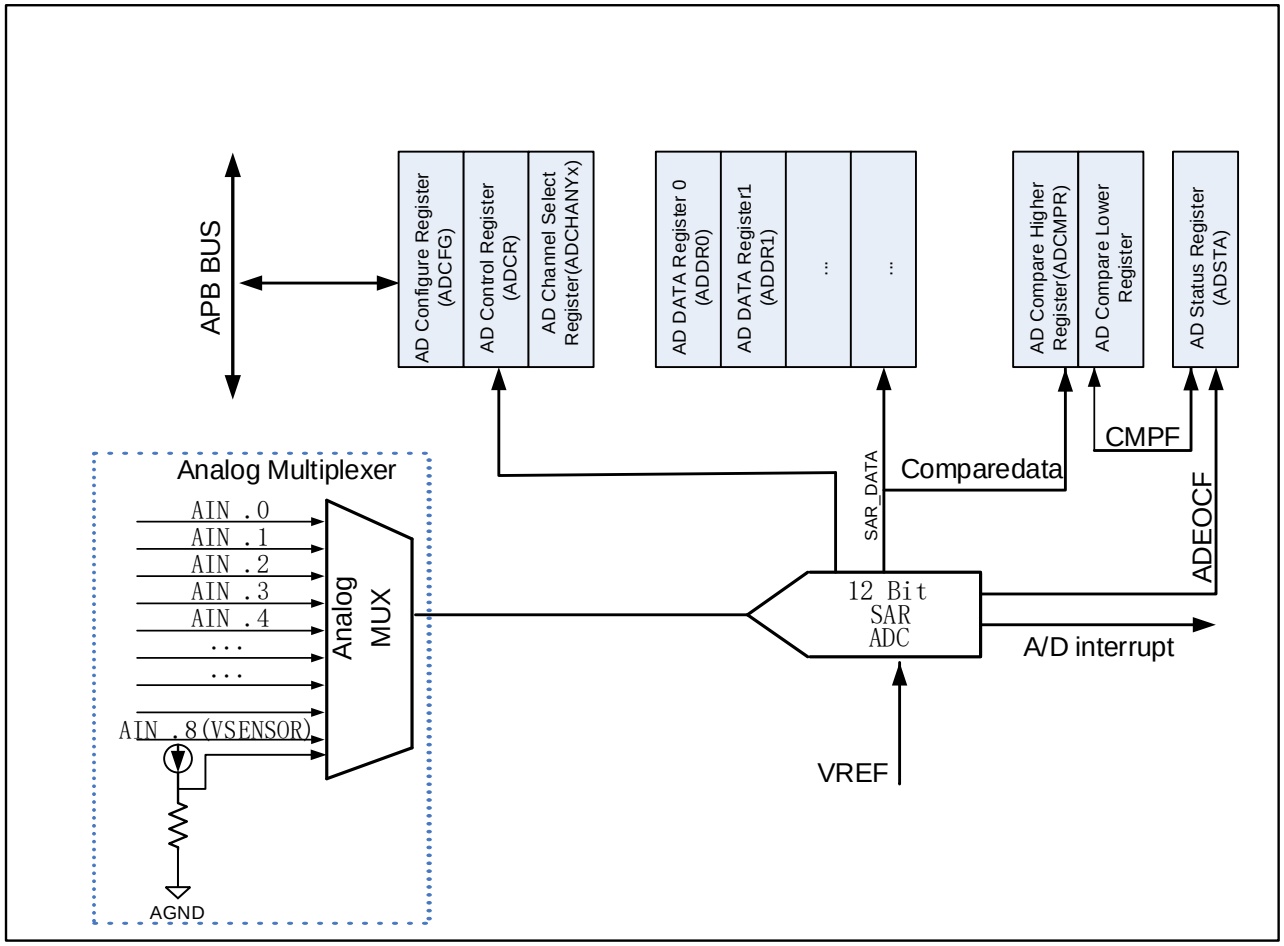

图 8-2 ADC 框图

注: V_SENSOR(内部参考电压) 通道在 ADC 的 AIN8 通道。

8.4.1 ADC 开关控制

通过设置 ADCFG 寄存器的 ADEN 位可给 ADC 上电。当第一次设置 ADEN 位时，它将 ADC 从断电状态下唤醒。

ADC 上电延迟一段时间后(大约 200ns)，设置 ADCR 寄存器的 ADST 位开始进行转换。通过清除 ADST 位可以停止转换，清除 ADEN 位可置于断电模式。

8.4.2 通道选择

ADC1 有 8 路外部输入通道和内部 1.2V 参考电压通道。每个外部输入通道都有独立的使能位，可通过设置 CHANY_NUM，ADC_CHANY0、 ADC_CHANY1 来设置。

8.5 任意通道工作模式

8.5.1 单次转换模式

在单次转换模式下，A/D 转换相应通道上只执行一次，具体流程如下：

软件设置寄存器 ADC_ANY_CFG，ADC_CHANY0，ADC_CHANY1，设置转换通道，置位 CHANY_MDEN。(单次转换模式，只需设置 CHANY_SEL0)

通过软件、外部触发输入及定时器溢出置位 ADCR 寄存器的 ADST，开始 A/D 转换。

A/D 转换完成时，A/D 转换的数据值将存储于数据寄存器 ADDATA 和 ADDRn 中。

A/D 转换完成时，状态寄存器 ADSTA 的 ADIF 位置 1。若此时控制寄存器 ADCR 的 ADIE 位置 1，将产生 AD 转换结束中断请求。

A/D 转换期间，ADST 位保持为 1。A/D 转换结束时，ADST 位自动清 0，A/D 转换器进入空闲模式。

若在 A/D 转换过程中，软件更新 ADC_ANY_CFG， ADC_CHANY0， ADC_CHANY1，硬件不会立即更新这些配置，只会在当前设置的通道都转换结束时更新，然后等待下一次软件置位 ADST。

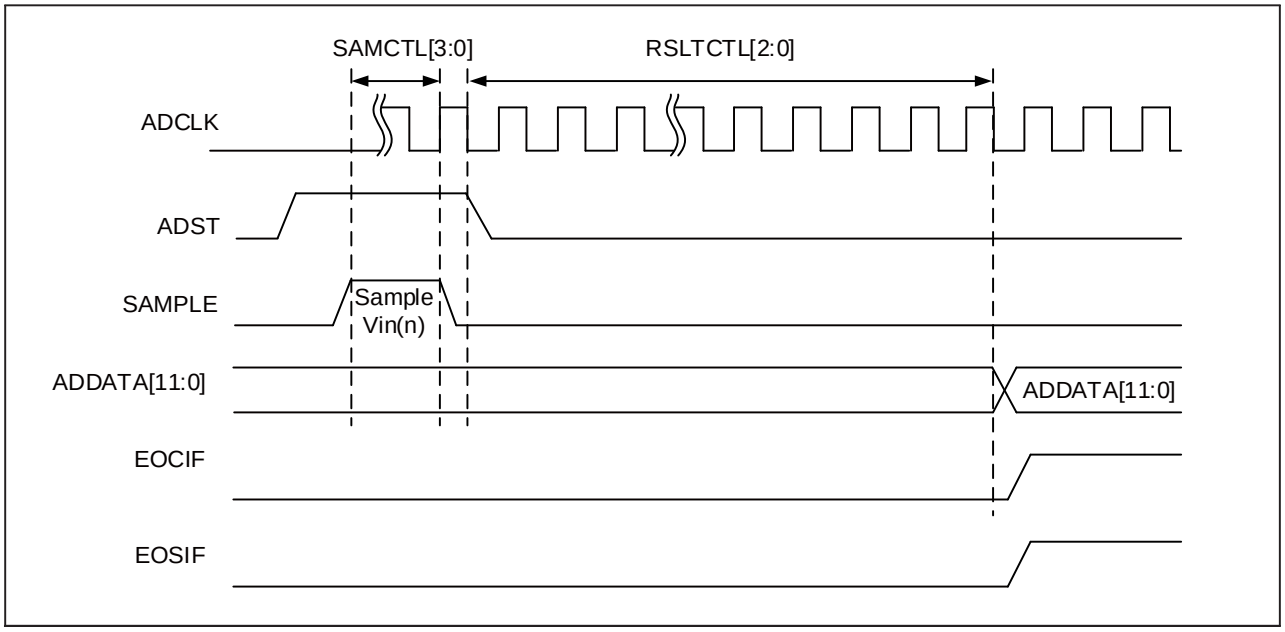

图 8-3 单次转换模式时序图

8.5.2 单周期扫描模式

在单周期扫描模式下，A/D 转换通道依软件配置执行一遍，具体流程如下：

软件设置寄存器 ADC_ANY_CFG，ADC_CHANY0，ADC_CHANY1，将需要转换的通道、数量设

置好，然后置位 CHANY_MDEN。

通过软件、外部触发置位 ADCR 寄存器的 ADST，外部触发可软件配置触发延时，A/D 转换方向从 CHANY_SEL0 到 CHANY_SEL8 转换通道数量由 CHANY_NUM 配置， 且 CHANY_SEL0 到 CHANY_SEL8 是任意配置的，可以完全相同，或完全不相同。

每路 A/D 转换完成时，A/D 转换的数据值将有序装载到相应通道的数据寄存器中，ADIF 转换结束标志被设置，若此时控制寄存器 ADCR 的 ADIE 位置 1，将产生 AD 转换结束中断请求。

A/D 转换结束后，ADST 位自动清 0，A/D 转换器进入空闲模式。

若在 A/D 转换过程中，软件更新 ADC_ANY_CFG， ADC_CHANY0， ADC_CHANY1，硬件不会立即更新这些配置，只会在当前设置的通道都转换结束时更新，然后等待下一次软件软件置位 ADST。

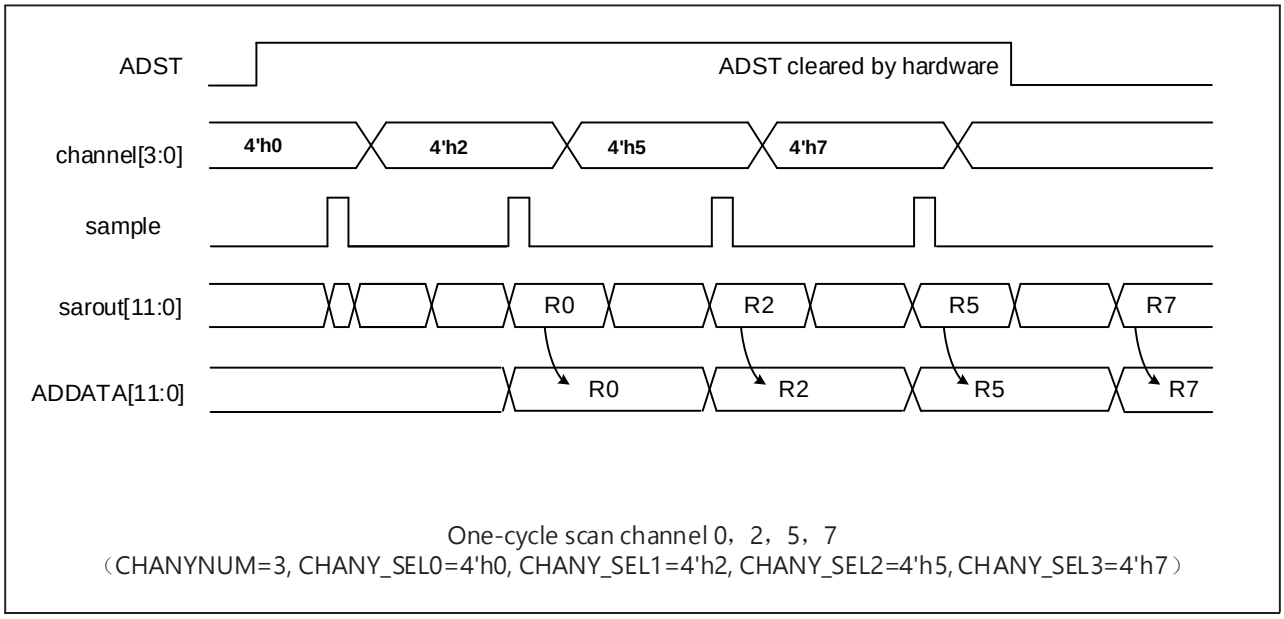

图 8-4 单周期扫描下通道转换时序图

8.5.3 连续扫描模式

在连续扫描模式下，A/D 转换通道依软件配置一直执行，直到软件禁止。具体流程如下：

软件设置寄存器 ADC_ANY_CFG，ADC_CHANY0，ADC_CHANY1，将需要转换的通道、数量设置好，然后置位 CHANY_MDEN

通过软件、外部触发置位 ADCR 寄存器的 ADST，外部触发可软件配置触发延时，A/D 转换方向从 CHANY_SEL0 到 CHANY_SEL8 转换通道数量由 CHANY_NUM 配置，且 CHANY_SEL0 到 CHANY_SEL8 是任意配置的，可以完全相同，或完全不相同。

每路 A/D 转换完成时，A/D 转换的数据值将有序装载到相应通道的数据寄存器中，ADIF 转换结束标志被设置，若此时控制寄存器 ADCR 的 ADIE 位置 1，将产生 AD 转换结束中断请求。

只要 ADST 位保持为 1，持续进行 A/D 转换。当 ADST 位被清 0，当前 A/D 转换完成后停止，A/D 转换器进入空闲状态。

若在 A/D 转换过程中，软件更新 ADC_ANY_CFG，ADC_CHANY0，ADC_CHANY1，硬件不会立即更新这些配置，只会在当前设置的通道都转换结束时更新，即下一个扫描周期开始新的通道转换。

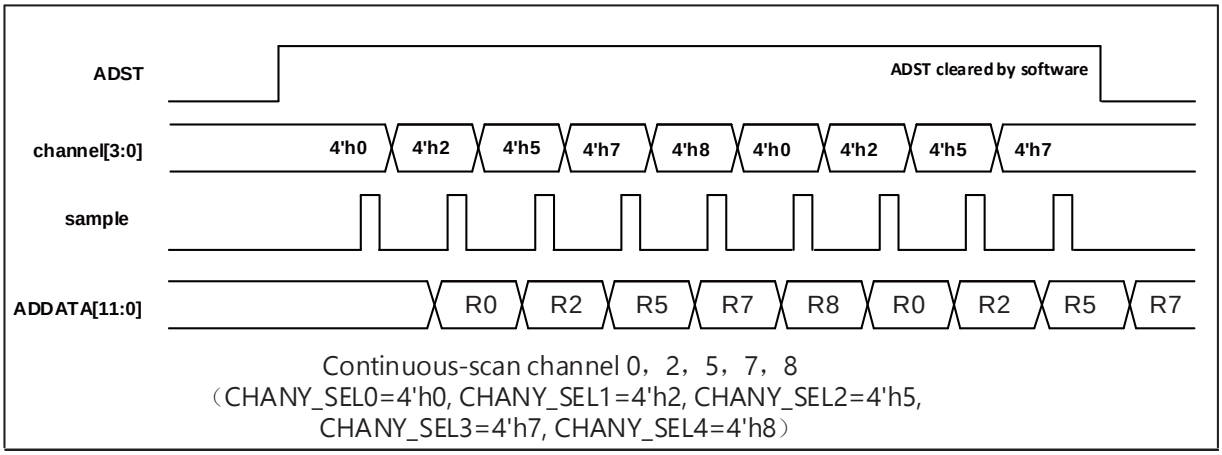

图 8-5 连续扫描模式通道转换时序图

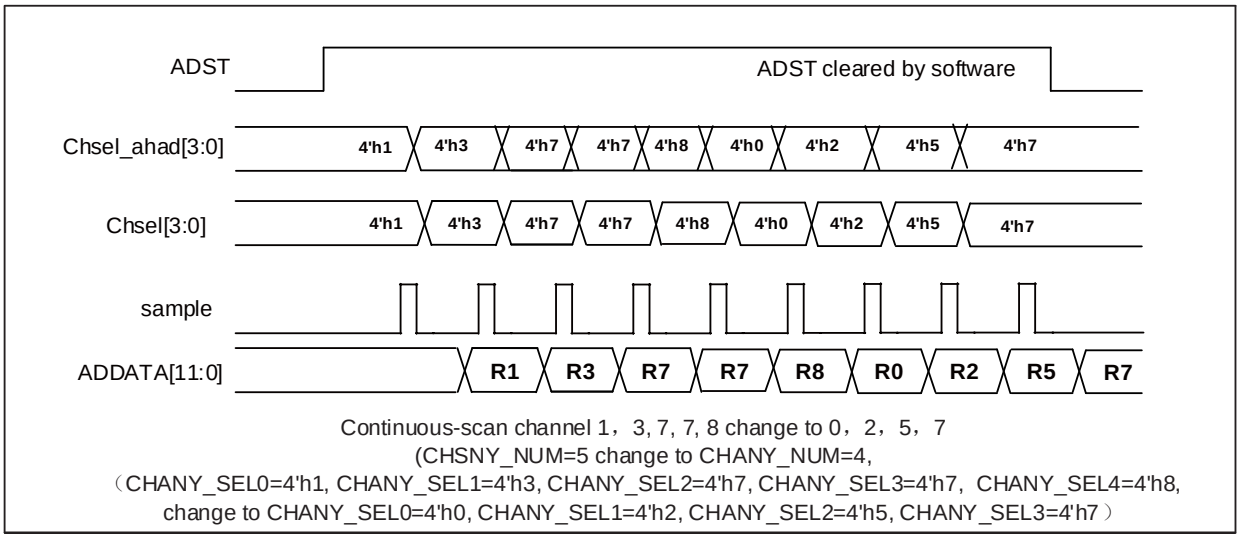

图 8-6 连续扫描模式动态更新配置时序图

8.6 数据对齐

ADCR 寄存器中的 ALIGN 位选择转换后数据储存的对齐方式。数据可以左对齐或右对齐， 如下图所示。

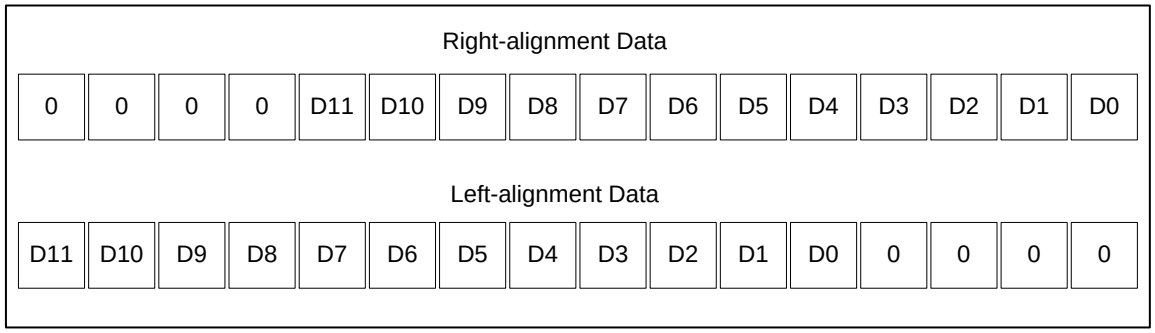

图 8-7 数据对其方式

8.7 可编程分辨率

ADC 转换有效位数可通过 ADC_CFG 寄存器中的 RSLTCTL[2：0] 位更改，以便加快数据转换速率，有效数据位是在 12 位数据高位对齐。

8.8 可编程采样时间

ADC 的时钟 ADCLK 由 PCLK1 分频得到，分频系数可通过设置 ADCFG 寄存器的 ADCPRE 位来确定，即 PCLK1 / (N + 2) 分频后作为 ADC 时钟。ADC 使用若干个 ADC_CLK 周期对输入电压采样，采样周期数目可以通过 ADC_CFG 寄存器中的 SAMCTL[3:0] 位更改。

设置 ADC 分辨率为 n 位 (n=8,9,10,11,12)，每个通道采样周期为 m。采样频率采样时间计算如下：

Fsample = FADCLK / (m + n + 0.5) 。

假设分辨率配置为 12Bit，每个通道采样周期为 3.5T，则 Fsample = FADCLK / 16。总转换时间如下计算：

TCONV = 采样时间 + 12.5 个转换周期

例如：

当 ADCCLK = 16MHz，采样时间为 3.5 周期

TCONV = 3.5 + 12.5 = 16 ，周期 = 1μs

8.9 外部触发转换

ADC 转换可以由外部事件触发 (例如定时器捕获，EXTI 线)。

在触发信号产生后，延时 N 个 PCLK1 的时钟周期再开始采样。如果是触发扫描模式，只有第一个通道采样被延时，其余通道是在上一个采样结束后立即开始。

如果设置了 ADCR 寄存器的 TRGEN 位，就可以使用外部事件触发转换。通过设置 TRGSEL 位可以选择外部触发源。

具体的外部触发源选择情况，可以参考 AD 控制寄存器（ADCR.TRGSEL）相关位的描述。

外部触发可设置延时控制，具体参考 AD 控制寄存器 (ADCR.TRGSHIFT) 相关位的描述。

8.10 内部基准参考电压

ADC 的内部信号源通道连接了一个内部基准参考电压，大小为 1.2V，此通道把 1.2v 的参考电压输出转换为数字值。

内部参考电压有单独的始能位，可通过设置寄存器的相应位开启或关闭。

8.11 窗口比较器模式下 AD 转换结果监控

比较模式下提供了上限和下限两个比较寄存器。可通过软件设定 CMPCH 位选择监控通道。当 CPMHDATA ≥ CPMLDATA 时，比较结果大于或等于 ADCMPR 寄存器的 CMPHDATA 指定值或者小于 CMPLDATA 指定值，状态寄存器 ADSTA 的 ADWIF 位置 1。

当 CPMHDATA < CPMLDATA 时，比较结果如果等于 CPMHDATA 指定值或者处于两个指定值之间，则状态寄存器 ADSTA 的 ADWIF 位置 1。

如果控制寄存器 ADCR 的 ADWIE 置位，将产生中断请求。

8.12 寄存器

8.12.1 寄存器总览

表 8-1 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	ADC_ADDATA	A/D 数据寄存器	0x00000000
0x04	ADC_ADCFG	A/D 配置寄存器	0x00000000
0x08	ADC_ADCR	A/D 控制寄存器	0x00000000
0x10	ADC_ADCMPR	A/D 模拟看门狗比较寄存器	0x00000000
0x14	ADC_ADSTA	A/D 状态寄存器	0x00000000
0x18~0x38	ADC_ADDR 0~8	A/D 通道数据寄存器	0x00000000
0x5C	ADC_CHANY0	A/D 任意通道通道选择寄存器 0	0x00000000
0x60	ADC_CHANY1	A/D 任意通道通道选择寄存器 1	0x00000000
0x64	ADC_ANY_CFG	A/D 任意通道配置寄存器	0x00000000
0x68	ADC_ANY_CR	A/D 任意通道控制寄存器	0x00000000

8.12.2 A/D 数据寄存器 (ADC_ADDATA)

地址偏移：0x00

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										VAILD	OVERRUN	CHANNELSEL			
										r	r	r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DATA
r

Bit	Field	Description
31:22	Reserved	保留，必须保持复位值。
21	VALID	数据有效标志位（只读）（Valid Flag） 1： DATA[11:0] 位数据有效 0： DATA[11:0] 位数据无效 相应模拟通道转换完成后，将该位置位，读 ADDATA 寄存器后，该位由硬件清除。
20	OVERRUN	数据覆盖标志位（只读）（Overrun Flag） 1： DATA[11:0] 数据被覆盖 0： DATA[11:0] 数据最近一次转换结果 新的转换结果装载至寄存器之前，若 DATA[11： 0] 的数据没有被读取，OVERRUN 将置 1。读 ADDATA 寄存器后，该位由硬件清除。
19:16	CHANNELSEL	该 4 位显示当前数据所对应的通道 (Channel selection) 0000 ： 通道 0 的转换数据 0001 ： 通道 1 的转换数据 0010 ： 通道 2 的转换数据 0011 ： 通道 3 的转换数据 0100 ： 通道 4 的转换数据 0101 ： 通道 5 的转换数据 0110 ： 通道 6 的转换数据 0111 ： 通道 7 的转换数据 1000 ： 内部参考电压的转换数据 其他 ：无效
15:0	DATA	12 位 A/D 转换结果（Transfer Data） 根据设置左对齐或者右对齐

8.12.3 A/D 配置寄存器 (ADC_ADCFG)

地址偏移：0x04

复位值：0x00000000

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.	ADCPREL	SAMCTL				RSLTCTL			ADCPREH			VSEN	Res.	ADWEN	ADEN
	rw	rw				rw			rw			rw		rw	rw

Bit	Field	Description
31:15	Reserved	保留，必须保持复位值。
14	ADCPREL	ADC 时钟预分频低位（ADC Prescaler Low Bits） 预分频系数 ADCPRE={ADCPREH, ADCPREL}
13: 10	SAMCTL	选择通道 x 的采样时间 (Channel x Sample time selection) 这些位用于独立地选择每个通道的采样时间。在采样周期 中通道选择位必须保持不变。 0000：2.5 周期　0100：42.5 周期 0001：8.5 周期　0101：56.5 周期 0010：14.5 周期　0110：72.5 周期 0011：29.5 周期　0111：240.5 周期 1000：3.5 周期　1001：4.5 周期 1010：5.5 周期　1011：6.5 周期 1100：7.5 周期　其他：保留
9: 7	RSLTCTL	ADC 转换数据分辨率选择（Data Resolution Selection） 000：12 位有效 001：11 位有效 010：10 位有效 011：9 位有效 100：8 位有效 其它：保留
6: 4	ADCPREH	ADC 时钟预分频高位（ADC Prescaler High Bits） 预分频系数 ADCPRE={ADCPREH, ADCPREL} ADC 时钟分频：div= (ADCPRE+2)
3	VSEN	内部参考电压传感器使能（Voltage Sensor Enable） 1：内部电压传感器使能 0：内部电压传感器禁用
2	Reserved	保留，必须保持复位值。
1	ADWEN	A/D 窗口比较器使能 (ADC window comparison enable) 1：A/D 窗口比较器使能 0：A/D 窗口比较器禁用
0	ADEN	A/D 转换使能 （ADC Enable） 1： 使能 0： 禁止

8.12.4 A/D 控制寄存器 (ADC_ADCR)

地址偏移：0x08

复位值：0x00000000

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

						rw				rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPCH				ALIGN	ADMD		ADST	Res.	TRGSELL			Res.	TRGEN	AWDIE	ADIE
rw				rw	rw		rw		rw				rw	rw	rw

Bit	Field	Description
31:26	Reserved	保留，必须保持复位值
25 : 24	TRG_EDGE	触发源边沿选择(Trigger Sources Edge Selection) 00：双沿触发 01：下沿触发 10：上沿触发 11：屏蔽触发
23:22	Reserved	保留，必须保持复位值。
21 : 19	TRGSHIFT	外部触发源采样延时（External Trigger Sources Shift For Sample） 在触发产生后，延时 N 个 PCLK1 的时钟周期再开始采样。 如果是触发扫描模式，其他通道是在上一个采样结束后立即开始。 000：不延时 001：4 个周期 010：16 个周期 011：32 个周期 100：64 个周期 101：128 个周期 110：256 个周期 111：512 个周期
18 :17	TRGSELH	外部触发源选择高位（External Trigger Sources Select For High Bits） TRGSEL={TRGSELH,TRGSELL}
16	Reserved	保留，必须保持复位值。
15 : 12	CMPCH	窗口比较通道选择 (Window comparison channel selection) 0000：选择比较通道 0 转换结果 0001：选择比较通道 1 转换结果 0010：选择比较通道 2 转换结果 0011：选择比较通道 3 转换结果 0100：选择比较通道 4 转换结果 0101：选择比较通道 5 转换结果 0110：选择比较通道 6 转换结果 0111：选择比较通道 7 转换结果 1000：选择比较内部参考电压转换结果 1111：所有扫描通道 其他：无效
11	ALIGN	数据对齐格式（Data Alignment） 1：左对齐 0：右对齐

Bit	Field	Description
10 : 9	ADMD	A/D 转换模式（ADC Mode） 00：单次转换 01：单周期扫描 10：连续扫描 11：保留 当改变转换模式时，软件要先清除 ADST 位。
8	ADST	A/D 转换开始（ADC Start） 1：转换开始 0：转换结束或空闲状态 ADST 置位有下列两种方式： 在单次模式或者单周期模式下，转换完成后， ADST 将被硬件自动清除。 在连续扫描模式下， A/D 转换将一直进行，直到软件写‘0’到该位或系统复位。
7	Reserved	保留，必须保持复位值。
6: 4	TRGSELL	外部触发源选择 (External trigger selection)，位 [18:17,6:4] 选择外部触发源 00000：TIM1_CC1 00001：TIM1_CC2 00010：TIM1_CC3 00011：Reserved 00100：TIM3_TRGO 00101：TIM1_CC4 和 TIM1_CC5 00110：TIM3_CC1 00111：EXTI 线 11 01000：TIM1_TRGO 01001：Reserved 01010：Reserved 01011：Reserved 01100：TIM3_CC4 01101：Reserved 01110：Reserved 01111：EXTI 线 15 10000：TIM1_CC4 10001：TIM1_CC5 其他：无效
3	Reserved	保留，必须保持复位值。
2	TRGEN	外部硬件触发源使能（External Trigger Sources Enable） 1：使用外部触发信号启动 A/D 转换 0：禁止外部触发信号启动 A/D 转换
1	AWDIE	A/D 窗口比较器中断使能 (ADC window comparator interrupt enable) 1: 使能 A/D 窗口比较器中断 0: 禁用 A/D 窗口比较器中断

Bit	Field	Description
0	ADIE	A/D 中断使能 (ADC interrupt enable) 1: 使能 A/D 中断 0: 禁用 A/D 中断 如果 ADIF 置位，A/D 转换结束后产生中断请求。

8.12.5 A/D 窗口比较寄存器 (ADC_ADCMPR)

地址偏移：0x10

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				CMPHDATA											
				rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				CMPLDATA											
				rw											

Bit	Field	Description
31:28	Reserved	保留，必须保持复位值
27: 16	CMPHDATA	比较数值上限 (Compare data high limit) 该 12 位数值将和指定通道的转换结果相比较。
15:12	Reserved	保留，必须保持复位值。
11 : 0	CMPLDATA	比较数值下限 (Compare data low limit) 该 12 位数值将和指定通道的转换结果相比较。

8.12.6 A/D 状态寄存器 (ADC_ADSTA)

地址偏移：0x14

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.			OVERRUN									Res.			VALID
			r												r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALID								CHANNEL				Res.	BUSY	AWDIF	ADIF
r								r					r	rc_w1	rc_w1

Bit	Field	Description
31：29	Reserved	保留，必须保持复位值

Bit	Field	Description
28:20	OVERRUN	通道 0~8 的数据覆盖标志位（Overrun Flag） 只读。通道 8 为电压传感器通道
19：17	Reserved	保留，必须保持复位值
16:8	VALID	通道 0~8 的数据有效标志位（Valid Flag） 只读。通道 8 为电压传感器通道
7:4	CHANNEL	当前转换通道（Current Convert Channel） 注：该 4 位在 BUSY = 1 时表示进行转换中的通道。 BUSY = 0 时表示可进行下次转换的通道。
3	Reserved	保留，必须保持复位值
2	BUSY	通道转换忙/空闲（Busy） 1: A/D 转换器忙碌 0: A/D 转换器空闲
1	AWDIF	比较标志位 (ADC window comparator interrupt flag) 当 CPMHDATA ≥ CPMLDATA 时，选择的 A/D 转换通道比较结果大于或等于 ADCMPR 寄存器的 CMPHDATA 指定值或者小于 CMPLDATA 指定值，状态寄存器 ADSTA 的 ADWIF 位置 1。 当 CPMHDATA < CPMLDATA 时，选择的 A/D 转换通道比较结果如果等于 CPMHDATA 指定值或者处于两个指定值之间，则状态寄存器 ADSTA 的 ADWIF 位置 1。 该标志位写‘1’清零。
0	ADIF	A/D 转换结束标志位 (ADC convert complete flag) 该位由硬件在通道组转换结束时设置，由软件清除。 1: A/D 转换完成 0: A/D 转换未完成 该标志位写‘1’清零。

8.12.7 A/D 数据寄存器 (ADC_ADDR0 ~ 8)

地址偏移：0x18~0x38

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										VAILD	OVERRUN	Res.			
										r	r				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA															
r															

Bit	Field	Description
31:22	Reserved	保留，必须保持复位值。

Bit	Field	Description
21	VALID	有效标志位（只读）（Valid Flag） 1： DATA[11:0]位数据有效 0： DATA[11:0]位数据无效 相应模拟通道转换完成后，将该位置位，读 ADDATA 寄存器后，该位由硬件清除。
20	OVERRUN	数据覆盖标志位 (只读)(Overrun flag) 1 ：DATA [11： 0] 数据被覆盖 0 ：DATA [11： 0] 数据最近一次转换结果 新的转换结果装载至寄存器之前，若 DATA[11： 0] 的数据没有被读取，OVERRUN 将置‘1’，读 ADDATA 寄存器后，该位由硬件清除。
19 : 16	Reserved	保留，必须保持复位值。
15 : 0	DATA	通道的 12 位 A/D 转换结果（Covert Data） 软件选择对齐方式。

8.12.8 A/D 任意通道通道选择寄存器 0(ADC_CHANY0)

地址偏移：0x5C

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHANY_SEL7				CHANY_SEL6				CHANY_SEL5				CHANY_SEL4			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANY_SEL3				CHANY_SEL2				CHANY_SEL1				CHANY_SEL0			
rw				rw				rw				rw			

Bit	Field	Description
31:28	CHANY_SEL7	可配置为通道 0 ~ 通道 8 中的任意通道。
27:24	CHANY_SEL6	可配置为通道 0 ~ 通道 8 中的任意通道。
23:20	CHANY_SEL5	可配置为通道 0 ~ 通道 8 中的任意通道。
19:16	CHANY_SEL4	可配置为通道 0 ~ 通道 8 中的任意通道。
15:12	CHANY_SEL3	可配置为通道 0 ~ 通道 8 中的任意通道。
11:8	CHANY_SEL2	可配置为通道 0 ~ 通道 8 中的任意通道。
7:4	CHANY_SEL1	可配置为通道 0 ~ 通道 8 中的任意通道。
3:0	CHANY_SEL0	可配置为通道 0 ~ 通道 8 中的任意通道。

注：单周期扫描或连续扫描模式下，硬件会启动 ADC_CHANY0 影子寄存器，在 ADC 未开始工作时，软件写 ADC_CHANY0 的话，也会写到其影子寄存器；在 ADC 工作期间，若更改 ADC_CHANY0 的值，只会更新其影子寄存器，且当 ADC 开始转换最后一个通道时，影子寄存器的值会更新至 ADC_CHANY0，这样即可完成动态切换通道。

8.12.9 A/D 任意通道通道选择寄存器 1(ADC_CHANY1)

地址偏移：0x60

复位值：0x00000000

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.												CHANY_SEL8			
												rw			

Bit	Field	Description
31:4	Reserved	保留，必须保持复位值。
3:0	CHANY_SEL8	可配置通道 0 ～通道 8 中的任意通道。

注：单周期扫描或连续扫描模式下，硬件会启动 ADC_CHANY1 影子寄存器，在 ADC 未开始工作时，软件写 ADC_CHANY1 的话，也会写到其影子寄存器；在 ADC 工作期间，若更改 ADC_CHANY1 的值，只会更新其影子寄存器，且当 ADC 开始转换最后一个通道时，影子寄存器的值会更新至 ADC_CHANY1，这样即可完成动态切换通道。

8.12.10 A/D 任意通道配置寄存器 (ADC_ANY_CFG)

地址偏移：0x64

复位值：0x00000000

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.												CHANY_NUM			
												rw			

Bit	Field	Description
31:4	Reserved	保留，必须保持复位值

Bit	Field	Description
3: 0	CHANY_NUM	转换通道数配置(Number of Any Channel Mode)： 0000：0 通道 0001：0~1 通道 0010：0~2 通道 0011：0~3 通道 0100：0~4 通道 0101：0~5 通道 0110：0~6 通道 0111：0~7 通道 1000：0~8 通道

注：单周期扫描或连续扫描模式下，硬件会启动 ADC_NUM 影子寄存器，在 ADC 未开始工作时，软件写 ADC_NUM 的话，也会写到其影子寄存器；在 ADC 工作期间，若更改 ADC_NUM 的值，只会更新其影子寄存器，且当 ADC 开始转换最后一个通道时，影子寄存器的值会更新至 ADC_NUM，这样即可完成动态切换通道。

8.12.11 A/D 任意通道控制寄存器 (ADC_ANY_CR)

地址偏移：0x68

复位值：0x00000000

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.															CHANY_MDEN
															rw

Bit	Field	Description
31:1	Reserved	保留，必须保持复位值
0	CHANY_ MDEN	任意通道转换使能位(Any Channel Mode Enable)： 1：使能 0：禁止 在使能后，配置 ADC 通道功能发生变化。由两部分共同控制，CHANY_NUM 是配置通道 0 ～ 通道 8 中的通道数，然后通道 0 ～通道 8 分别由 CHANY_SEL0 ～ CHANY_SEL8 配置为任意 ADC 通道。

注：在任意通道模式，且单周期/连续扫描模式下，关闭 ADC 时，必须先禁止 ADC_ADCR 的 ADST 位，然后判断 ADC_ADSTA 的 BUSY 位是否为 0，即等到 ADC 转换完成，再禁止 ADC_ANY_CR 的 CHANY_MDEN 位。

9 TIM1 高级控制定时器

9.1 简介

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。

它适合多种用途，能够提供比较输出、PWM 输出、死区时间可编程的互补输出、单脉冲模式输出等多种功能。

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) 和通用定时器 (TIMx) 是完全独立的，它们不共享任何资源。它们可以同步操作，具体描述参看通用定时器同步的章节。

9.2 主要特征

TIM1 定时器的功能包括：

- 16 位向上、向下、向上/下自动装载寄存器
- 16 位可编程(可以实时修改) 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
- 5 个比较通道（通道 1~4 支持输出，通道 5 仅提供中断及内部触发信号）
- 4 个输出通道，通道 1/2/3 有互补输出通道，通道 4 无互补输出通道
  - 输出比较
  - PWM 生成 (边缘或中间对齐模式)
  - 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联的同步电路
- 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

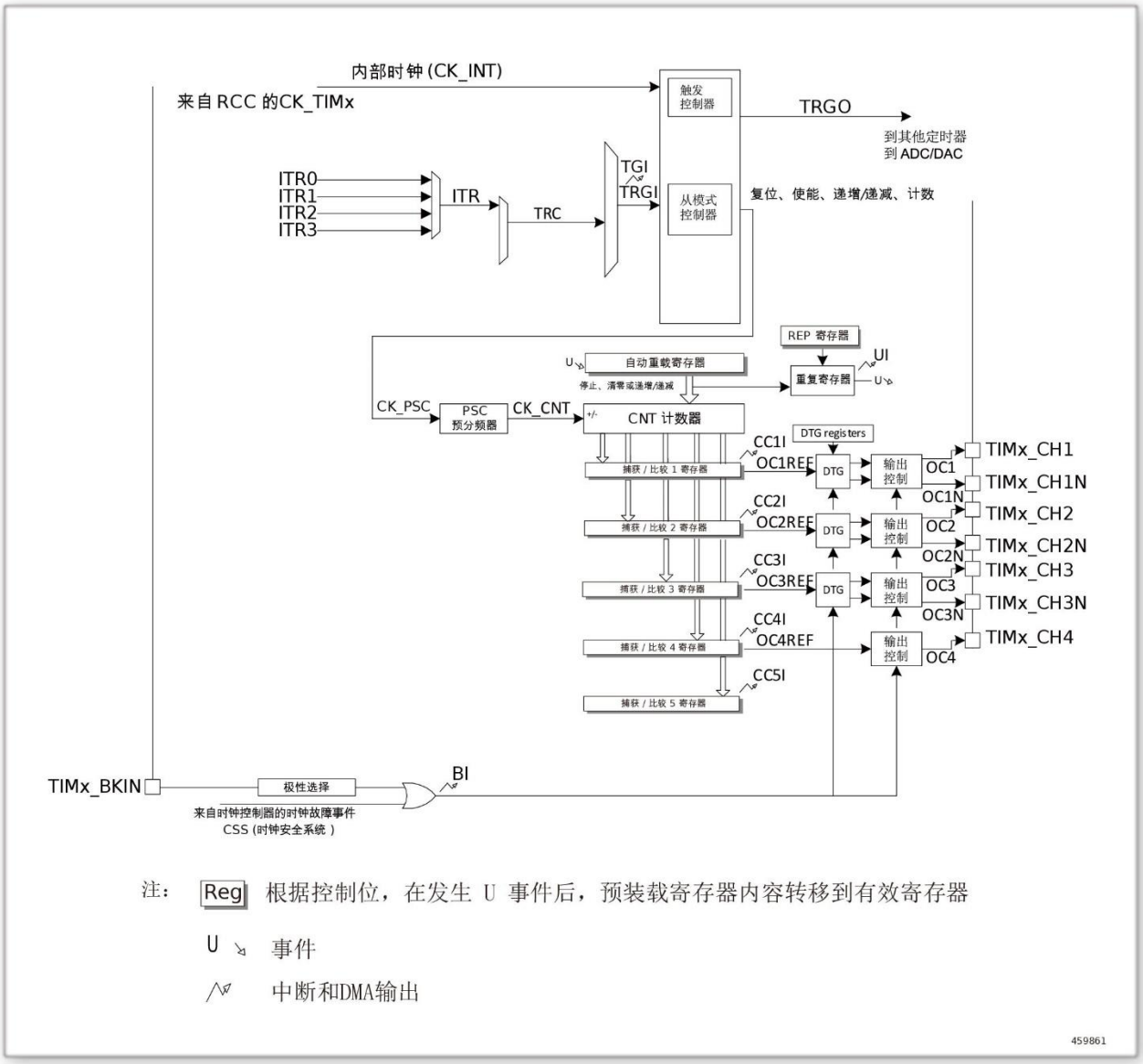

图 9-1 高级控制定时器框图

9.3 功能描述

9.3.1 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 (TIMx_CNT)

- 预分频器寄存器 (TIMx_PSC)
- 自动装载寄存器 (TIMx_ARR)
- 重复次数寄存器 (TIMx_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 TIMx_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发送更新事件 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 TIMx_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 TIMx_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，请参见控制器的从模式描述)。

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

9.3.1.1 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 (TIMx_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。

下面两个图分别给出了在预分频器运行时，更改计数器参数的例子。当配置 TIM1_SMCR 寄存器的 SMS=000、关闭从模式时，计数器使能打开，预分频器的时钟直接由内部时钟驱动。此时计数器时钟为内部时钟分频后的时钟。

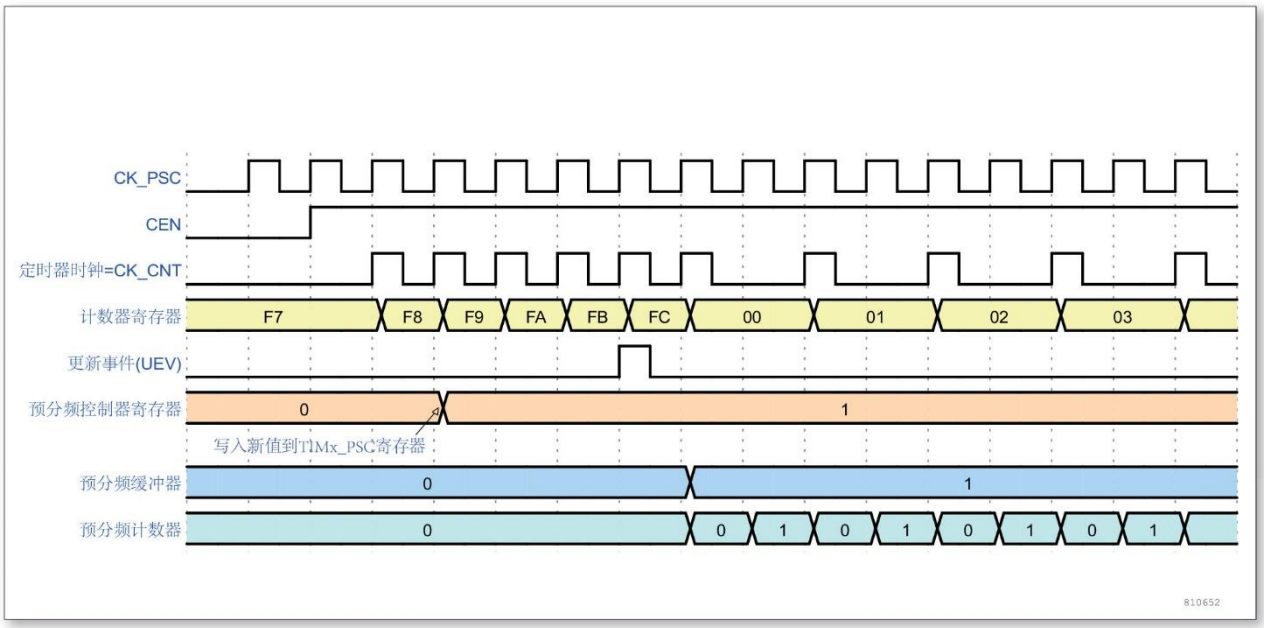

图 9-2 当预分频器的参数从 1 变到 2 时，计数器的时序图

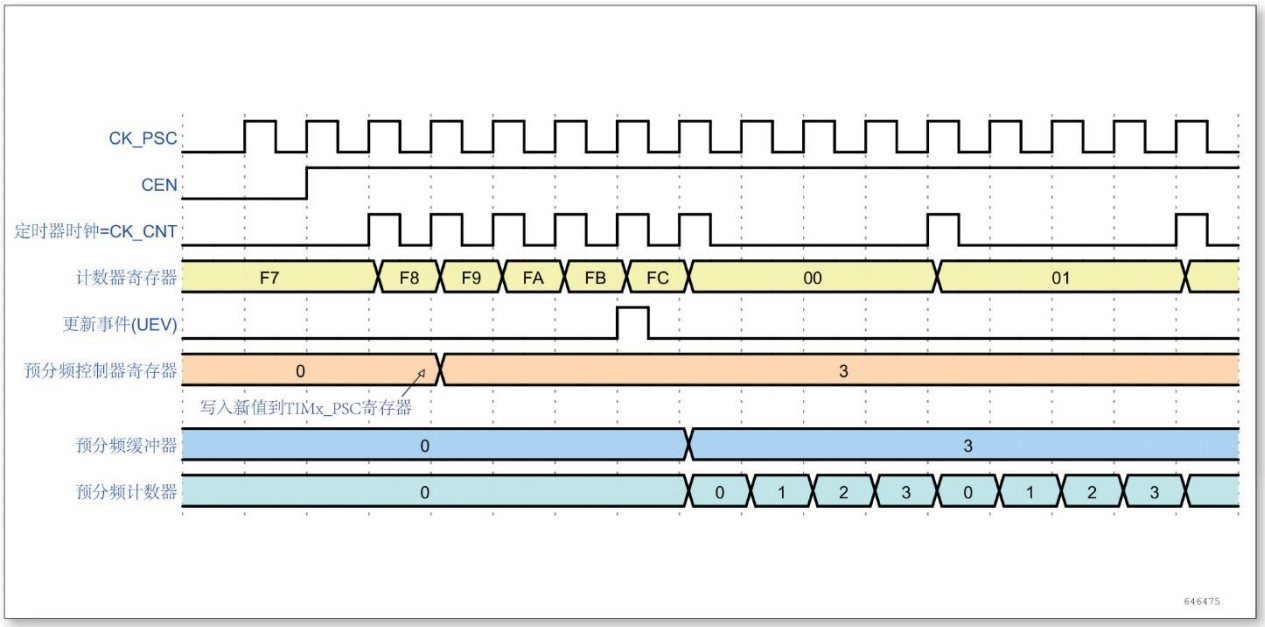

图 9-3 当预分频器的参数从 1 变到 4 时，计数器的时序图

9.3.2 计数模式

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

设置 TIMx_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请 0(但预分频器的数值不变)。此外，如果设置了 TIMx_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)。

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 (TIMx_SR 寄存器中的 UIF 位)。

- 重复计数器被重新加载为 TIMx_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 (TIMx_ARR)。
- 预分频器的缓冲区被置入预装载寄存器的值 (TIMx_PSC 寄存器的内容)。

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

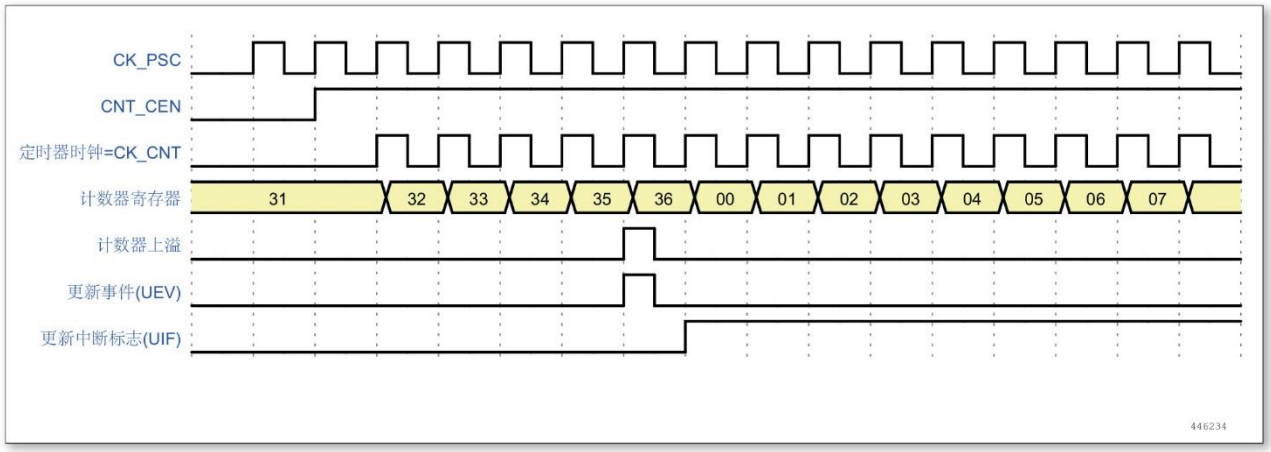

图 9-4 计数器时序图，内部时钟分频因子为 1

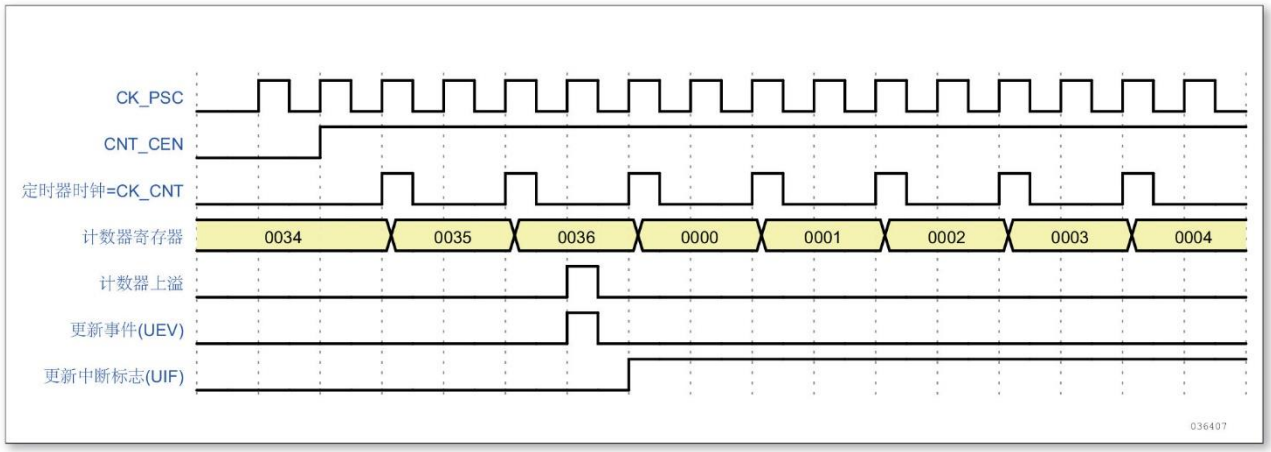

图 9-5 计数器时序图，内部时钟分频因子为 2

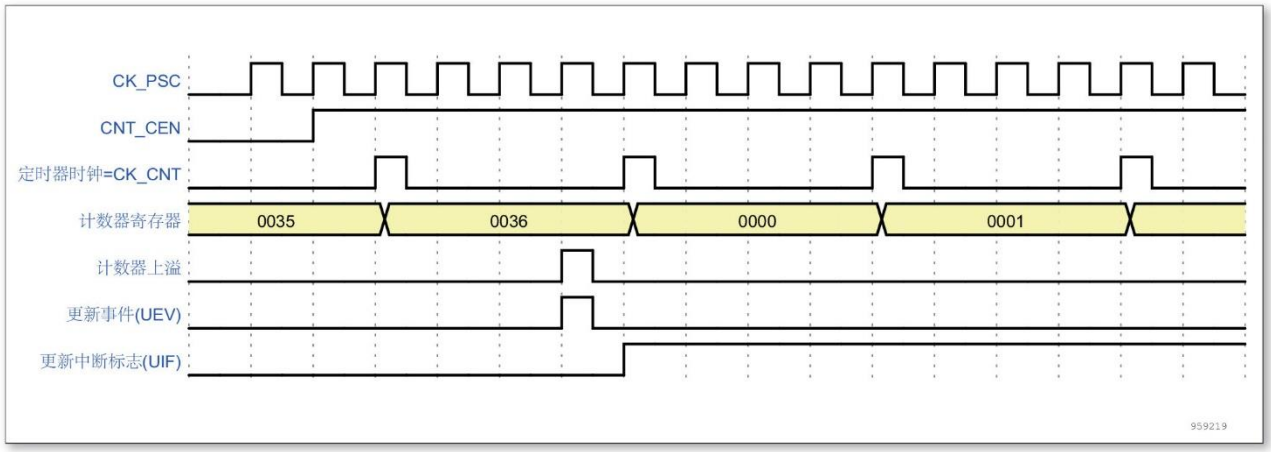

图 9-6 计数器时序图，内部时钟分频因子为 4

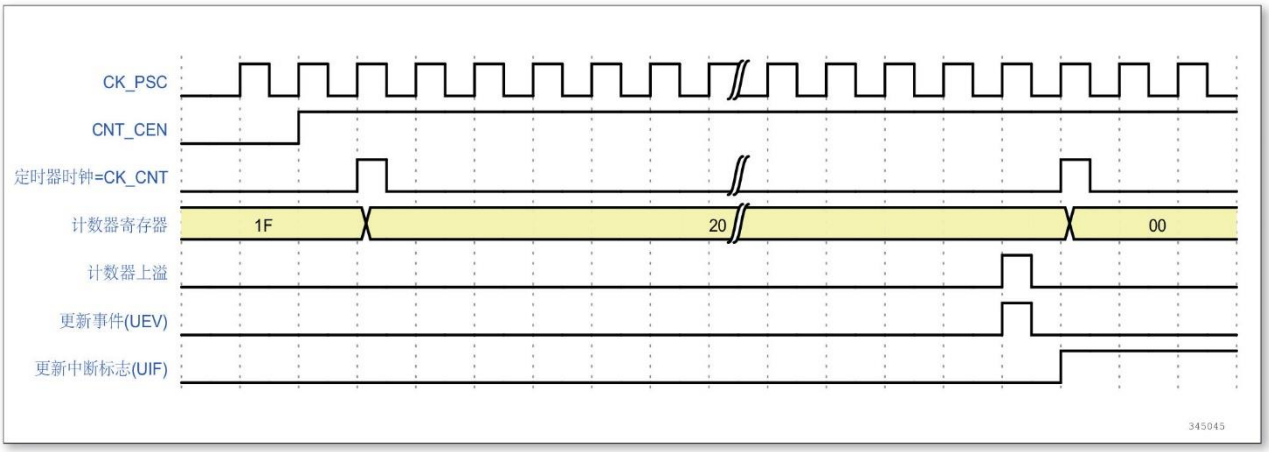

图 9-7 计数器时序图，内部时钟分频因子为 N

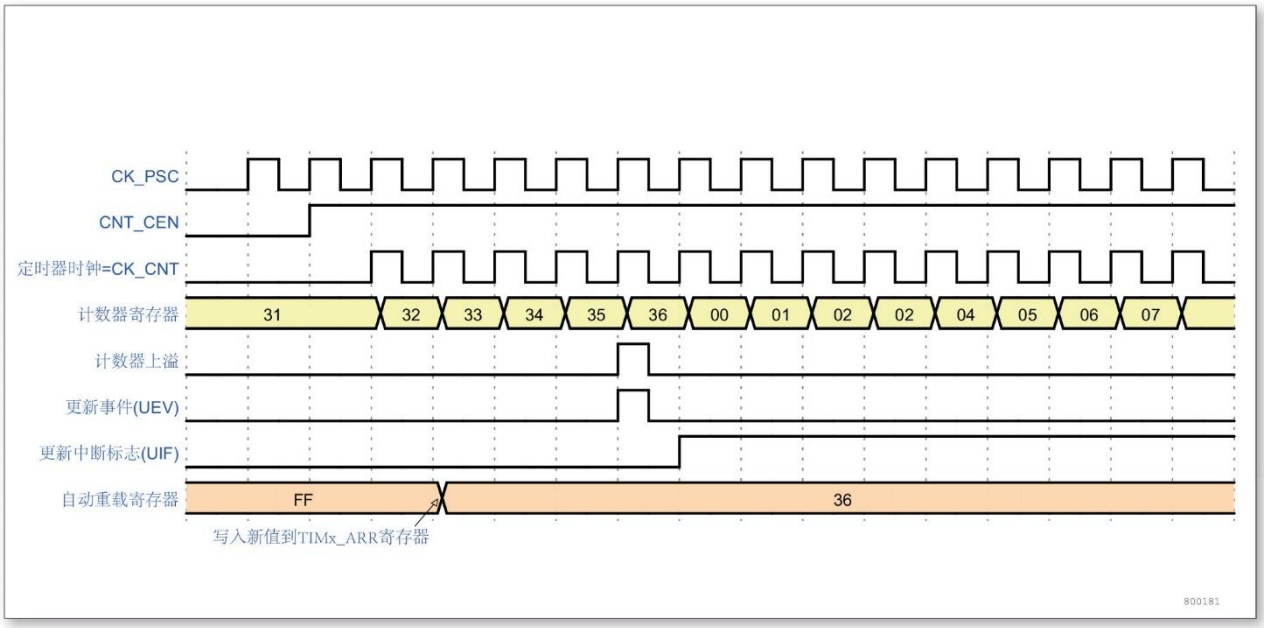

图 9-8 计数器时序图，当 ARPE = 0 时的更新事件 (TIMx_ARR 没有预装入)

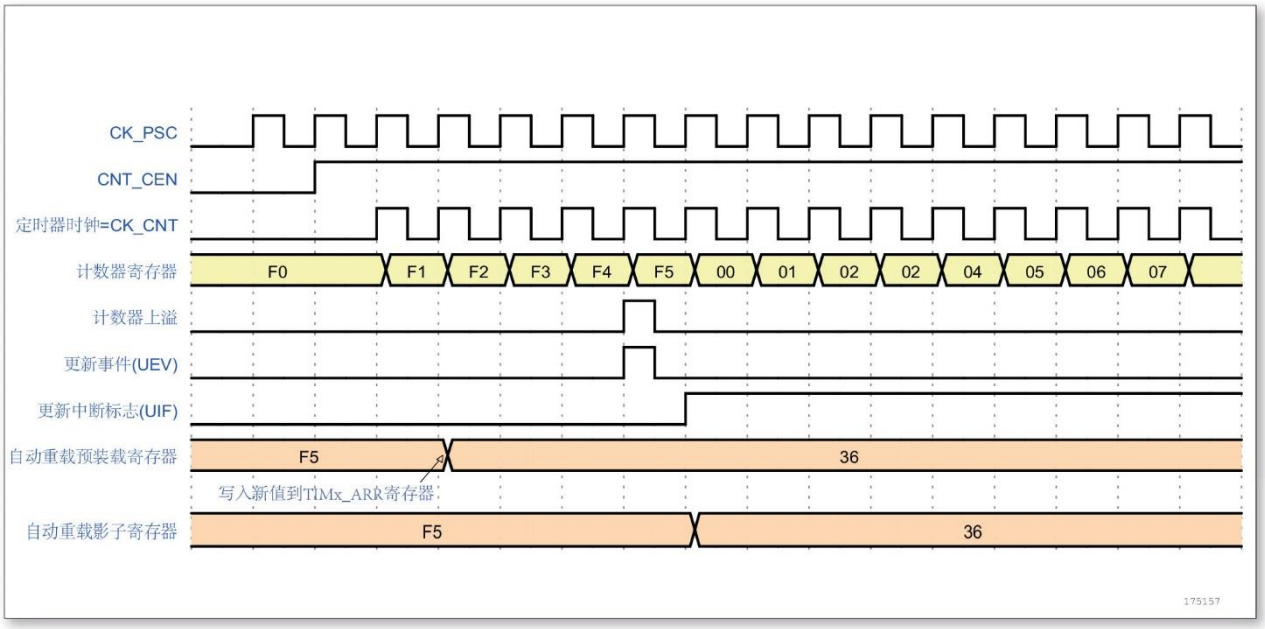

图 9-9 计数器时序图，当 ARPE = 1 时的更新事件 (预装入了 TIMx_ARR)

9.3.2.2 向下计数模式

在向下模式中，计数器从自动装入的值 (TIMx_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 (TIMx_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 TIMx_EGR 寄存器中设置 UG 位 (通过软件方式或者使用从模式控制器) 也同样可以产生一个更新事件。

设置 TIMx_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 (但预分频器的速率不能被修改)。

此外，如果设置了 TIMx_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)。

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 (TIMx_SR 寄存器中的 UIF 位) 也被设置。

- 重复计数器被重置为 TIMx_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 (TIMx_PSC 寄存器的值)。
- 当前的自动加载寄存器被更新为预装载值 (TIMx_ARR 寄存器中的内容)。

注：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 TIMx_ARR = 0x36 时，计数器在不同时钟频率下的操作例子。

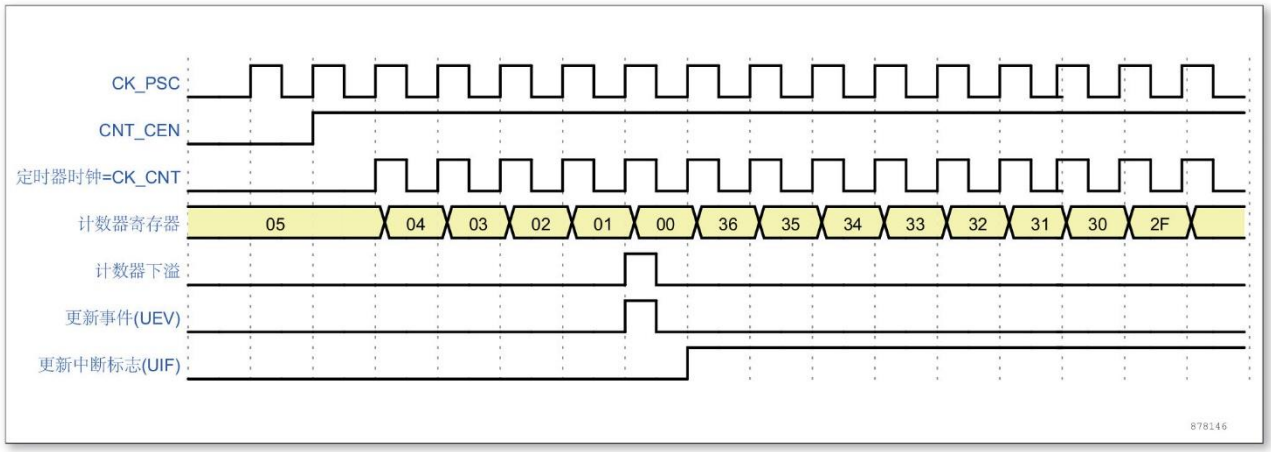

图 9-10 计数器时序图，内部时钟分频因子为 1

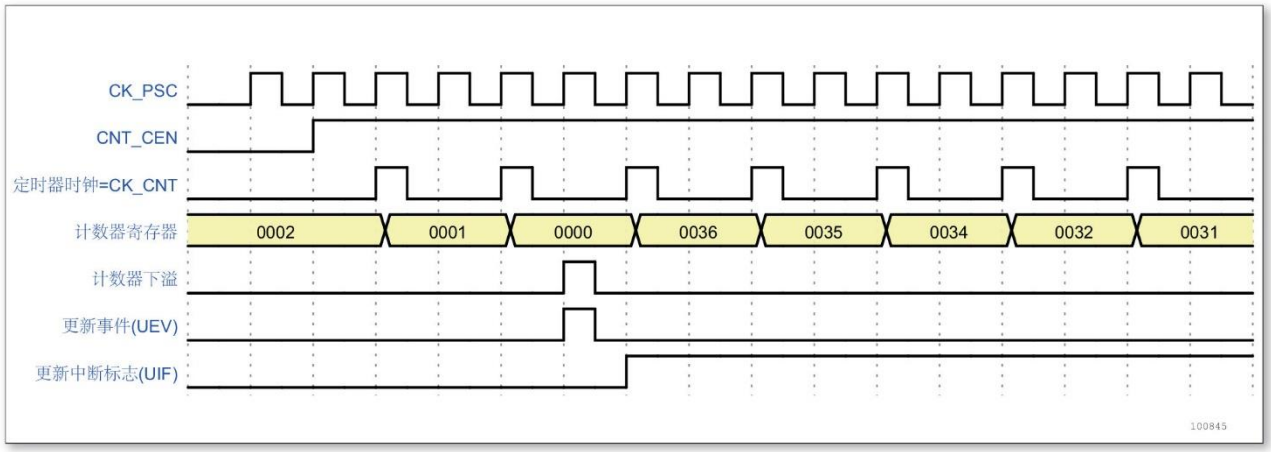

图 9-11 计数器时序图，内部时钟分频因子为 2

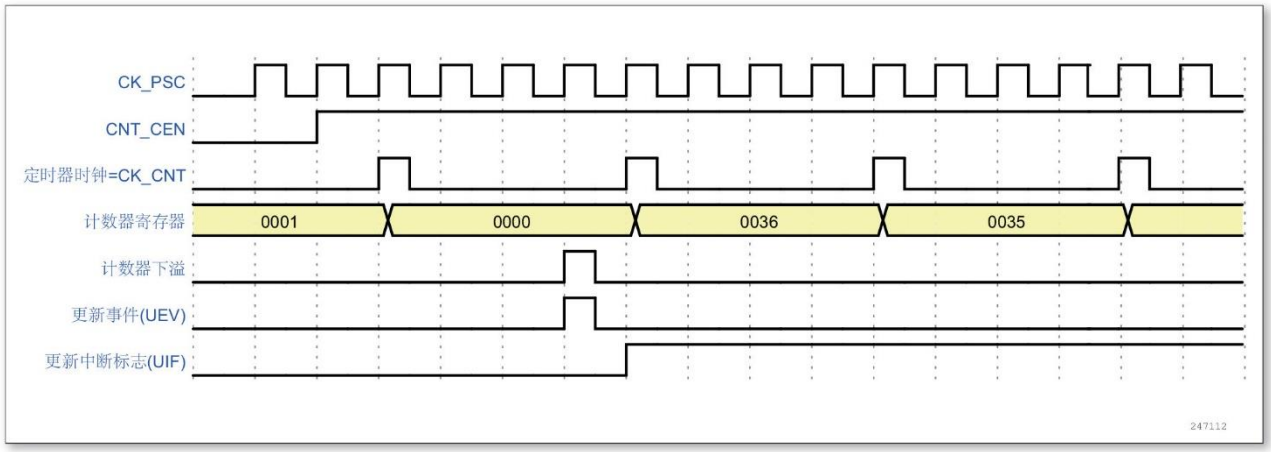

图 9-12 计数器时序图，内部时钟分频因子为 4

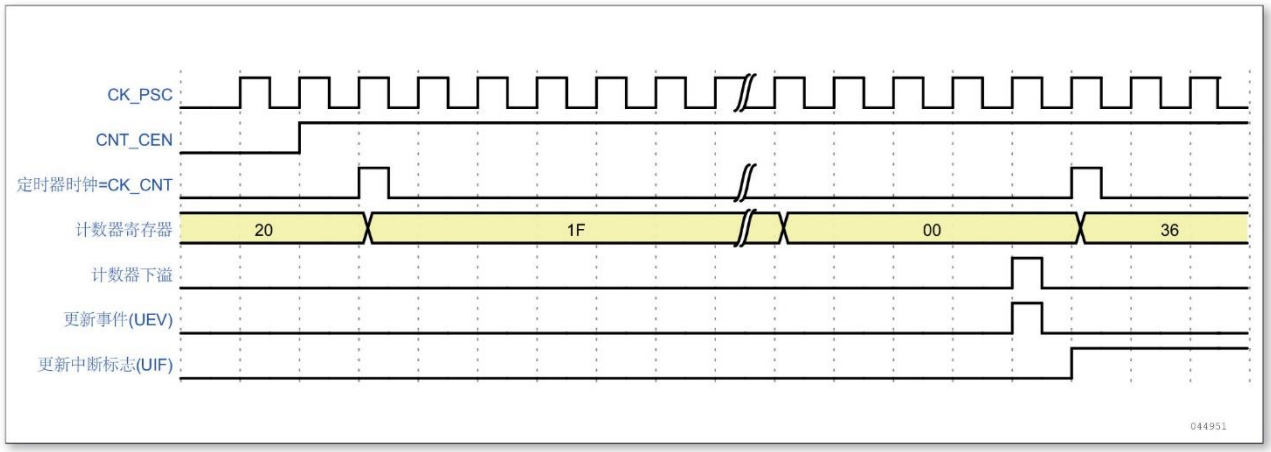

图 9-13 计数器时序图，内部时钟分频因子为 N

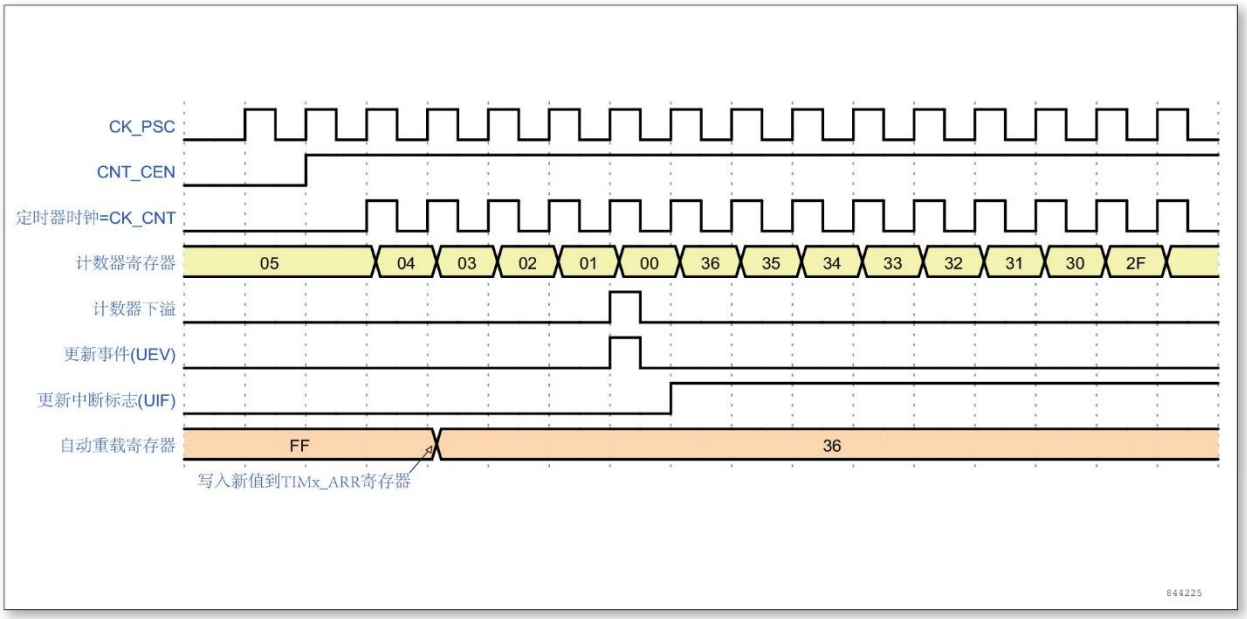

图 9-14 计数器时序图，当没有使用重复计数器时的更新事件

9.3.2.3 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 (TIMx_ARR 寄存器)- 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 TIMx_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

更新事件可以产生在每次计数上溢和每次计数下溢；也可以通过 (软件或者使用从模式控制器) 设置 TIMx_EGR 寄存器中的 UG 位产生。此时，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 TIMx_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 TIMx_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)。

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 (TIMx_SR 寄存器中的 UIF 位) 也被设置。

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

注：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：

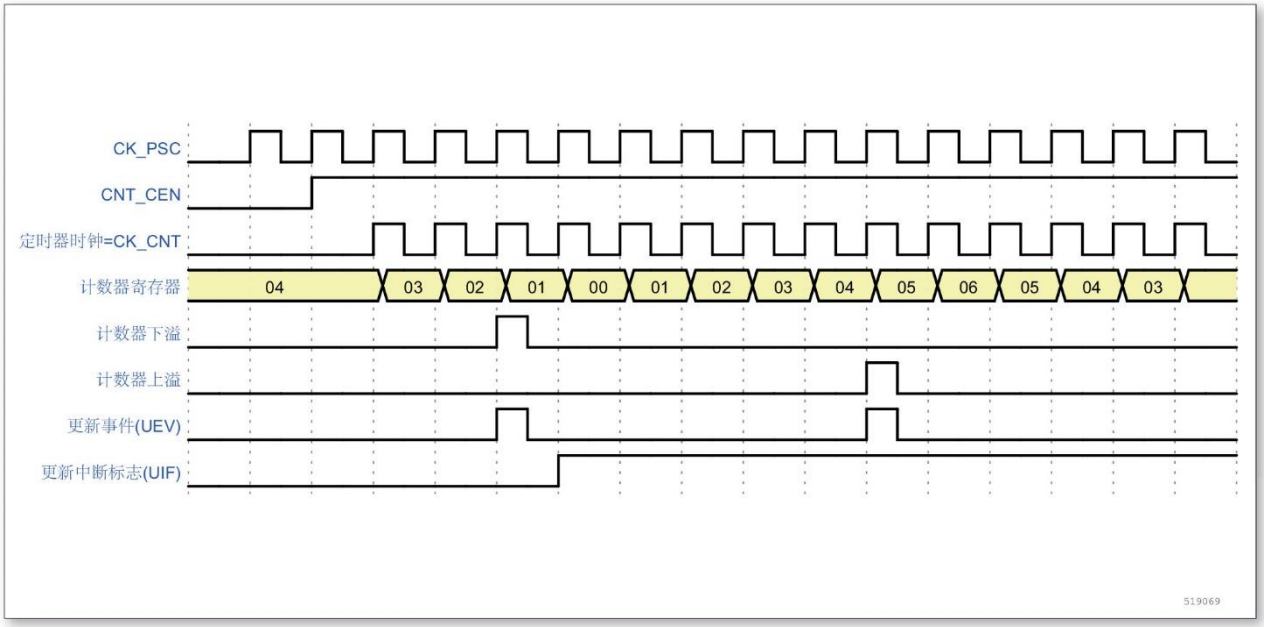

图 9-15 计数器时序图，内部时钟分频因子为 1， TIMx_ARR = 0x06

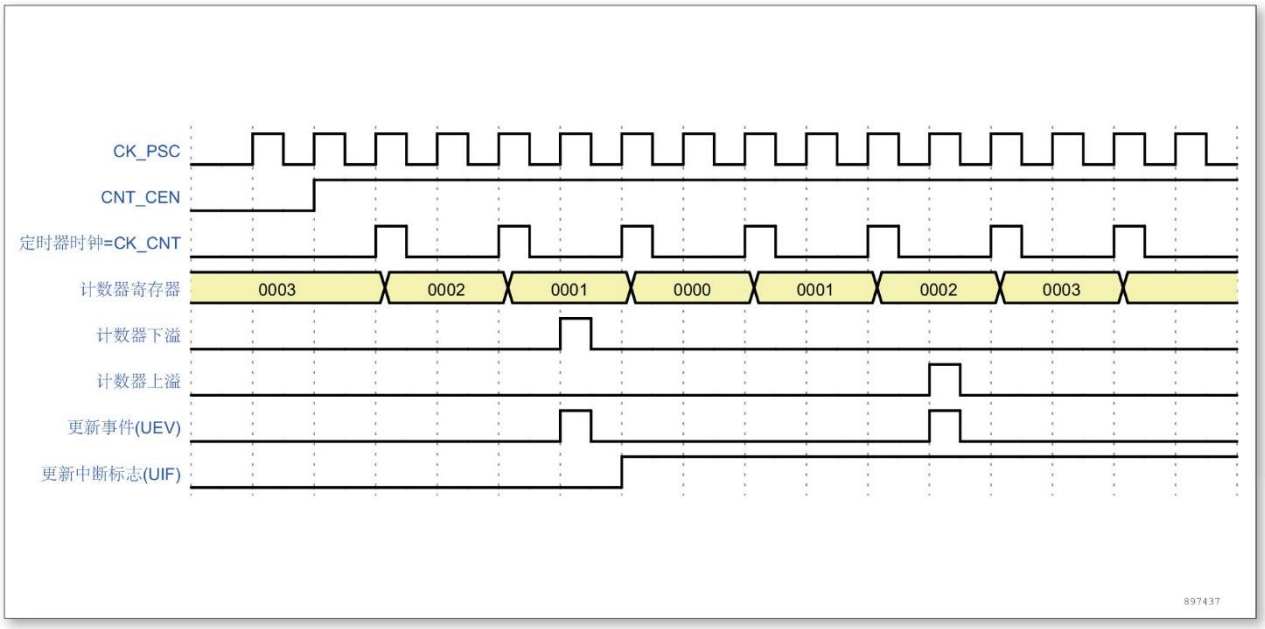

图 9-16 计数器时序图，内部时钟分频因子为 2

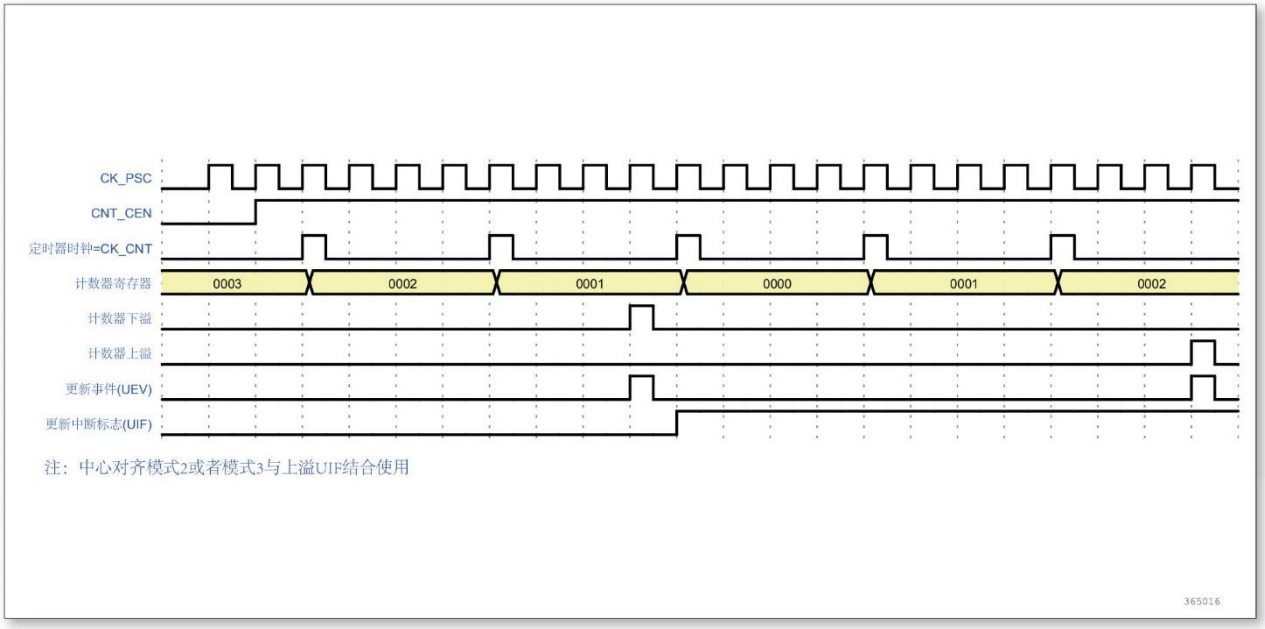

图 9-17 计数器时序图，内部时钟分频因子为 4， TIMx_ARR = 0x03

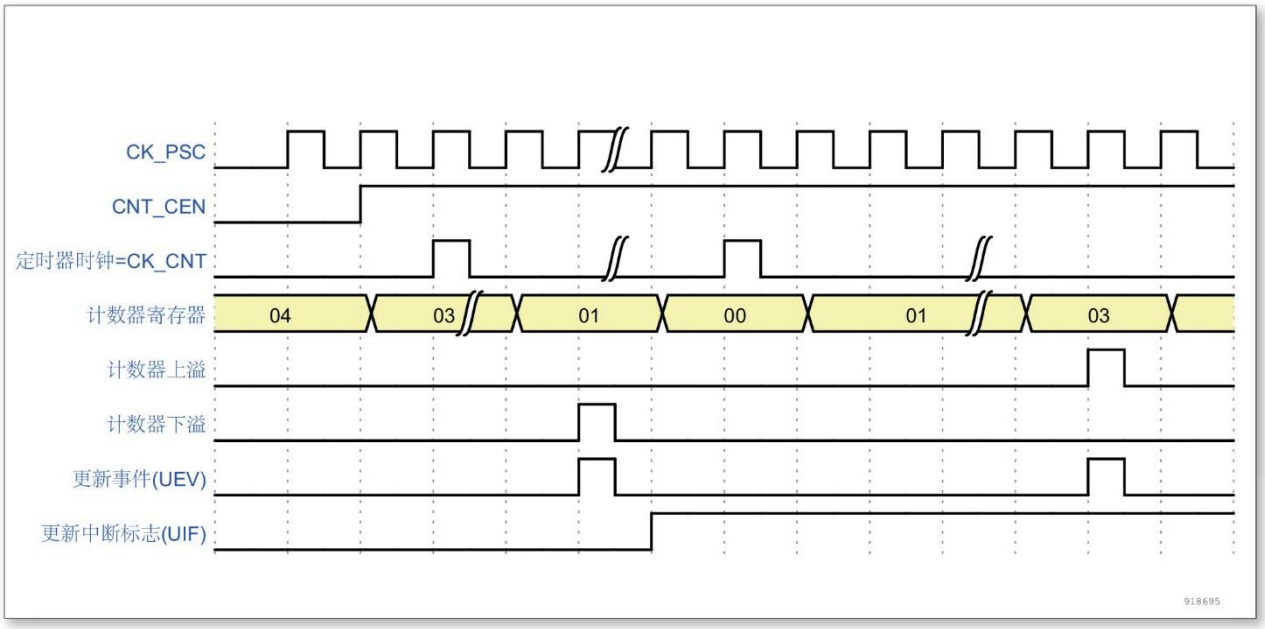

图 9-18 计数器时序图，内部时钟分频因子为 N

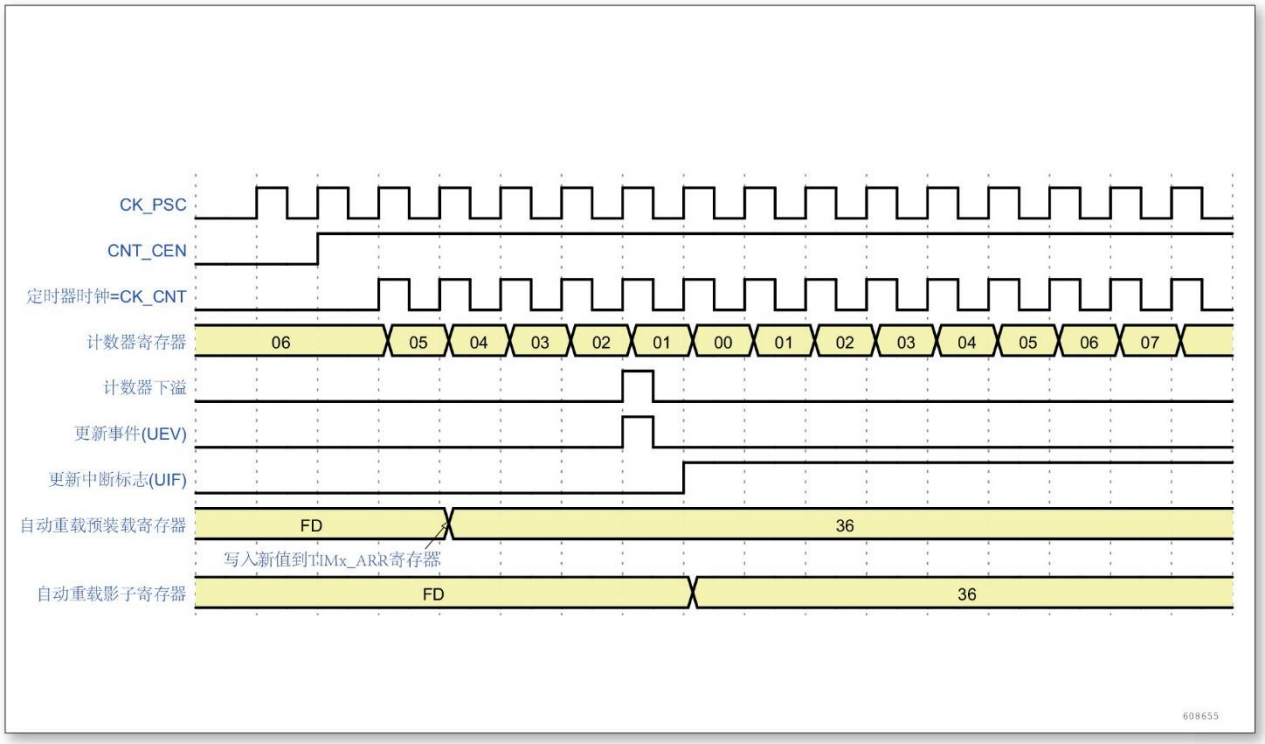

图 9-19 计数器时序图， ARPE = 1 时的更新事件 (计数器下溢)

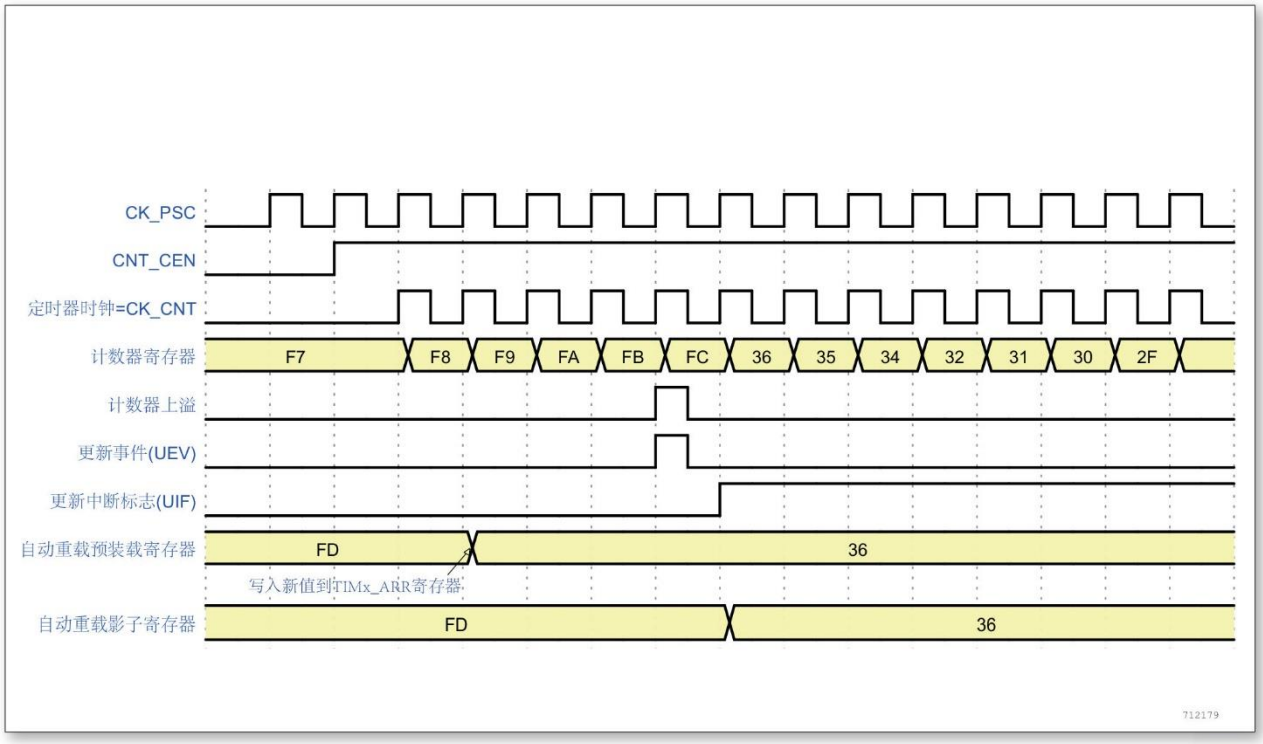

图 9-20 计数器时序图， ARPE = 1 时的更新事件 (计数器上溢)

9.3.3 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 (TIMx_ARR 自动重载入寄存器，TIMx_PSC 预装载寄存器，还有比较寄存器 TIMx_CCRx)，N 是 TIMx_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 2xTck。

重复计数器是自动加载的，重复速率是由 TIMx_RCR 寄存器的值定义 (参看图 48)。当更新事件由软件产生 (通过设置 TIMx_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 TIMx_RCR 寄存器中的内容被重载入到重复计数器。

新增 REP_CNT 寄存器，重复计数器的值实时写入 REP_CNT 中 (Repetition counter value of real-

time writing)，用于修改重复计数模式下，实时的将更新中断标志位 (UIF) 的检测点移位，即可将 UIF 左移 (REP-REP_CNT) 个相位 (相减值为负值时右移)。在更新事件 UG 后写入这些位 (注意在更新事件前写入 REP_CNT 将会使移位无效)。

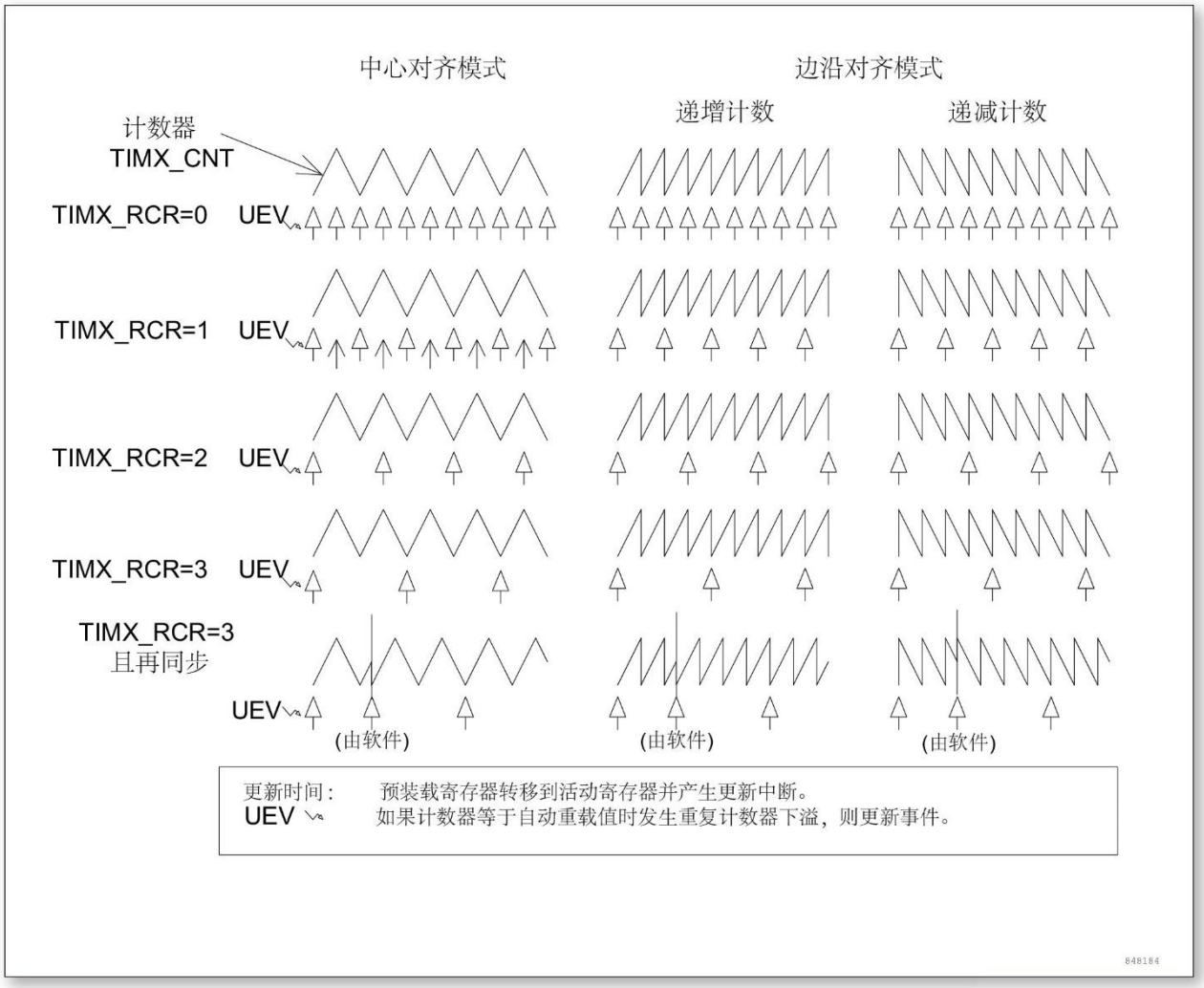

图 9-21 不同模式下更新速率的例子，及 TIMx_RCR 的寄存器设置

9.3.4 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟 (CK_INT)。
- 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 Timer3 的预分频器。

9.3.4.1 内部时钟源 (CK_INT)

如果禁止了从模式控制器 (SMS = 000)，则 CEN、DIR(TIMx_CR1 寄存器) 和 UG 位 (TIMx_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)。当 CEN 位被写成 1 时，预分频器的时钟由内部时钟 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。

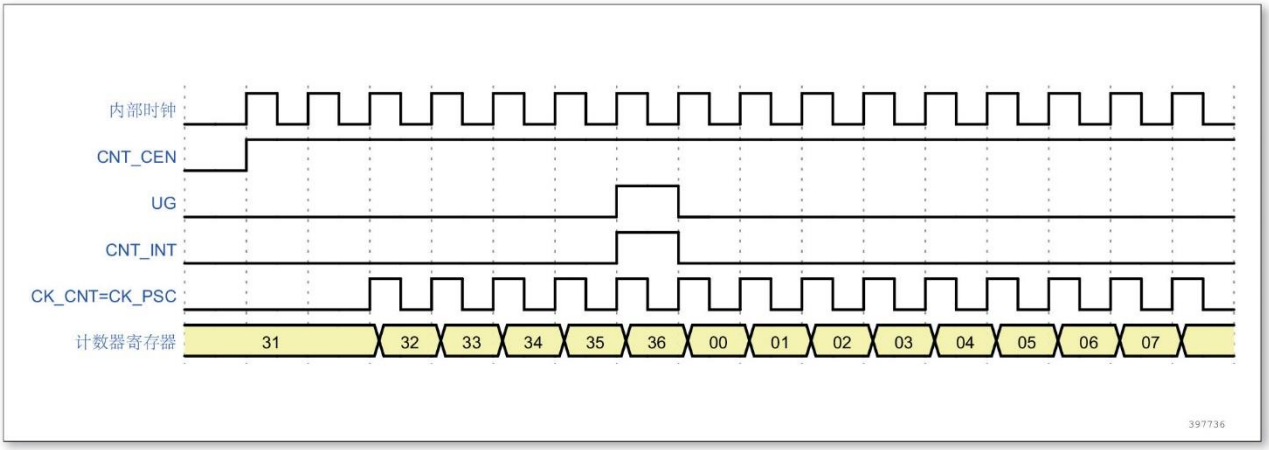

图 9-22 一般模式下的控制电路，内部时钟分频因子为 1

9.3.5 比较通道

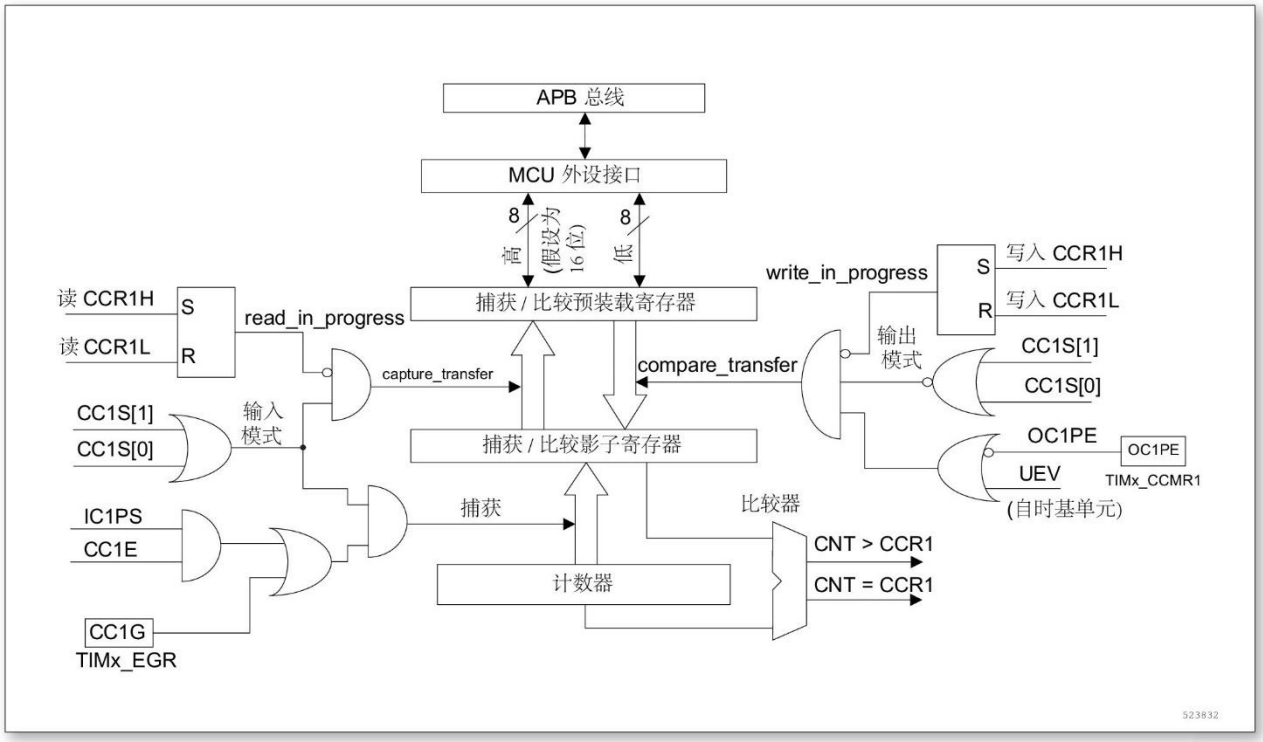

图 9-23 比较通道 1 的主电路

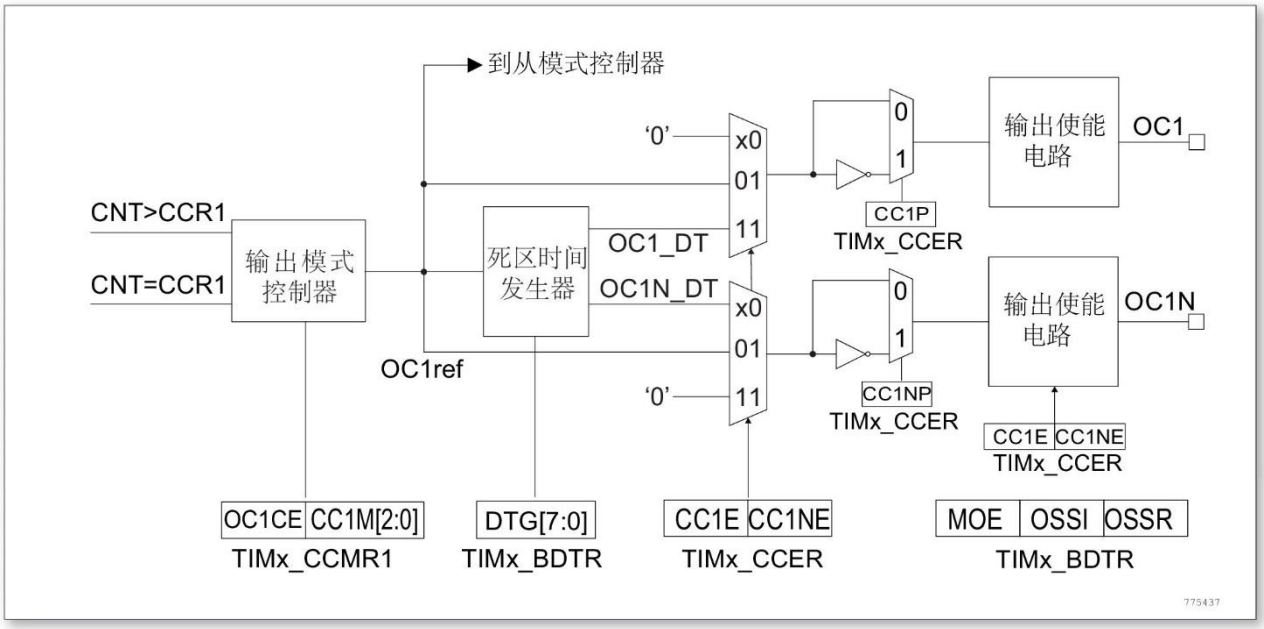

图 9-24 比较通道的输出部分 (通道 1 至 3)

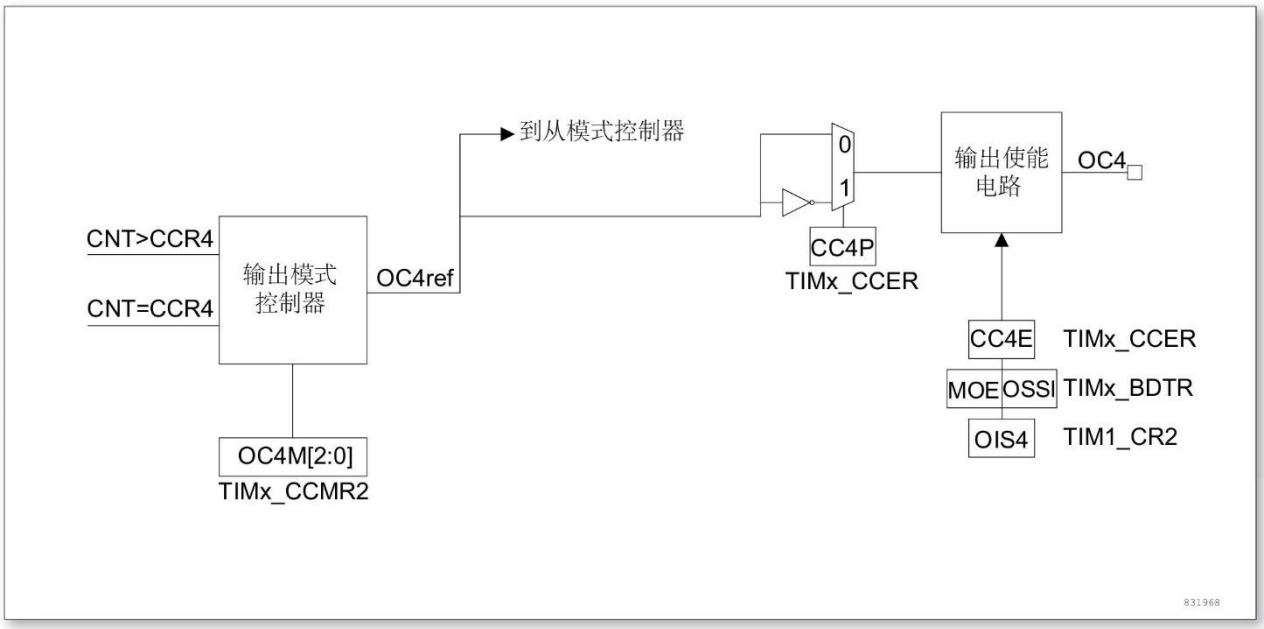

图 9-25 比较通道的输出部分 (通道 4)

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。

9.3.6 强制输出模式

在输出模式 (TIMx_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 TIMx_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 TIMx_CCMRx 寄存器中的 OCxM = 100，可强置 OCxREF 信号为低。该模式下，在 TIMx_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。

9.3.7 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何时一段给定的时间已经到时。 当计数器与比较寄存器的内容相同时，输出比较功能做如下操作：

- 将输出比较模式 (TIMx_CCMRx 寄存器中的 OCxM 位) 和输出极性 (TIMx_CCER 寄存器中的 CCxP 位) 定义的值输出到对应的管脚上。在比较匹配时，输出管脚可以保持它的电平 (OCxM = 000)、被设置成有效电平 (OCxM = 001)、被设置成无有效电平 (OCxM= 010) 或进行翻转 (OCxM = 011)。
- 设置中断状态寄存器中的标志位 (TIMx_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽 (TIMx_DIER 寄存器中的 CCxIE 位)，则产生一个中断。TIMx_CCMRx 中的 OCxPE 位选择 TIMx_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：

- 选择计数器时钟 (内部，外部，预分频器)
- 将相应的数据写入 TIMx_ARR 和 TIMx_CCRx 寄存器中
- 如果要产生一个中断请求，设置 CCxIE 位
- 选择输出模式，例如：
    - 要求计数器与 CCRx 匹配时翻转 OCx 的输出管脚，设置 OCxM = 011
    - 置 OCxPE = 0 禁用预装载寄存器
    - 置 CCxP = 0 选择极性为高电平有效
    - 置 CCxE = 1 使能输出
- 设置 TIMx_CR1 寄存器的 CEN 位启动计数器

TIMx_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 TIMx_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。

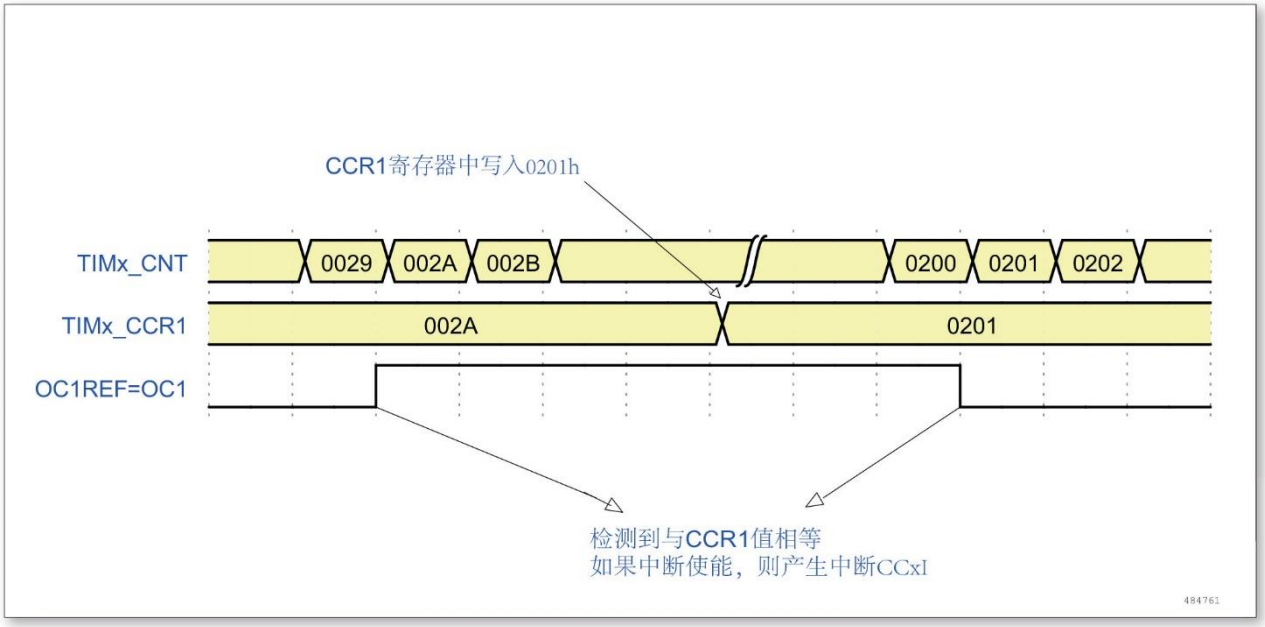

图 9-26 输出比较模式，翻转 OC1

9.3.8 PWM 模式

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

在 TIMx_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 TIMx_CCMRx 寄存器的 OCxPE 位使能相应的预装载寄存器，最后还要设置 TIMx_CR1 寄存器的 ARPE 位使能自动重装载的预装载寄存器 (在向上计数或中心对称模式中)。

因为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 TIMx_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 TIMx_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 (TIMx_CCER 和 TIMx_BDTR 寄存器中) CCxE、CCxNE 、MOE、OSSI 和 OSSR 位的组合控制。详见 TIMx_CCER 寄存器的描述。

在 PWM 模式(模式 1 或模式 2) 下，TIMx_CNT 和 TIMx_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 TIMx_CCRx ≤ TIMx_CNT 或者 TIMx_CNT ≤ TIMx_CCRx。

根据 TIMx_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。

9.3.8.1 PWM 边沿对齐模式

9.3.8.1.1 向上计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。参看计数模式章节。

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

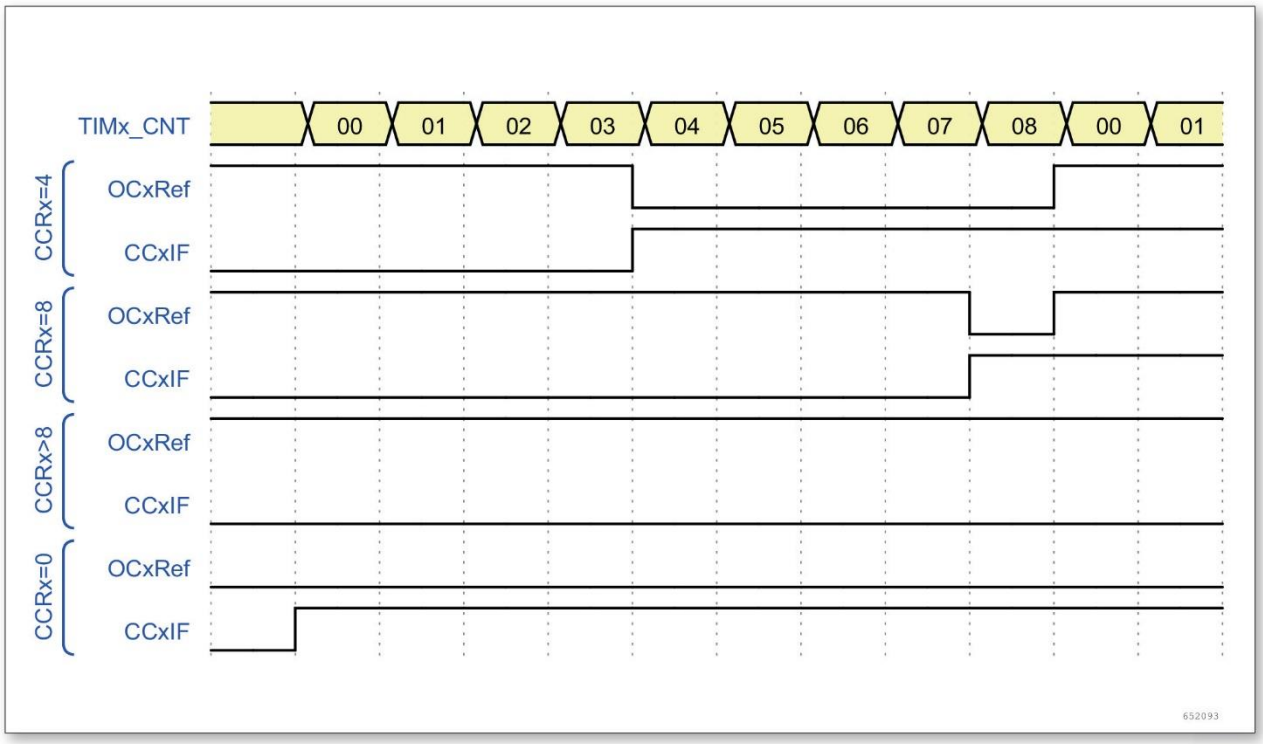

图 9-27 边沿对齐的 PWM 波形 (ARR = 8)

9.3.8.1.2 向下计数配置

当 TIMx_CR1 寄存器的 DIR 位为高时执行向下计数。参看计数模式章节。

在 PWM 模式 1，当 TIMx_CNT > TIMx_CCRx 时参考信号 OCxREF 为低，否则为高。如果 TIMx_CCRx 中的比较值大于 TIMx_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。

9.3.8.2 PWM 中央对齐模式

当 TIMx_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 位的设置，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’。TIMx_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。参看计数模式章节的中央对齐模式。

下图给出了一些中央对齐的 PWM 波形的例子

- TIMx_ARR = 8
- PWM 模式 1
- TIMx_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

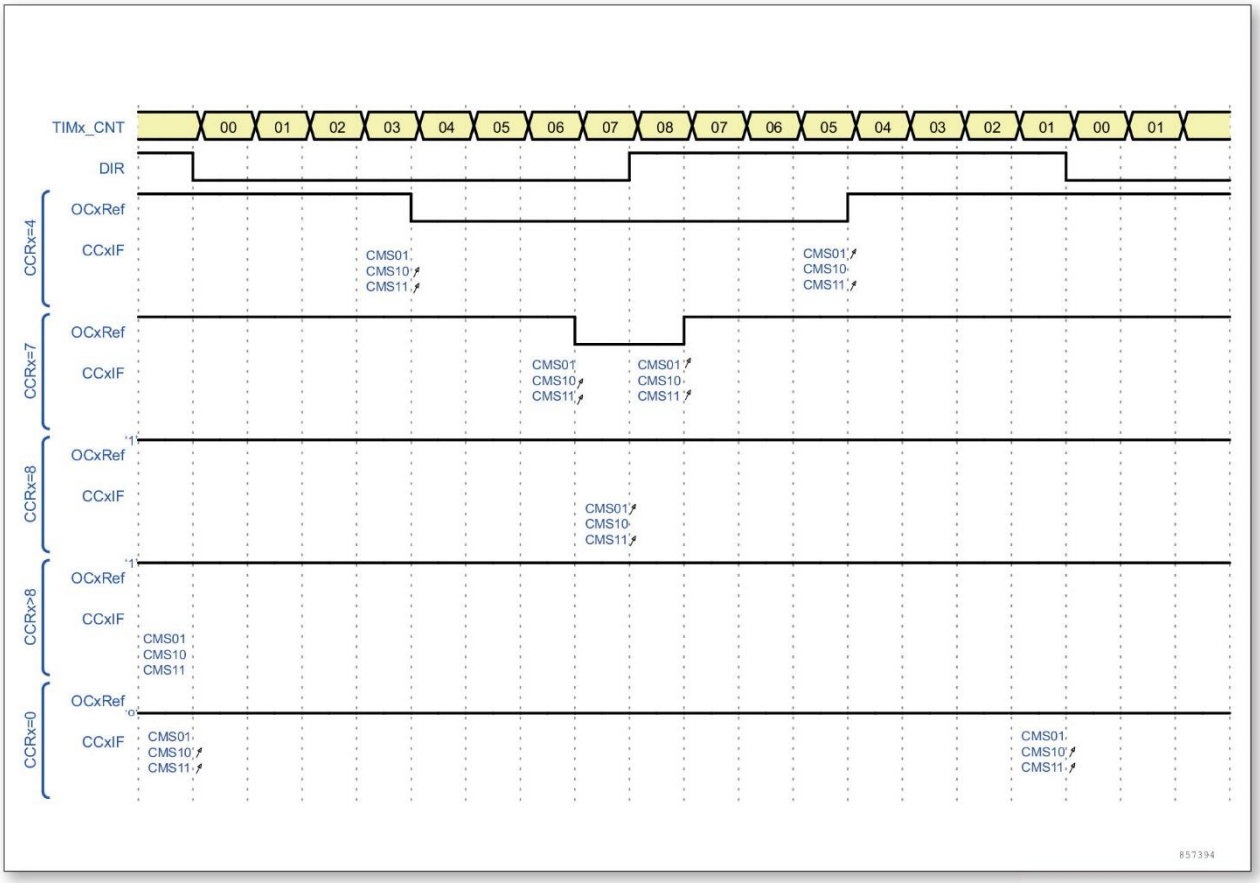

图 9-28 中央对齐的 PWM 波形 (APR = 8)

9.3.8.3 PWM 中央对齐模式下移相功能

新增 PDER(通道 x 输出 PWM 移相使能位) 和 CCRxFALL(通道 x 在 PWM 中央对齐模式向下计数时的比较值)，允许 5 个通道输出 PWM 移相。开启 PDER 寄存器的 PWM 移相使能，根据需要移动相位，配置 CCRxFALL 以及 CCRx，即可实现 PWM 输出可编程的移相波形，可左移或是右移。

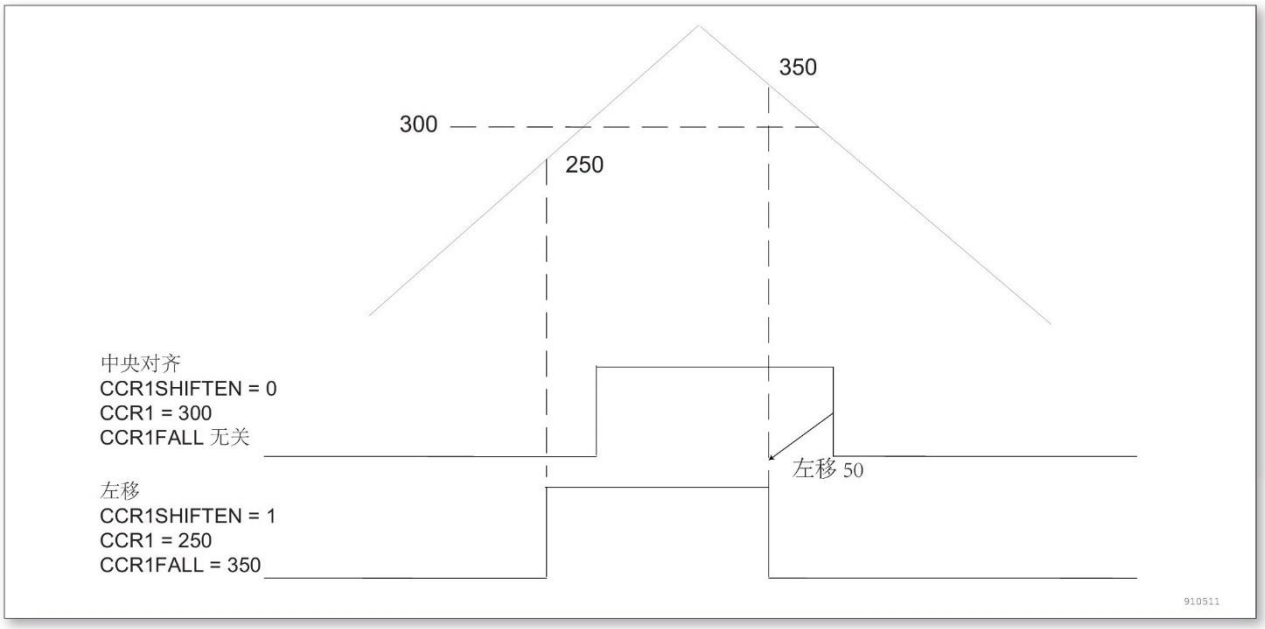

图 9-29 移相功能示意图

使用中央对齐模式的提示：

- 进入中央对齐模式时，使用当前的上/下计数配置；这就意味着计数器向上还是向下计数取决于 TIMx_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 (TIMx_CNT > TIMx_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 TIMx_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 TIMx_EGR 位中的 UG 位)，不要在计数进行过程中修改计数器的值。

9.3.9 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 TIMx_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：TIMx_CCER 寄存器的 CCxE 和 CCxNE 位，TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位，详见表 9-3 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。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。如果 OCx 和 OCxN 为高有效：

- 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。(配置 CCxP = 0、CCxNP = 0、MOE = 1、CCxE = 1 并且 CCxNE = 1)。

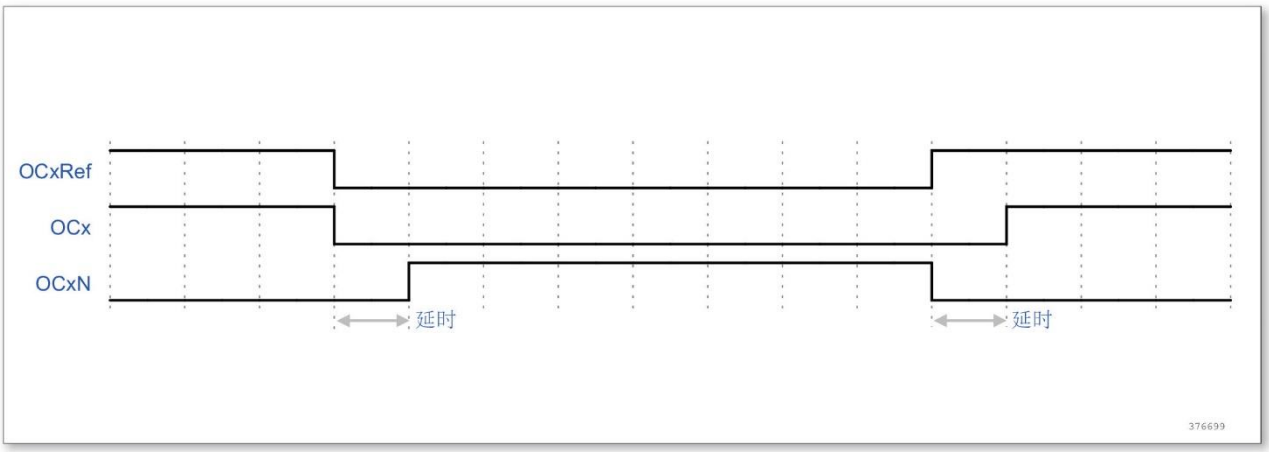

图 9-30 带死区插入的互补输出

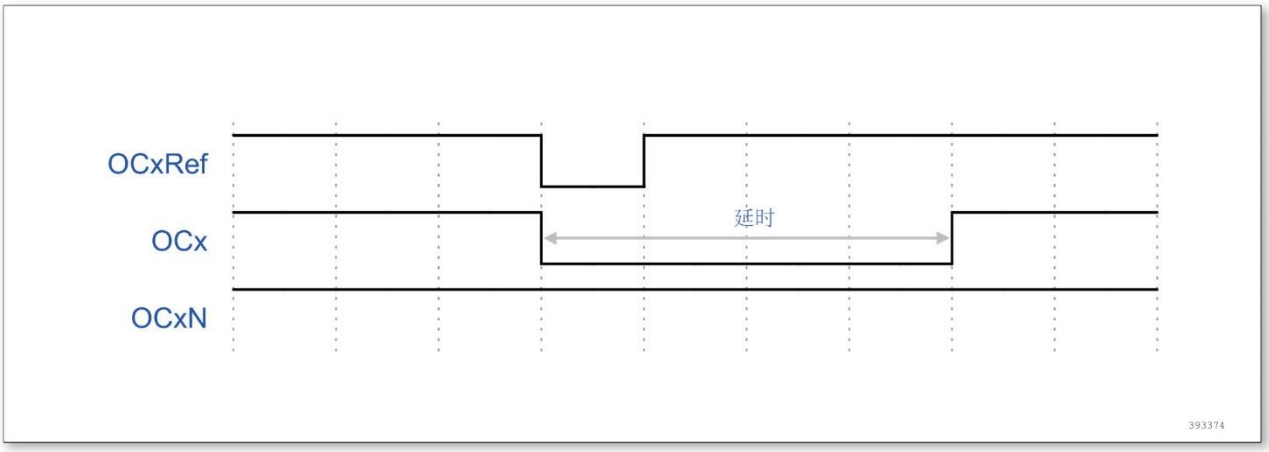

图 9-31 死区波形延迟大于负脉冲

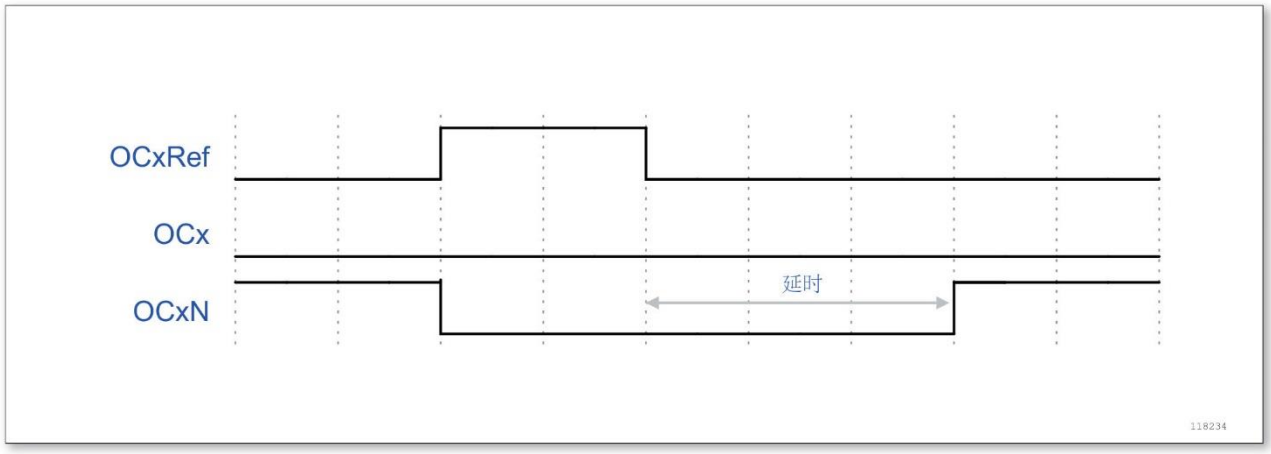

图 9-32 死区波形延迟大于正脉冲

每一个通道的死区延时都是相同的，是由 TIMx_BDTR 寄存器中的 DTG 位编程配置。详见寄存器章节中的延时计算详细描述。

9.3.9.1 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 TIMx_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。

注：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。

9.3.10 使用刹车功能

当使用刹车功能时，依据相应的控制位 (TIMx_BDTR 寄存器中的 MOE 、OSSI 和 OSSR 位，TIMx_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。详见寄存器表中带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。

刹车源既可以是刹车输入管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 TIMx_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 可以被同时修改。

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 TIMx_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。
- 当 MOE = 0 时，每一个输出通道输出的电平由 TIMx_CR2 寄存器中的 OISx 位决定。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，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；或当 CCxE 与 CCxNE 之一变高时，使能输出变为高。
- 如果设置了 TIMx_DIER 寄存器中的 BIE 位，当刹车状态标志 (TIMx_SR 寄存器中的 BIF 位) 为‘1’时，则产生一个中断。
- 如果设置了 TIMx_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。

注：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 TIMx_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 TIMx_BDTR 寄存器中的 LOCK 位，从三级保护中选择一种，参见寄存器章节详细描述。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：

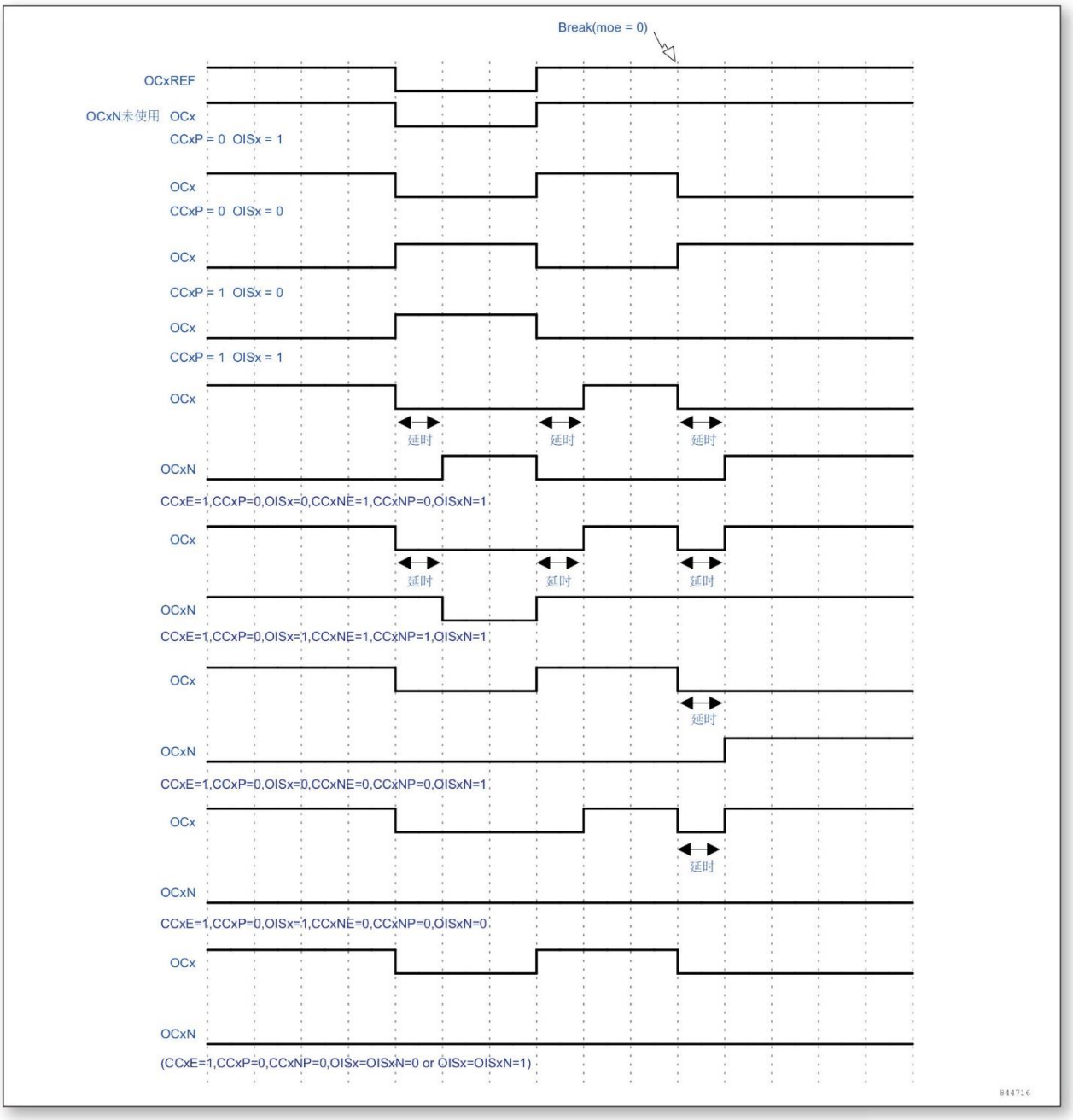

图 9-33 响应刹车的输出

9.3.11 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。这样你就可以预先设置好下一步骤配置，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 TIMx_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 (TIMx_SR 寄存器中的 COMIF 位)，这时如果已设置了

TIMx_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。

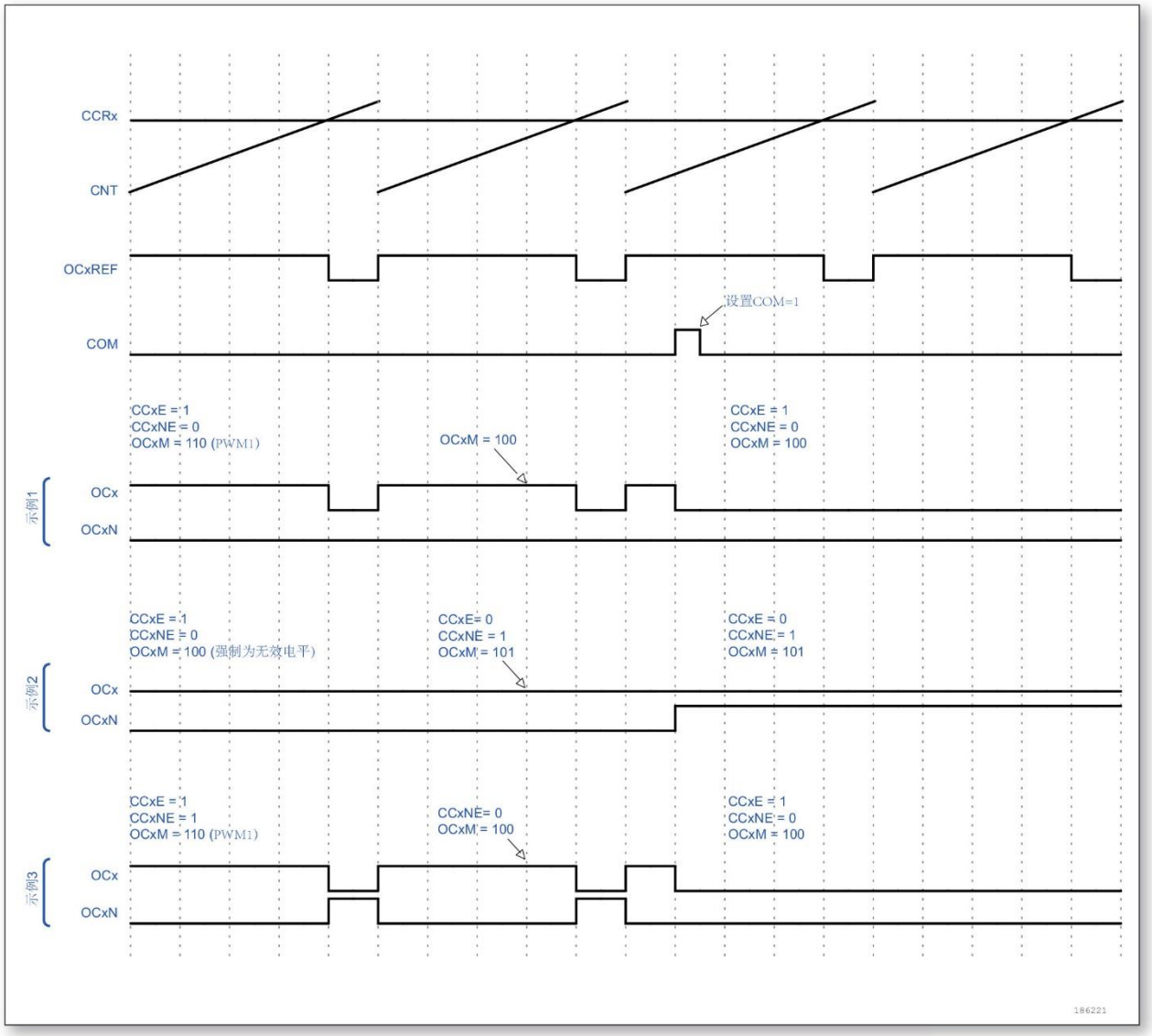

图 9-34 产生六步 PWM，使用 COM 的例子 (OSSR = 1)

9.3.12 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 TIMx_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

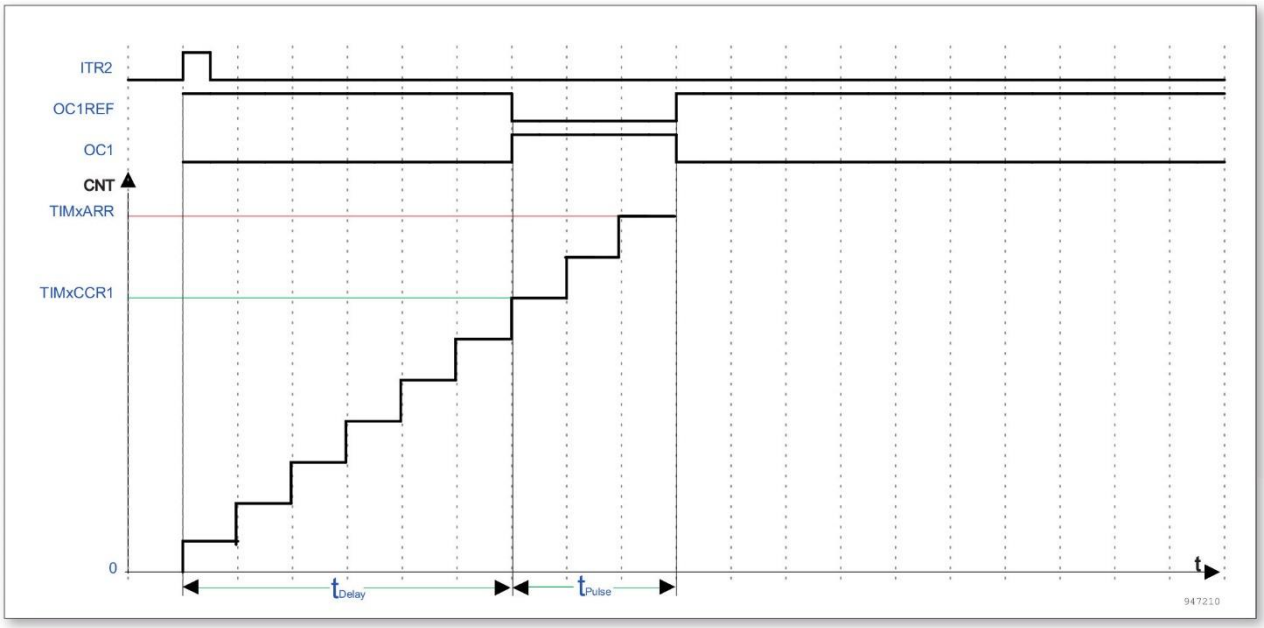

图 9-35 单脉冲模式的例子

例如，在 ITR2 检测到上升沿，延迟 tDELAY 之后，在 OC2 上产生一个长度为 tPULSE 的正脉冲。

- 配置 ITR2 作为触发源。
- 配置 TIMx_SMCR 寄存器中的 TS =010，ITR2 作为从模式控制器的触发 (TRGI)。
- 配置 TIMx_SMCR 寄存器中的 SMS = 110，选择触发模式，ITR1 使能计数器工作。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)。
- tDELAY 由 TIMx_CCR1 寄存器中的值定义。
- tPULSE 由自动装载值和比较值之间的差值定义 (TIMx_ARR-TIMx_CCR1)。

例如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 TIMx_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 TIMx_CCMR1 中的 OC1PE = 1 和 TIMx_CR1 寄存器中的 ARPE；然后在 TIMx_CCR1 寄存器中填写比较值，在 TIMx_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 ITR2 上的一个外部触发事件。本例中，CC1P = 1。

在这个例子中，TIMx_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 TIMx_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，如果要以最小延时输出波形，可以设置 TIMx_CCMRx 寄存器中的 OCxFE 位；此时强制 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。

9.3.13 定时器同步

TIMx 定时器能够在多种模式下和内部触发同步：复位模式、门控模式和触发模式。

9.3.13.1 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，此时如果 TIMx_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 (TIMx_ARR， TIMx_CCRx) 都被更新了。

例如，ITR2 触发计数器重启：

- 配置 TIMx_SMCR 寄存器的 SMS = 100，从模式选择复位模式；配置 TIMx_SMCR 寄存器的 TS = 010，选择 ITR2 作为同步计数器的触发输入。
- 配置 TIMx_CR1 寄存器的 DIR=0，选择计数方向为递增计数；配置 PSC=0，不分频；配置 CEN=1，使能计数器。

计数器开始依据内部时钟计数，然后正常运转直到 ITR2 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 (TIMx_SR 寄存器中的 TIF 位) 被设置，根据 TIMx_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 TIMx_ARR = 0x36 时的动作。

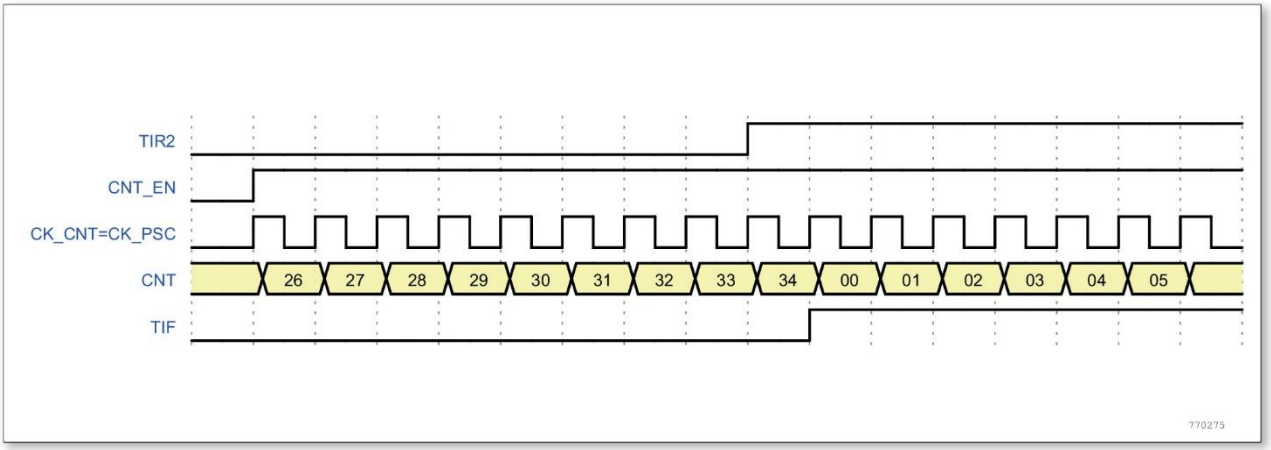

图 9-36 复位模式下的控制电路

9.3.13.2 从模式：门控模式

计数器的使能依赖于选中的输入端的电平。例如，计数器只在 ITR2 为低时计数：

- 配置 TIMx_SMCR 寄存器的 SMS=101，从模式选择为门控模式；配置 TIMx_SMCR 寄存器的

TS=010，选择 ITR2 作为同步计数器的触发输入。

- 配置 TIMx_CR1 寄存器的 DIR=0，选择计数方向为递增计数；配置 PSC=0，不分频；配置 CEN=1，使能计数器。

只要 ITR2 为低，计数器开始依据内部时钟计数，当 ITR2 变高时停止计数。当计数器开始或停止时都设置 TIMx_SR 中的 TIF 标置。

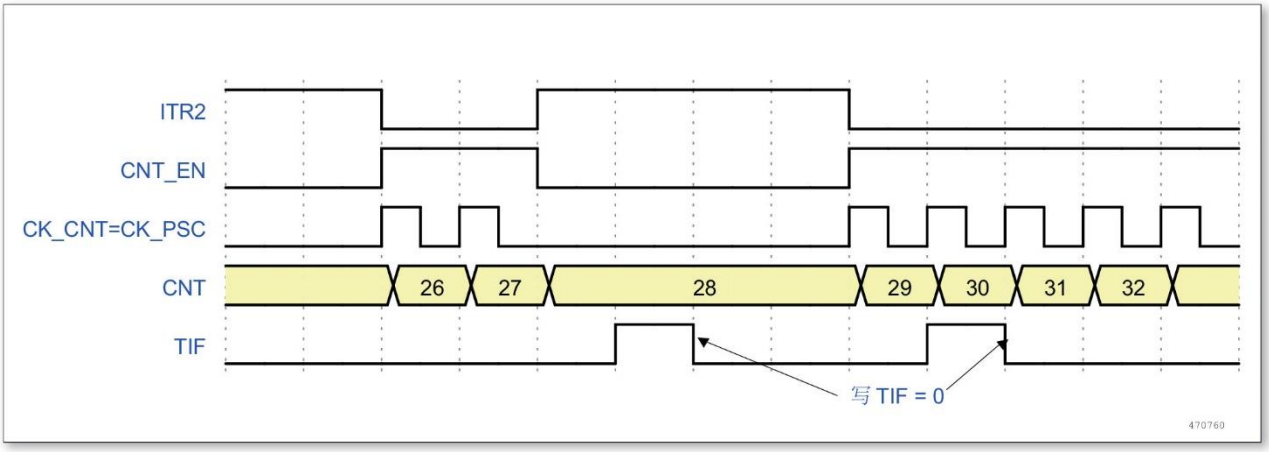

图 9-37 门控模式下的控制电路

9.3.13.3 从模式：触发模式

计数器的使能依赖于选中的输入端上的事件。例如，计数器在 ITR2 的上升沿开始计数：

- 配置 TIMx_SMCR 寄存器的 SMS = 110，从模式选择为触发模式; 配置 TIMx_SMCR 寄存器的 TS=010，选择 ITR2 作为计数器的触发输入。
- 配置 TIMx_CR1 寄存器的 DIR=0，选择计数方向为递增计数；配置 PSC=0，不分频。当 ITR2 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。

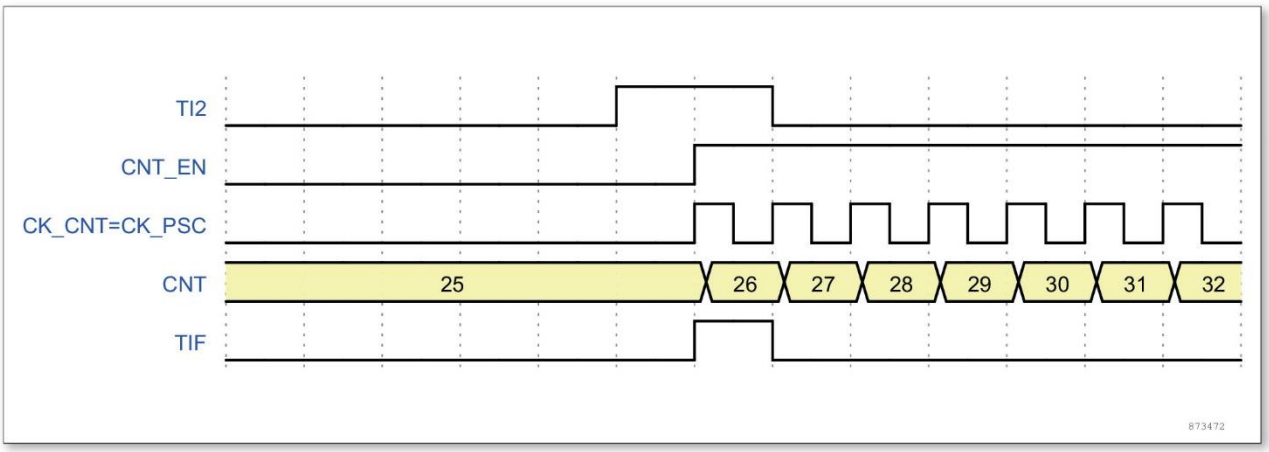

图 9-38 触发器模式下的控制电路

9.3.14 调试模式

当微控制器进入调试模式时 (CPU 核心停止)，根据 DBG 模块中 DBG_TIM1_STOP 的设置，TIMx 计数器可以或者继续正常操作，或者停止。详见随后的调试章节。

9.4 寄存器

表 9-1 TIM1 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	TIM1_CR1	控制寄存器 1	0x0000
0x04	TIM1_CR2	控制寄存器 2	0x0000
0x08	TIM1_SMCR	从模式控制寄存器	0x0000
0x0C	TIM1_DIER	中断使能寄存器	0x0000 0000
0x10	TIM1_SR	状态寄存器	0x0000 0000
0x14	TIM1_EGR	事件产生寄存器	0x0000 0000
0x18	TIM1_CCMR1	比较模式寄存器 1	0x0000
0x1C	TIM1_CCMR2	比较模式寄存器 2	0x0000
0x20	TIM1_CCER	比较使能寄存器	0x0000 0000
0x24	TIM1_CNT	计数器	0x0000
0x28	TIM1_PSC	预分频率器	0x0000
0x2C	TIM1_ARR	自动装载寄存器	0x0000
0x30	TIM1_RCR	重复计数寄存器	0x0000
0x34	TIM1_CCR1	比较寄存器 1	0x0000
0x38	TIM1_CCR2	比较寄存器 2	0x0000
0x3C	TIM1_CCR3	比较寄存器 3	0x0000
0x40	TIM1_CCR4	比较寄存器 4	0x0000
0x44	TIM1_BDTR	刹车和死区寄存器	0x0000 0000
0x54	TIM1_CCMR3	比较模式寄存器 3	0x0000
0x58	TIM1_CCR5	比较寄存器 5	0x0000
0x5C	TIM1_PDER	PWM 移相使能寄存器	0x0000
0x60 ~ 0x70	TIM1_CCRxFALL	PWM 移相递减计数比较寄存器	0x0000

9.4.1 TIM1_CR1 控制寄存器 1

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
Reserved						CKD		ARPE	CMS		DIR	OPM	URS	UDIS	CEN

	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
15：10	Reserved	保留，必须保持复位值。
9：8	CKD	时钟分频（clock division） 定义定时器时钟（INT_CK）频率与死区时间计数器所用的时钟之间的分频比例。 00：$t_{DTS} = t_{INT\_CK}$ 01：$t_{DTS} = 2x\ t_{INT\_CK}$ 10：$t_{DTS} = 4x\ t_{INT\_CK}$ 11：保留，不要使用这个配置
7	ARPE	自动重装载预装载使能（Auto-reload preload enable） 0：关闭 TIM1_ARR 寄存器的影子寄存器 1：使能 TIM1_ARR 寄存器的影子寄存器
6：5	CMS	中央对齐模式选择（Center-aligned mode selection） 00：边沿对齐模式。计数方向取决于 DIR 位 01：中央对齐模式 1。计数器交替地递增和递减计数。通道为输出模式，只在计数器递减计数时比较中断标志位被置 1 10：中央对齐模式 2。计数器交替地递增和递减计数。通道为输出模式，只在计数器递增计数时比较中断标志位被置 1 11：中央对齐模式 3。计数器交替地递增和递减计数。通道为输出模式，在计数器递增和递减计数时比较中断标志位均被置 1 注：计数过程中，不允许更改此位。
4	DIR	计数方向（Direction） 0：计数器递增计数 1：计数器递减计数 注：当计数器配置为中央对齐模式时，该位为只读。
3	OPM	单脉冲模式（One pulse mode） 0：禁止单脉冲模式，在发生更新事件时，计数器继续计数 1：使能单脉冲模式，在发生下一次更新事件或软件清除 CEN 位时，计数器停止计数
2	URS	更新请求源（Update request source） 软件配置该位，选择更新事件源。 0：以下事件可产生一个更新中断： - 计数器上溢/下溢 - 设置 UG 位 - 从模式控制器产生的更新 1：只有计数器上溢/下溢才产生一个更新中断

Bit	Field	Description
1	UDIS	禁止更新（Update disable） 该位用来允许或禁止更新事件的产生 0：允许更新事件（UEV） 1：禁止更新事件。不产生更新事件，影子寄存器（ARR、PSC、CCRx）保持值不变。如果设置了 EGR_UG 位为 1，计数器和预分频器被初始化，如果从模式控制器接收到硬件复位，计数器将被初始化。
0	CEN	计数器使能（Counter enable） 0：禁止计数器 1：使能计数器 注：在软件设置了 CEN 位后，外部时钟、门控模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

9.4.2 TIM1_CR2 控制寄存器 2

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	Res.	MMS			Res.	CCUS	Res.	CCPC
	rw	rw	rw	rw	rw	rw	rw		rw				rw		rw

Bit	Field	Description
15	Reserved	保留，必须保持复位值。
14	OIS4	输出空闲状态 4（OC4 输出）。参见 OIS1 位。
13	OIS3N	输出空闲状态 3（OC3N 输出）。参见 OIS1N 位。
12	OIS3	输出空闲状态 3（OC3 输出）。参见 OIS1 位。
11	OIS2N	输出空闲状态 2（OC2N 输出）。参见 OIS1N 位。
10	OIS2	输出空闲状态 2（OC2 输出）。参见 OIS1 位。
9	OIS1N	输出空闲状态 1（OC1N 输出）（Output Idle state 1） 0：当 MOE = 0 时，死区后 OC1N = 0 1：当 MOE = 0 时，死区后 OC1N = 1 注：已经设置了 LOCK（TIM1_BKR 寄存器）级别 1、2 或 3 后，该位不能被修改。
8	OIS1	输出空闲状态 1（OC1 输出）（Output Idle state 1） 0：当 MOE=0，若 OC1N 有效，则在死区时间后 OC1 = 0 1：当 MOE=0，若 OC1N 有效，则在死区时间后 OC1 = 1 注：已经设置了 LOCK（TIM1_BKR 寄存器）级别 1、2 或 3 后，该位不能被修改。
7	Reserved	保留，必须保持复位值。

Bit	Field	Description
6：4	MMS	主模式选择（Master mode selection） 这些位控制 TRGO 信号的选择，用于选择在主模式下送到从定时器的同步信息： 000：复位　TIM1_EGR 寄存器的 UG 位触发一次 TRGO 脉冲。 001：使能 用于控制在一定时间内使能从定时器或同时启动多个定时器。计数器使能信号 CNT_EN 被用于作为触发输出（TRGO），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 TRGO。 011：比较脉冲　发生一次比较成功时，触发输出送出一个 TRGO 信号。 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）
3	Reserved	保留，必须保持复位值。
2	CCUS	比较控制更新源选择（compare control update selection） 0：CCPC=1 时，只能配置 COMG=1 更新。 1：CCPC=1 时，可以通过配置 COMG=1 或检测到 TRGI 上的一个上升沿更新。 注：此位只在通道为互补输出时有效。
1	Reserved	保留，必须保持复位值。
0	CCPC	比较预装载控制位（compare preloaded control） 0：CCxE，CCxNE 和 OCxM 位预装载禁用 1：CCxE，CCxNE 和 OCxM 位预装载使能 注：此位只在通道为互补输出时有效。

9.4.3 TIM1_SMCR 从模式控制寄存器

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
Reserved								MSM	TS			Res.	SMS		
								rw	rw				rw		

Bit	Field	Description
15：8	Reserved	保留，必须保持复位值。

Bit	Field	Description
7	MSM	主/从 模 式（Master/slave mode） 0：无作用 1：触发输入（TRGI） 事件被延迟，以实现当前定时器（通过 TRGO） 与它的从定时器间的完美同步，该功能可以把几个定时器同步到一个单一的外部事件。
6：4	TS	触发选择（Trigger selection） 触发输入源选择。 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：保留 101：保留 110：保留 111：保留 注：从模式使能后这些位不能修改。
3	Reserved	保留，必须保持复位值。
2：0	SMS	从模式选择（Slave mode selection） 当选择了外部信号作为触发源，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：保留 010：保留 011：保留 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，并且产生一个更新事件。 101：门控模式 - 当触发输入（TRGI） 为高时，计数器开始计数。当触发输入变为低时，计数器停止计数（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 9-2 TIMx 内部触发连接

从定时器	ITR0	ITR1	ITR2	ITR3
TIM1	-	-	TIM3	-
TIM3	TIM1	-	-	-

9.4.4 TIM1_DIER 中断使能寄存器

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
Reserved															CC5IE
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31：17	Reserved	保留，必须保持复位值。
16	CC5IE	允许比较 5 中断（Compare 5 interrupt enable） 0：禁止比较 5 中断 1：允许比较 5 中断
15：8	Reserved	保留，必须保持复位值。
7	BIE	允许刹车中断（Break interrupt enable） 0：禁止刹车中断 1：允许刹车中断
6	TIE	允许触发中断（Trigger interrupt enable） 0：禁止触发中断 1：允许触发中断
5	COMIE	允许 COM 中断（COM interrupt enable） 0：禁止 COM 中断 1：允许 COM 中断
4	CC4IE	允许比较 4 中断（Compare 4 interrupt enable） 0：禁止比较 4 中断 1：允许比较 4 中断
3	CC3IE	允许比较 3 中断（Compare 3 interrupt enable） 0：禁止比较 3 中断 1：允许比较 3 中断
2	CC2IE	允许比较 2 中断（Compare 2 interrupt enable） 0：禁止比较 2 中断 1：允许比较 2 中断
1	CC1IE	允许比较 1 中断（Compare 1 interrupt enable） 0：禁止比较 1 中断 1：允许比较 1 中断
0	UIE	允许更新事件中断（Update interrupt enable） 0：禁止更新事件中断 1：允许更新事件中断

9.4.5 TIM1_SR 状态寄存器

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
Reserved															CC5IF
															rw0c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
								r_w0c							

Bit	Field	Description
31：17	Reserved	保留，必须保持复位值。
16	CC5IF	比较 5 中断标记（Compare 5 interrupt flag） 参考 CC1IF 描述。
15：8	Reserved	保留，必须保持复位值。
7	BIF	刹车中断标记（Break interrupt flag） 当刹车输入有效，由硬件对该位置 1。如果刹车输入无效，则该位可由软件清 0 0：无刹车事件产生 1：刹车输入上检测到有效电平
6	TIF	触发器中断标记（Trigger 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。它由软件清 0。 0：无触发器事件产生 1：触发器中断产生
5	COMIF	COM 中断标记（COM interrupt flag） 当产生 COM 事件（比较控制位 CCxE、CCxNE、OCxM 已被更新） 时该位由硬件置 1。它由软件清 0。 0：无 COM 事件产生 1：COM 中断产生
4	CC4IF	比较 4 中断标记（Compare 4 interrupt flag） 参考 CC1IF 描述。
3	CC3IF	比较 3 中断标记（Compare 3 interrupt flag） 参考 CC1IF 描述。
2	CC2IF	比较 2 中断标记（Compare 2 interrupt flag） 参考 CC1IF 描述。
1	CC1IF	比较 1 中断标记（Compare 1 interrupt flag） 当计数器值与比较值匹配时该位由硬件置 1（在中央对齐模式下根据 TIM1_CR1.CMS[1:0]的选择来置位）。它由软件清 0。 0：无匹配发生 1：TIM1_CNT 的值与 TIM1_CCR1 的值匹配

Bit	Field	Description
0	UIF	更新中断标记（Update interrupt flag） 当产生更新事件时该位由硬件置 1。它由软件清 0。 0：无更新中断发生 1：发生更新中断 当寄存器被更新时该位由硬件置 1： - 若 TIM1_CR1 寄存器的 UDIS=0，且 REP_CNT=0，当计数器产生上溢/下溢事件时。 -若 TIM1_CR1 寄存器的 UDIS=0、URS=0，当 TIM1_EGR 寄存器的 UG=1 时。 - 若 TIM1_CR1 寄存器的 UDIS=0、URS=0，从模式控制器产生更新事件时。

9.4.6 TIM1_EGR 事件产生寄存器

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
Reserved															CC5G
															w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
								w	w	w	w	w	w	w	w

Bit	Field	Description
31：17	Reserved	保留，必须保持复位值。
16	CC5G	产生比较 5 事件（Compare 5 generation） 参考 CC1G 描述。
15：8	Reserved	保留，必须保持复位值。
7	BG	产生刹车事件（Break generation） 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，由硬件清除。
6	TG	产生触发事件（Trigger generation） 0：无动作 1：产生触发事件，TIM1_SR 寄存器的 TIF = 1，若开启对应的中断， 则产生相应的中断，由硬件自动清 0。

Bit	Field	Description
5	COMG	比较事件，产生控制更新（Compare control update generation） 0：无动作 1：比较事件控制更新产生，由硬件自动清 0，当 CCPC=1，允许更新 CCxE、CCxNE、OCxM 位。 注：该位只对拥有互补输出的通道有效。
4	CC4G	产生比较 4 事件（Compare 4 generation） 参考 CC1G 描述。
3	CC3G	产生比较 3 事件（Compare 3 generation） 参考 CC1G 描述。
2	CC2G	产生比较 2 事件（Compare 2 generation） 参考 CC1G 描述。
1	CC1G	产生通道 1 比较事件（Compare 1 generation） 该位由软件置 1，用于产生一个比较事件，由硬件自动清 0。 0：无动作 1：通道 CC1 上产生一个比较事件： CC1IF 置 1，若开启对应的中断，则产生相应的中断。
0	UG	产生更新事件（Update generation） 0：无动作 1：初始化计数器，并产生一个更新事件。由硬件自动清 0，如果选择了中央对齐或递增计数模式，计数器被清 0；如果选择递减计数模式，计数器将载入自动重载值。预分频计数器将同时被清除。

9.4.7 TIM1_CCMR1 比较模式寄存器 1

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
Res.	OC2M			OC2PE	OC2FE	Reserved			OC1M			OC1PE	OC1FE	Reserved	
	rw			rw	rw				rw			rw	rw		

Bit	Field	Description
15	Reserved	保留，必须保持复位值。
14：12	OC2M	通道 2 比较输出模式（Output compare 2 mode） 参考 OC1M 的描述。
11	OC2PE	通道 2 比较输出预装载使能（Output compare 2 preload enable） 参考 OC1PE 的描述。
10	OC2FE	通道 2 比较输出快速使能（Output compare 2 fast enable） 参考 OC1FE 的描述。

Bit	Field	Description
9：7	Reserved	保留，必须保持复位值。
6：4	OC1M	通道 1 比较输出模式（Output compare 1 mode） 该位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效，而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 000：冻结。TIM1_CCR1 与 TIM1_CNT 间的比较结果对 OC1REF 不起作用。 001：匹配时设置为高。当 TIM1_CNT 的值与 TIM1_CCR1 的值相同时，强制 OC1REF 为高电平。 010：匹配时设置为低。当 TIM1_CNT 的值与 TIM1_CCR1 的值相同时，强制 OC1REF 为低电平。 011：匹配时翻转。当 TIM1_CCR1=TIM1_CNT 时，翻转 OC1REF 的电平。 100：强制为低。强制 OC1REF 为低电平。 101：强制为高。强制 OC1REF 为高电平。 110：PWM 模式 1。在递增计数时，当 TIM1_CNT<TIM1_CCR1 时强制 OC1REF 为高电平，否则为低电平;在递减计数时，当 TIM1_CNT > TIM1_CCR1 时强制 OC1REF 为低电平，否则为高电平。 111：PWM 模式 2。在递增计数时，当 TIM1_CNT<TIM1_CCR1 时通道 1 为强制 OC1REF 为低电平，否则为高电平;在递减计数时，当 TIM1_CNT > TIM1_CCR1 时强制 OC1REF 为高电平，否则为低电平。 注 1：当 LOCK 级别设为 3（TIM1_BDTR 寄存器中的 LOCK 位）时，该位不能被修改。 注 2：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。
3	OC1PE	通道 1 比较输出预装载使能（Output compare 1 preload enable） 0：禁止 TIM1_CCR1 寄存器的预装载功能，写入 TIM1_CCR1 寄存器的数值立即生效。 1：开启 TIM1_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIM1_CCR1 的预装载值在更新事件到来时生效。 注 1：当 LOCK 级别设为 3（TIM1_BDTR 寄存器中的 LOCK 位）并且时，该位不能被修改。 注 2：若该位置 1，在单脉冲模式下（TIM1_CR1 寄存器的 OPM= 1），是否设定预装载寄存器无影响；其它情况下，需要设定预装载寄存器，否则后续动作不确定。
2	OC1FE	通道 1 比较输出快速使能（Output compare 1 fast enable） 该位为 1 时，若通道配置为 PWM 模式，会加快比较输出对触发时间的响应。输出通道将触发输入信号的有效边沿的作用等同于发生了一次比较匹配，此时 OC 被设置为比较电平，与比较结果无关。 0：禁止通道 1 比较输出快速使能 1：开启通道 1 比较输出快速使能
1：0	Reserved	保留，必须保持复位值。

9.4.8 TIM1_CCMR2 比较模式寄存器 2

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
Res.	OC4M			OC4PE	OC4FE	Reserved			OC3M			OC3PE	OC3FE	Reserved	
	rw			rw	rw				rw			rw	rw		

Bit	Field	Description
15	Reserved	保留，必须保持复位值。
14：12	OC4M	通道 4 比较输出模式（Output compare 4 mode） 参考 OC3M 的描述
11	OC4PE	通道 4 比较输出预装载使能（Output compare 4 preload enable） 参考 OC3PE 的描述
10	OC4FE	通道 4 比较输出快速使能（Output compare 4 fast enable） 参考 OC3FE 的描述
9：7	Reserved	保留，必须保持复位值。
6：4	OC3M	通道 3 比较输出模式（Output compare 3 mode） 该位定义了输出参考信号 OC3REF 的动作，而 OC3REF 决定了 OC3、OC3N 的值。OC3REF 是高电平有效，而 OC3、OC3N 的有效电平取决于 CC3P、CC3NP 位。 000：冻结。TIM1_CCR3 与 TIM1_CNT 间的比较结果对 OC3REF 不起作用 001：匹配时设置为高。当 TIM1_CNT 的值与 TIM1_CCR3 的值相同时，强制 OC3REF 为高电平 010：匹配时设置为低。当 TIM1_CNT 的值与 TIM1_CCR3 的值相同时，强制 OC3REF 为低电平 011：匹配时翻转。当 TIM1_CCR3=TIM1_CNT 时，翻转 OC3REF 的电平 100：强制为低。强制 OC3REF 为低电平 101：强制为高。强制 OC3REF 为高电平 110：PWM 模式 1。在递增计数时，当 TIM1_CNT<TIM1_CCR3 时强制 OC3REF 为高电平，否则为低电平;在递减计数时，当 TIM1_CNT > TIM1_CCR3 时强制 OC3REF 为低电平，否则为高电平。 111：PWM 模式 2。在递增计数时，当 TIM1_CNT<TIM1_CCR3 时强制 OC3REF 为低电平，否则为高电平;在递减计数时，当 TIM1_CNT>TIM1_CCR3 时强制 OC3REF 为高电平，否则为低电平。 注 1：当 LOCK 级别设为 3（TIM1_BDTR 寄存器中的 LOCK 位）时，该位不能被修改。 注 2：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时，OC3REF 电平才改变。

Bit	Field	Description
3	OC3PE	通道 3 比较输出预装载使能（Output compare 3 preload enable） 0：禁止 TIM1_CCR3 寄存器的预装载功能，写入 TIM1_CCR3 寄存器的数值立即生效 1：开启 TIM1_CCR3 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIM1_CCR3 的预装载值在更新事件到来时生效 注 1：当 LOCK 级别设为 3（TIM1_BDTR 寄存器中的 LOCK 位）时，该位不能被修改。 注 2：仅在单脉冲模式下（TIM1_CR1 寄存器的 OPM= 1），无需设定预装载寄存器，其它情况下，需要设定预装载寄存器，否则后续动作不确定。
2	OC3FE	通道 3 比较输出快速使能（Output compare 3 fast enable） 该位为 1 时，若通道配置为 PWM 模式，会加快比较输出对触发时间的响应。输出通道将触发输入信号的有效边沿的作用等同于发生了一次比较匹配，此时 OC 被设置为比较电平，与比较结果无关。 0：禁止通道 3 比较输出快速使能 1：开启通道 3 比较输出快速使能
1：0	Reserved	保留，必须保持复位值。

9.4.9 TIM1_CCER 比较使能寄存器

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
Reserved		CC 4P	CC 4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
		rw													

Bit	Field	Description
15：14	Reserved	保留，必须保持复位值。
13	CC4P	通道 4 输出极性（Compare 4 output polarity） 参考 CC1P 的描述。
12	CC4E	通道 4 输出使能（Compare 4 output enable） 参考 CC1E 的描述。
11	CC3NP	通道 3 互补输出极性（Compare 3 complementary output polarity） 参考 CC1NP 的描述。
10	CC3NE	通道 3 互补输出使能（Compare 3 complementary output enable） 参考 CC1NE 的描述。
9	CC3P	通道 3 输出极性（Compare 3 output polarity） 参考 CC1P 的描述。
8	CC3E	通道 3 输出使能（Compare 3 output enable） 参考 CC1E 的描述。

Bit	Field	Description
7	CC2NP	通道 2 互补输出极性（Compare 2 complementary output polarity） 参考 CC1NP 的描述。
6	CC2NE	通道 2 互补输出使能（Compare 2 complementary output enable） 参考 CC1NE 的描述。
5	CC2P	通道 2 输出极性（Compare 2 output polarity） 参考 CC1P 的描述。
4	CC2E	通道 2 输出使能（Compare 2 output enable） 参考 CC1E 的描述。
3	CC1NP	通道 1 互补输出极性（Compare 1 complementary output polarity） 此位定义了输出信号极性： 0：OC1N 高电平有效 1：OC1N 低电平有效 注：当 LOCK 级别（TIM1_BDTR 寄存器中的 LCCK 位） 设为 3 或 2 时，该位不能被修改。
2	CC1NE	通道 1 互补输出使能（Compare 1 complementary output enable） 0：关闭通道 1 互补输出。OC1N 禁止输出。 1：开启通道 1 互补输出。 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。
1	CC1P	通道 1 输出极性（Compare 1 output polarity） 此位定义了输出信号极性： 0：OC1 高电平有效 1：OC1 低电平有效 注：当 LOCK 级别（TIM1_BDTR 寄存器中的 LCCK 位） 设为 3 或 2 时，该位不能被修改。
0	CC1E	通道 1 输出使能（Compare 1 output enable） 0：关闭。OC1 禁止输出 1：开启。OC1 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 CC1NE 位的值。

表 9-3 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	
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

控制位					输出状态	
		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
		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		

注 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。

注 2：与互补通道 OCx 和 OCxN 相连的外部 I/O 管脚的状态，取决于 OCx 和 OCxN 通道状态和 GPIO 寄存器。

9.4.10 TIM1_CNT 计数器

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															
rw															

Bit	Field	Description
15：0	CNT	计数器的值（Count value）

9.4.11 TIM1_PSC 预分频器

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															
rw															

Bit	Field	Description
15：0	PSC	预分频器的值（Prescaler value） 计数器的时钟频率（ck_cnt）=f_{CK_PSC}/（PSC+1） 当发生更新事件时，PSC 的值装入当前预分频寄存器。

9.4.12 TIM1_ARR 自动预装载寄存器

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															
rw															

Bit	Field	Description
15：0	ARR	自动预装载值（Auto-reload value） 这些位定义了计数器的自动预装载值。当自动预装载的值为 0 时，计数器不工作。

9.4.13 TIM1_RCR 重复计数寄存器

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
REP_CNT								REP							
rw								rw							

Bit	Field	Description
15：8	REP_CNT	重复计数器实时写入的值（Repetition counter value of real-time writing） 在重复计数模式下，写入该位可以实时地将更新中断标志位（UIF）的检测点移位。 注：在更新事件后写入该位，在更新事件前写入 REP_CNT 将会被 REP 的值覆盖，使移位无效。
7：0	REP	重复计数器的值（Repetition counter value） 重复计数器的值定义了更新事件的产生速率。重复计数器计数值递减为 0 时产生更新事件。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。 对 REP 值的写入在下次更新事件发生时生效，所以在 PWM 模式中，（REP+1） 对应着： 在边沿对齐模式下，PWM 周期的数目 在中央对齐模式下，PWM 半周期的数目

9.4.14 TIM1_CCR1 比较寄存器 1

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															
rw															

Bit	Field	Description
15：0	CCR1	通道 1 比较的值（ompare 1 value） 如果在 TIM1_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对应的当前比较影子寄存器中。否则只有当更新事件发生时，此预装载值才传输至对应的当前比较影子寄存器中。当前比较影子寄存器参与同计数器 TIM1_CNT 的比较，并将比较结果反映到 OC1 端口的输出信号上。

9.4.15 TIM1_CCR2 比较寄存器 2

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
rw

Bit	Field	Description
15：0	CCR2	通道 2 比较的值（Compare 2 value） 参考 CCR1 的描述。

9.4.16 TIM1_CCR3 比较寄存器 3

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															
rw															

Bit	Field	Description
15：0	CCR3	通道 3 比较的值（Compare 3 value） 参考 CCR1 的描述。

9.4.17 TIM1_CCR4 比较寄存器 4

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															
rw															

Bit	Field	Description
15：0	CCR4	通道 4 比较的值（Compare 4 value） 参考 CCR1 的描述。

9.4.18 TIM1_BDTR 刹车和死区寄存器

偏移地址：0x44

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK		DTG							
rw	rw	rw	rw	rw	rw	rw		rw							

注：根据锁定设置，DOE、AOE、BKP、BKE、OSSI、OSSR 和 DTG 位均可被写保护，有必要在第一次写入 TIM1_BDTR 寄存器时对它们进行配置，详见互补输出和死区插入章节。

Bit	Field	Description
31：17	Reserved	保留，必须保持复位值。
16	DOE	直接输出（Direct output enable） 当刹车有效、MOE 置零后，有效。 0：刹车输入后，等待一个死区时间后输出空闲状态（输出使能信号关闭） 1：立即输出空闲状态（输出使能信号关闭） 注：当 LOCK 级别（TIM1_BDTR 寄存器中的 LOCK 位）设为 1 时，该位不能被修改。
15	MOE	主输出使能（Main output enable） 当通道 x 配置为输出时，根据 AOE 位的设置值，该位可以由软件清 0 或被自动置 1。当刹车输入有效时，该位被硬件异步清 0。 0：禁止 OCx 和 OCxN 输出或强制为空闲状态（输出使能信号关闭） 1：如果设置了相应的使能位（TIM1_CCER 寄存器的 CCxE、CCxNE 位），则开启 OCx 和 OCxN 输出
14	AOE	自动输出使能（AutoMatic output enable） 0：MOE 不能被硬件置 1 1：MOE 能被软件置 1 或刹车无效时在下一个更新事件被硬件自动置 1 注：当 LOCK 级别（TIM1_BDTR 寄存器中的 LOCK 位）设为 1 时，该位不能被修改。
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 注：当 LOCK 级别（TIM1_BDTR 寄存器中的 LOCK 位）设为 1 时，该位不能被修改。
12	BKE	刹车功能使能（Break enable） 0：禁止刹车输入 1：开启刹车输入 注：当 LOCK 级别（TIM1_BDTR 寄存器中的 LOCK 位）设为 1 时，该位不能被修改。

Bit	Field	Description
11	OSSR	运行模式下“关闭状态”选择（Off-state selection for Run mode） 该位仅适用于当 MOE = 1 且通道为互补输出。 0：当定时器不工作时，禁止 OC/OCN 输出 1：当定时器不工作时，如果 CCxE = 1 或 CCxNE = 1，首先开启 OC/OCN 并输出无效电平，然后置位 OC/OCN 输出使能信号。 注：当 LOCK 级别（TIM1_BDTR 寄存器中的 LOCK 位）设为 2 时，该位不能被修改。
10	OSSI	空闲模式下“关闭状态”选择（Off-state selection for Idle mode） 该位仅适用于当 MOE = 0 且通道设为输出时。 0：当定时器不工作时，禁止 OC/OCN 输出。 1：当定时器不工作时，如果 CCxE = 1 或 CCxNE = 1 ，首先 OC/OCN 输出无效电平，然后置位 OC/OCN 输出使能信号。 注：当 LOCK 级别（TIM1_BDTR 寄存器中的 LOCK 位）设为 2 时，该位不能被修改。
9：8	LOCK	锁定设置（Lock configuration） 该位定义了寄存器的写保护功能。 00：写保护功能关闭，寄存器无写保护 01：锁定级别 1，不能写入 TIM1_BDTR 寄存器的 DOE、DTG、BKE、BKP、AOE 位和 TIM1_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 注：在系统复位后，LOCK 位只能写一次，当写入 TIM1_BDTR 寄存器后，LOCK 被写保护。
7：0	DTG	死区发生器设置（Dead-time generator setup） 这些位定义了插入互补输出之间的死区持续时间。 DTG[7: 5] = 0xx: DT = (DTG[7: 0] + 1) × tdtg, tdtg = tDTS; DTG[7: 5] = 10x: DT = (DTG[5: 0] + 1 + 64) × tdtg, tdtg = 2 × tDTS; DTG[7: 5] = 110: DT = (DTG[4: 0] + 1 + 32) × tdtg, tdtg = 8 × tDTS; DTG[7: 5] = 111: DT = (DTG[4: 0] + 1 + 32) × tdtg, tdtg = 16 × tDTS; 例：若 tDTS = 125ns(8MHz)，可能的死区时间为： 125ns 到 15875ns(步长时间为 125ns)， 16μs 到 31750ns(步长时间为 250ns)， 32μs 到 63μs(步长时间为 1μs)， 64μs 到 126μs(步长时间为 2μs)。 注：当 LOCK 级别（TIM1_BDTR 寄存器中的 LOCK 位） 设为 1、2 或 3 时，不能修改这些位。

9.4.19 TIM1_CCMR3 比较模式寄存器 3

偏移地址：0x54

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												OC5PE	Reserved		
												rw			

Bit	Field	Description
15：4	Reserved	保留，必须保持复位值。
3	OC5PE	比较输出 5 预装载使能（Output compare 5 preload enable） 0：禁止 TIM1_CCR5 寄存器的预装载功能，写入 TIM1_CCR5 寄存器的数值立即生效。 1：开启 TIM1_CCR5 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIM1_CCR5 的预装载值在更新事件到来时生效。 注 1：当 LOCK 级别设为 3（TIM1_BDTR 寄存器中的 LOCK 位）时，该位不能被修改。
2：0	Reserved	保留，必须保持复位值。

9.4.20 TIM1_CCR5 比较寄存器 5

偏移地址：0x58

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR5															
rw															

Bit	Field	Description
15：0	CCR5	比较 5 的值（Compare 5 value） CC5 通道只能配置为输出： 如果在 TIM1_CCMR3 寄存器（OC5PE 位）中未选择预装载功能，写入的数值会立即传输至对应的当前比较影子寄存器中。否则只有当更新事件发生时，此预装载值才传输至对应的当前比较影子寄存器中。当前比较影子寄存器参与同计数器 TIM1_CNT 的比较，由于 CC5 通道为内部通道，无法输出至引脚，比较结果用于内部触发事件。

9.4.21 TIM1_PDER PWM 移相使能寄存器

偏移地址：0x5C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										CCR5_SHIFT_EN	CCR4_SHIFT_EN	CCR3_SHIFT_EN	CCR2_SHIFT_EN	CCR1_SHIFT_EN	Res.
										rw	rw	rw	rw	rw	

Bit	Field	Description
15：6	Reserved	保留，必须保持复位值。
5	CCR5_SHIFT_EN	允许通道 5 输出 PWM 移相使能位 0：禁止通道 5 输出 PWM 移相 1：允许通道 5 输出 PWM 移相 具体见 CCRxFALL 寄存器描述移相操作
4	CCR4_SHIFT_EN	允许通道 4 输出 PWM 移相使能位 0：禁止通道 4 输出 PWM 移相 1：允许通道 4 输出 PWM 移相 具体见 CCRxFALL 寄存器描述移相操作
3	CCR3_SHIFT_EN	允许通道 3 输出 PWM 移相使能位 0：禁止通道 3 输出 PWM 移相 1：允许通道 3 输出 PWM 移相 具体见 CCRxFALL 寄存器描述移相操作
2	CCR2_SHIFT_EN	允许通道 2 输出 PWM 移相使能位 0：禁止通道 2 输出 PWM 移相 1：允许通道 2 输出 PWM 移相 具体见 CCRxFALL 寄存器描述移相操作
1	CCR1_SHIFT_EN	允许通道 1 输出 PWM 移相使能位 0：禁止通道 1 输出 PWM 移相 1：允许通道 1 输出 PWM 移相 具体见 CCRxFALL 寄存器描述移相操作
0	Reserved	保留，必须保持复位值。

9.4.22 TIM1_CCRxFALL PWM 移相递减计数比较寄存器

偏移地址：0x60 ~ 0x70

复位值：0x0000

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

CCRxFALL
rw

Bit	Field	Description
15：0	CCRxFALL	通道 x 在 PWM 中央对齐模式递减计数时的比较值PWM 移相功能：开启 PDER 寄存器的 PWM 移相使能，根据需要移动相位，配置 CCRxFALL 以及 CCRx，即可实现 PWM 输出可编程的移相波形，可左移或是右移。

10 TIM3 16 位通用定时器

10.1 简介

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。

它适合多种用途，能够提供比较输出、 PWM 输出、死区时间可编程的互补输出、单脉冲模式输出等多种功能。

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) 和通用定时器 (TIMx) 是完全独立的，它们不共享任何资源。它们可以同步操作，具体描述参看通用定时器同步的章节。

10.2 主要特征

通用 TIMx(TIM3) 定时器功能包括：

- 16 位向上、向下、向上/向下自动装载计数器
- 16 位可编程 (可以实时修改) 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 个独立的通道
  - 输入捕获
  - 输出比较
  - PWM 生成 (边缘或中间对齐模式)
  - 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
- 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

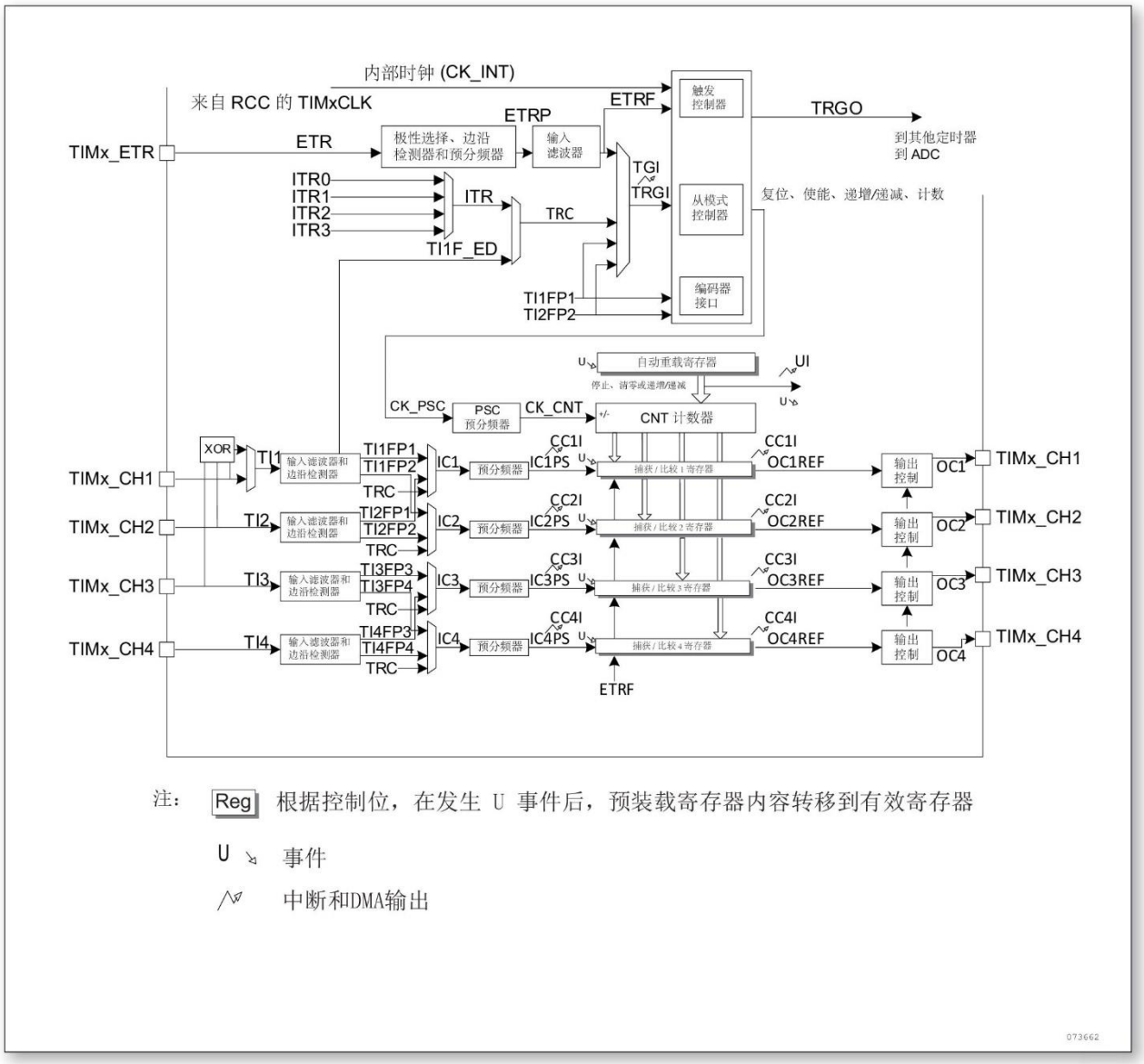

图 10-1 通用定时器框图

10.3 功能描述

10.3.1 时基单元

可编程通用定时器的主要部分是一个 16 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，在计数器运行时仍可以读写，时基单元包含：

- 计数器寄存器 (TIMx_CNT)

- 预分频器寄存器 (TIMx_PSC)
- 自动装载寄存器 (TIMx_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 TIMx_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 TIMx_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 TIMx_CR1 寄存器中的计数器使能位 (CEN) 时，CK_CNT 才有效。 (有关计数器使能的细节，请参见控制器的从模式描述)。

10.3.1.1 预分频器描述

预分频器可以将计数器的时钟频率按 1 ~ 65536 之间的任意值分频。它是基于一个 (在 TIMx_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，它能够在工作时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

下面两个图分别给出了在预分频器运行时，更改计数器参数的例子。

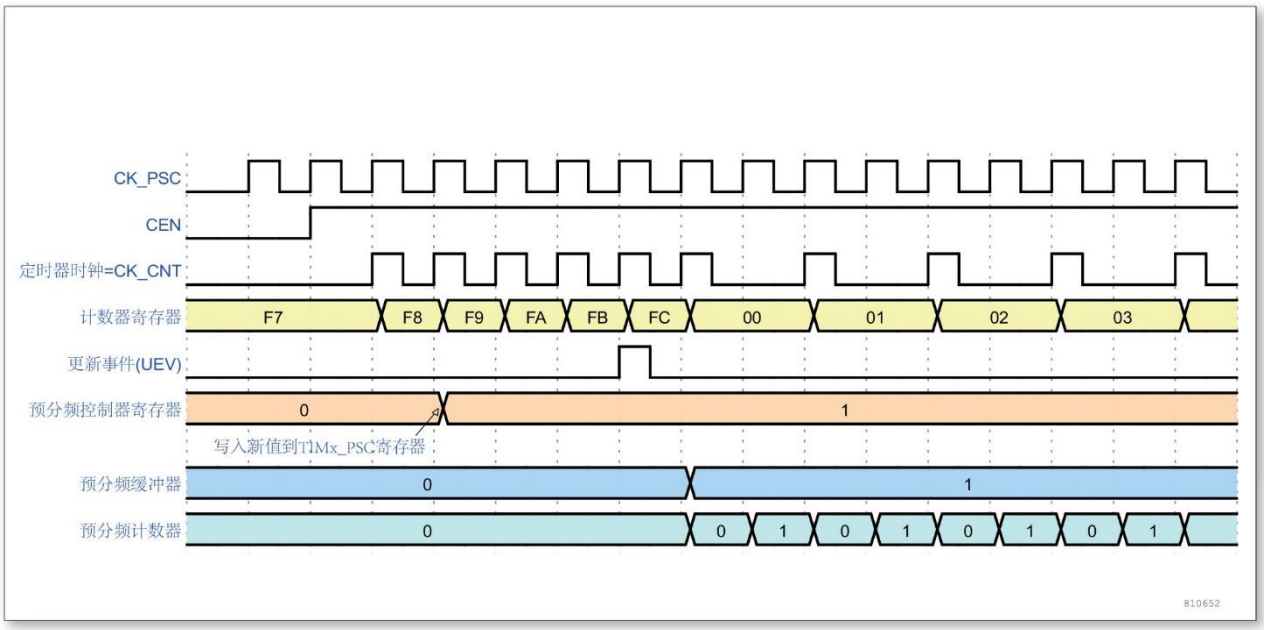

图 10-2 当预分频器的参数从 1 变到 2 时，计数器的时序图

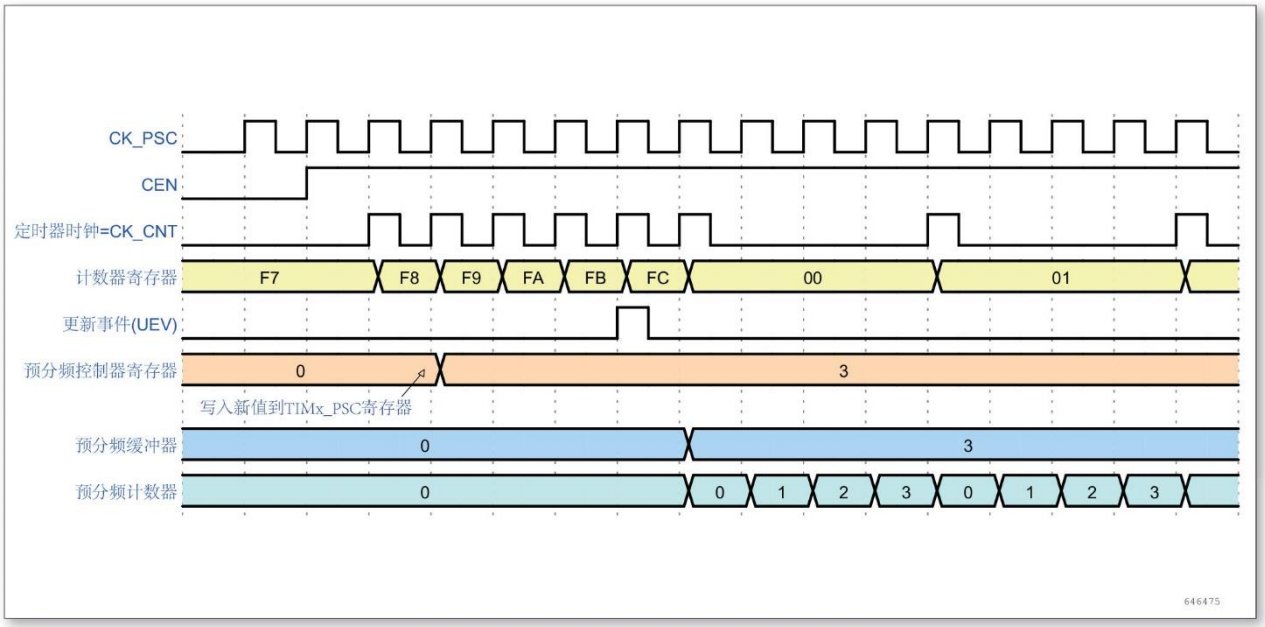

图 10-3 当预分频器的参数从 1 变到 4 时，计数器的时序图

10.3.2 计数模式

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

设置 TIMx_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 0(但预分频器的数值不变)。此外，如果设置了 TIMx_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 (即不产生中断请求)。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 (TIMx_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 (TIMx_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 (TIMx_ARR)

下图给出一些例子，当 TIMx_ARR = 0x36 时计数器在不同时钟频率下的动作：

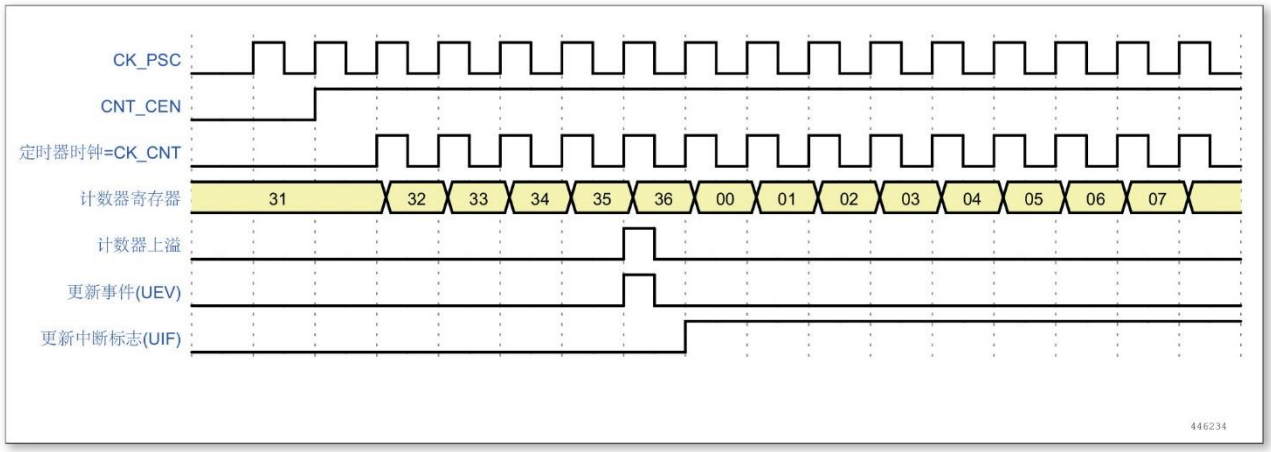

图 10-4 计数器时序图，内部时钟分频因子为 1

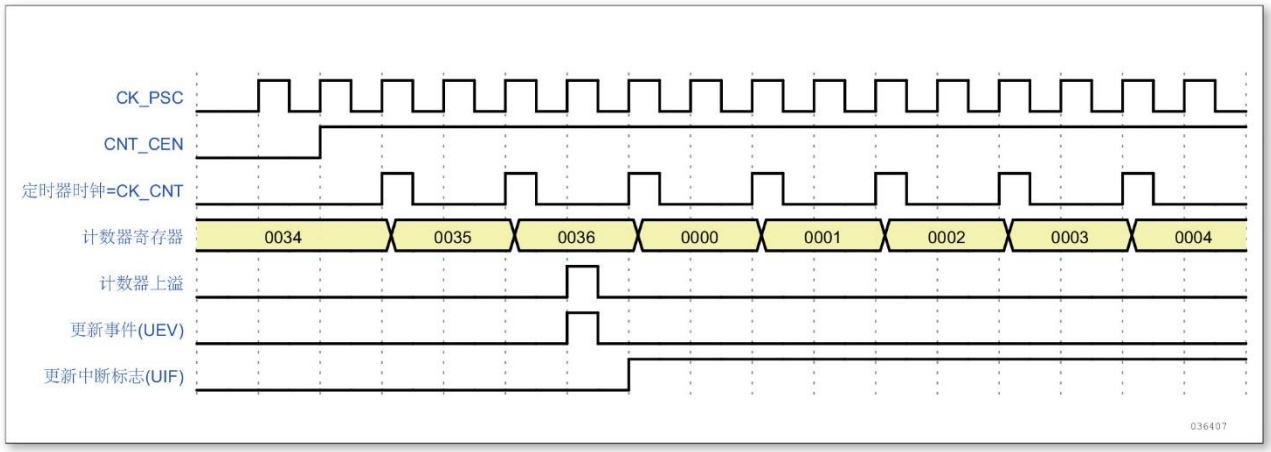

图 10-5 计数器时序图，内部时钟分频因子为 2

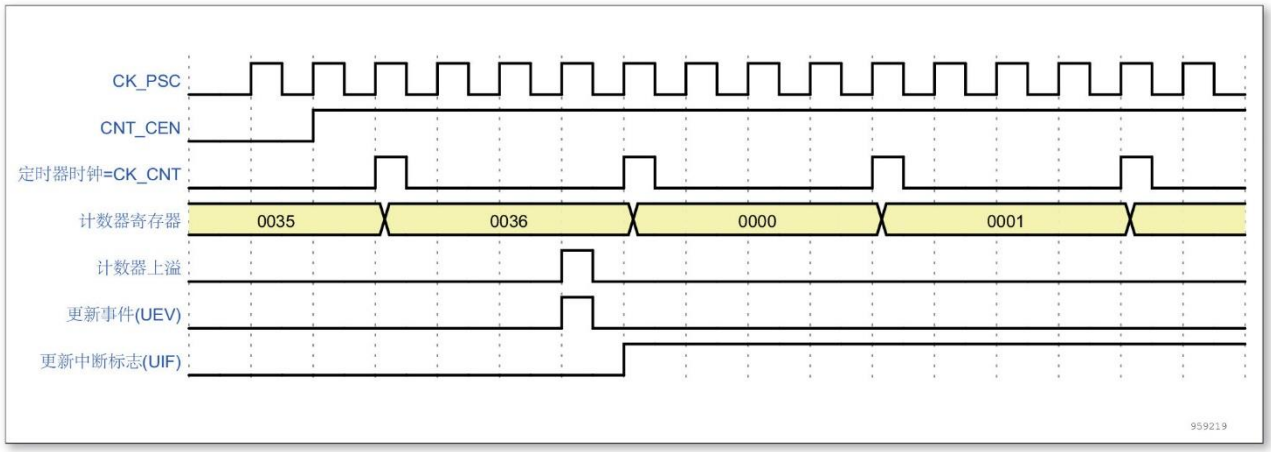

图 10-6 计数器时序图，内部时钟分频因子为 4

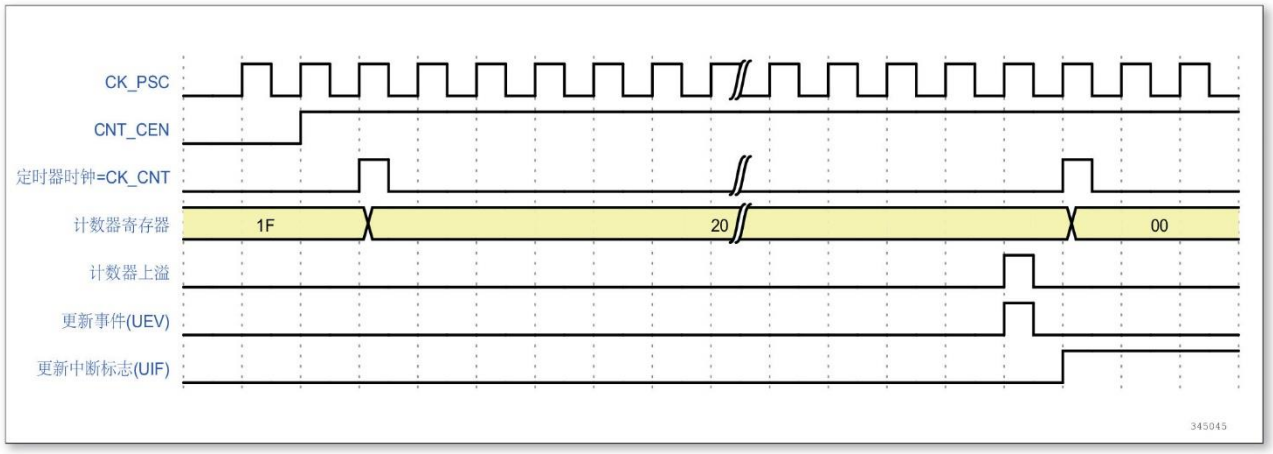

图 10-7 计数器时序图，内部时钟分频因子为 N

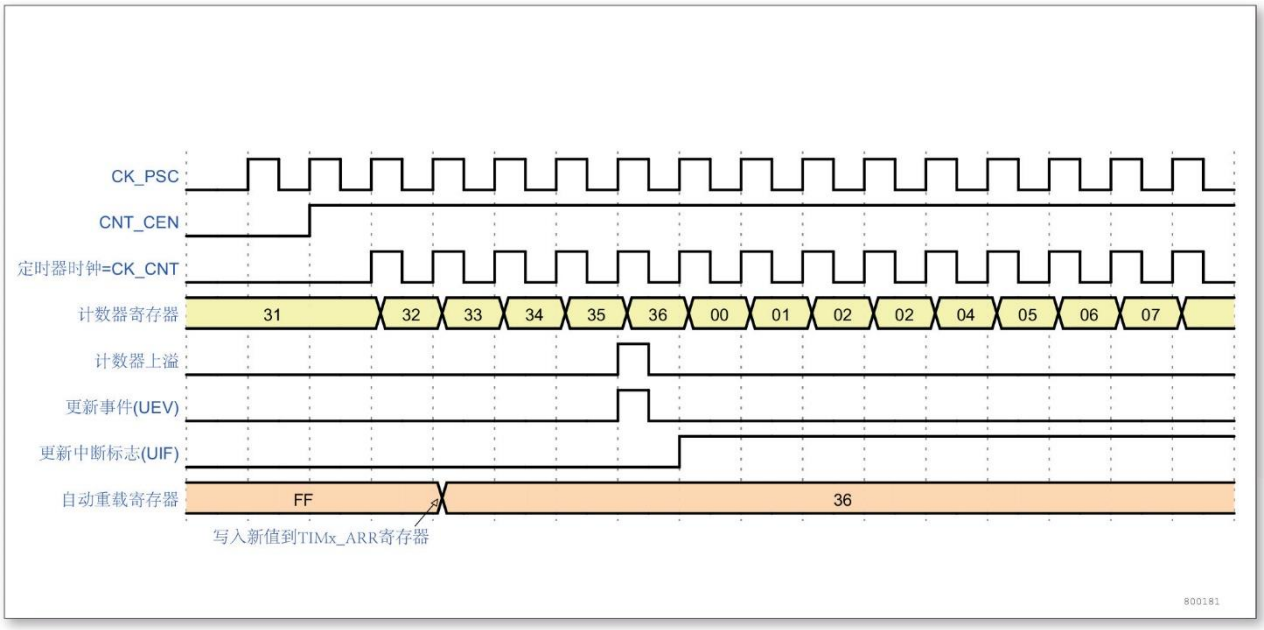

图 10-8 计数器时序图，当 ARPE = 0 时的更新事件 (TIMx_ARR 没有预装入)

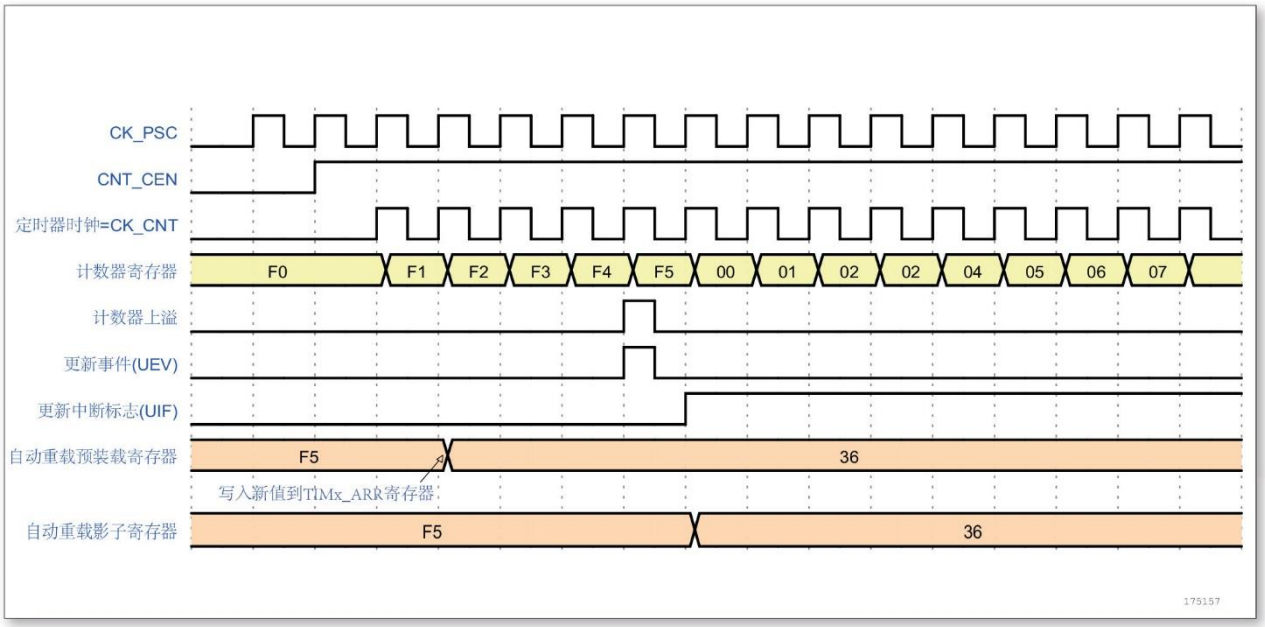

图 10-9 计数器时序图，当 ARPE = 1 时的更新事件 (预装入了 TIMx_ARR)

10.3.2.2 向下计数模式

在向下模式中，计数器从自动装入的值 (TIMx_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 TIMx_EGR 寄存器中设置 UG 位 (通过软件方式或者使用从模式控制器) 也同样可以产生一个更新事件。

设置 TIMx_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。但是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 (但预分频器的速率不能被修改)。

此外，如果设置了 TIMx_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 (因此不产生中断请求)，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 (TIMx_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 (TIMx_PSC 寄存器的值)。
- 当前的自动加载寄存器被更新为预装载值 (TIMx_ARR 寄存器中的内容)。

注：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 TIMx_ARR = 0x36 时，计数器在不同时钟频率下的操作实例：

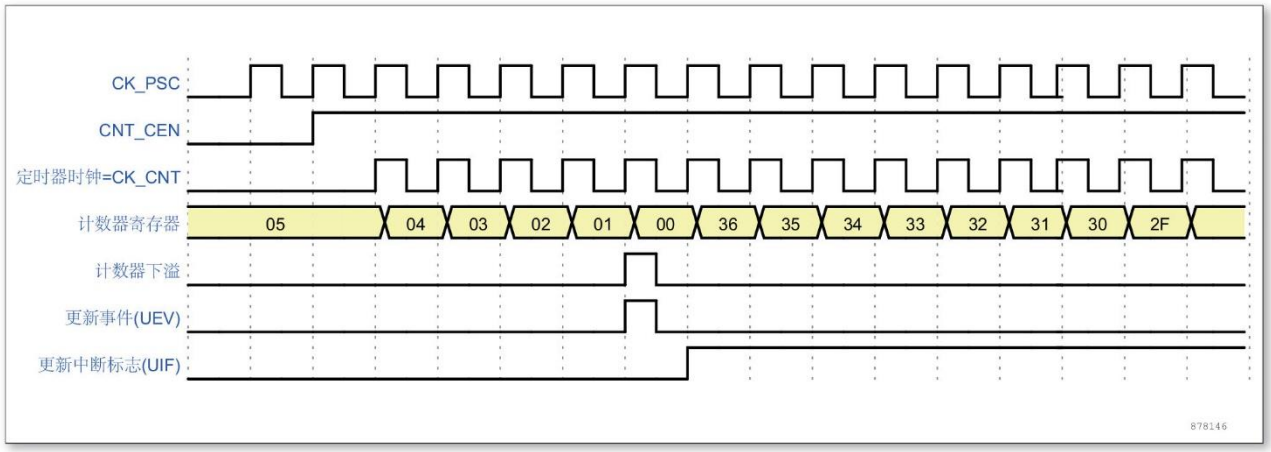

图 10-10 计数器时序图，内部时钟分频因子为 1

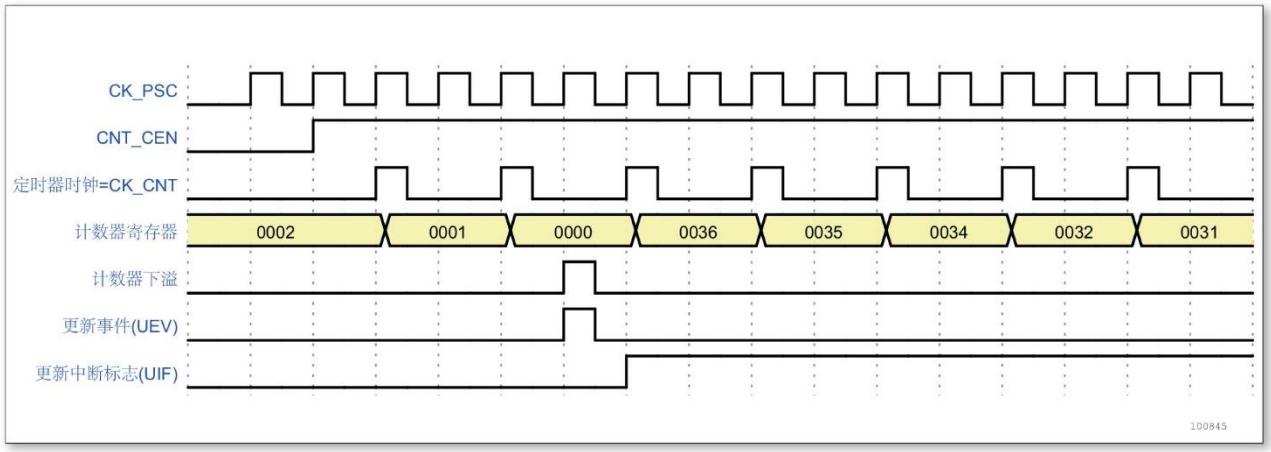

图 10-11 计数器时序图，内部时钟分频因子为 2

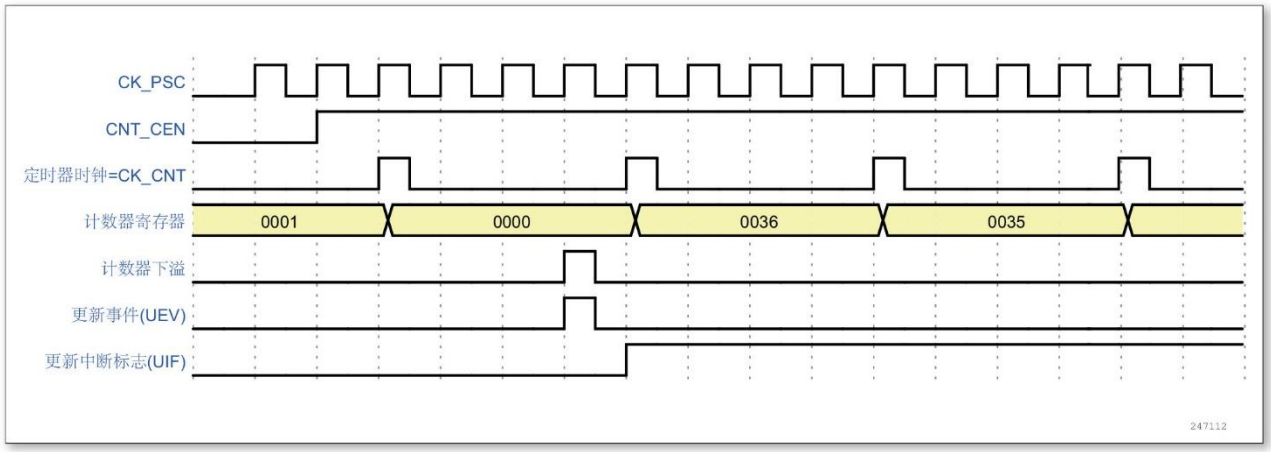

图 10-12 计数器时序图，内部时钟分频因子为 4

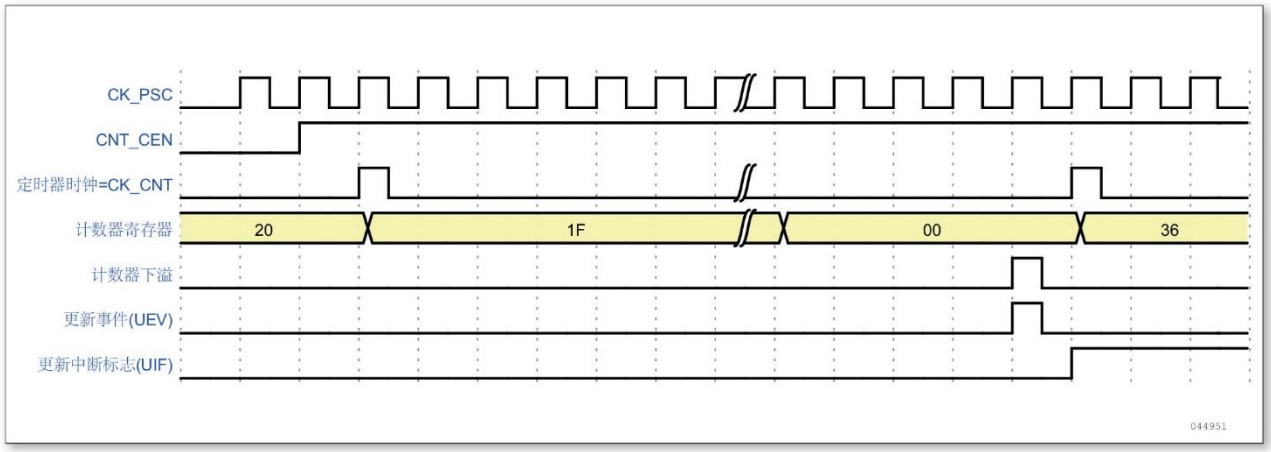

图 10-13 计数器时序图，内部时钟分频因子为 N

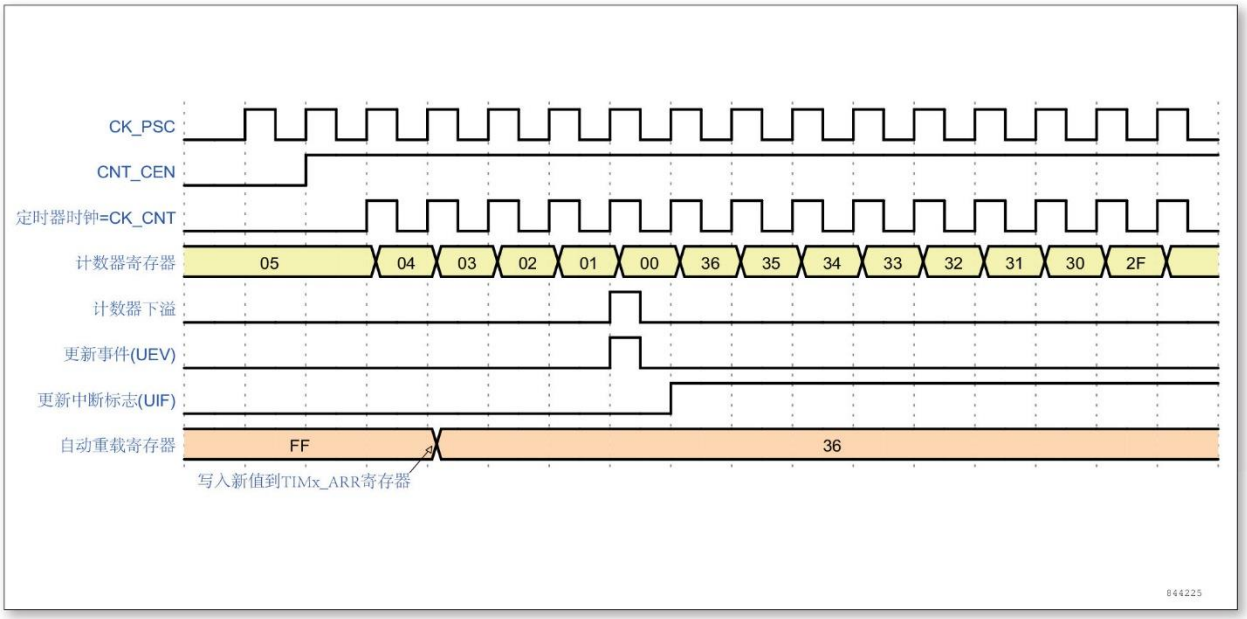

图 10-14 计数器时序图，当没有使用重复计数器时的更新事件

10.3.2.3 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 (TIMx_ARR 寄存器)-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 TIMx_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。更新事件可以产生在每次计数溢出和每次计数下溢；也可以通过 (软件或者使用从模式控制器) 设置 TIMx_EGR 寄存器中的 UG 位产生，此时，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 TIMx_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。但是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 TIMx_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 (因此不产生中断请求)，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 (TIMx_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 (TIMx_PSC 寄存器) 的值
- 当前的自动加载寄存器被更新为预装载值 (TIMx_ARR 寄存器中的内容)

注：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：

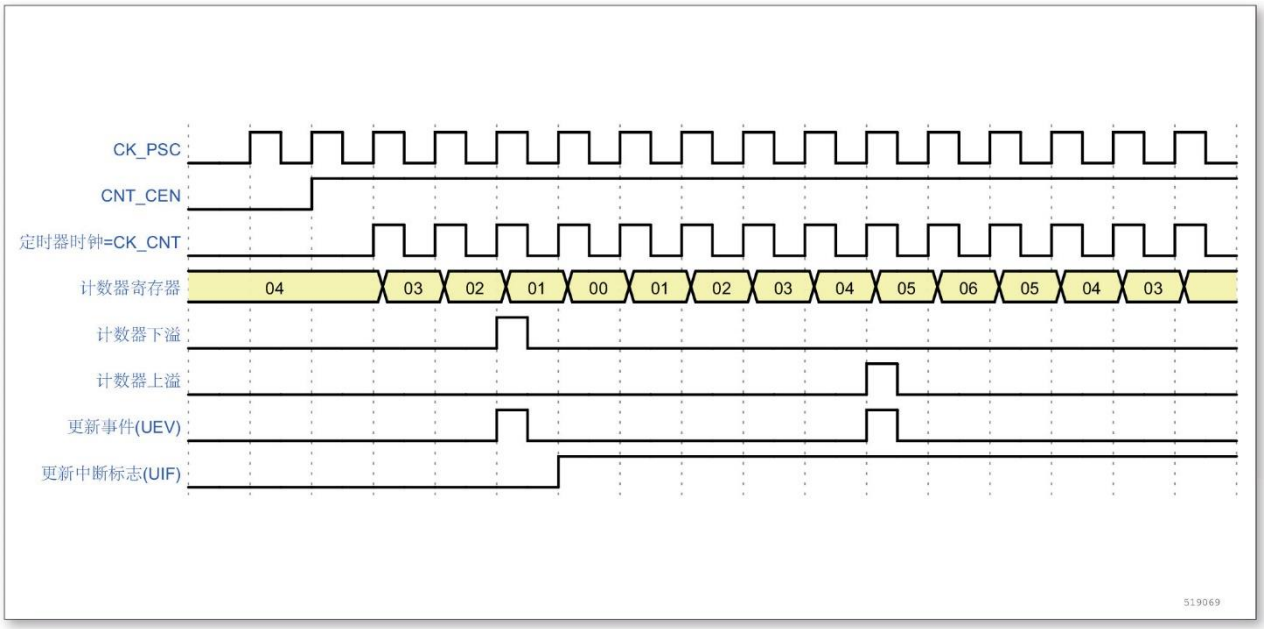

图 10-15 计数器时序图，内部时钟分频因子为 1， TIMx_ARR = 0x06

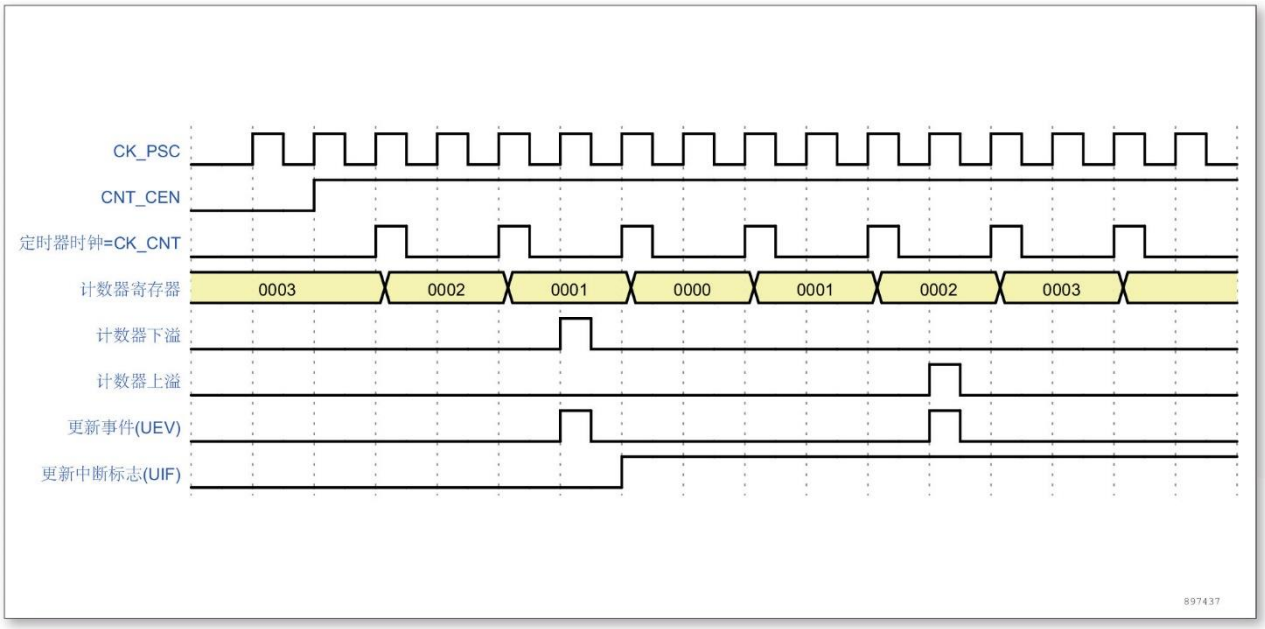

图 10-16 计数器时序图，内部时钟分频因子为 2

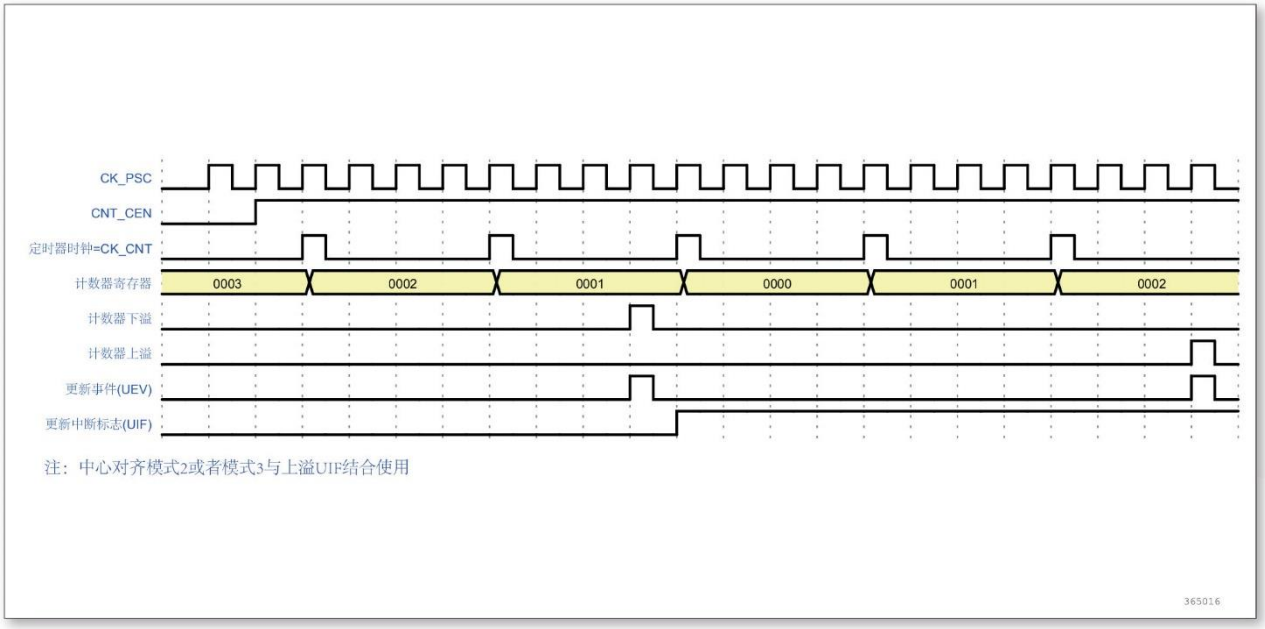

图 10-17 计数器时序图，内部时钟分频因子为 4， TIMx_ARR = 0x03

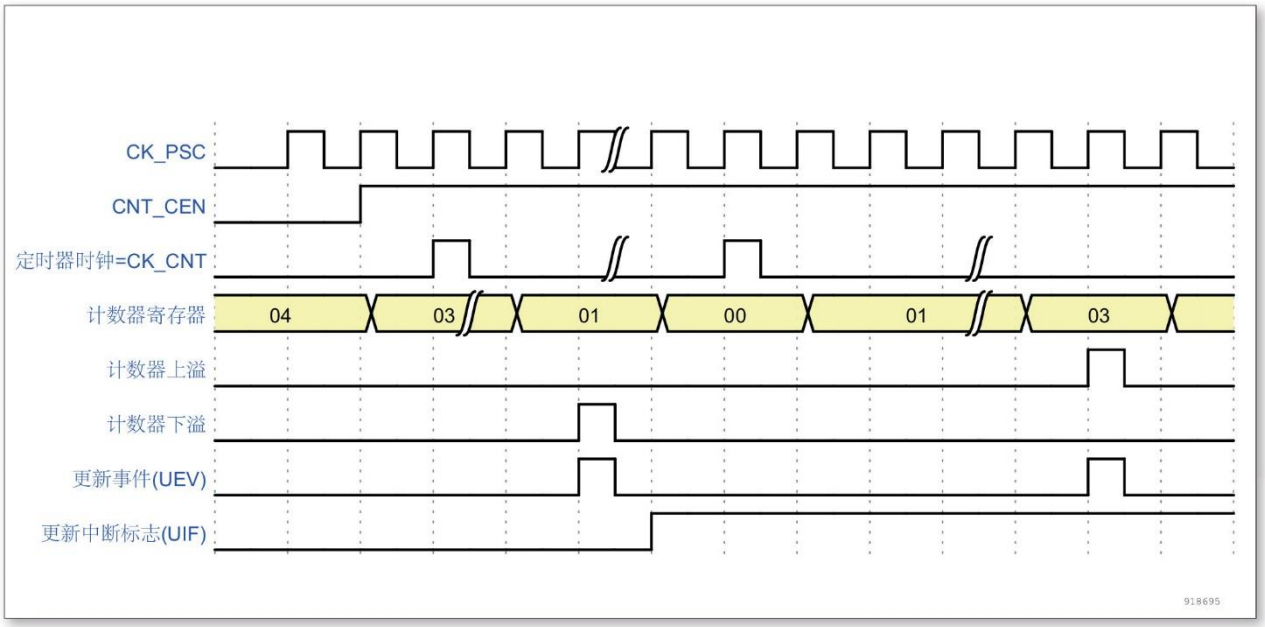

图 10-18 计数器时序图，内部时钟分频因子为 N

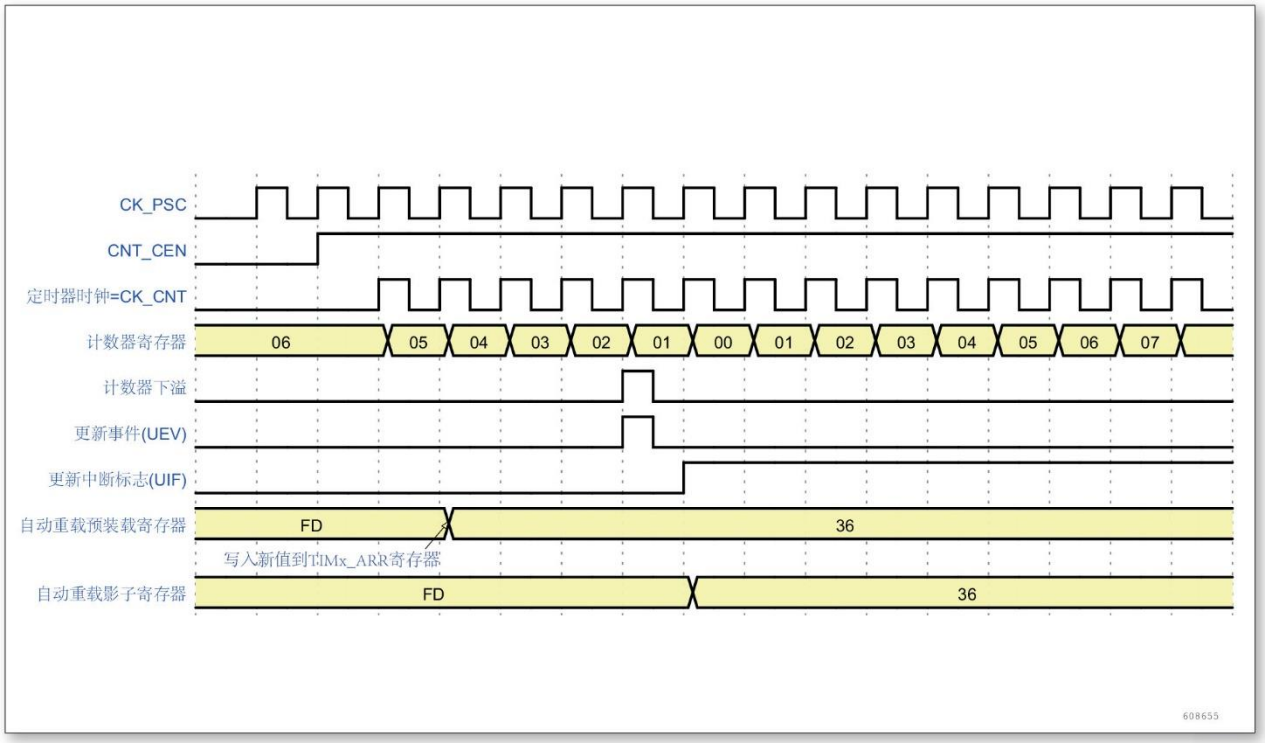

图 10-19 计数器时序图， ARPE = 1 时的更新事件 (计数器下溢)

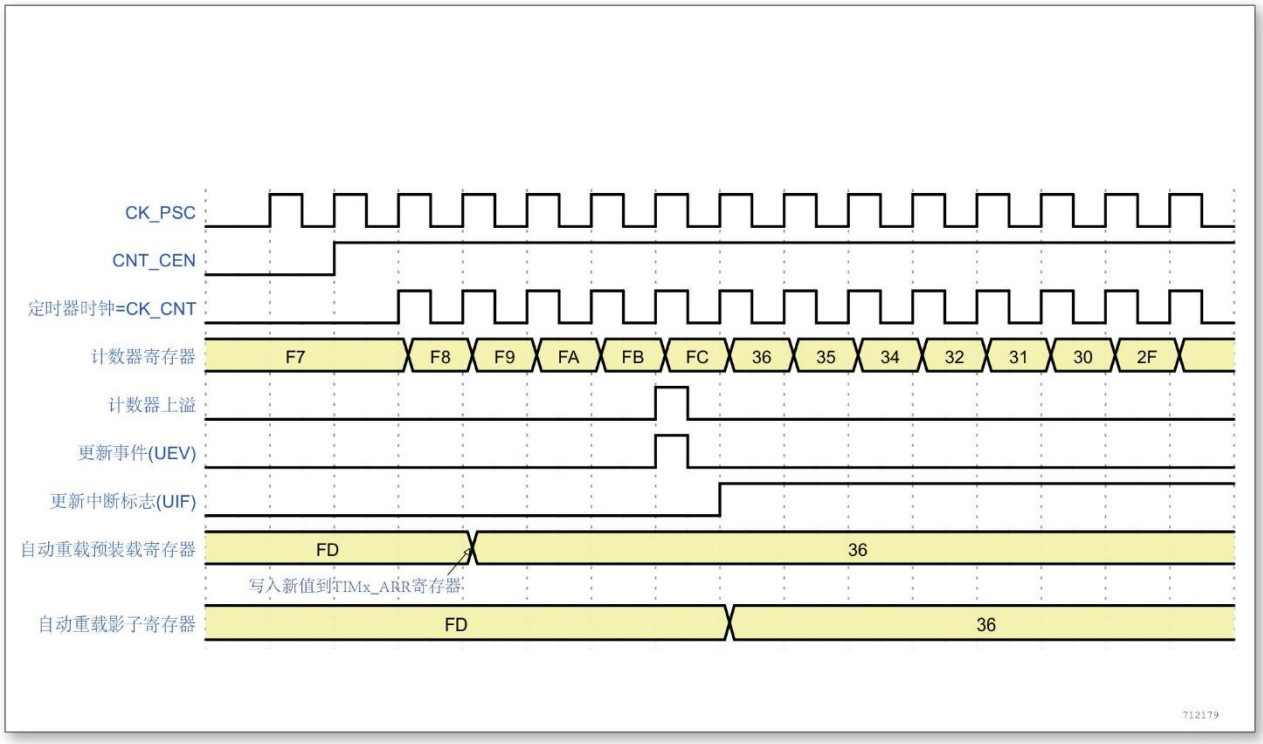

图 10-20 计数器时序图， ARPE = 1 时的更新事件 (计数器上溢)

10.3.3 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)
- 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 Timer3 的预分频器。

10.3.3.1 内部时钟源 (CK_INT)

如果禁止了从模式控制器 (SMS = 000)，则 CEN、DIR(TIMx_CR1 寄存器) 和 UG 位 (TIMx_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)。当 CEN 位被写成 1 时，预分频器的时钟由内部时钟 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。

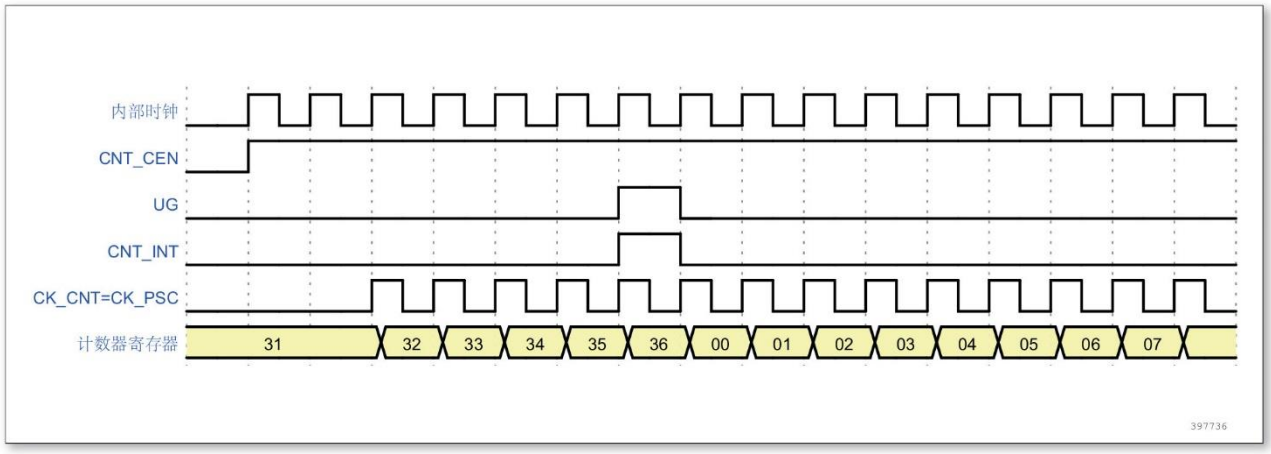

图 10-21 一般模式下的控制电路，内部时钟分频因子为 1

10.3.3.2 外部时钟源模式 1

当 TIMx_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。

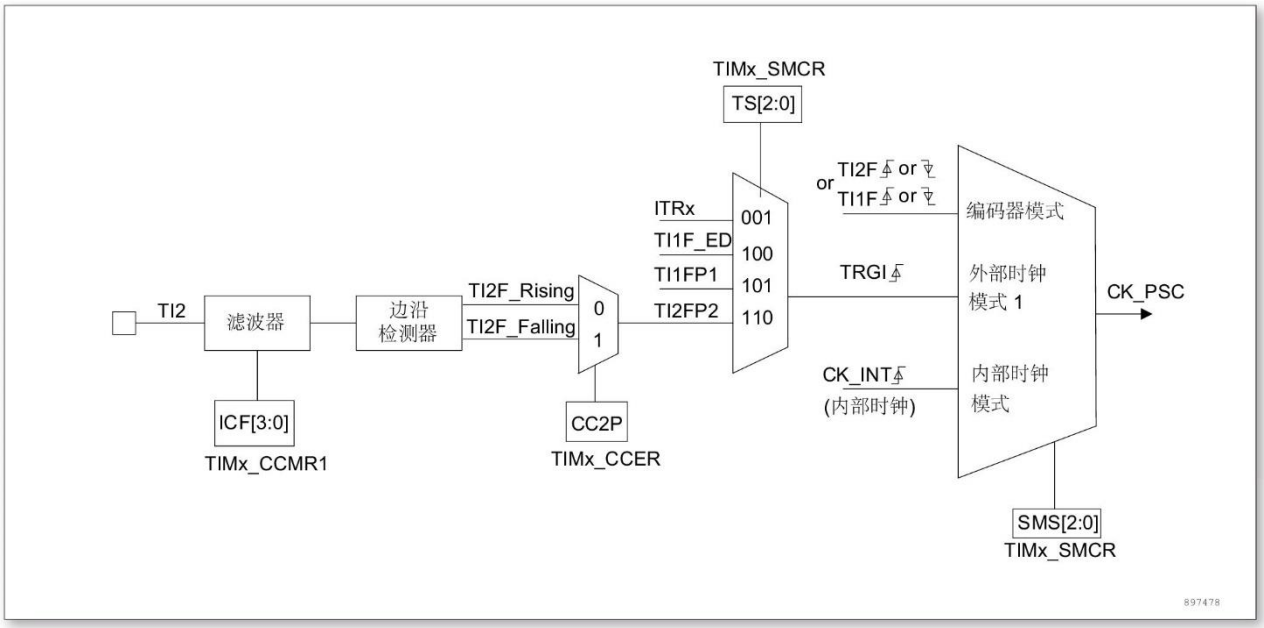

图 10-22 TI2 外部时钟连接例子

例如，要配置向上计数器在 T12 输入端的上升沿计数，使用下列步骤：

- 配置 TIMx_CCMR1 寄存器 CC2S = 01，配置通道 2 检测 TI2 输入的上升沿
- 配置 TIMx_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) 注：捕获预分频器不用作触发，所以不需要对它进行配置
- 配置 TIMx_CCER 寄存器的 CC2P = 0，选定上升沿极性

- 配置 TIMx_SMCR 寄存器的 SMS = 111，选择定时器外部时钟模式 1
- 配置 TIMx_SMCR 寄存器中的 TS = 110，选定 TI2 作为触发输入源
- 设置 TIMx_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。

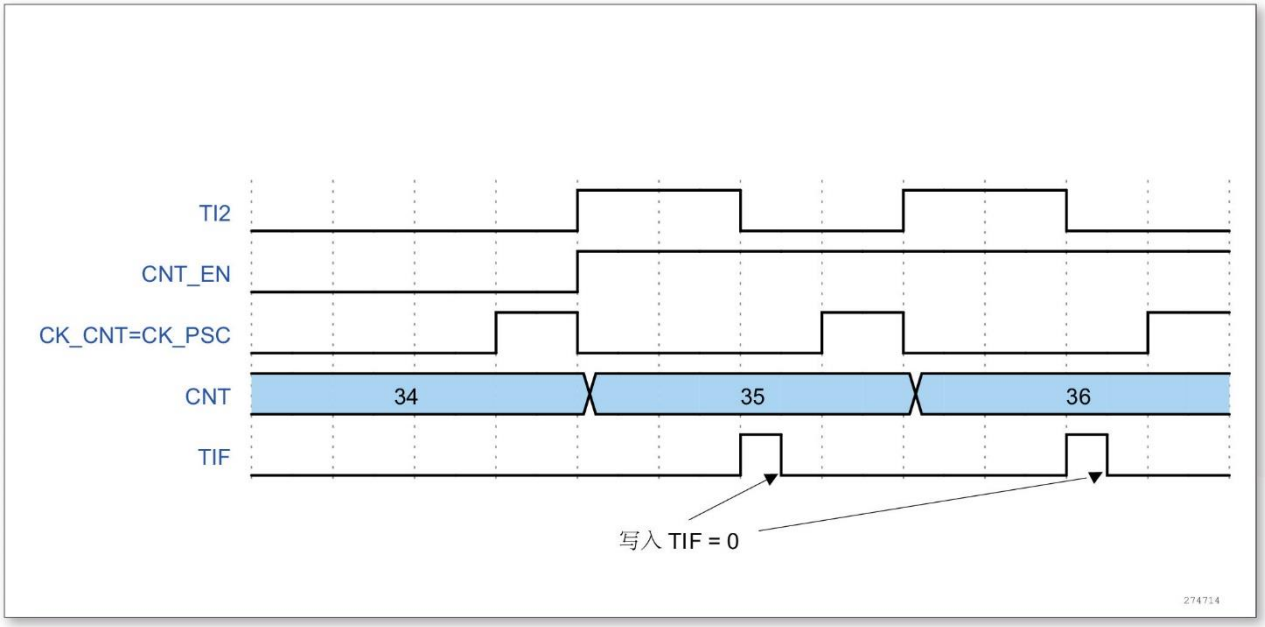

图 10-23 外部时钟模式 1 下的控制电路

10.3.4 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缘监测器产生一个信号 (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)。

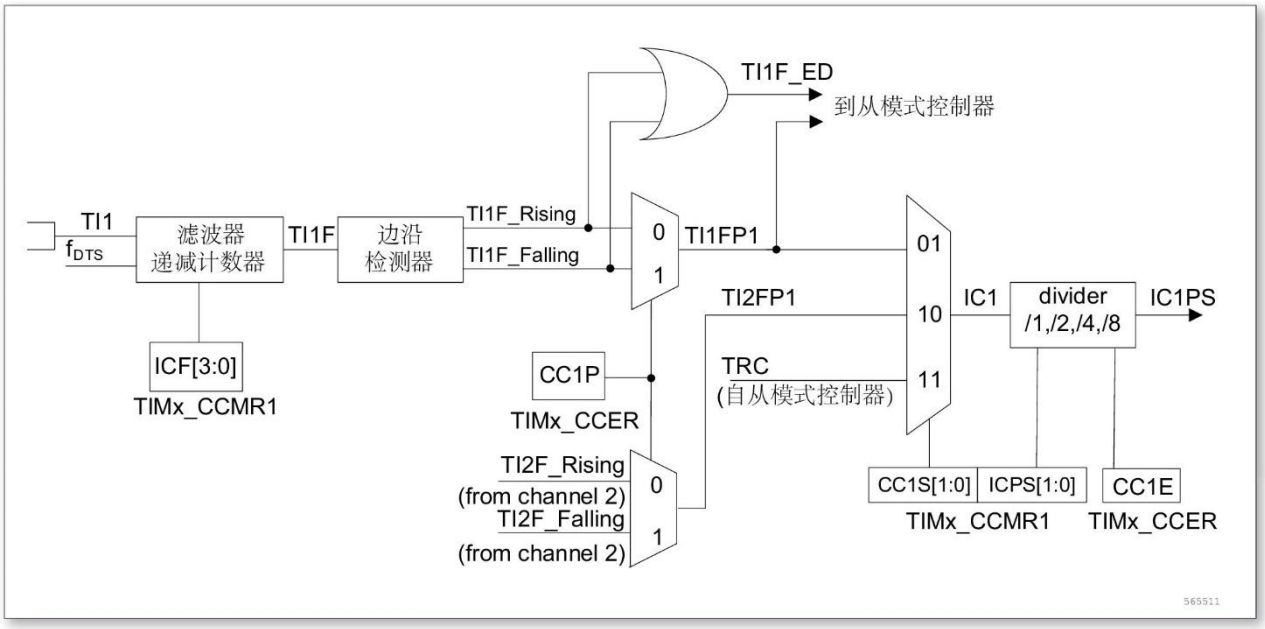

图 10-24 捕获/比较通道 (如：通道 1 输入部分)

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。

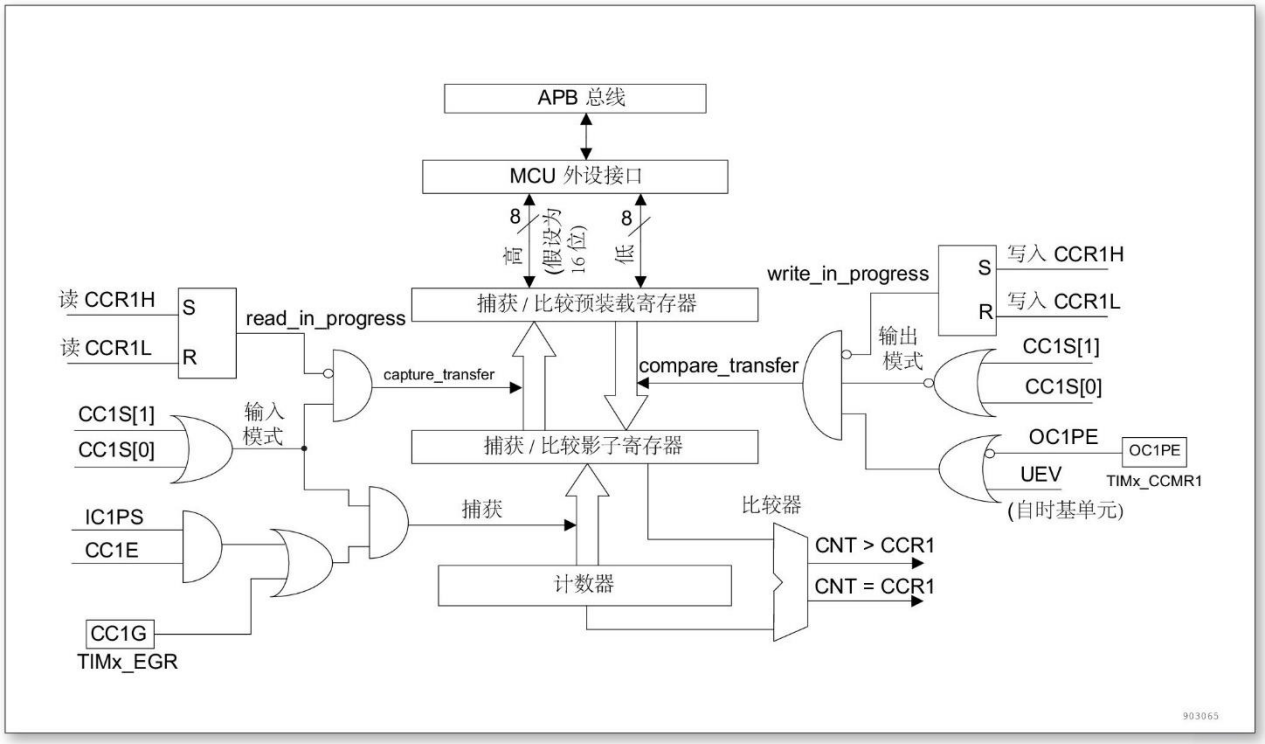

图 10-25 捕获/比较通道 1 的主电路

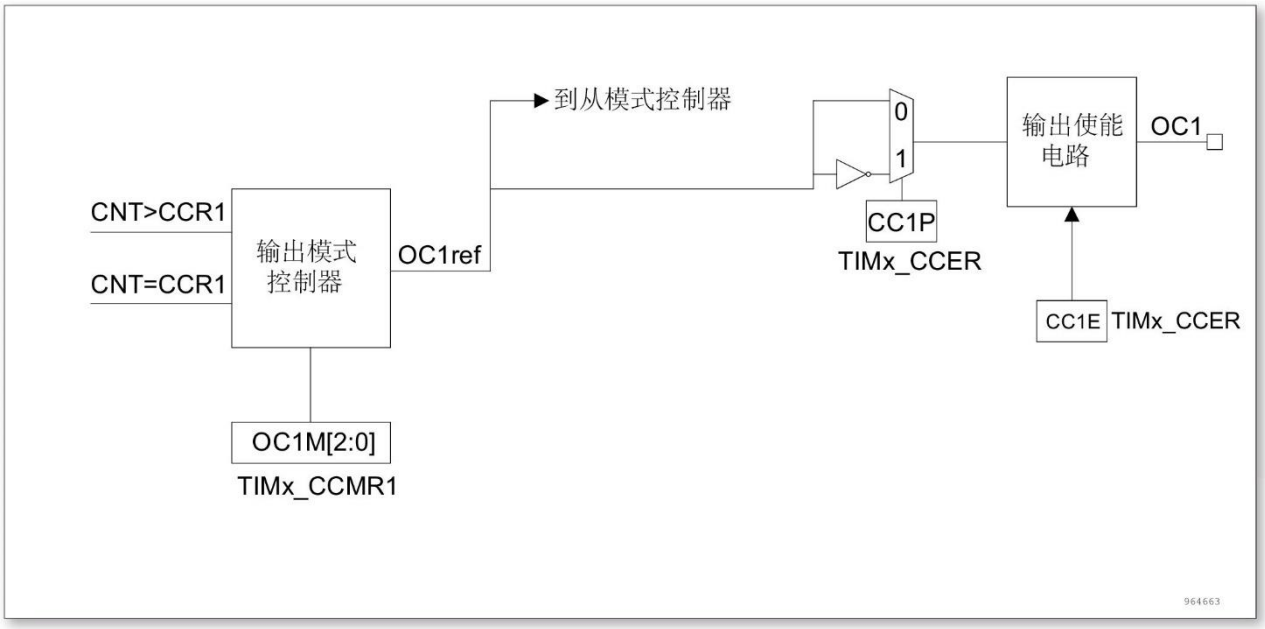

图 10-26 捕获/比较通道的输出部分 (通道 1)

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。

10.3.5 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 (TIMx_CCRx) 中。当捕获事件发生时，相应的 CCxIF 标志 (TIMx_SR 寄存器) 被置 1，如果开放了中断，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(TIMx_SR 寄存器) 被置 1。写 CCxIF = 0 可清除 CCxIF，或读取存储在 TIMx_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 TIMx_CCR1 寄存器中，步骤如下：

- 选择有效输入端： TIMx_CCR1 必须连接到 TI1 输入，所以写入 TIMx_CCR1 寄存器中的 CC1S = 01，当 CC1S 不为 00 时，通道被配置为输入，并且 TM1_CCR1 寄存器变为只读。
- 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 TIMx_CCMRx 寄存器中的 ICxF 位)。假设输入信号在最多 5 个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 TIMx_CCMR1 寄存器中写入 IC1F = 0011。
- 选择 TI1 通道的有效转换边沿，在 TIMx_CCER 寄存器中写入 CC1P = 0(上升沿)。
- 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 TIMx_CCMR1 寄存器的 IC1PS = 00)。

- 设置 TIMx_CCER 寄存器的 CC1E = 1，允许捕获计数器的值到捕获寄存器中。
- 如果需要，通过设置 TIMx_DIER 寄存器中的 CC1IE 位允许相关中断请求。

当一个输入捕获时:

- 当产生有效的电平转换时，计数器的值被传送到 TIMx_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。

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

10.3.6 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：

- 两个 ICx 信号被映射同一个 TIx 输入
- 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 (TIMx_CCR1 寄存器) 和占空比(TIMx_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)

- 选择 TIMx_CCR1 的有效输入：置 TIMx_CCMR1 寄存器的 CC1S = 01(选择 TI1)
- 选择 TI1FP1 的有效极性 (用来捕获数据到 TIMx_CCR1 中和清除计数器)：置 CC1P =0(上升沿有效)。
- 选择 TIMx_CCR2 的有效输入：置 TIMx_CCMR1 寄存器的 CC2S = 10(选择 TI1)。
- 选择 TI1FP2 的有效极性 (捕获数据到 TIMx_CCR2)：置 CC2P = 1(下降沿有效)。
- 选择有效的触发输入信号：置 TIMx_SMCR 寄存器中的 TS = 101(选择 TI1FP1)。
- 配置从模式控制器为复位模式：置 TIMx_SMCR 中的 SMS = 100。
- 使能捕获：置 TIMx_CCER 寄存器中 CC1E = 1 且 CC2E = 1。

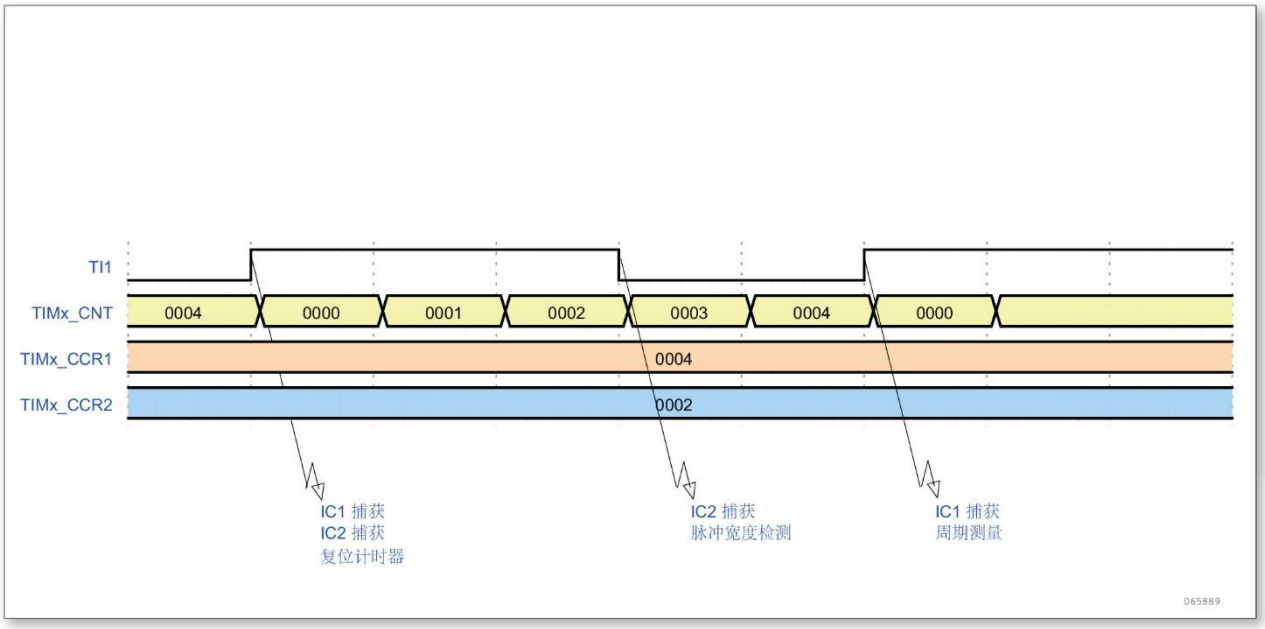

图 10-27 PWM 输入模式时序

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 TIMx_CH1 / TIMx_CH2 信号。

10.3.7 强制输出模式

在输出模式 (TIMx_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 TIMx_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 TIMx_CCMRx 寄存器中的 OCxM = 100，可强置 OCxREF 信号为低。该模式下，在 TIMx_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。

10.3.8 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何时一段给定的时间已经到时。 当计数器与比较寄存器的内容相同时，输出比较功能做如下操作：

- 将输出比较模式 (TIMx_CCMRx 寄存器中的 OCxM 位) 和输出极性 (TIMx_CCER 寄存器中的 CCxP 位) 定义的值输出到对应的管脚上。在比较匹配时，输出管脚可以保持它的电平

(OCxM = 000)、被设置成有效电平 (OCxM = 001)、被设置成无有效电平 (OCxM= 010) 或进行翻转 (OCxM = 011)。

- 设置中断状态寄存器中的标志位 (TIMx_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽 (TIMx_DIER 寄存器中的 CCxIE 位)，则产生一个中断。

TIMx_CCMRx 中的 OCxPE 位选择 TIMx_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：

- 选择计数器时钟 (内部，外部，预分频器)
- 将相应的数据写入 TIMx_ARR 和 TIMx_CCRx 寄存器中
- 如果要产生一个中断请求，设置 CCxIE 位
- 选择输出模式，例如：设置 OCxM = '011'、 OCxPE = '0'、 CCxP = '0'和 CCxE ='1'，当计数器 CNT 与 CCRx 匹配时翻转 OCx 的输出管脚， CCRx 预装载未用，开启 OCx 输出且高电平有效
- 设置 TIMx_CR1 寄存器的 CEN 位启动计数器

TIMx_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 TIMx_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。

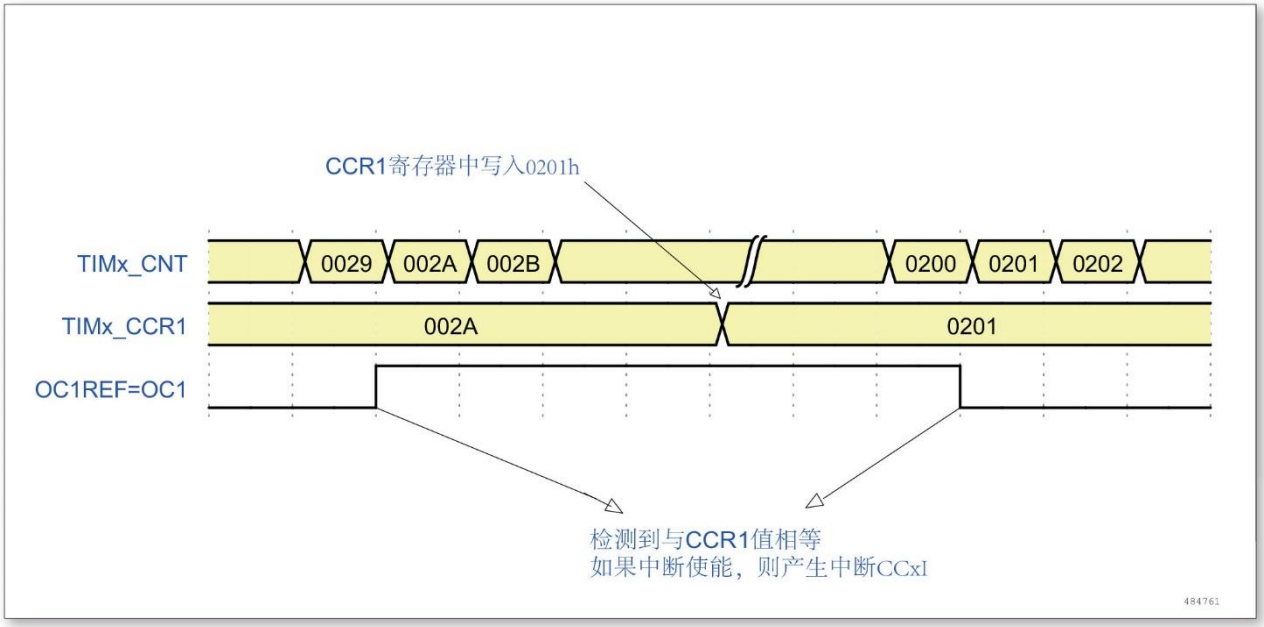

图 10-28 输出比较模式，翻转 OC1

10.3.9 PWM 模式

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

在 TIMx_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 TIMx_CCMRx 寄存器 OCxPE 位以使能相应的预装载寄存器，最后还要设置 TIMx_CR1 寄存器的 ARPE 位使能自动重装载的预装载寄存器 (在向上计数或中心对称模式中)。

因为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 TIMx_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 TIMx_CCER 寄存器中的 CCxP 位设置，它可以设置为高电平有效活或低电平有效。 TIMx_CCER 寄存器中的 CCxE 位控制 OCx 输出使能。详见 TIMx_CCERx 寄存器的描述。

在 PWM 模式 (模式 1 或模式 2) 下， TIMx_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。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 (TIMx_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 TIMx_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。

10.3.9.1 PWM 边沿对齐模式

10.3.9.1.1 向上计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。参看计数模式章节。

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

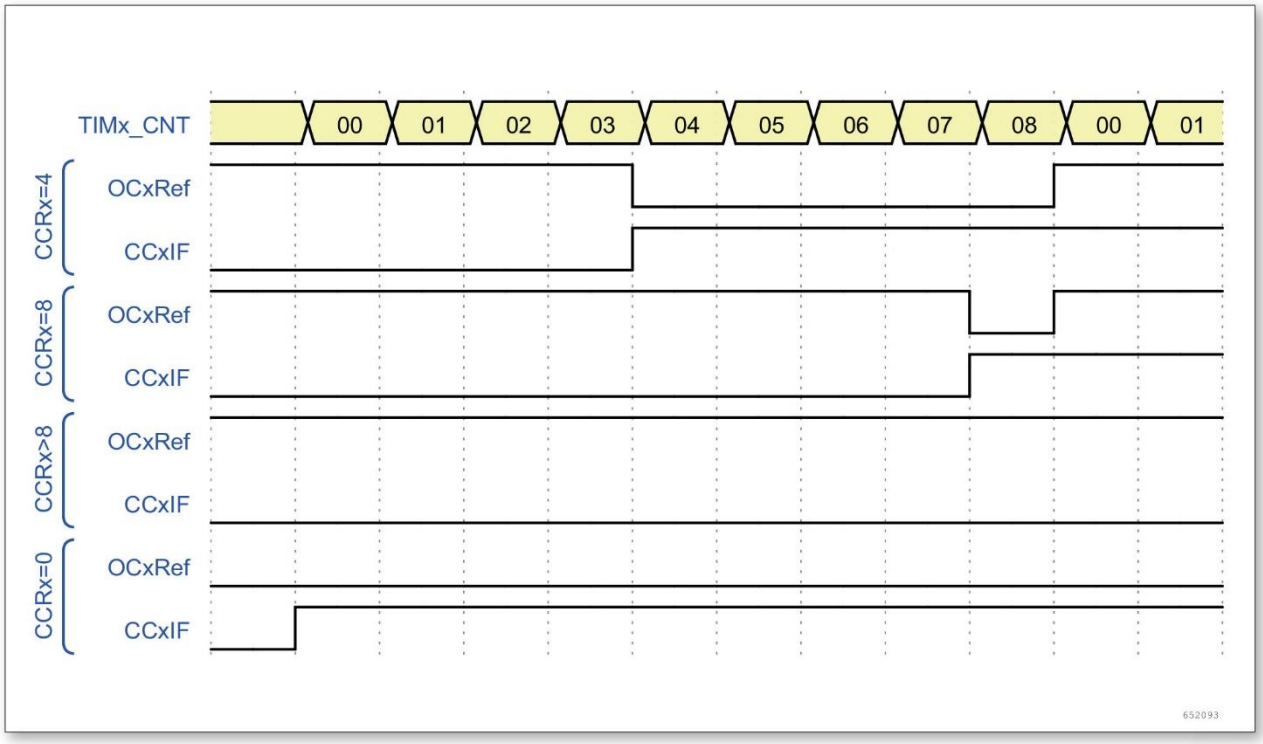

图 10-29 边沿对齐的 PWM 波形 (ARR = 8)

10.3.9.1.2 向下计数配置

当 TIMx_CR1 寄存器的 DIR 位为高时执行向下计数。参看计数模式章节。

在 PWM 模式 1，当 TIMx_CNT > TIMx_CCRx 时参考信号 OCxREF 为低，否则为高。如果 TIMx_CCRx 中的比较值大于 TIMx_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。

10.3.9.2 PWM 中央对齐模式

当 TIMx_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 位的设置，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’。TIMx_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。参看计数模式章节的中央对齐模式。

下图给出了一些中央对齐的 PWM 波形的例子

- TIMx_ARR = 8
- PWM 模式 1
- TIMx_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

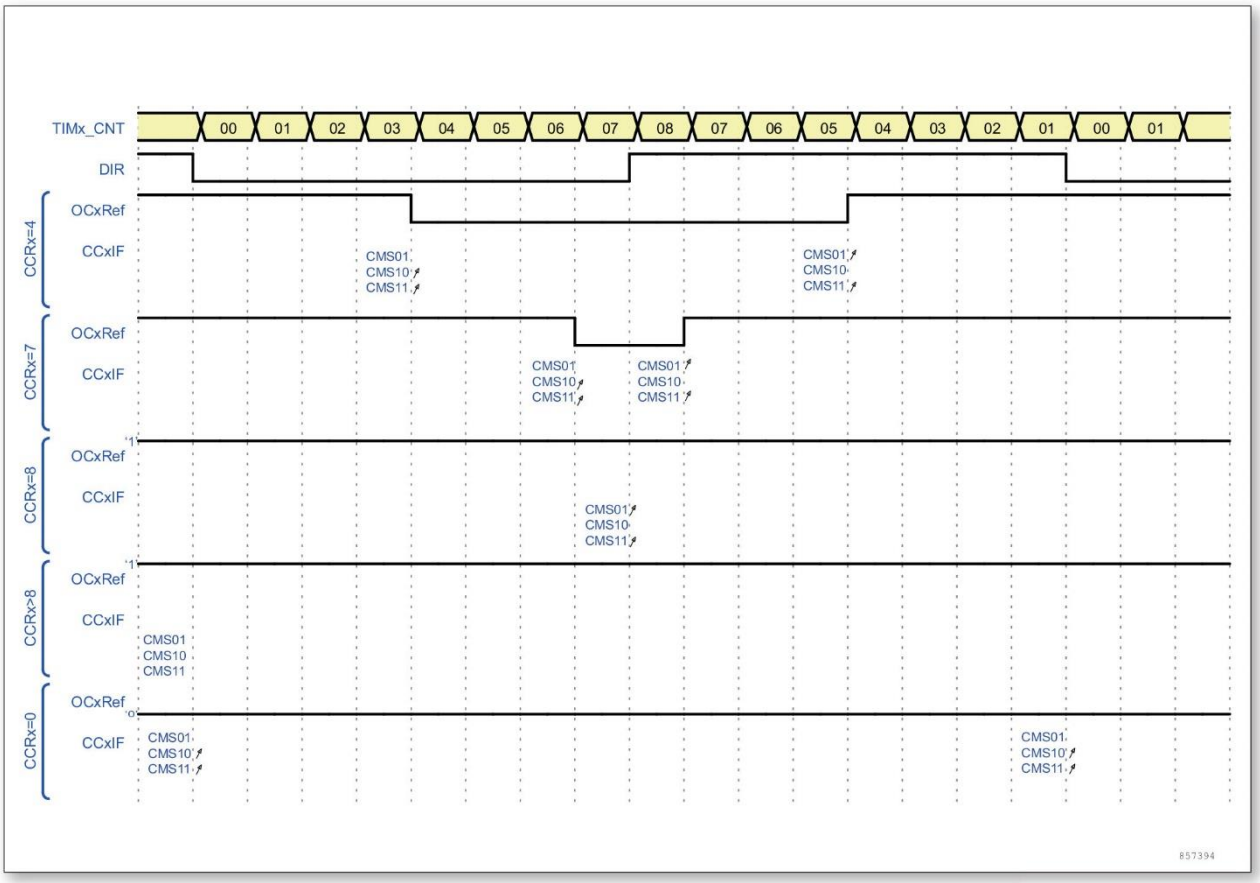

图 10-30 中央对齐的 PWM 波形 (APR = 8)

使用中央对齐模式的提示：

- 进入中央对齐模式时，使用当前的上/下计数配置；这就意味着计数器向上还是向下计数取决于 TIMx_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 (TIMx_CNT > TIMx_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 TIMx_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 TIMx_EGR 位中的 UG 位)，不要在计数进行过程中修改计数器的值。

10.3.10 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 TIMx_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

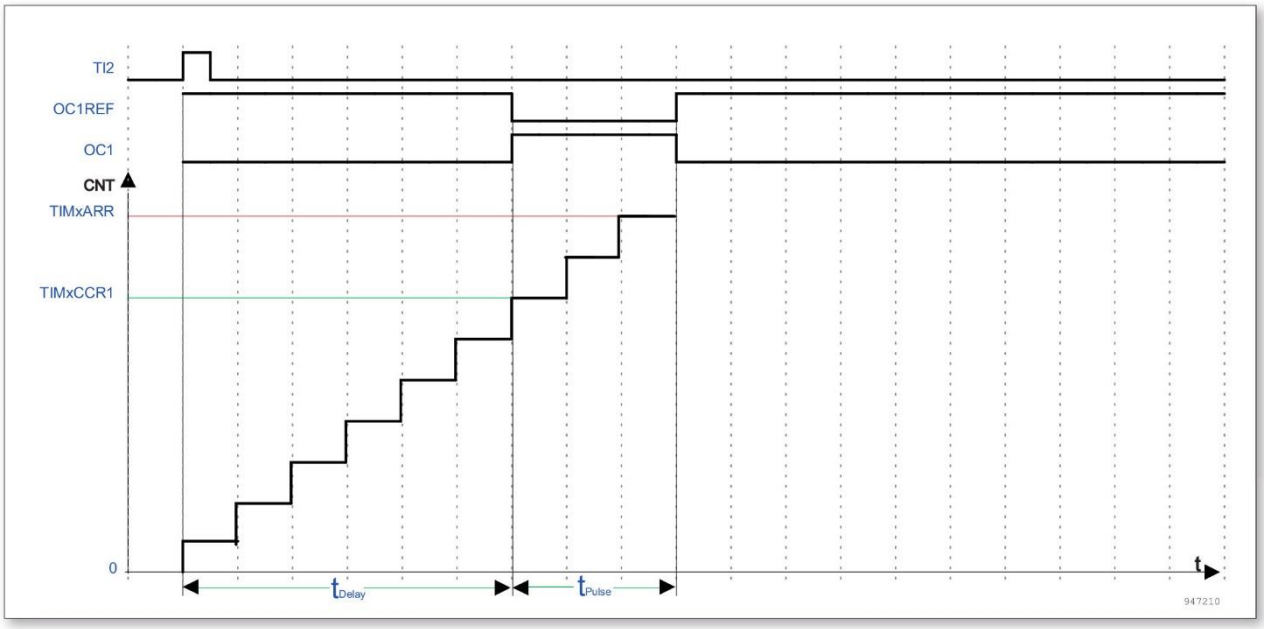

图 10-31 单脉冲模式的例子

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

假定 TI2FP2 作为触发 1：

- 置 TIMx_CCMR1 寄存器中的 CC2S = 01，把 TI2FP2 映像到 TI2
- 置 TIMx_CCER 寄存器中的 CC2P = 0，使 TI2FP2 能够检测上升沿
- 置 TIMx_SMCR 寄存器中的 TS = 110， TI2FP2 作为从模式控制器的触发 (TRGI)
- 置 TIMx_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)。

- tDELAY 由写入 TIMx_CCR1 寄存器中的值定义。
- tPULSE 由自动装载值和比较值之间的差值定义 (TIMx_ARR - TIMx_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是要产生一个从 1 到 0 的波形；首先要置 TIMx_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 TIMx_CCMR1 中的 OC1PE = 1 和 TIMx_CR1 寄存器中的 ARPE；然后在 TIMx_CCR1 寄存器中填写比较值，在 TIMx_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 = 1。

在这个例子中， TIMx_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 TIMx_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 tDELAY。

如果要以最小延时输出波形，可以设置 TIMx_CCMRx 寄存器中的 OCxFE 位；此时强制 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。

10.3.11 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 TIMx_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 TIMx_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。下表，假定计数器已经启动 (TIMx_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；如果没有滤波和变相，则 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 TIMx_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 TIMx_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 TIMx_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 10-1 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		TI1FP2 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

有效边沿	相对信号的电平 (TI1FP1 对应 TI2，TI2FP2 对应 TI1)	TI1FP1 信号		TI1FP2 信号	
		上升	下降	上升	下降
	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，一般使用比较器将编码器的差动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：

- C1S = ‘01’ (TIMx_CCMR1 寄存器， IC1FP1 映射到 TI1)
- CC2S = ‘01’ (TIMx_CCMR2 寄存器， IC2FP2 映射到 TI2)
- CC1P = ‘0’ (TIMx_CCER 寄存器， IC1FP1 不反相， IC1FP1 = TI1)
- CC2P = ‘0’ (TIMx_CCER 寄存器， IC2FP2 不反相， IC2FP2 = TI2)
- SMS = ‘011’ (TIMx_SMCR 寄存器，所有的输入均在上升沿和下降沿有效)
- CEN = ‘1’ (TIMx_CR1 寄存器，计数器使能)

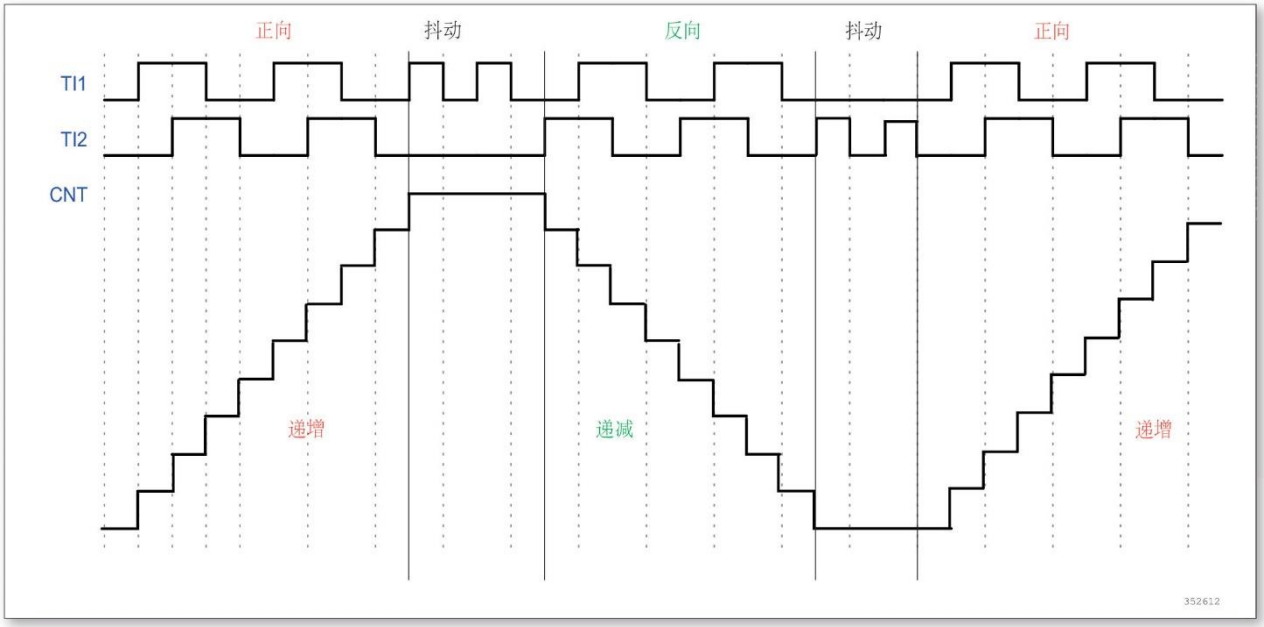

图 10-32 编码器模式下的计数器操作实例

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)

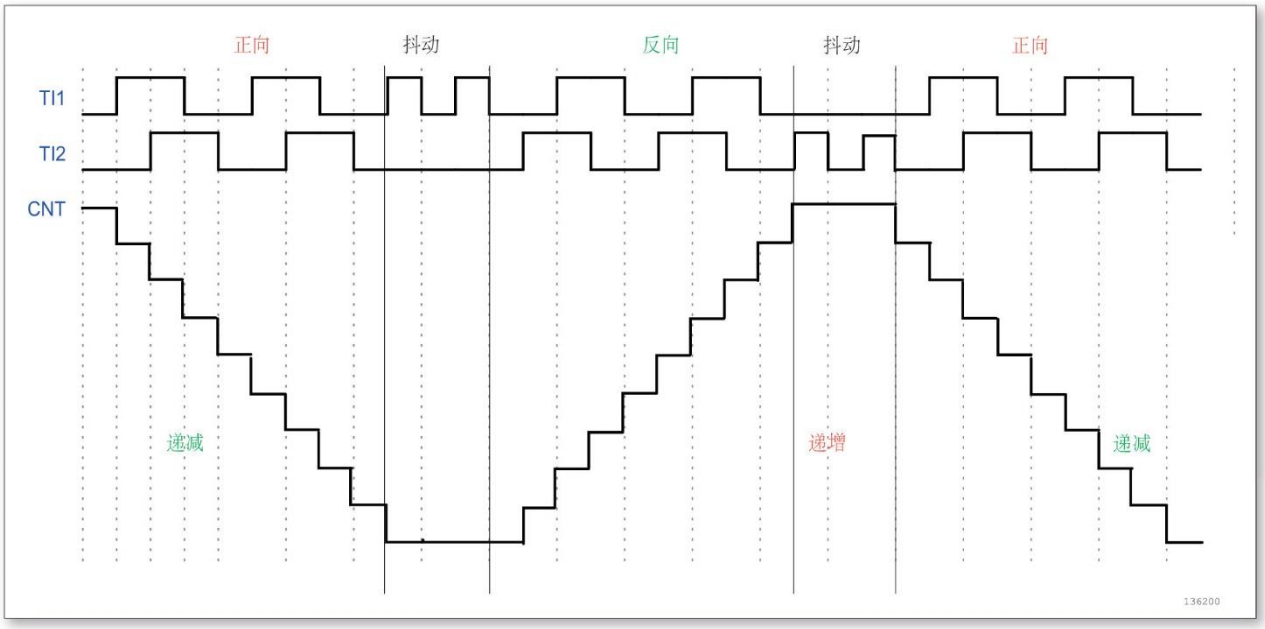

图 10-33 IC1FP1 反相的编码器接口模式实例

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定时器测量两个编码器事件的间隔，可以获得动态的信息 (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)。

10.3.12 定时器输入异或功能

TIMx_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 TIMx_CH1、 TIMx_CH2 和 TIMx_CH3。

10.3.13 定时器和外部触发的同步

TIMx 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。

10.3.13.1 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 TIMx_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 (TIMx_ARR ，TIMx_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 TIMx_CCMR1 寄存器中 CC1S = 01。置 TIMx_CCER 寄存器中 CC1P = 0 以确定极性 (只检测上升沿)

- 置 TIMx_SMCR 寄存器中 SMS = 100，配置定时器为复位模式；置 TIMx_SMCR 寄存器中 TS = 101，选择 TI1 作为输入源
- 置 TIMx_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 (TIMx_SR 寄存器中的 TIF 位) 被设置，根据 TIMx_DIER 寄存器中 TIE(中断使能) 位，产生一个中断请求。

下图显示当自动重装载寄存器 TIMx_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。

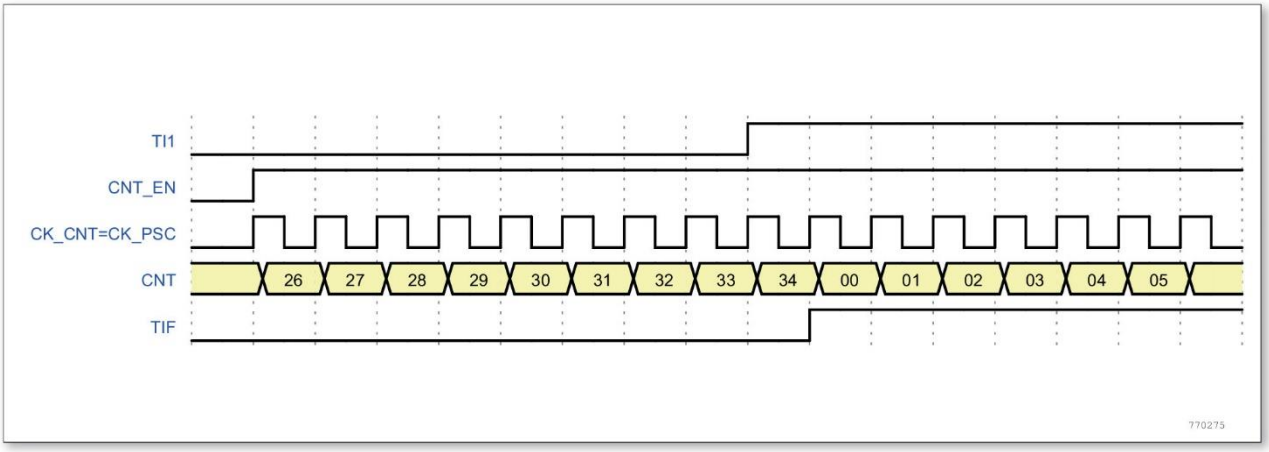

图 10-34 复位模式下的控制电路

10.3.13.2 从模式：门控模式

计数器的使能依赖于选中的输入端的电平。

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 TIMx_CCMR1 寄存器中 CC1S = 01。置 TIMx_CCER 寄存器中 CC1P= 1 以确定极性 (只检测低电平)。
- 置 TIMx_SMCR 寄存器中 SMS = 101，配置定时器为门控模式；置 TIMx_SMCR 寄存器中 TS = 101，选择 TI1 作为输入源。
- 置 TIMx_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 TIMx_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。

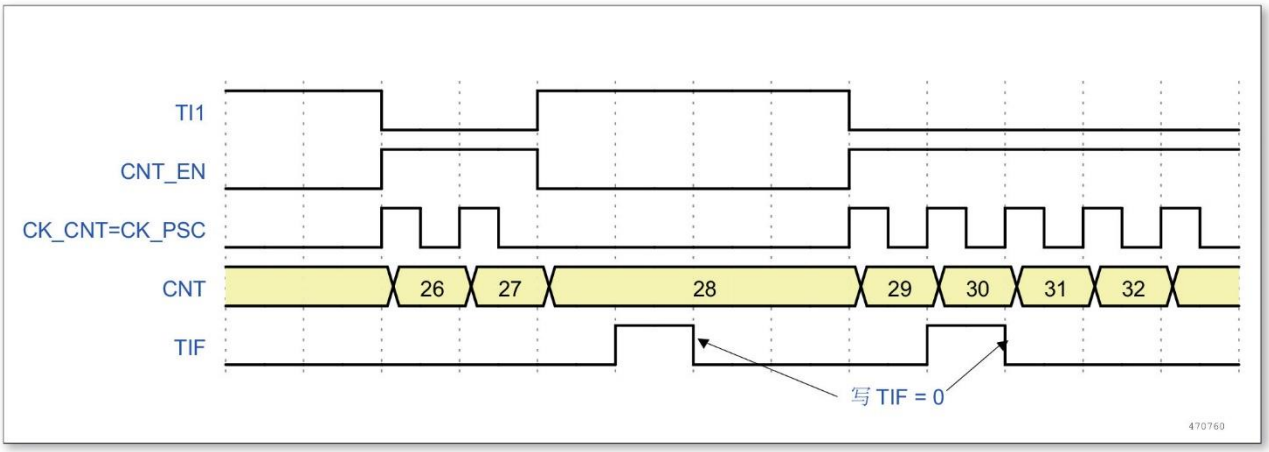

图 10-35 门控模式下的控制电路

10.3.13.3 从模式：触发模式

计数器的使能依赖于选中的输入端上的事件。

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 TIMx_CCMR1 寄存器中 CC2S = 01。置 TIMx_CCER 寄存器中 CC1P= 1 以确定极性 (只检测低电平)。
- 置 TIMx_SMCR 寄存器中 SMS = 110，配置定时器为触发模式；置 TIMx_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。

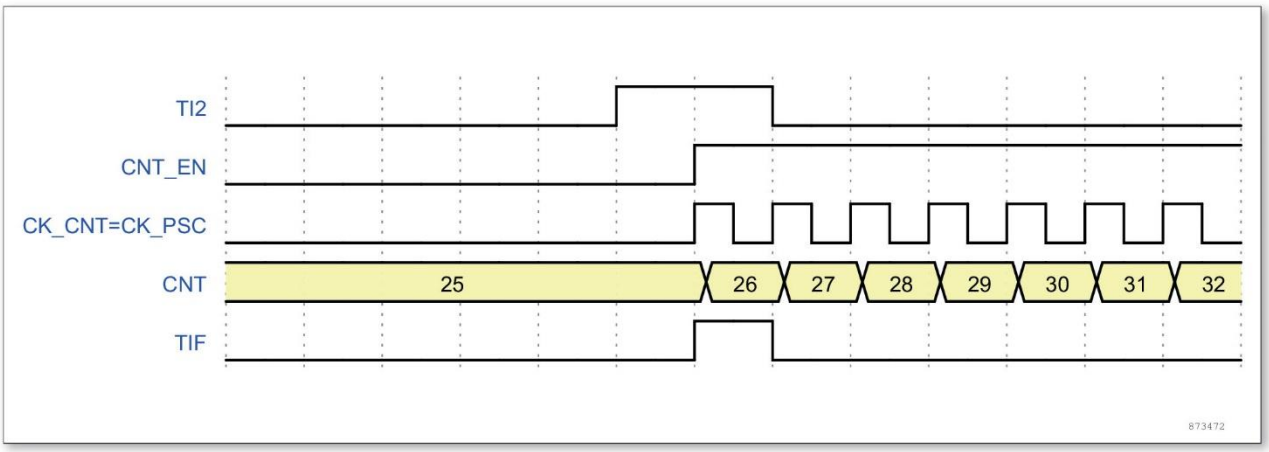

图 10-36 触发器模式下的控制电路

10.3.14 定时器同步

所有 TIMx 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。

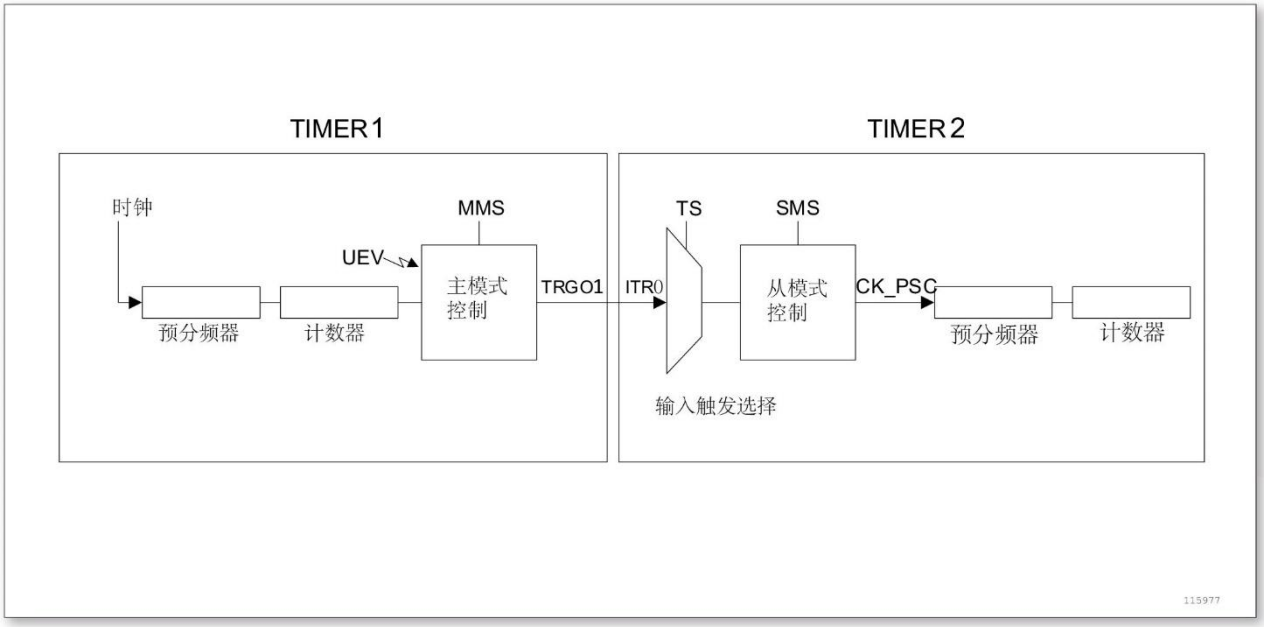

图 10-37 主/从定时器的例子

10.3.14.1 使用一个定时器作为另一个定时器的预分频器

如：可以配置定时器 1 作为定时器 3 的预分频器。参考上图，进行下述操作：

- 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。
- 连接定时器 1 的 TRGO1 输出至定时器 3，设置 TIM3_SMCR 寄存器的 TS = ‘000’，配置定时器 3 为使用 ITR1 作为内部触发的从模式。
- 然后把从模式控制器置于外部时钟模式 1(TIM3_SMCR 寄存器的 SMS = 111)；这样定时器 3 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。
- 最后，必须设置相应 (TIMx_CR1 寄存器) 的 CEN 位分别启动两个定时器。

注：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 3 的计数器。

10.3.14.2 使用一个定时器使能另一个定时器

在这个例子中，定时器 3 的运行受由定时器 1 的输出比较控制。参考下图。只当定时器 1 的 OC1REF 为高时定时器 3 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。

- 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)
- 配置定时器 1 的 OC1REF 波形 (TIM1_CCMR1 寄存器)
- 配置定时器 3 从定时器 1 获得输入触发 (TIM3_SMCR 寄存器的 TS = 001)
- 配置定时器 3 为门控模式 (TIM3_SMCR 寄存器的 SMS = 101)
- 置 TIM3_CR1 寄存器的 CEN = 1 以使能定时器 3
- 置 TIM1_CR1 寄存器的 CEN = 1 以使能定时器 1

注：定时器 3 的时钟不与定时器 1 的时钟同步，这个模式只影响定时器 3 计数器的使能信号。

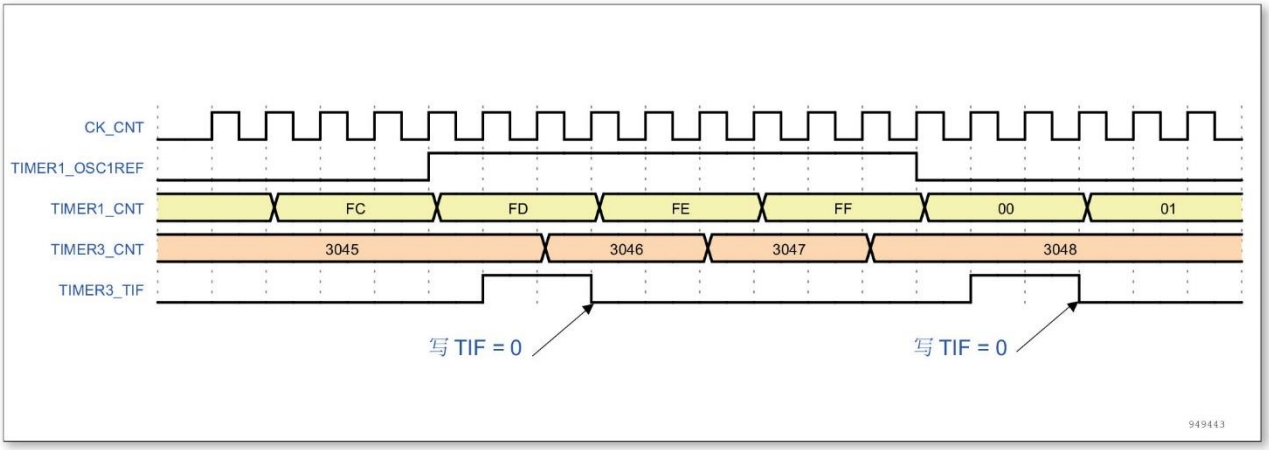

图 10-38 定时器 1 的 OC1REF 控制定时器 3

在上图的例子中，在定时器 3 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 TIMx_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 3。定时器 1 是主模式并从 0 开始，定时器 3 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 3 随即停止。

- 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)。
- 配置定时器 1 的 OC1REF 波形 (TIM1_CCMR1 寄存器)。
- 配置定时器 3 从定时器 1 获得输入触发 (TIM3_SMCR 寄存器的 TS = 000)
- 配置定时器 3 为门控模式 (TIM3_SMCR 寄存器的 SMS = 101)

- 置 TIM1_EGR 寄存器的 UG = 1，复位定时器 1。
- 置 TIM3_EGR 寄存器的 UG = 1，复位定时器 3。
- 写 0xE7 至定时器 3 的计数器 (TIM3_CNT)，初始化它为 0xE7。
- 置 TIM3_CR1 寄存器的 CEN = 1 以使能定时器 3。
- 置 TIM1_CR1 寄存器的 CEN = 1 以启动定时器 1。
- 置 TIM1_CR1 寄存器的 CEN = 0 以停止定时器 1。

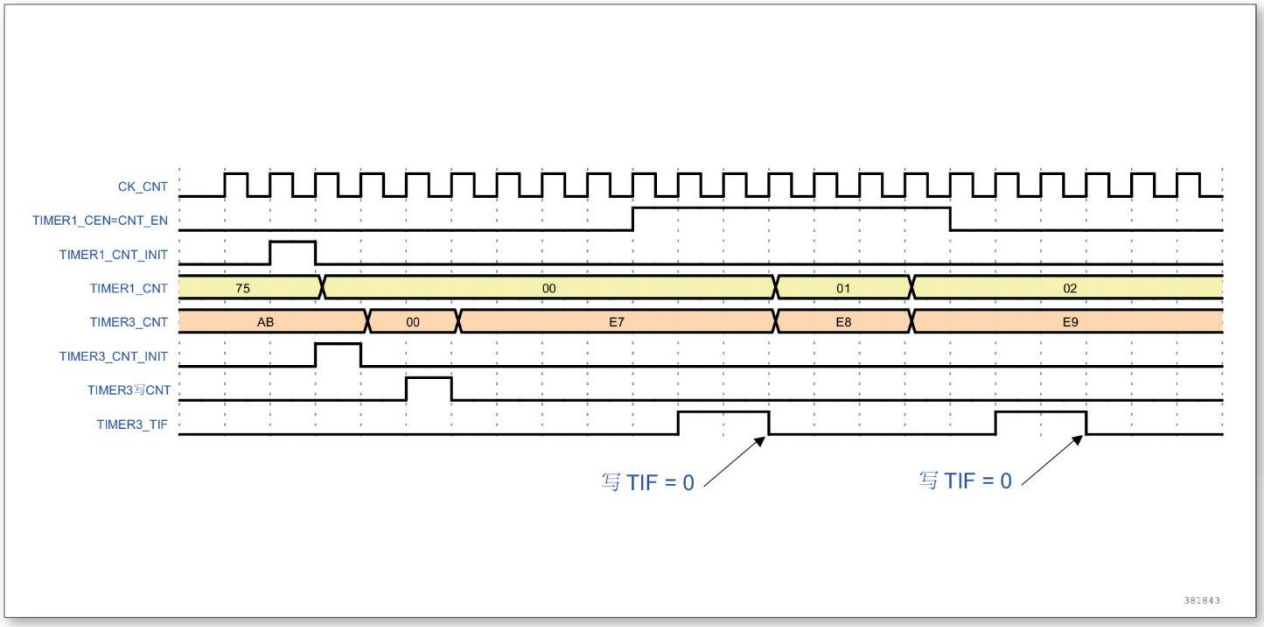

图 10-39 通过使能定时器 1 可以控制定时器 3

10.3.14.3 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 3。参考下图。当定时器 1 产生更新事件时，定时器 3 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 3 的 CEN 位被自动地置 1，同时计数器开始计数直到写 0 到 TIM3_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)。

- 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)。
- 配置定时器 1 的周期 (TIM1_ARR 寄存器)。
- 配置定时器 3 从定时器 1 获得输入触发 (TIM3_SMCR 寄存器的 TS = 000) 。
- 配置定时器 3 为触发模式 (TIM3_SMCR 寄存器的 SMS = 110) 。
- 置 TIM1_CR1 寄存器的 CEN = 1 以启动定时器 1。

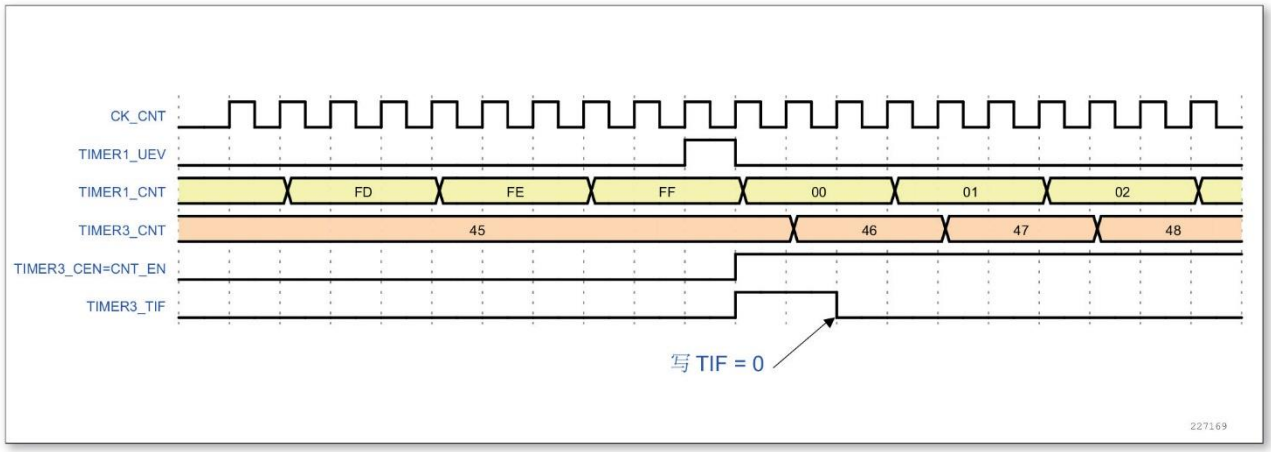

图 10-40 使用定时器 1 的更新触发定时器 3

在上一个例子中，可以在启动计数之前初始化两个计数器。下图显示在相同配置情况下，使用触发模式而不是门控模式 (TIM3_SMCR 寄存器的 SMS = 110) 的动作。

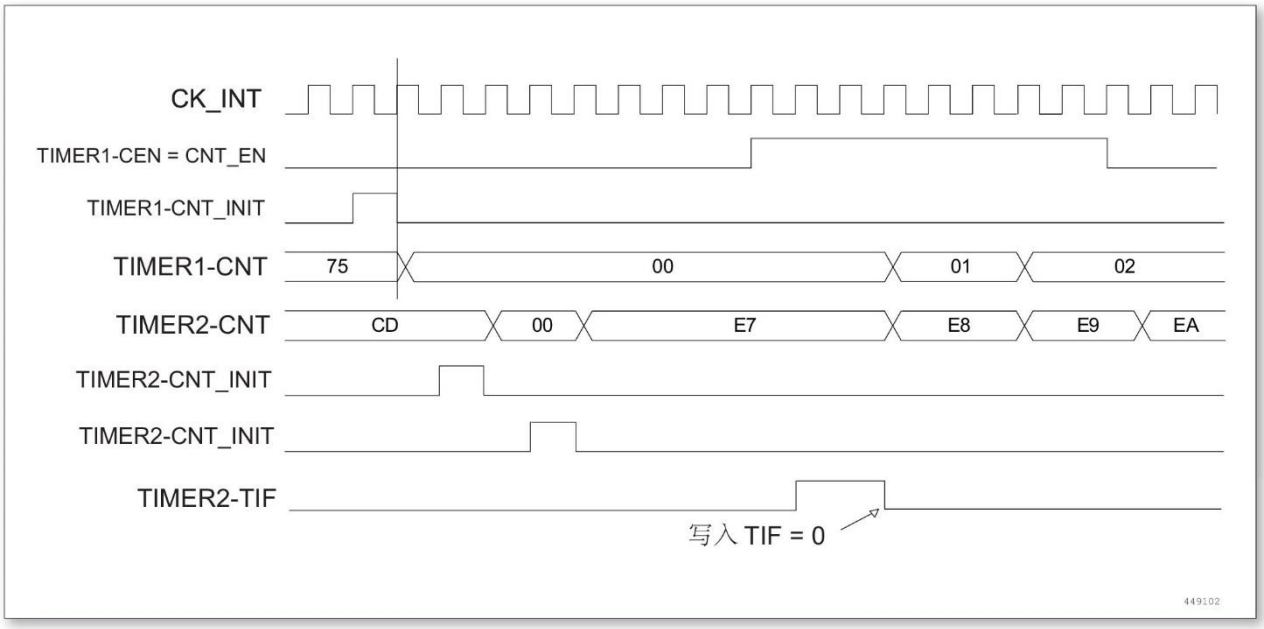

图 10-41 利用定时器 1 的使能触发定时器 3

10.3.14.4 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 的使能定时器 3。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 3 为主)：

- 配置定时器 1 为主模式，送出它的使能作为触发输出 (TIM1_CR2 寄存器的 MMS='001')
- 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')
- 配置定时器 1 为触发模式 (TIM1_SMCR 寄存器的 SMS='110')

- 配置定时器 3 从定时器 1 获得输入触发 (TIM3_SMCR 寄存器的 TS = 000)
- 配置定时器 3 为触发模式 (TIM3_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。

注：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 (TIMx_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。

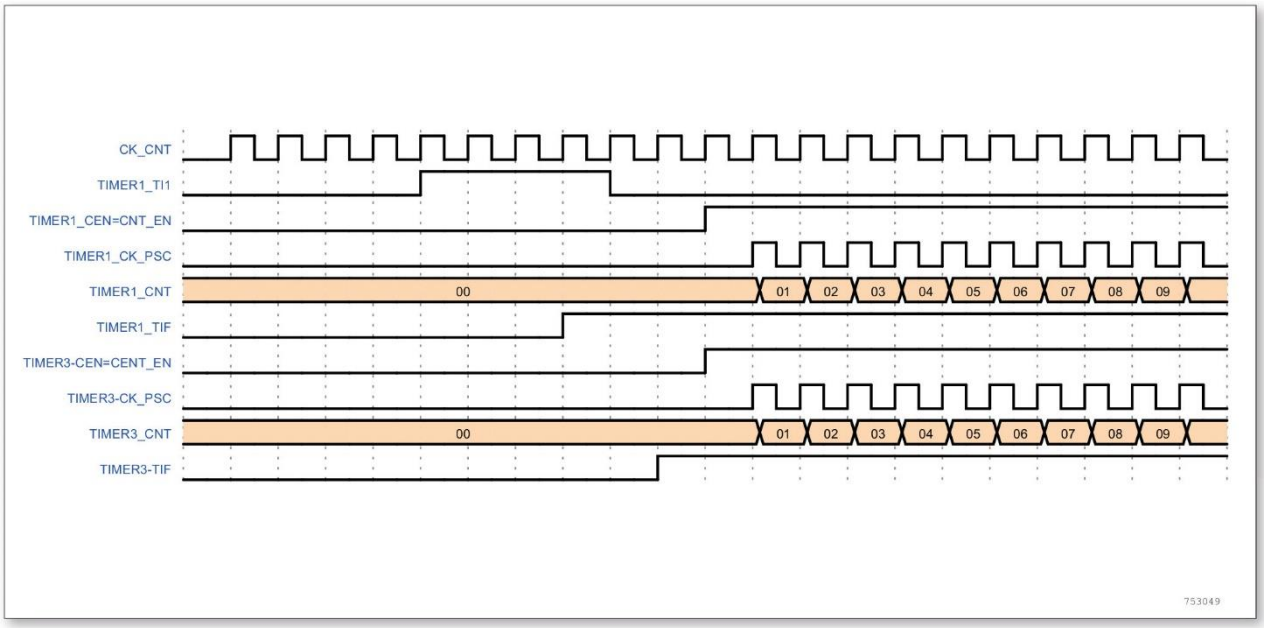

图 10-42 使用定时器 1 的 TI1 输入触发定时器 1 和定时器 3

10.3.15 调试模式

当微控制器进入调试模式时 (CPU 核心停止)，根据 DBG 模块中 DBG_TIM3_STOP 的设置，TIMx 计数器可以或者继续正常操作，或者停止。详见随后的调试章节。

10.4 寄存器

表 10-2 TIMx 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	TIMx_CR1	控制寄存器 1	0x0000
0x04	TIMx_CR2	控制寄存器 2	0x0000
0x08	TIMx_SMCR	从模式控制寄存器	0x0000
0x0C	TIMx_DIER	中断使能寄存器	0x0000

Offset	Acronym	Register Name	Reset
0x10	TIMx_SR	状态寄存器	0x0000
0x14	TIMx_EGR	事件产生寄存器	0x0000
0x18	TIMx_CCMR1	捕获/比较模式寄存器 1	0x0000
0x1C	TIMx_CCMR2	捕获/比较模式寄存器 2	0x0000
0x20	TIMx_CCER	捕获/比较使能寄存器	0x0000
0x24	TIMx_CNT	计数器	0x0000
0x28	TIMx_PSC	预分频率器	0x0000
0x2C	TIMx_ARR	自动装载寄存器	0x0000
0x34	TIMx_CCR1	捕获/比较寄存器 1	0x0000
0x38	TIMx_CCR2	捕获/比较寄存器 2	0x0000
0x3C	TIMx_CCR3	捕获/比较寄存器 3	0x0000
0x40	TIMx_CCR4	捕获/比较寄存器 4	0x0000
0x50	TIMx_OR	输入选项寄存器	0x0000

10.4.1 TIMx_CR1 控制寄存器 1

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
Reserved						CKD		ARPE	CMS		DIR	OPM	URS	UDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

Bit	Field	Description
15：10	Reserved	保留，必须保持复位值。
9：8	CKD	时钟分频（clock division） 定义定时器时钟（INT_CK）频率与死区时间计数器、数字滤波器（TIx）所用的时钟之间的分频比例。 00：$t_{DTS} = t_{INT\_CK}$ 01：$t_{DTS} = 2x\ t_{INT\_CK}$ 10：$t_{DTS} = 4x\ t_{INT\_CK}$ 11：保留，不要使用这个配置
7	ARPE	自动重装载预装载使能（Auto-reload preload enable） 0：关闭 TIMx_ARR 寄存器的影子寄存器 1：使能 TIMx_ARR 寄存器的影子寄存器

Bit	Field	Description
6：5	CMS	中央对齐模式选择（Center-aligned mode selection） 00：边沿对齐模式。计数方向取决于 DIR 位 01：中央对齐模式 1。计数器交替地递增和递减计数。通道为输出模式，只在计数器递减计数时比较中断标志位被置 1 10：中央对齐模式 2。计数器交替地递增和递减计数。通道为输出模式，只在计数器递增计数时比较中断标志位被置 1 11：中央对齐模式 3。计数器交替地递增和递减计数。通道为输出模式，在计数器递增和递减计数时比较中断标志位均被置 1 注：计数过程中，不允许更改此位。
4	DIR	计数方向（Direction） 0：计数器递增计数 1：计数器递减计数 注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。
3	OPM	单脉冲模式（One pulse mode） 0：禁止单脉冲模式，在发生更新事件时，计数器继续计数 1：使能单脉冲模式，在发生下一次更新事件或软件清除 CEN 位时，计数器停止计数
2	URS	更新请求源（Update request source） 软件配置该位，选择更新事件源。 0：以下事件可产生一个更新中断请求： - 计数器上溢/下溢 - 设置 UG 位 - 从模式控制器产生的更新 1：只有计数器上溢/下溢才产生一个更新中断请求
1	UDIS	禁止更新（Update disable） 该位用来允许或禁止更新事件的产生 0：允许更新事件（UEV） 1：禁止更新事件。不产生更新事件，影子寄存器（ARR、PSC、CCRx）保持值不变。如果设置了 EGR_UG 位为 1，或者从模式控制器接收到硬件复位，计数器和预分频器被初始化。
0	CEN	计数器使能（Counter enable） 0：禁止计数器 1：使能计数器 注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

10.4.2 TIMx_CR2 控制寄存器 2

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
Reserved								TI1S	MMS			Reserved			
								rw	rw						

Bit	Field	Description
15：8	Reserved	保留，必须保持复位值。
7	TI1S	TI1 选择（TI1 selection） 0：TIMx_CH1 管脚连到 TI1 输入 1：TIMx_CH1、TIMx_CH2 和 TIMx_CH3 管脚经异或后作为 TI1 输入
6：4	MMS	主模式选择（Master mode selection） 这些位控制 TRGO 信号的选择，用于选择在主模式下送到从定时器的同步信息： 000：复位　TIMx_EGR 寄存器的 UG 位触发一次 TRGO 脉冲。 001：使能 用于控制在一定时间内使能从定时器或同时启动多个定时器。计数器使能信号 CNT_EN 被用于作为触发输出（TRGO），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 TRGO。 011：捕获/比较脉冲　发生一次捕获或一次比较成功时，触发输出送出一个 TRGO 信号。 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）
3：0	Reserved	保留，必须保持复位值。

10.4.3 TIMx_SMCR 从模式控制寄存器

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
Reserved								MSM	TS			Res.	SMS		
								rw	rw				rw		

Bit	Field	Description
15：8	Reserved	保留，必须保持复位值。

Bit	Field	Description
7	MSM	主/从 模 式（Master/slave mode） 0：无作用 1：触发输入（TRGI） 事件被延迟，以实现当前定时器（通过 TRGO） 与它的从定时器间的完美同步，该功能可以把几个定时器同步到一个单一的外部事件。
6：4	TS[2:0]	触发选择（Trigger selection） 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：保留 更多有关 ITRx 的细节，参见下表。 注：从模式使能后这些位不能修改。
3	Reserved	保留，必须保持复位值。
2：0	SMS	从模式选择（Slave mode selection） 当选择了外部信号作为触发源，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 的边沿递增/递减计数。 010：编码器模式 2- 根据 TI2FP2 的电平，计数器在 TI1FP1 的边沿递增/递减计数。 011：编码器模式 3 - 根据另一个输入的电平，计数器在 TI1FP1 和 TI2FP2 的边沿递增/递减计数。 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，并且产生一个更新事件。 101：门控模式 - 当触发输入（TRGI） 为高时，计数器开始计数。当触发输入变为低时，计数器停止计数（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。 注：如果 TI1F_ED 被选为触发输入（TS = 100） 时，不要使用门控模式。这是因为，TI1F_ED 在每次 TI1F 变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。

表 10-3 TIMx 内部触发连接

从定时器	ITR0	ITR1	ITR2	ITR3
TIM3	TIM1	-	-	-

10.4.4 TIMx_DIER 中断使能寄存器

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
Reserved									TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
									rw		rw	rw	rw	rw	rw

Bit	Field	Description
15：7	Reserved	保留，必须保持复位值。
6	TIE	允许触发中断（Trigger interrupt enable） 0：禁止触发中断 1：允许触发中断
5	Reserved	保留，必须保持复位值。
4	CC4IE	允许捕获/比较 4 中断（Capture/Compare 4 interrupt enable） 0：禁止捕获/比较 4 中断 1：允许捕获/比较 4 中断
3	CC3IE	允许捕获/比较 3 中断（Capture/Compare 3 interrupt enable） 0：禁止捕获/比较 3 中断 1：允许捕获/比较 3 中断
2	CC2IE	允许捕获/比较 2 中断（Capture/Compare 2 interrupt enable） 0：禁止捕获/比较 2 中断 1：允许捕获/比较 2 中断
1	CC1IE	允许捕获/比较 1 中断（Capture/Compare 1 interrupt enable） 0：禁止捕获/比较 1 中断 1：允许捕获/比较 1 中断
0	UIE	允许更新事件中断（Update interrupt enable） 0：禁止更新事件中断 1：允许更新事件中断

10.4.5 TIMx_SR 状态寄存器

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
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved		TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			r_w0c						r_w0c		r_w0c				

Bit	Field	Description
15：13	Reserved	保留，必须保持复位值。
12	CC4OF	捕获/比较 4 重复捕获标记（Capture/Compare 4 overcapture flag） 参考 CC1OF 描述。
11	CC3OF	捕获/比较 3 重复捕获标记（Capture/Compare 3 overcapture flag） 参考 CC1OF 描述。
10	CC2OF	捕获/比较 2 重复捕获标记（Capture/Compare 2 overcapture flag） 参考 CC1OF 描述。
9	CC1OF	捕获/比较 1 重复捕获标记（Capture/Compare 1 overcapture flag） 仅当通道 1 被配置为输入捕获，CC1IF 已经为 1 后，捕获事件再次发生时，该标记可由硬件置 1。写 0 可清除该位。 0：无重复捕获产生 1：重复捕获产生
8：7	Reserved	保留，必须保持复位值。
6	TIF	触发器中断标记（Trigger 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。它由软件清 0。 0：无触发器事件产生 1：触发器中断产生
5	Reserved	保留，必须保持复位值。
4	CC4IF	捕获/比较 4 中断标记（Capture/Compare 4 interrupt flag） 参考 CC1IF 描述。
3	CC3IF	捕获/比较 3 中断标记（Capture/Compare 3 interrupt flag） 参考 CC1IF 描述。
2	CC2IF	捕获/比较 2 中断标记（Capture/Compare 2 interrupt flag） 参考 CC1IF 描述。
1	CC1IF	捕获/比较 1 中断标记（Capture/Compare 1 interrupt flag） 通道 1 为输出模式： 当计数器值与比较值匹配时该位由硬件置 1（在中央对齐模式下根据 TIMx_CR1.CMS[1:0]的选择来置位）。它由软件清 0。 0：无匹配发生 1：TIMx_CNT 的值与 TIMx_CCR1 的值匹配 通道 1 为输入模式： 当发生捕获事件时该位由硬件置 1，由软件清 0 或读取 TIMx_CCR1 的值清 0。 0：无输入捕获产生 1：计数器值已被捕获至 TIMx_CCR1

Bit	Field	Description
0	UIF	更新中断标记（Update interrupt flag） 当产生更新事件时该位由硬件置 1。它由软件清 0。 0：无更新中断发生 1：发生更新中断 当寄存器被更新时该位由硬件置 1： - 若 TIMx_CR1 寄存器的 UDIS=0，且 REP_CNT=0，当计数器产生上溢/下溢事件时。 -若 TIMx_CR1 寄存器的 UDIS=0、URS=0，当 TIMx_EGR 寄存器的 UG=1 时。 - 若 TIMx_CR1 寄存器的 UDIS=0、URS=0，从模式控制器产生更新事件时。

10.4.6 TIMx_EGR 事件产生寄存器

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
Reserved									TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
									w		w	w	w	w	w

Bit	Field	Description
15：7	Reserved	保留，必须保持复位值。
6	TG	产生触发事件（Trigger generation） 0：无动作 1：产生触发事件，TIMx_SR 寄存器的 TIF = 1，若开启对应的中断， 则产生相应的中断，由硬件自动清 0。
5	Reserved	保留，必须保持复位值。
4	CC4G	产生捕获/比较 4 事件（Capture/Compare 4 generation） 参考 CC1G 描述。
3	CC3G	产生捕获/比较 3 事件（Capture/Compare 3 generation） 参考 CC1G 描述。
2	CC2G	产生捕获/比较 2 事件（Capture/Compare 2 generation） 参考 CC1G 描述。

Bit	Field	Description
1	CC1G	产生通道 1 捕获/比较事件（Capture/Compare 1 generation） 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：通道 CC1 上产生一个捕获/比较事件： 若通道 CC1 配置为输出：CC1IF 置 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，CC1IF 置 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。
0	UG	产生更新事件（Update generation） 0：无动作 1：初始化计数器，并产生一个更新事件。由硬件自动清 0，如果选择了中央对齐或递增计数模式，计数器被清 0；如果选择递减计数模式，计数器将载入自动重载值。预分频计数器将同时被清除。

10.4.7 TIMx_CCMR1 捕获/比较模式寄存器 1

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
Res.	OC2M			OC2PE	OC2FE	CC2S		Res.	OC1M			OC1PE	OC1FE	CC1S	
IC2F				IC2PSC		CC2S		IC1F				IC1PSC		CC1S	
rw	rw			rw	rw	rw		rw	rw			rw	rw	rw	

通道可用于输入（捕获模式）或输出（比较模式），通道的方向由相应的 CCxS 定义。该寄存器 CCxS 以外其它位的作用在输入模式和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。

比较输出模式：

Bit	Field	Description
15	Reserved	保留，必须保持复位值。
14：12	OC2M	通道 2 比较输出模式（Output compare 2 mode） 参考 OC1M 的描述。
11	OC2PE	通道 2 比较输出预装载使能（Output compare 2 preload enable） 参考 OC1PE 的描述。
10	OC2FE	通道 2 比较输出快速使能（Output compare 2 fast enable） 参考 OC1FE 的描述。

Bit	Field	Description
9：8	CC2S	通道 2 捕获/比较选择（Capture/Compare 2 selection） 该位定义通道的方向和输入信号的选择，只有在通道关闭时这些位才可写入： 00：通道 2 被配置为输出 01：通道 2 被配置为输入，IC2 映射在 TI2 上 10：通道 2 被配置为输入，IC2 映射在 TI1 上 11：通道 2 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）
7	Reserved	保留，必须保持复位值。
6：4	OC1M	通道 1 比较输出模式（Output compare 1 mode） 该位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1 的值。OC1REF 是高电平有效，而 OC1 的有效电平取决于 CC1P 位。 000：冻结。TIMx_CCR1 与 TIMx_CNT 间的比较结果对 OC1REF 不起作用。 001：匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时，强制 OC1REF 为高电平。 010：匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时，强制 OC1REF 为低电平。 011：匹配时翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。 100：强制为低。强制 OC1REF 为低电平。 101：强制为高。强制 OC1REF 为高电平。 110：PWM 模式 1。在递增计数时，当 TIMx_CNT<TIMx_CCR1 时强制 OC1REF 为高电平，否则为低电平;在递减计数时，当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为低电平，否则为高电平。 111：PWM 模式 2。在递增计数时，当 TIMx_CNT<TIMx_CCR1 时通道 1 为强制 OC1REF 为低电平，否则为高电平;在递减计数时，当 TIMx_CNT > TIMx_CCR1 时强制 OC1REF 为高电平，否则为低电平。 注：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。
3	OC1PE	通道 1 比较输出预装载使能（Output compare 1 preload enable） 0：禁止 TIMx_CCR1 寄存器的预装载功能，写入 TIMx_CCR1 寄存器的数值立即生效。 1：开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1 的预装载值在更新事件到来时生效。 注：若该位置 1，在单脉冲模式下（TIMx_CR1 寄存器的 OPM= 1），是否设定预装载寄存器无影响；其它情况下，需要设定预装载寄存器，否则后续动作不确定。

Bit	Field	Description
2	OC1FE	通道 1 比较输出快速使能（Output compare 1 fast enable） 该位为 1 时，若通道配置为 PWM 模式，会加快捕获/比较输出对触发时间的响应。输出通道将触发输入信号的有效边沿的作用等同于发生了一次比较匹配，此时 OC 被设置为比较电平，与比较结果无关。 0：禁止通道 1 比较输出快速使能 1：开启通道 1 比较输出快速使能
1：0	CC1S	通道 1 捕获/比较选择（Capture/Compare 1 selection） 该位定义通道的方向和输入信号的选择，只有在通道关闭时这些位才可写入： 00：通道 1 被配置为输出 01：通道 1 被配置为输入，IC1 映射在 TI1 上 10：通道 1 被配置为输入，IC1 映射在 TI2 上 11：通道 1 被配置为输入，IC1 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）

输入捕获模式：

Bit	Field	Description
15：12	IC2F	输入捕获 2 滤波器（Input capture 2 filter） 参考 IC1F 的描述
11：10	IC2PSC	输入/捕获 2 预分频器（Input capture 2 prescaler） 参考 IC1PSC 的描述
9：8	CC2S	通道 2 捕获/比较选择（Capture/Compare 2 selection） 该位定义通道的方向和输入信号的选择，只有在通道关闭时这些位才可写入： 00：通道 2 被配置为输出 01：通道 2 被配置为输入，IC2 映射在 TI2 上 10：通道 2 被配置为输入，IC2 映射在 TI1 上 11：通道 2 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）

Bit	Field	Description
7：4	IC1F	通道 1 输入捕获滤波器（Input capture 1 filter） 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 IC1 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，以 f_{DTS} 采样 0001：采样频率 $f_{sampling}=f_{INT\_CK}$，N=2 0010：采样频率 $f_{sampling}=f_{INT\_CK}$，N=4 0011：采样频率 $f_{sampling}=f_{INT\_CK}$，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
3：2	IC1PSC	通道 1 输入/捕获预分频器（Input capture 1 prescaler） 该位定义了 IC1 的预分频系数。当 CC1E=0（TIMx_CCER 寄存器中） 时，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
1：0	CC1S	通道 1 捕获/比较选择（Capture/Compare 1 selection） 该位定义通道的方向和输入信号的选择，只有在通道关闭时这些位才可写入： 00：通道 1 被配置为输出 01：通道 1 被配置为输入，IC1 映射在 TI1 上 10：通道 1 被配置为输入，IC1 映射在 TI2 上 11：通道 1 被配置为输入，IC1 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）

10.4.8 TIMx_CCMR2 捕获/比较模式寄存器 2

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
Res.	OC4M			OC4PE	OC4FE	CC4S		Res.	OC3M			OC3PE	OC3FE	CC3S	
IC4F				IC4PSC		CC4S		IC3F				IC3PSC		CC3S	
rw	rw			rw	rw	rw		rw	rw			rw	rw	rw	

比较输出模式：

Bit	Field	Description
15	Reserved	保留，必须保持复位值。
14：12	OC4M	通道 4 比较输出模式（Output compare 4 mode） 参考 OC3M 的描述
11	OC4PE	通道 4 比较输出预装载使能（Output compare 4 preload enable） 参考 OC3PE 的描述
10	OC4FE	通道 4 比较输出快速使能（Output compare 4 fast enable） 参考 OC3FE 的描述
9：8	CC4S	通道 4 捕获/比较选择（Capture/Compare 4 selection） 该位定义通道的方向和输入信号的选择，只有在通道关闭时这些位才可写入： 00：通道 4 被配置为输出 01：通道 4 被配置为输入，IC4 映射在 TI4 上 10：通道 4 被配置为输入，IC4 映射在 TI3 上 11：通道 4 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）
7	Reserved	保留，必须保持复位值。

Bit	Field	Description
6：4	OC3M	通道 3 比较输出模式（Output compare 3 mode） 该位定义了输出参考信号 OC3REF 的动作，而 OC3REF 决定了 OC3 的值。OC3REF 是高电平有效，而 OC3 的有效电平取决于 CC3P 位。 000：冻结。TIMx_CCR3 与 TIMx_CNT 间的比较结果对 OC3REF 不起作用 001：匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时，强制 OC3REF 为高电平 010：匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR3 的值相同时，强制 OC3REF 为低电平 011：匹配时翻转。当 TIMx_CCR3=TIMx_CNT 时，翻转 OC3REF 的电平 100：强制为低。强制 OC3REF 为低电平 101：强制为高。强制 OC3REF 为高电平 110：PWM 模式 1。在递增计数时，当 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为高电平，否则为低电平;在递减计数时，当 TIMx_CNT > TIMx_CCR3 时强制 OC3REF 为低电平，否则为高电平。 111：PWM 模式 2。在递增计数时，当 TIMx_CNT<TIMx_CCR3 时强制 OC3REF 为低电平，否则为高电平；在递减计数时，当 TIMx_CNT>TIMx_CCR3 时强制 OC3REF 为高电平，否则为低电平。 注：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时，OC3REF 电平才改变。
3	OC3PE	通道 3 比较输出预装载使能（Output compare 3 preload enable） 0：禁止 TIMx_CCR3 寄存器的预装载功能，写入 TIMx_CCR3 寄存器的数值立即生效 1：开启 TIMx_CCR3 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR3 的预装载值在更新事件到来时生效 注：仅在单脉冲模式下（TIMx_CR1 寄存器的 OPM= 1），无需设定预装载寄存器，其它情况下，需要设定预装载寄存器，否则后续动作不确定。
2	OC3FE	通道 3 比较输出快速使能（Output compare 3 fast enable） 该位为 1 时，若通道配置为 PWM 模式，会加快捕获/比较输出对触发时间的响应。输出通道将触发输入信号的有效边沿的作用等同于发生了一次比较匹配，此时 OC 被设置为比较电平，与比较结果无关。 0：禁止通道 3 比较输出快速使能 1：开启通道 3 比较输出快速使能
1：0	CC3S	通道 3 捕获/比较选择（Capture/Compare 3 selection） 该位定义通道的方向和输入信号的选择，只有在通道关闭时这些位才可写入： 00：通道 3 被配置为输出 01：通道 3 被配置为输入，IC3 映射在 TI3 上 10：通道 3 被配置为输入，IC3 映射在 TI4 上 11：通道 3 被配置为输入，IC3 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）

输入捕获模式：

Bit	Field	Description
15：12	IC4F	输入捕获 4 滤波器（Input capture 4 filter） 参考 IC3F 的描述
11：10	IC4PSC	输入/捕获 4 预分频器（Input capture 4 prescaler） 参考 IC3PSC 的描述
9：8	CC4S	通道 4 捕获/比较选择（Capture/Compare 4 selection） 该位定义通道的方向和输入信号的选择，只有在通道关闭时这些位才可写入： 00：通道 4 被配置为输出 01：通道 4 被配置为输入，IC4 映射在 TI4 上 10：通道 4 被配置为输入，IC4 映射在 TI3 上 11：通道 4 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）
7：4	IC3F	通道 3 输入捕获滤波器（Input capture 3 filter） 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 IC1 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，以 f_{DTS} 采样 0001：采样频率 $f_{sampling}=f_{INT\_CK}$，N=2 0010：采样频率 $f_{sampling}=f_{INT\_CK}$，N=4 0011：采样频率 $f_{sampling}=f_{INT\_CK}$，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

Bit	Field	Description
3：2	IC3PSC	通道 3 输入/捕获预分频器（Input capture 3 prescaler） 该位定义了 IC3 的预分频系数。当 CC3E=0（TIMx_CCER 寄存器中） 时，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
1：0	CC3S	通道 3 捕获/比较选择（Capture/Compare 3 selection） 该位定义通道的方向和输入信号的选择，只有在通道关闭时这些位才可写入： 00：通道 3 被配置为输出 01：通道 3 被配置为输入，IC3 映射在 TI3 上 10：通道 3 被配置为输入，IC3 映射在 TI4 上 11：通道 3 被配置为输入，IC3 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）

10.4.9 TIMx_CCER 捕获/比较使能寄存器

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
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

Bit	Field	Description
15	CC4NP	通道 4 输入捕获极性（Capture 4 polarity） 参考 CC1NP 的描述。
14	Reserved	保留，必须保持复位值。
13	CC4P	通道 4 输入/捕获输出极性（Capture/Compare 4 output polarity） 参考 CC1P 的描述。
12	CC4E	通道 4 输入/捕获输出使能（Capture/Compare 4 output enable） 参考 CC1E 的描述。
11	CC3NP	通道 3 输入捕获极性（Capture 3 polarity） 参考 CC1NP 的描述。
10	Reserved	保留，必须保持复位值。
9	CC3P	通道 3 输入/捕获输出极性（Capture/Compare 3 output polarity） 参考 CC1P 的描述。

Bit	Field	Description
8	CC3E	通道 3 输入/捕获输出使能（Capture/Compare 3 output enable） 参考 CC1E 的描述。
7	CC2NP	通道 2 输入捕获极性（Capture 2 polarity） 参考 CC1NP 的描述。
6	Reserved	保留，必须保持复位值。
5	CC2P	通道 2 输入捕获输出极性（Capture/Compare 2 output polarity） 参考 CC1P 的描述。
4	CC2E	通道 2 输入/捕获输出使能（Capture/Compare 2 output enable） 参考 CC1E 的描述。
3	CC1NP	通道 1 输入捕获极性（Capture 1 polarity） 通道 1 配置为输出时，此位无效。 通道 1 配置为输入时，CC1P/CC1NP 配合使用定义了输入信号极性和电平，详细参考 ICx 极性/电平选择表。
2	Reserved	保留，必须保持复位值。
1	CC1P	通道 1 输入/捕获输出极性（Capture/Compare 1 output polarity） 通道 1 配置为输出时，此位定义了输出信号极性： 0：OC1 高电平有效 1：OC1 低电平有效 通道 1 配置为输入时，CC1P/CC1NP 配合使用定义了输入信号极性和电平，详细参考 ICx 极性/电平选择表。
0	CC1E	通道 1 输入/捕获输出使能（Capture/Compare 1 output enable） 通道 1 配置为输出时： 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： 该位决定了输入捕获功能是否启用。 0：捕获禁止 1：捕获使能

输入模式下，ICx 的极性/电平选择如下表所示：

表 10-4 ICx 极性/电平选择表

CCxP	CCxNP	ICx 极性/电平
0	0	上升沿有效/高电平有效
1	0	下降沿有效/低电平有效
1	1	上升沿或下降沿有效/低电平有效
0	1	保留

10.4.10 TIMx_CNT 计数器

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															
rw															

Bit	Field	Description
15：0	CNT	计数器的值（Count value）

10.4.11 TIMx_PSC 预分频器

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															
rw															

Bit	Field	Description
15：0	PSC	预分频器的值（Prescaler value） 计数器的时钟频率（ck_cnt）=f_{CK_PSC}/（PSC+1） 当发生更新事件时，PSC 的值装入当前预分频寄存器。

10.4.12 TIMx_ARR 自动预装载寄存器

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															
rw															

Bit	Field	Description
15：0	ARR	自动预装载值（Auto-reload value） 这些位定义了计数器的自动预装载值。当自动预装载的值为 0 时，计数器不工作。

10.4.13 TIMx_CCR1 捕获/比较寄存器 1

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															
rw															

Bit	Field	Description
15：0	CCR1	通道 1 捕获/比较的值（Capture/Compare 1 value） 通道 1 配置为输入： 上一次捕获事件发生时捕获的计数器值存放于 CCR1（此时 CCR1 寄存器为只读）。 通道 1 配置为输出： 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对应的当前捕获/比较影子寄存器中。否则只有当更新事件发生时，此预装载值才传输至对应的当前捕获/比较影子寄存器中。当前捕获/比较影子寄存器参与同计数器 TIMx_CNT 的比较，并将比较结果反映到 OC1 端口的输出信号上。

10.4.14 TIMx_CCR2 捕获/比较寄存器 2

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															
rw															

Bit	Field	Description
15：0	CCR2	通道 2 捕获/比较的值（Capture/Compare 2 value） 参考 CCR1 的描述。

10.4.15 TIMx_CCR3 捕获/比较寄存器 3

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															
rw															

Bit	Field	Description
15：0	CCR3	通道 3 捕获/比较的值（Capture/Compare 3 value） 参考 CCR1 的描述。

10.4.16 TIMx_CCR4 捕获/比较寄存器 4

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															
rw															

Bit	Field	Description
15：0	CCR4	通道 4 捕获/比较的值（Capture/Compare 4 value） 参考 CCR1 的描述。

10.4.17 TIMx_OR 输入选项寄存器

偏移地址：0x50

复位值：0x0000

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

Bit	Field	Description
15:2	Reserved	保留，必须保持复位值。
1:0	ETR_RMP	ETR 复用 00：保留 01：LSI 时钟输入 10：保留 11：OSCIN_128 分频时钟输入

11 TIM14 基本定时器

11.1 简介

基本定时器 TIM14 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。它适合多种用途，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 (输出比较和 PWM)。

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。

基本定时器 TIM14 是完全独立的，不共享任何资源。

11.2 主要特征

- 16 位自动装载计数器
- 16 位可编程 (可以实时修改) 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
- 独立通道
  - 输入捕获
  - 输出比较
  - PWM 生成 (边缘对齐模式)
- 如下事件发生时产生中断
  - 更新：计数器溢出，计数器初始化 (通过软件)
  - 输入捕获
  - 输出比较

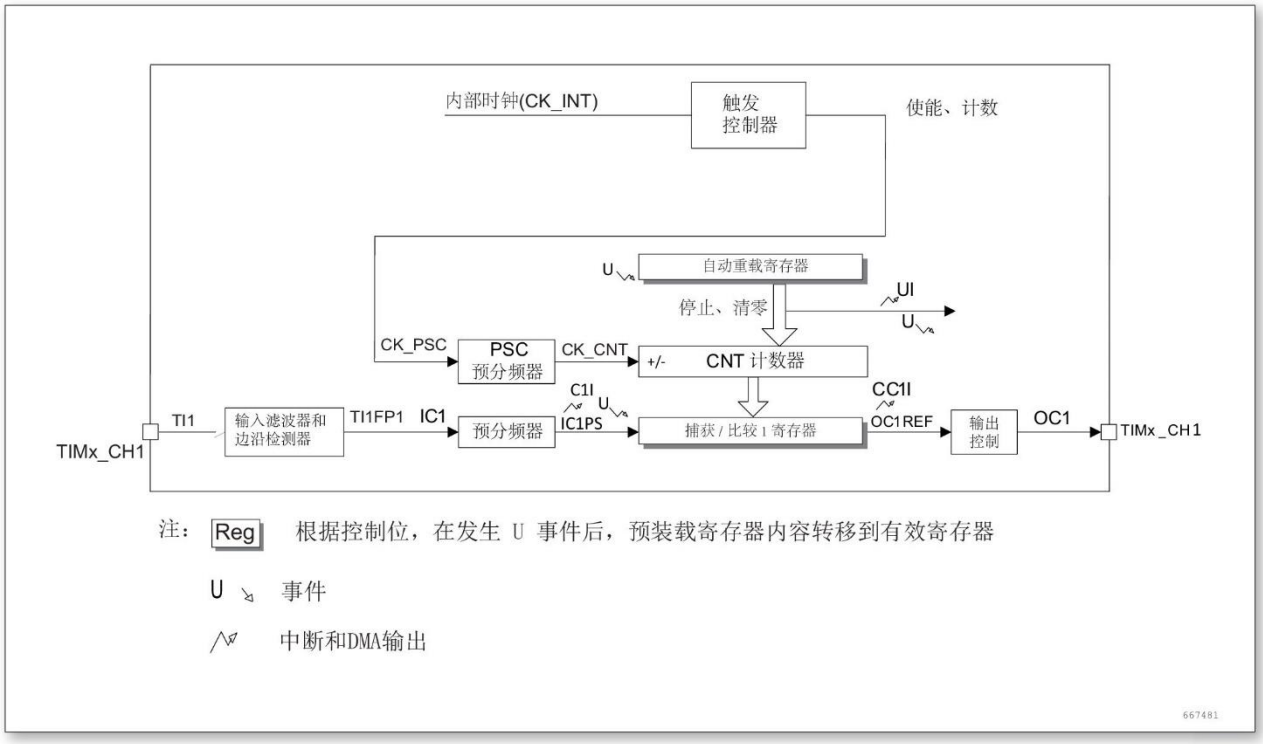

图 11-1 基本定时器框图

11.3 功能描述

11.3.1 时基单元

可编程基本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。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，在计数器运行时仍可以读写，时基单元包含：

- 计数器寄存器 (TIM14_CNT)
- 预分频器寄存器 (TIM14_PSC)
- 自动装载寄存器 (TIM14_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 TIM14_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并当 TIM14_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 TIM14_CR1 寄存器中的计数器使能位 (CEN) 时， CK_CNT 才有效。 (有关计数器使能的细节，请参见控制器的从模式描述)。

注：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。

11.3.1.1 预分频器描述

预分频器可以将计数器的时钟频率按 1 ～ 65536 之间的任意值分频。它是基于一个 (在 TIM14_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。新的预分频器的参数在下一次更新事件到来时被采用。

下面几张图分别给出了在预分频器运行时，更改计数器参数的例子。

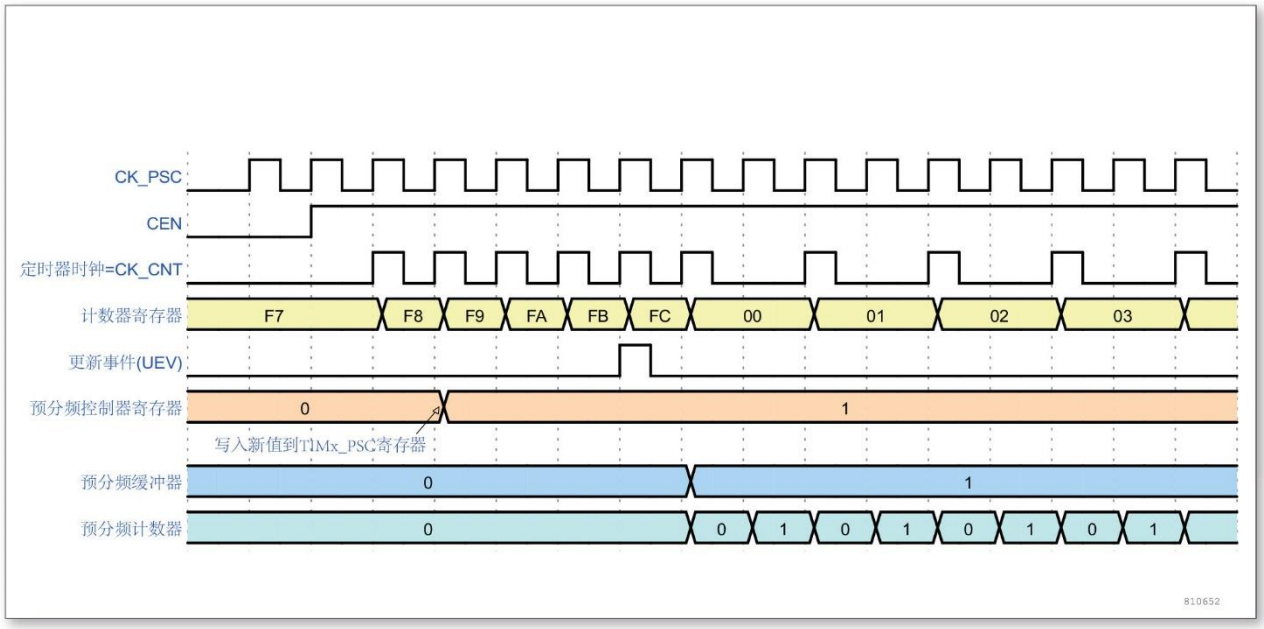

图 11-2 当预分频器的参数从 1 变到 2 时，计数器的时序图

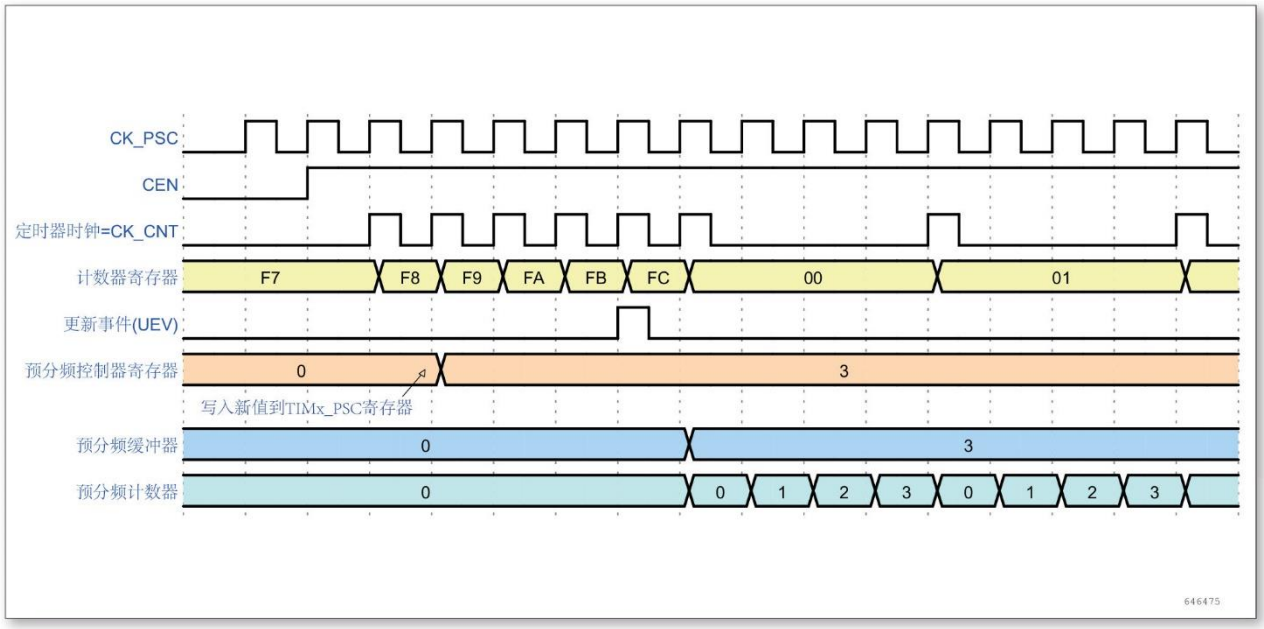

图 11-3 当预分频器的参数从 1 变到 4 时，计数器的时序图

11.3.2 计数模式

11.3.2.1 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 (TIM14_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。

在 TIM14_EGR 寄存器中设置 UG 位也同样可以产生一个更新事件。

设置 TIM14_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请 0(但预分频器的数值不变)。此外，如果设置了 TIM14_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 (TIM14_SR 寄存器中的 UIF 位)。

- 自动装载影子寄存器被重新置入预装载寄存器的值 (TIM14_ARR)
- 预分频器的缓冲区被置入预装载寄存器的值 (TIM14_PSC 寄存器的内容)

下图给出一些例子，当 TIM14_ARR = 0x36 时计数器在不同时钟频率下的动作：

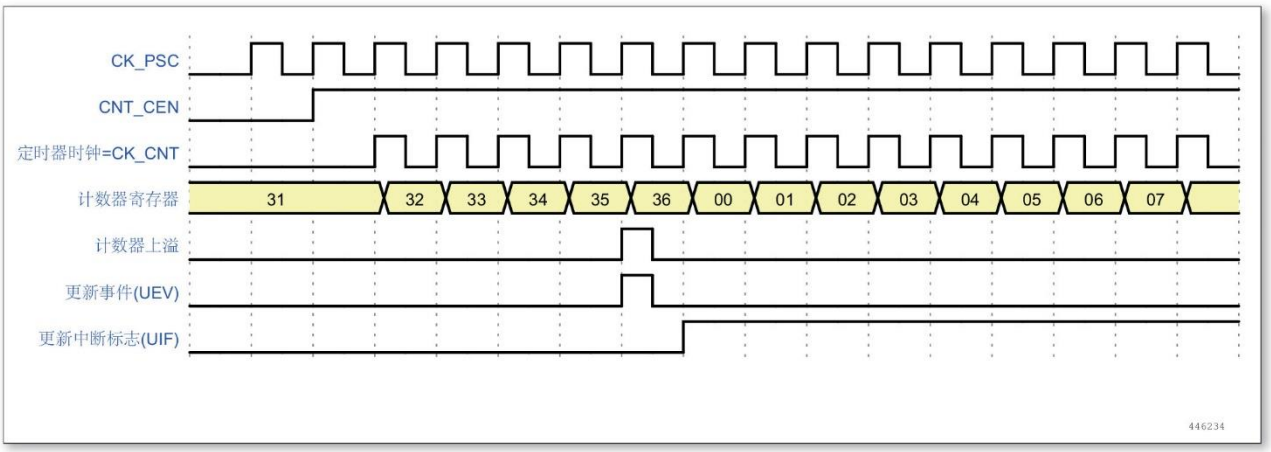

图 11-4 计数器时序图，内部时钟分频因子为 1

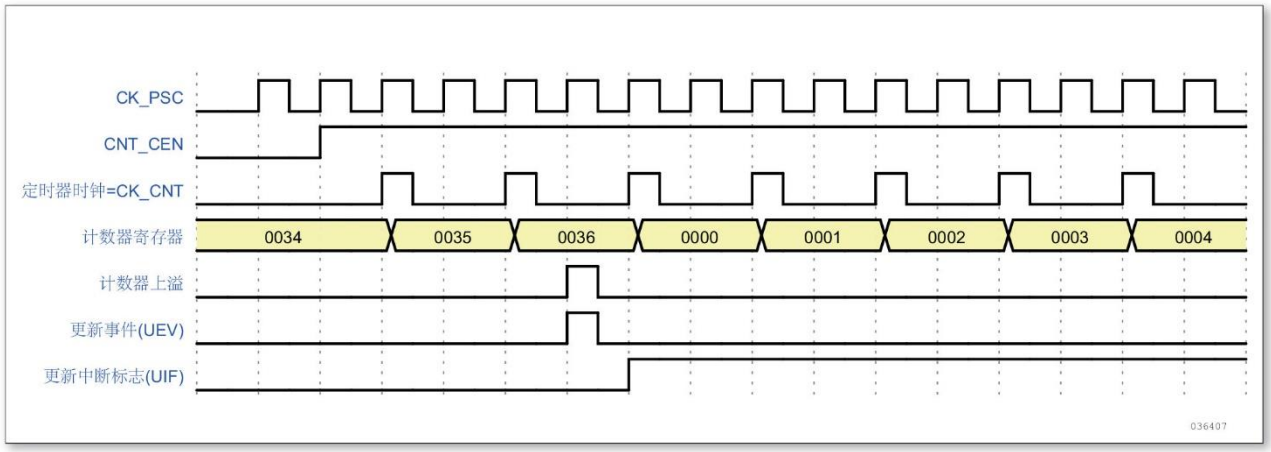

图 11-5 计数器时序图，内部时钟分频因子为 2

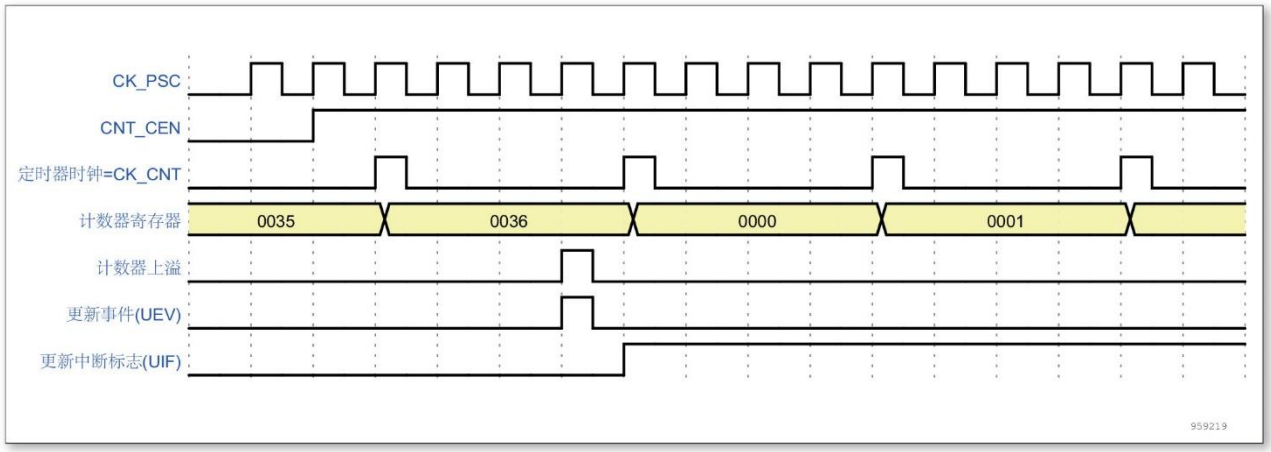

图 11-6 计数器时序图，内部时钟分频因子为 4

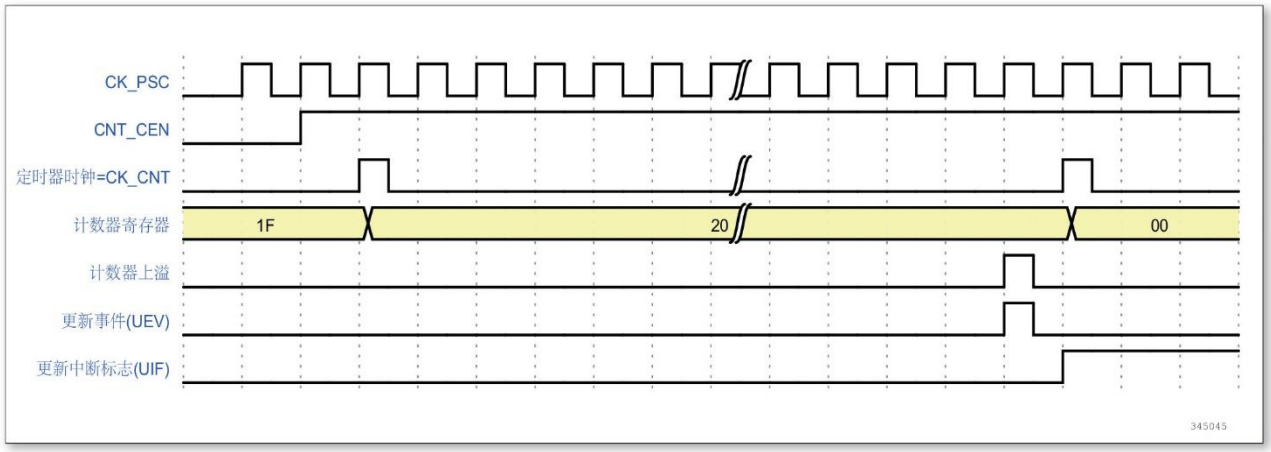

图 11-7 计数器时序图，内部时钟分频因子为 N

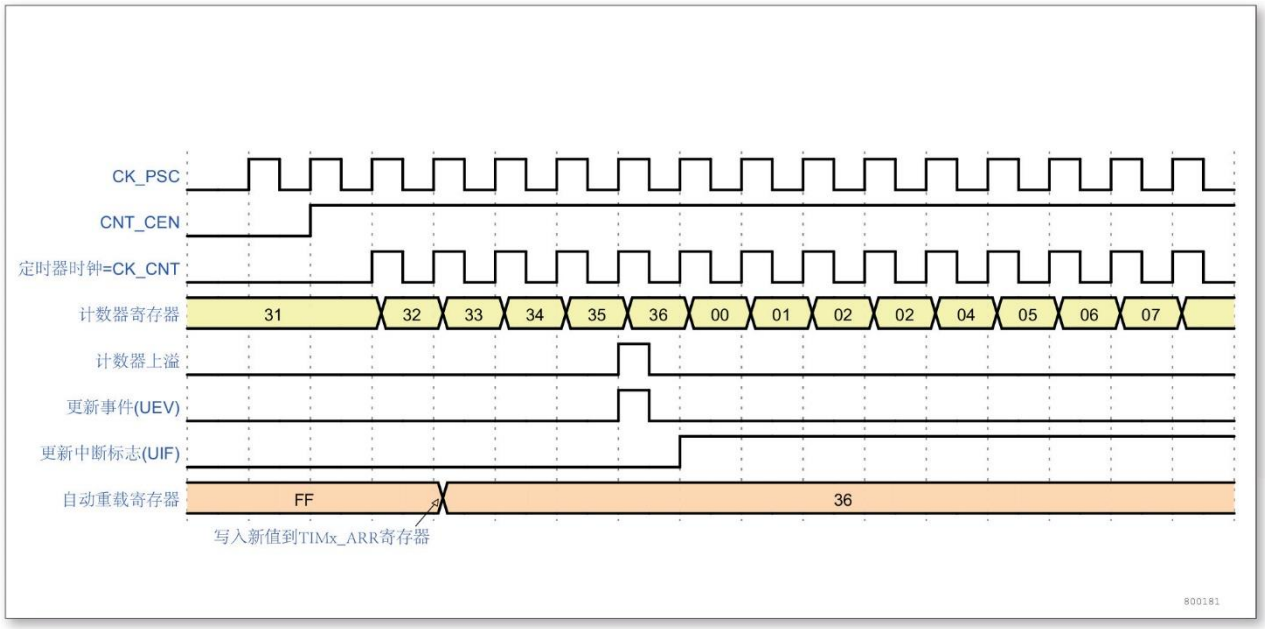

图 11-8 计数器时序图，当 ARPE = 0 时的更新事件 (TIM14_ARR 没有预装载)

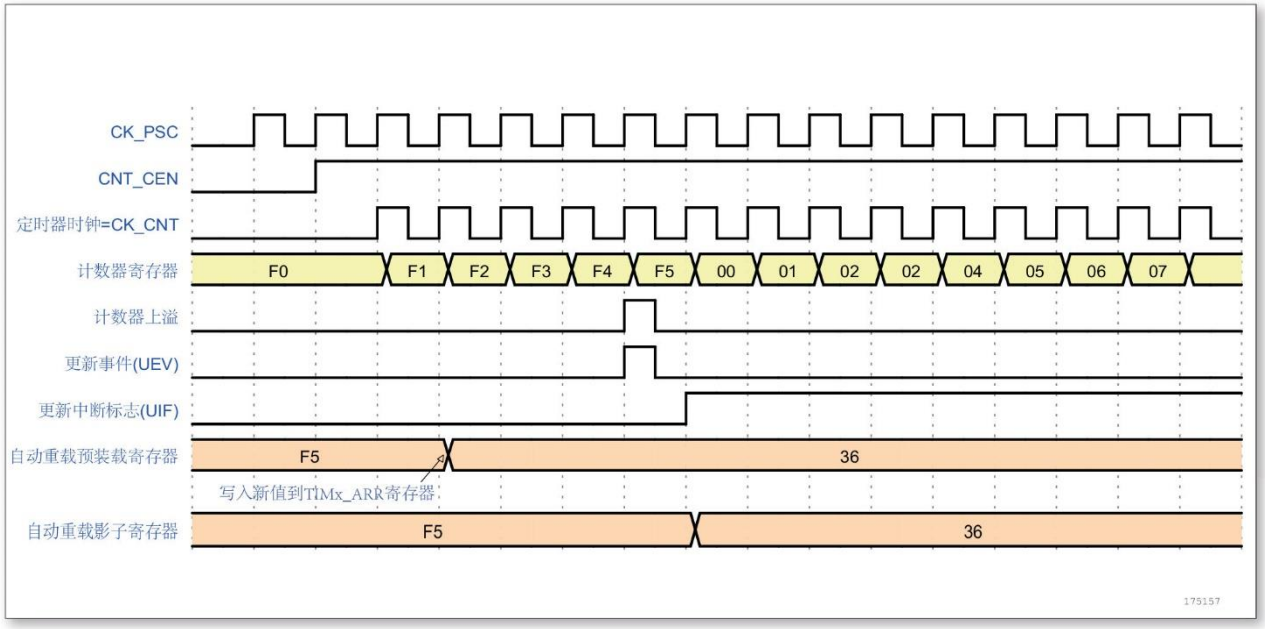

图 11-9 计数器时序图，当 ARPE = 1 时的更新事件 (预装载了 TIM14_ARR)

11.3.3 时钟源

计数器时钟由内部时钟 (CK_INT) 提供.

CEN(TIM14_CR1 寄存器) 和 UG 位 (TIM14_EGR 寄存器) 是事实上的控制位，并且只能被软件修改 (除了 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。

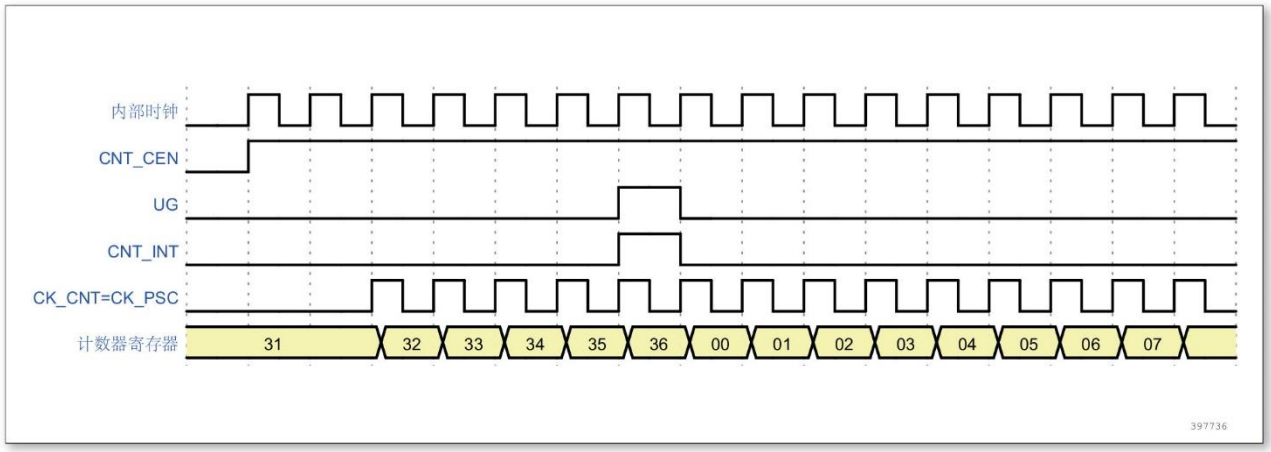

图 11-10 一般模式下的控制电路，内部时钟分频因子为 1

11.3.4 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缘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)。

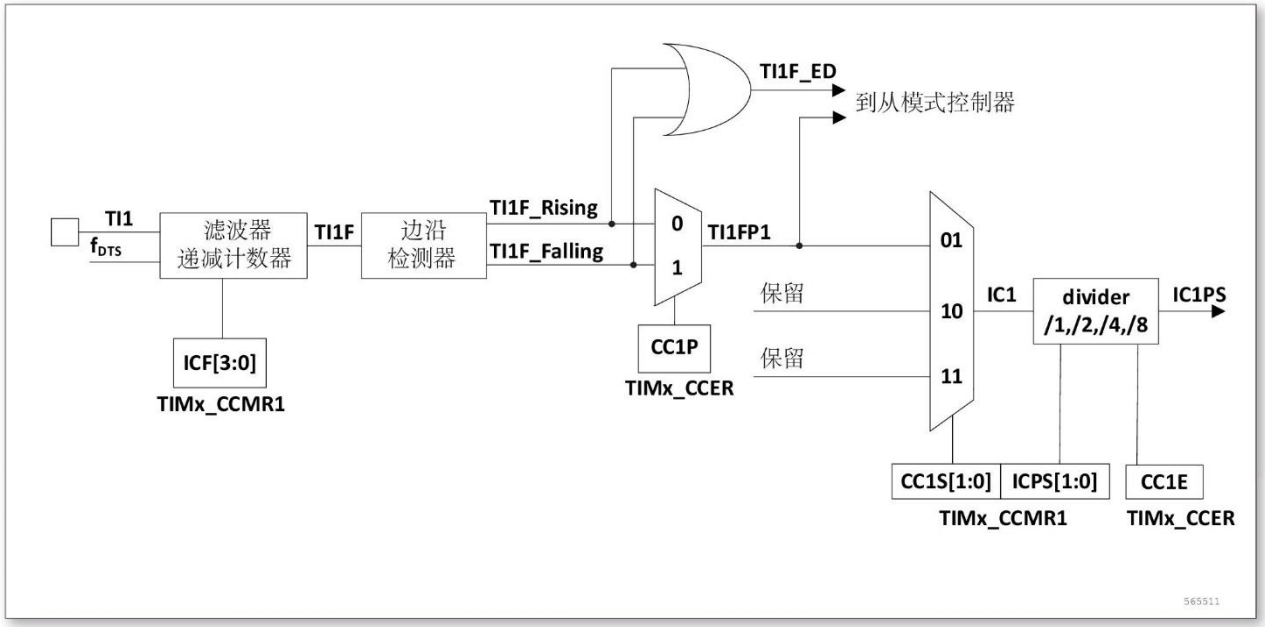

图 11-11 捕获/比较通道 (如：通道 1 输入部分)

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。

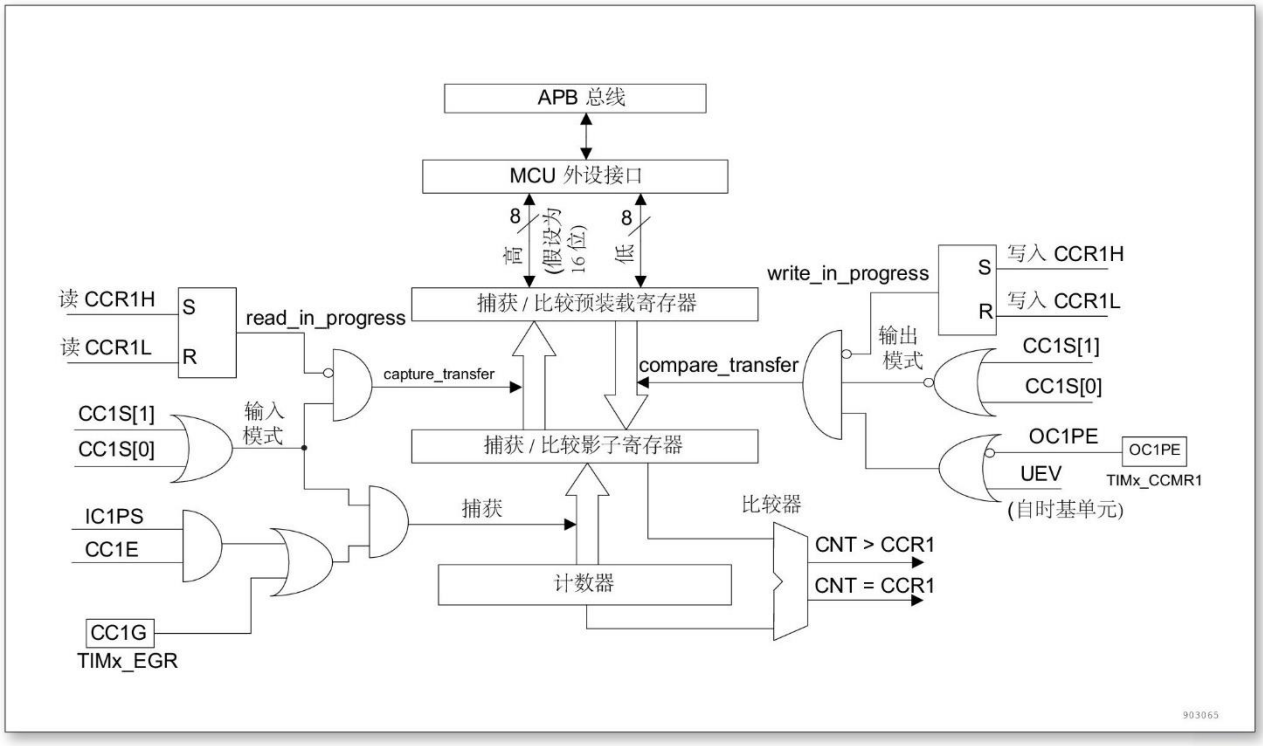

图 11-12 捕获/比较通道 1 的主电路

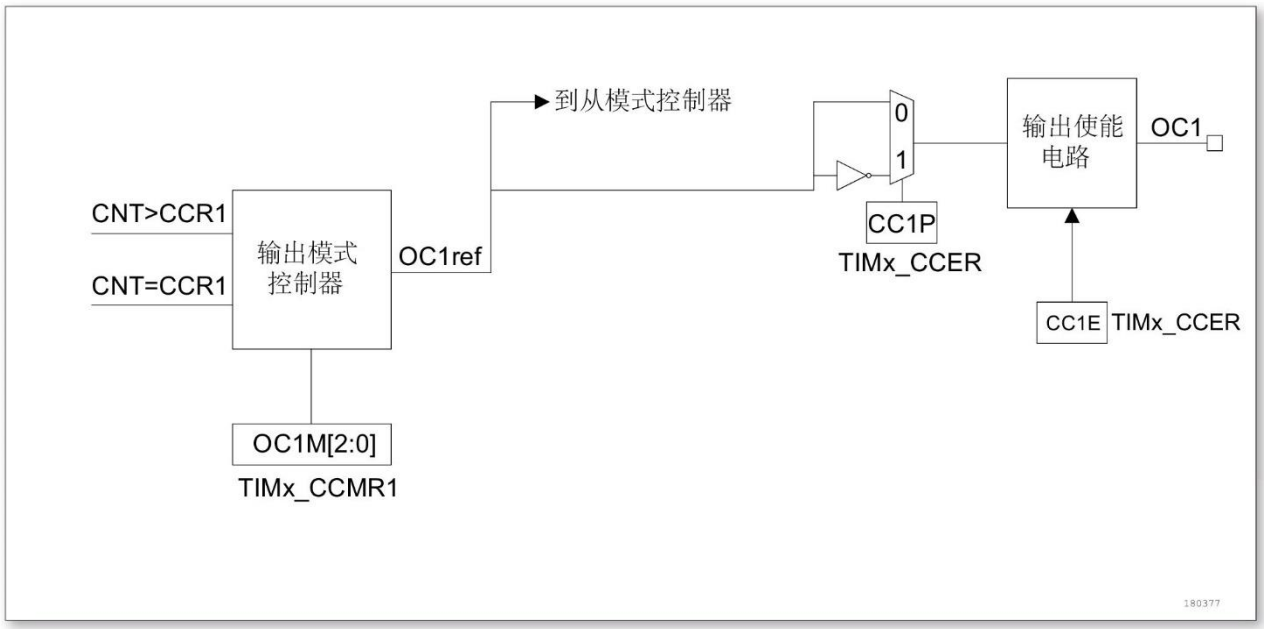

图 11-13 捕获/比较通道的输出部分 (通道 1)

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。

11.3.5 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 (TIM14_CCRx) 中。当捕获事件发生时，相应的 CCxIF 标志 (TIM14_SR 寄存器)被置 1，如果开放了中断，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(TIM14_SR 寄存器) 被置 1。写 CCxIF = 0 可清除 CCxIF，或读取存储在 TIM14_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 TIM14_CCR1 寄存器中，步骤如下：

- 选择有效输入端：TIM14_CCR1 必须连接到 TI1 输入，所以写入 TIM14_CCMR1 寄存器中的 CC1S = 01，当 CC1S 不为 00 时，通道被配置为输入，并且 TIM14_CCR1 寄存器变为只读。
- 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 TIM14_CCMRx 寄存器中的 ICxF 位)。假设输入信号在最多 5 个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 TIM14_CCMR1 寄存器中写入 IC1F = 0011。
- 选择 TI1 通道的有效转换边沿，在 TIM14_CCER 寄存器中写入 CC1P = 0 和 CC1NP= 0(上升沿)。
- 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 TIM14_CCMR1 寄存器的 IC1PS = 00)。
- 设置 TIM14_CCER 寄存器的 CC1E = 1，允许捕获计数器的值到捕获寄存器中。
- 如果需要，通过设置 TIM14_DIER 寄存器中的 CC1IE 位允许相关中断请求。

当一个输入捕获时:

- 产生有效的电平转换时，计数器的值被传送到 TIM14_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。

注：设置 TIM14_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断请求。

11.3.6 强制输出模式

在输出模式 (TIM14_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 TIM14_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 TIM14_CCMRx 寄存器中的 OCxM = 100，可强置 OCxREF 信号为低。该模式下，在 TIM14_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。

11.3.7 输出比较模式

此功能是用来控制一个输出波形或者指示何时一段给定的的时间已经到时。

当计数器与捕获/比较寄存器的内容相同时，输出比较功能做如下操作：

- 将输出比较模式 (TIM14_CCMRx 寄存器中的 OCxM 位) 和输出极性 (TIM14_CCER 寄存器中的 CCxP 位) 定义的值输出到对应的管脚上。在比较匹配时，输出管脚可以保持它的电平 (OCxM = 000)、被设置成有效电平 (OCxM = 001)、被设置成无有效电平(OCxM = 010) 或进行翻转 (OCxM = 011)。
- 设置中断状态寄存器中的标志位 (TIM14_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽 (TIM14_DIER 寄存器中的 CCXIE 位)，则产生一个中断。

TIM14_CCMRx 中的 OCxPE 位选择 TIM14_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：

- 选择计数器时钟 (内部，外部，预分频器)
- 将相应的数据写入 TIM14_ARR 和 TIM14_CCRx 寄存器中
- 如果要产生一个中断请求，设置 CCxIE 位
- 选择输出模式，
    - 写 OCxM = ‘011’，当计数器 CNT 与 CCRx 匹配时翻转 OCx 的输出引脚
    - 写 OCxPE = ‘0’，禁止预装载
    - 写 CCxP = ‘0’，选择高电平有效
    - 写 CCxE = ‘1’，允许输出
- 设置 TIM14_CR1 寄存器的 CEN 位启动计数器

TIM14_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 TIM14_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。

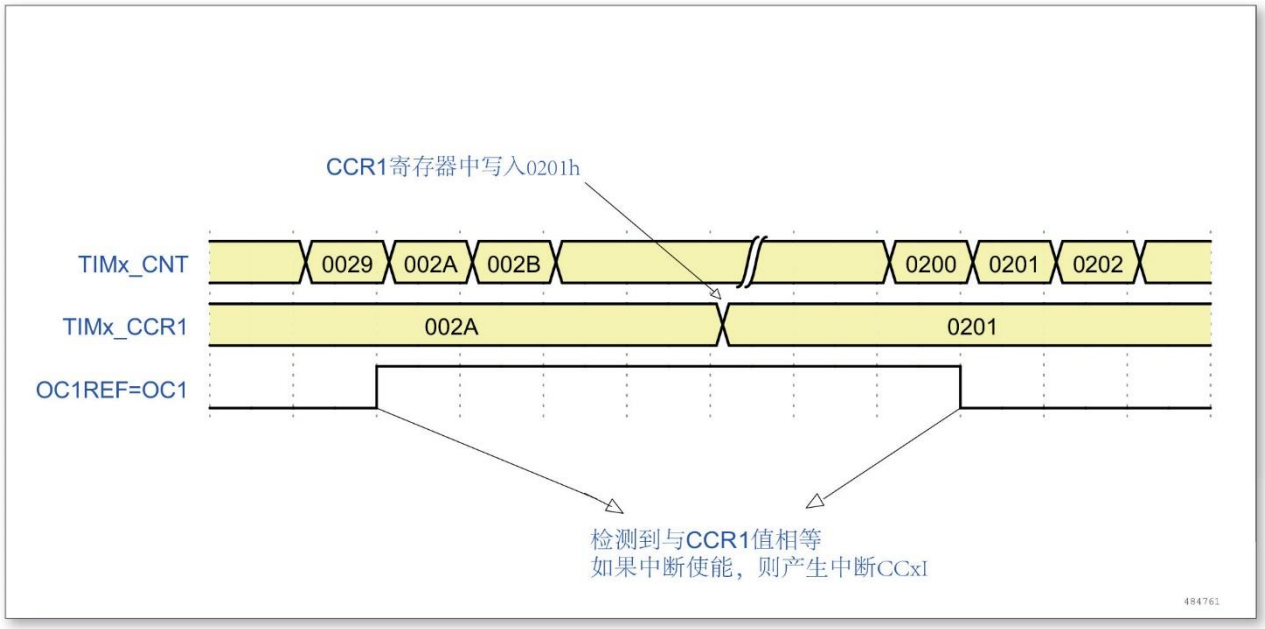

图 11-14 输出比较模式，翻转 OC1

11.3.8 PWM 模式

脉冲宽度调制模式可以产生一个由 TIM14_ARR 寄存器确定频率、由 TIM14_CCRx 寄存器确定占空比的信号。

在 TIM14_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 TIM14_CCMRx 寄存器 OCxPE 位以使能相应的预装载寄存器，最后还要设置 TIM14_CR1 寄存器的 ARPE 位使能自动重装载的预装载寄存器 (在向上计数模式中)。

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 TIM14_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 TIM14_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。 TIM14_CCER 寄存器中的 CCxE 位控制 OCx 输出使能。详见 TIM14_CCERx 寄存器的描述。

在 PWM 模式 (模式 1 或模式 2) 下， TIM14_CNT 和 TIM14_CCRx 始终在进行比较是否符合 TIM14_CNT ≤ TIM14_CCRx 。

因为本计数器是向上计数只能产生边沿对齐的 PWM 模式。

11.3.8.1 PWM 边沿对齐模式

下面是一个 PWM 模式 1 的例子。当 TIM14_CNT < TIM14_CCRx 时 PWM 信号参考 OCxREF 为高，否则为低。如果 TIM14_CCRx 中的比较值大于自动重装载值 (TIM14_ARR)，则 OCxREF 保持

为‘1’。如果比较值为 0，则 OCxREF 保持为‘0’。下图为 TIM14_ARR = 8 时边沿对齐的 PWM 波形实例。

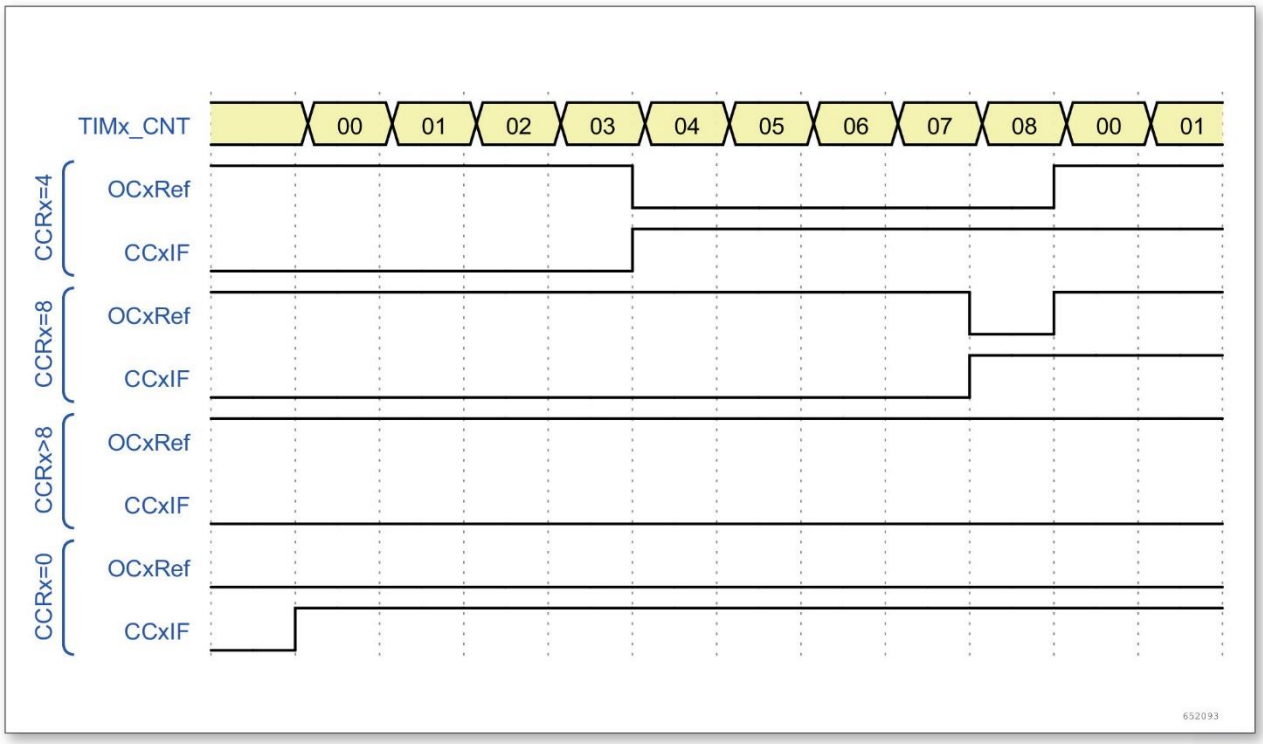

图 11-15 边沿对齐的 PWM 波形 (ARR = 8)

11.3.9 调试模式

当微控制器进入调试模式 (CPU 核心停止)，根据 DBG 模块中 DBG_TIM14_STOP 的设置， TIM14 计数器或者继续正常操作，或者停止。

11.4 寄存器

表 11-1 TIMx 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	TIMx_CR1	控制寄存器 1	0x0000
0x0C	TIMx_DIER	中断使能寄存器	0x0000
0x10	TIMx_SR	状态寄存器	0x0000
0x14	TIMx_EGR	事件产生寄存器	0x0000
0x18	TIMx_CCMR1	捕获/比较模式寄存器 1	0x0000
0x20	TIMx_CCER	捕获/比较使能寄存器	0x0000
0x24	TIMx_CNT	计数器	0x0000
0x28	TIMx_PSC	预分频率器	0x0000

Offset	Acronym	Register Name	Reset
0x2C	TIMx_ARR	自动装载寄存器	0x0000
0x34	TIMx_CCR1	捕获/比较寄存器 1	0x0000
0x44	TIMx_BDTR	刹车和死区寄存器	0x0000

11.4.1 TIMx_CR1 控制寄存器 1

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
Reserved						CKD		ARPE	Reserved				URS	UDIS	CEN
						rw		rw					rw	rw	rw

Bit	Field	Description
15：10	Reserved	保留，必须保持复位值。
9：8	CKD	时钟分频（clock division） 定义定时器时钟（INT_CK）频率与数字滤波器（TI1）所用的时钟之间的分频比例。 00：$t_{DTS} = t_{INT\_CK}$ 01：$t_{DTS} = 2x\ t_{INT\_CK}$ 10：$t_{DTS} = 4x\ t_{INT\_CK}$ 11：保留，不要使用这个配置
7	ARPE	自动重装载预装载使能（Auto-reload preload enable） 0：关闭 TIMx_ARR 寄存器的影子寄存器 1：使能 TIMx_ARR 寄存器的影子寄存器
6：3	Reserved	保留，必须保持复位值。
2	URS	更新请求源（Update request source） 软件配置该位，选择更新事件源。 0：以下事件可产生一个更新中断请求： - 计数器上溢 - 设置 UG 位 1：只有计数器上溢才产生一个更新中断请求
1	UDIS	禁止更新（Update disable） 该位用来允许或禁止更新事件的产生 0：允许更新事件（UEV） 1：禁止更新事件。不产生更新事件，影子寄存器（ARR、PSC、CCR1）保持值不变。如果设置了 EGR_UG 位为 1，计数器和预分频器被初始化。
0	CEN	计数器使能（Counter enable） 0：禁止计数器 1：使能计数器

11.4.2 TIMx_DIER 中断使能寄存器

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
Reserved														CC1IE	UIE
														rw	rw

Bit	Field	Description
15：2	Reserved	保留，必须保持复位值。
1	CC1IE	允许捕获/比较 1 中断（Capture/Compare 1 interrupt enable） 0：禁止捕获/比较 1 中断 1：允许捕获/比较 1 中断
0	UIE	允许更新事件中断（Update interrupt enable） 0：禁止更新事件中断 1：允许更新事件中断

11.4.3 TIMx_SR 状态寄存器

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
Reserved						CC1OF	Reserved							CC1IF	UIF
						r_w0c								r_w0c	r_w0c

Bit	Field	Description
15：10	Reserved	保留，必须保持复位值。
9	CC1OF	捕获/比较 1 重复捕获标记（Capture/Compare 1 overcapture flag） 仅当通道 1 被配置为输入捕获，CC1IF 已经为 1 后，捕获事件再次发生时，该标记可由硬件置 1。写 0 可清除该位。 0：无重复捕获产生 1：重复捕获产生
8：2	Reserved	保留，必须保持复位值。

Bit	Field	Description
1	CC1IF	捕获/比较 1 中断标记（Capture/Compare 1 interrupt flag） 通道 1 为输出模式： 当计数器值与比较值匹配时该位由硬件置 1，它由软件清 0。 0：无匹配发生 1：TIMx_CNT 的值与 TIMx_CCR1 的值匹配 通道 1 为输入模式： 当发生捕获事件时该位由硬件置 1，由软件清 0 或读取 TIMx_CCR1 的值清 0。 0：无输入捕获产生 1：计数器值已被捕获至 TIMx_CCR1
0	UIF	更新中断标记（Update interrupt flag） 当产生更新事件时该位由硬件置 1。它由软件清 0。 0：无更新中断发生 1：发生更新中断 更新事件包含计数器上溢、设置 UG=1。

11.4.4 TIMx_EGR 事件产生寄存器

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
Reserved														CC1G	UG
														w	w

Bit	Field	Description
15：2	Reserved	保留，必须保持复位值。
1	CC1G	产生通道 1 捕获/比较事件（Capture/Compare 1 generation） 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：通道 CC1 上产生一个捕获/比较事件： 若通道 CC1 配置为输出：CC1IF 置 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： CC1IF 置 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。
0	UG	产生更新事件（Update generation） 0：无动作 1：初始化计数器，并产生一个更新事件。由硬件自动清 0

11.4.5 TIMx_CCMR1 捕获/比较模式寄存器 1

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
Reserved								Res.	OC1M			OC1PE	Res.	CC1S	
								IC1F				IC1PSC		CC1S	
								rw	rw			rw	rw	rw	

通道可用于输入（捕获模式）或输出（比较模式），通道的方向由相应的 CC1S 定义。该寄存器 CC1S 以外其它位的作用在输入模式和输出模式下不同。OC1x 描述了通道在输出模式下的功能，IC1x 描述了通道在输入模式下的功能。

比较输出模式：

Bit	Field	Description
15：7	Reserved	保留，必须保持复位值。
6：4	OC1M	通道 1 比较输出模式（Output compare 1 mode） 该位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1 的值。OC1REF 是高电平有效，而 OC1 的有效电平取决于 CC1P 位。 000：冻结。TIMx_CCR1 与 TIMx_CNT 间的比较结果对 OC1REF 不起作用。 001：匹配时设置为高。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时，强制 OC1REF 为高电平。 010：匹配时设置为低。当 TIMx_CNT 的值与 TIMx_CCR1 的值相同时，强制 OC1REF 为低电平。 011：匹配时翻转，当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。 100：强制为低。强制 OC1REF 为低电平。 101：强制为高。强制 OC1REF 为高电平。 110：PWM 模式 1。在递增计数时，当 TIMx_CNT<TIMx_CCR1 时强制 OC1REF 为高电平，否则为低电平。 111：PWM 模式 2。在递增计数时，当 TIMx_CNT<TIMx_CCR1 时通道 1 为强制 OC1REF 为低电平，否则为高电平。 注：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在比较输出模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。

Bit	Field	Description
3	OC1PE	通道 1 比较输出预装载使能（Output compare 1 preload enable） 0：禁止 TIMx_CCR1 寄存器的预装载功能，写入 TIMx_CCR1 寄存器的数值立即生效。 1：开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1 的预装载值在更新事件到来时生效。 注：若该位置 1，在单脉冲模式下（TIMx_CR1 寄存器的 OPM= 1），是否设定预装载寄存器无影响；其它情况下，需要设定预装载寄存器，否则后续动作不确定。
2	Reserved	保留，必须保持复位值。
1：0	CC1S	通道 1 捕获/比较选择（Capture/Compare 1 selection） 该位定义通道的方向和输入信号的选择，只有在通道关闭时这些位才可写入： 00：通道 1 被配置为输出 01：通道 1 被配置为输入 10：保留 11：保留

输入捕获模式：

Bit	Field	Description
15：8	Reserved	保留，必须保持复位值。
7：4	IC1F	通道 1 输入捕获滤波器（Input capture 1 filter） 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 IC1 输入信号的采样频率和数字滤波器的长度。 0000：无滤波器，以 f_{DTS} 采样 0001：采样频率 $f_{sampling}=f_{INT\_CK}$，N=2 0010：采样频率 $f_{sampling}=f_{INT\_CK}$，N=4 0011：采样频率 $f_{sampling}=f_{INT\_CK}$，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

Bit	Field	Description
3：2	IC1PSC	通道 1 输入/捕获预分频器（Input capture 1 prescaler） 该位定义了 IC1 的预分频系数。当 CC1E=0（TIMx_CCER 寄存器中） 时，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
1：0	CC1S	通道 1 捕获/比较选择（Capture/Compare 1 selection） 该位定义通道的方向和输入信号的选择，只有在通道关闭时这些位才可写入： 00：通道 1 被配置为输出 01：通道 1 被配置为输入 10：保留 11：保留

11.4.6 TIMx_CCER 捕获/比较使能寄存器

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
Reserved												CC1NP	Res.	CC1P	CC1E
												rw		rw	rw

Bit	Field	Description
15：4	Reserved	保留，必须保持复位值。
3	CC1NP	通道 1 输入捕获极性（Capture 1 polarity） 通道 1 配置为输出时，此位无效。 通道 1 配置为输入时，CC1P/CC1NP 配合使用定义了输入信号极性和电平，详细参考 IC1 极性/电平选择表。
2	Reserved	保留，必须保持复位值。
1	CC1P	通道 1 输入/捕获输出极性（Capture/Compare 1 output polarity） 通道 1 配置为输出时，此位定义了输出信号极性： 0：OC1 高电平有效 1：OC1 低电平有效 通道 1 配置为输入时，CC1P/CC1NP 配合使用定义了输入信号极性和电平，详细参考 IC1 极性/电平选择表。

Bit	Field	Description
0	CC1E	通道 1 输入/捕获输出使能（Capture/Compare 1 output enable） 通道 1 配置为输出时： 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： 该位决定了输入捕获功能是否启用。 0：捕获禁止 1：捕获使能

输入模式下，IC1 的极性/电平选择如下表所示：

表 11-2 IC1 极性/电平选择表

CC1P	CC1NP	IC1 极性/电平
0	0	上升沿有效/高电平有效
1	0	下降沿有效/低电平有效
1	1	上升沿或下降沿有效/低电平有效
0	1	保留

11.4.7 TIMx_CNT 计数器

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															
rw															

Bit	Field	Description
15：0	CNT	计数器的值（Count value）

11.4.8 TIMx_PSC 预分频器

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															
rw															

Bit	Field	Description
15：0	PSC	预分频器的值（Prescaler value） 计数器的时钟频率（ck_cnt）=f_{CK_PSC}/（PSC+1） 当发生更新事件时，PSC 的值装入当前预分频寄存器。

11.4.9 TIMx_ARR 自动预装载寄存器

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															
rw															

Bit	Field	Description
15：0	ARR	自动预装载值（Auto-reload value） 这些位定义了计数器的自动预装载值。当自动预装载的值为 0 时，计数器不工作。

11.4.10 TIMx_CCR1 捕获/比较寄存器 1

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															
rw															

Bit	Field	Description
15：0	CCR1	通道 1 捕获/比较的值（Capture/Compare 1 value） 通道 1 配置为输入： 上一次捕获事件发生时捕获的计数器值存放于 CCR1（此时 CCR1 寄存器为只读）。 通道 1 配置为输出： 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对应的当前捕获/比较影子寄存器中。否则只有当更新事件发生时，此预装载值才传输至对应的当前捕获/比较影子寄存器中。当前捕获/比较影子寄存器参与同计数器 TIMx_CNT 的比较，并将比较结果反映到 OC1 端口的输出信号上。

11.4.11 TIMx_BDTR 刹车和死区寄存器

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	Reserved														
rw															

Bit	Field	Description
15	MOE	主输出使能 （Main output enable） 0：禁止 OC1 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 CC1E），则开启 OC1 输出
14：0	Reserved	保留，必须保持复位值。

12 IWDG 独立看门狗

12.1 IWDG 简介

内置独立看门狗，提供了更高的安全性、时间的精确性和使用的灵活性。可用来检测和解决由软件错误引起的故障，当计数器达到给定的超时值时，产生系统复位。独立看门狗 (IWDG) 由专门的低速时钟 (LSI) 驱动，即使主时钟发生故障它也仍然有效。

IWDG 最适合应用于那些需要看门狗作为一个正在主程序外，能够完全独立工作，并且对时间精度要求低的场合。

12.2 IWDG 主要性能

- 自由运行的递减计数器
- 时钟由独立的振荡器提供 (可在停机模式下工作)
- 看门狗被激活后，则在计数器计数至 0x0000 时产生复位。

12.3 IWDG 功能描述

下图为独立看门狗模块的功能框图。

在键寄存器 (IWDG_KR) 中写入 0xCCCC。开始启动独立看门狗；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 (IWGD_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。

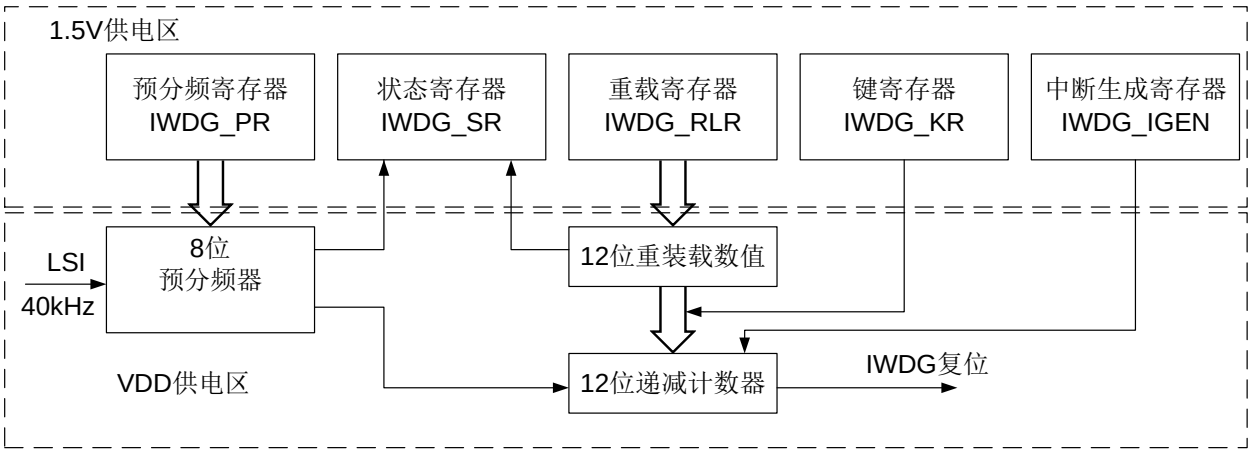

图 12-1 独立看门狗框图

注：看门狗功能处于 VDD 供电区，即在停机模式时仍能正常工作。

表 12-1 IWDG 超时时间（ 以 LSI 时钟频率 40kHz 为例）

预分频系数	PR[2:0] 位	最短时间（ms） RL[11:0]=0x000	最长时间（ms）
/4	0	0.1	409.6
/8	1	0.2	819.2
/16	2	0.4	1638.4
/32	3	0.8	3276.8
/64	4	1.6	6553.6
/128	5	3.2	13107.2
/256	（6 或 7）	6.4	26214.4

注：这些时间是按照 40KHz 时钟给出。实际上，MCU 内部的振荡器频率会在 30KHz 到 60KHz 之间变化。此外，即使振荡器的频率是精确的，确切的时序仍然依赖于 APB 接口时钟与振荡器时钟之间的相位差，因此总会有一个完整的振荡器周期是不确定的。

12.3.1 硬件看门狗

如果用户在选择字节中 (请参考“嵌入式闪存”章节) 启动了‘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；如果在计数器计数结束前，若软件没有向键寄存器写入相应的值，则系统会产生复位。

12.3.2 寄存器访问保护

IWDG_PR、IWDG_RLR 和 IWDG_IGEN 寄存器具有写保护功能。要修改这三个寄存器的值，必须先向 IWDG_KR 寄存器中写入 0x5555。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 0xAAAA) 也会启动写保护功能。状态寄存器指示预分频值和递减计数器是否正在被更新。

12.3.3 调试模式

当微控制器进入调试模式时 (CPU 核心停止)，根据调试模块中的 DBG_IWDG_STOP 配置位的状态，IWDG 的计数器能够继续工作或停止。详见调试模块的章节。

12.4 寄存器

12.4.1 寄存器总览

表 12-2 IWDG 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	IWDG_KR	键寄存器	0x00000000

Offset	Acronym	Register Name	Reset
0x04	IWDG_PR	预分频寄存器	0x00000000
0x08	IWDG_RLR	重装载寄存器	0x00000FFF
0x0C	IWDG_SR	状态寄存器	0x00000000
0x10	IWDG_CR	控制寄存器	0x00000000
0x14	IWDG_IGEN	中断生成寄存器	0x00000FFF
0x18	IWDG_CNT	计数寄存器	0x00000000

12.4.2 IWDG_KR 键寄存器

偏移地址：0x00

复位值：0x0000 0000 (在待机模式复位)

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

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

Bit	Field	Description
31：16	Reserved	保留，必须保持复位值
15：0	KEY	键值 (只写寄存器，读出值为 0x0000)(Key value) 软件必须以一定的间隔写入 0xAAAA，否则，当计数器为 0 时，看门狗会产生复位。 写入 0x5555 表示允许访问 IWDG_PR、IWDG_RLR、IWDG_IGEN 和 IWDG_CR[IRQ_SEL] 寄存器。 写入 0xCCCC，启动看门狗工作。

12.4.3 IWDG_PR 预分频寄存器

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.													PR		
													rw		

Bit	Field	Description
31:3	Reserved	始终读为 0
2：0	PR	预分频因子 (Prescaler divider) 这些位具有写保护设置。通过设置这些位来选择计数器时钟的预分频因子。要改变预分频因子，IWDG_SR 寄存器 的 PVU 位必须为 0。 000：预分频因子 = 4　100：预分频因子 = 64 001：预分频因子 = 8　101：预分频因子 = 128 010：预分频因子 = 16　110：预分频因子 = 256 011：预分频因子 = 32　111：预分频因子 = 256 注意：对此寄存器进行读操作，将从 VDD 电压域返回预分频值。 如果写操作正在进行，则读回的值可能是无效的。因此，只有对 IWDG_SR 寄存器的 PVU 位为 0 时，读出的值才有效。

12.4.4 IWDG_RLR 重装载寄存器

偏移地址：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											
				rw											

Bit	Field	Description
31:12	Reserved	保留，必须保持复位值
11：0	RL	看门狗计数器重装载值 (Watchdog counter reload value) 这些位具有写保护。用于定义看门狗计数器的重装载值，每当向 IWDG_KR 寄存器写入 0xAAAA 时，重装载值会被传 送到计数器中。随后计数器从这个值开始递减计数。看门 狗超时周期可通过次重装载值和时钟预分频值来计算。 注：对此寄存器进行读操作，将从 VDD 电压域返回预分频值。如果写操作正在进行，则读回的值可能是无效的。因此，只有当 IWDG_SR 寄存器的 RUV 位为 0 时，读出的值才有效。

12.4.5 IWDG_SR 状态寄存器

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.												UPDATE	IVU	RVU	PVU
												r	r	r	r

Bit	Field	Description
31:4	Reserved	保留，必须保持复位值
3	UPDATE	看门狗重装值更新标记 此位由硬件置‘1’用来指示 IWDG_KR 寄存器中写入 0xAAAA。当看门狗的重装值写入计数器后，计数器被更新， 此位由硬件清‘0’。
2	IVU	看门狗中断生成值更新 (Watchdog Interrupt Generate value update) 此位由硬件置‘1’用来指示中断生成值的更新正在进行中。 当在 VDD 域中的中断生成值更新结束后，此位由硬件清‘0’(最多需要 5 个 40KHz 的振荡器周期) 中断生成值只有在 IVU 位被清‘0’后才可更新。
1	RVU	看门狗计数器重装载值更新 (Watchdog counter reload value update) 此位由硬件置‘1’用来指示重装载值的更新正在进行中。 当在 VDD 域中的重装载更新结束后，此位由硬件清‘0’(最多需要 5 个 40KHz 的振荡器周期) 重装载值只有在 RVU 位被清‘0’后才可更新。
0	PVU	看门狗预分频更新 (Watchdog prescaler value update) 此位由硬件置‘1’用来指示预分频值的更新正在进行中。当 在 VDD 域中的预分频值更新结束后，此位由硬件清‘0’(最多需要 5 个 40KHz 的振荡器周期) 预分频值只有在 RVU 位被清‘0’后才可更新。

注：如果在应用程序中使用多个重装载值、预分频值或中断生成值，则必须在 RVU 位被清除后才能重新改变预装载值，在 PVU 位被清除后才能重新改变预分频值，必须在 IVU 位被清除后才能重新改变中断生成值。然而，在预分频或重装值更新后，不必等待 RVU 或 PVU 复位，可以继续执行下面的代码。

12.4.6 IWDG_CR 控制寄存器

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.														IRQ_CLR	IRQ_SEL
														rw	rw

Bit	Field	Description
31:2	Reserved	保留，必须保持复位值
1	IRQ_CLR	IWDG 中断清除 1：写 1 清除中断 0：无操作
0	IRQ_SEL	IWDG 溢出操作选择 1：溢出后产生中断 0：溢出后产生复位

12.4.7 IWDG_IGEN 中断生成寄存器

偏移地址：0x14

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.				IGEN											
				rw											

Bit	Field	Description
31:12	Reserved	保留，必须保持复位值
11：0	IGEN	IWDG 中断生成值 (Watchdog Interrupt Generate value) 这些位具有写保护。 用于定义看门狗计数器产生中断的阈值，每当计数器值递减等于阈值时，会产生中断。 只有当 IWDG_SR 寄存器中的 IVU 位为 0 时，才能对此寄存器进行修改。 注：对此寄存器进行读操作，将从 VDD 电压域返回中断生成值。如果写操作正在进行，返回值可能无效。只有当 IWDG_SR 寄存 器的 IVU 位为 0 时，返回值值才能保证有效。

12.4.8 IWDG_CNT 计数寄存器

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.													IWDG_CNT		
													r		

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

r	r

Bit	Field	Description
31：19	Reserved	保留，始终读为 0。
18：8	IWDG_CNT	IWDG 计数器 counter 的当前值
7：0	IWDG_PS	IWDG 时钟分频计数器的当前值

13 SPI 串行外设接口

13.1 简介

SPI（Serial Peripheral Interface）接口广泛用于不同设备之间的板级通讯，如扩展串行 Flash，ADC 等。许多 IC 制造商生产的器件都支持 SPI 接口。

SPI 允许 MCU 与外部设备以全双工、同步、串行方式通信。应用软件可以通过查询状态或 SPI 中断来通信。

13.2 功能框图

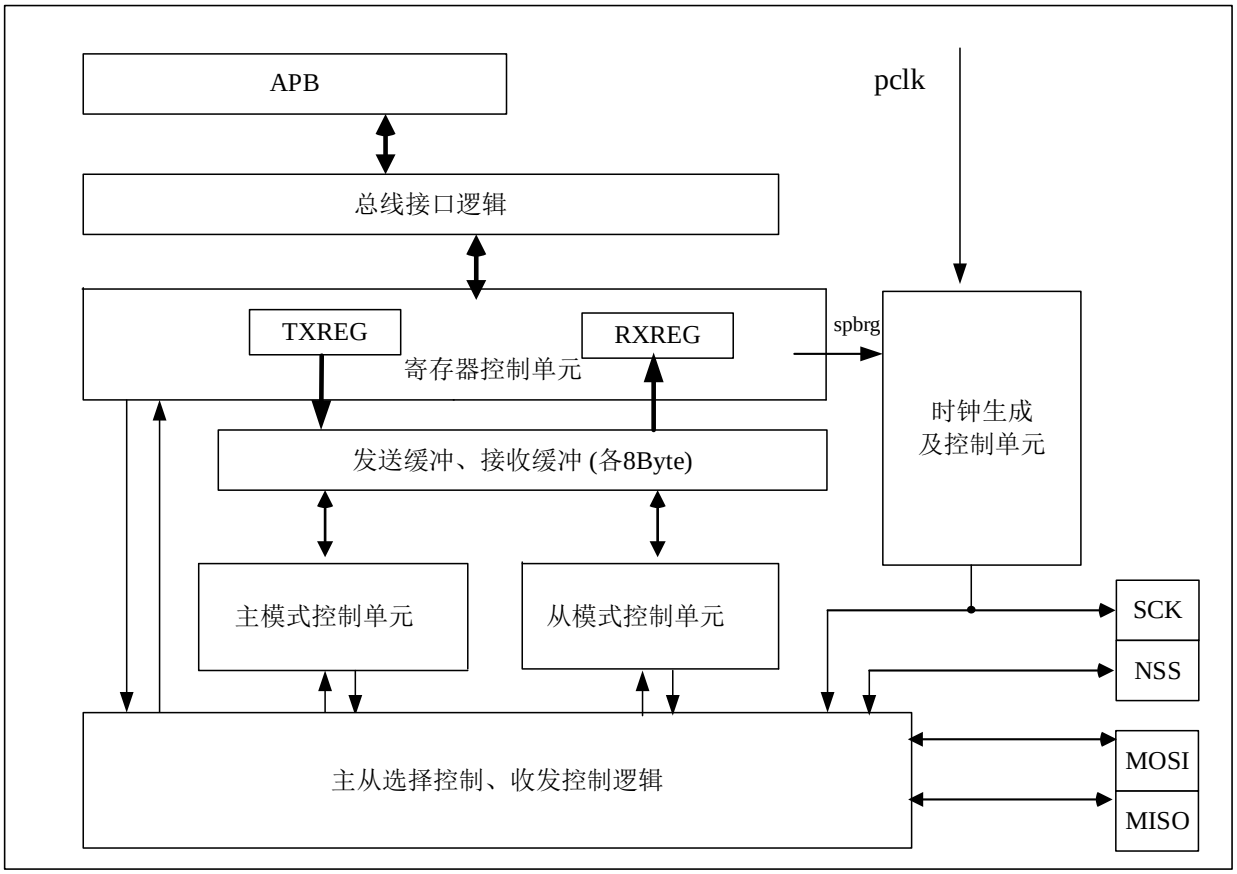

图 13-1 SPI 功能框图

13.3 功能描述

13.3.1 概述

SPI 支持同时接收和发送 1~32 位数据，但是仅在 8bit 传输时支持 LSB 和 MSB，1~32 位传输只支持 LSB 传输，可以被配置为从模式或在一个主机环境下的主模式。软件通过配置通用控制寄存器（CCTL）中的 CPOL 位和 CPHA 位，选择时钟与数据之间有四种不同的传输时序；并可配置 LSBFE 位选择使用 MSB 在前或者 LSB 在前来传输数据。

SPI 在时钟 SCK 的上升沿或下降沿发送数据，在相反的时钟有效沿接收数据。

使用 SPI 交换数据，要求必须在接收数据完成时进行数据读取操作，即使该数据不是有效数据；并且要求主机和与其通信的从机之间具有相同的时钟相位和极性。

SPI 通过 4 个引脚与外部器件相连：

- MISO：主设备输入、从设备输出引脚。传输方向为从设备发送到主设备。
- MOSI：主设备输出、从设备输入引脚。传输方向为主设备发送到从设备。
- SCK：串口时钟，由主设备产生并提供给从设备。
- NSS：从设备选择。该引脚为 SPI 协议下的可选功能，用于主设备选择与之通信的从设备（多从设备时），实现主从设备间的一对一单独通信，并能避免数据线上设备之间的冲突。当软件配置激活 NSS 引脚功能后，配置 SPI_GCTL.MODE 为‘1’使 SPI 工作于主模式，再配置 SPI_NSSR.NSS 为‘0’使 NSS 引脚输出低电平，从而允许与主设备连接的从设备（配置 SPI_GCTL.MODE 为‘0’）和它进行数据通信。

下图示意主、从设备之间一对一通信时的连接情况：

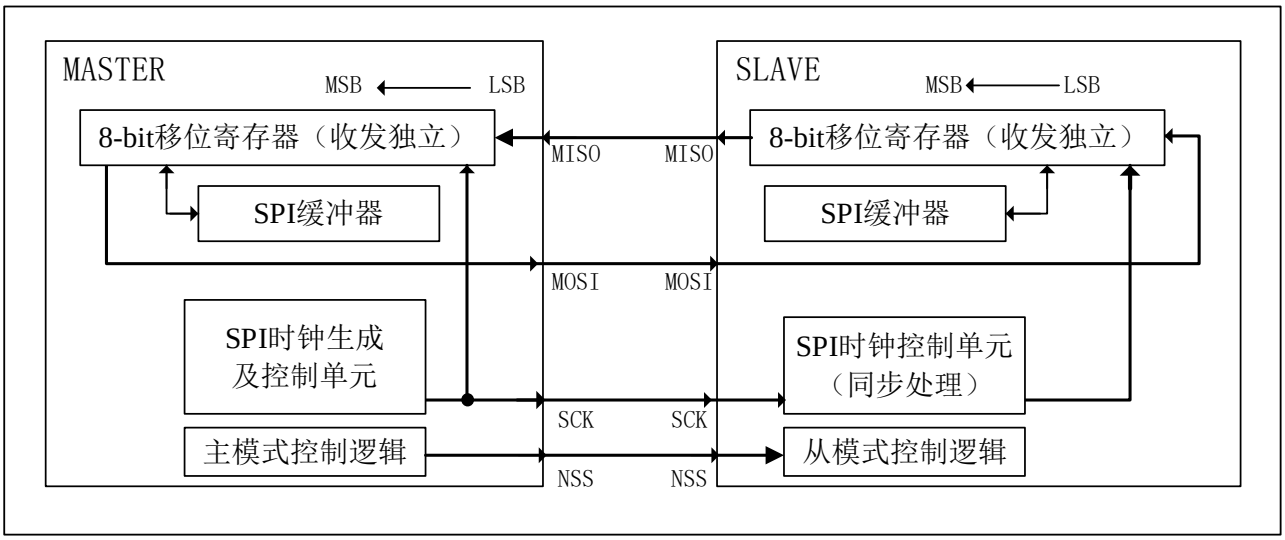

图 13-2 单主和单从应用

SPI 主、从设备的同名引脚互连，图中示意数据方向为从最高位到最低位的方式串行通信。

主设备负责发起通信请求，从设备负责响应，从设备通过 SCK 引脚得到主设备提供的时钟信号，从

而使得主、从设备均使用同一个时钟进行同步的全双工通信。

对于从设备而言，MOSI 引脚输入来自主设备的发送数据，MISO 引脚输出响应数据传给主设备。

13.3.1.1 时钟信号的极性和相位

SPI_CCTL 寄存器的 CPOL 和 CPHA 位分别控制时钟的极性和相位，通过软件配置可分别得到 4 种不同的时钟/数据时序关系。

时钟极性指的是 SCK 时钟空闲状态下电平保持何种状态：如果配置 CPOL 位为‘0’，在空闲状态下 SCK 时钟保持为低电平；反之，则在空闲状态下 SCK 时钟保持为高电平。主、从设备都会受到 CPOL 控制位的影响。

时钟相位决定输入数据采样的时序：如果配置 CPHA 位为‘0’，第一个数据位会在 SCK 的第二个时钟边沿被采样；反之，则第一个数据位会在 SCK 的第一个时钟边沿被采样。

另外 CPHASEL 位在系统上电复位后为‘0’，当软件调整此位配置为‘1’后将会切换数据采样时序，也就是使得 CPHA 位的功能发生切换。例如，CPHASEL=1，CPHA=0 时：第一个数据位会在 SCK 的第一个时钟边沿被采样（CPOL 位为‘0’为上升沿；为‘1’则是下降沿）。

因此，需要根据 CPOL、CPHA 及 CPHASEL 位的组合配置来期望使用的时钟/数据时序关系。

时序配置需要注意以下几点：

- SPI 工作时不能修改 CPOL/CPHA（如需修改，首先关闭 SPI 的使能位 SPIEN）。
- 主、从设备同步通信，因此双方时序配置应保持一致。
- 空闲状态时 SCK 的电平情况必须和 CPOL 为配置的极性一致。

下图示意在 SPI 传输下的不同 CPHA 与 CPOL 位组合的 4 种情况（配置 CPHASEL 位为‘1’时），以及主/从设备的 SCK/MISO/MOSI/NSS 引脚的时序关系。

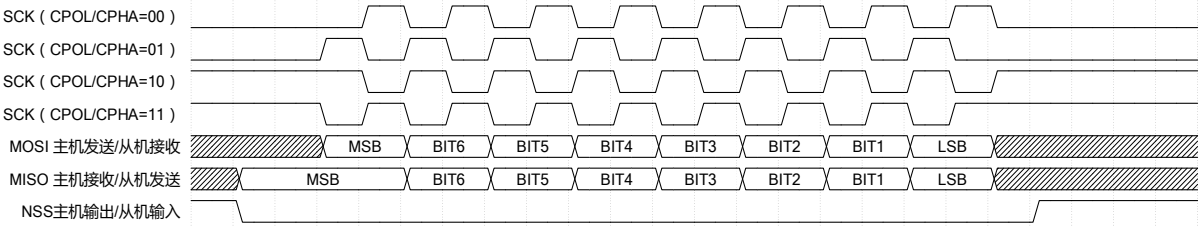

图 13-3 数据时钟时序图

需注意，配置 SPI_GCTL.NSS=1，即 NSS 引脚功能由硬件自动控制时，在数据通信完成后硬件将自动拉高此引脚（上图例中所示）；当配置 SPI_GCTL.NSS=0，NSS 输出状态则由从机片选寄存器 SPI_NSS 的 NSS 位进行控制（需软件切换 NSS 引脚的输出状态）。

13.3.1.2 高速传输

针对高速传输时板级延时敏感的情况，可配置 SPI_CCTL 寄存器中 TXEDGE/RXEDGE 位对数据发

送/接收采样的时间点进行调整。高速传输，针对主设备的波特率发生器配置在 SPBRG<=4 且输出 SCK 时钟较快（>=10MHz）；低速传输，针对主设备波特率发生器配置在 SPBRG>4 且输出 SCK 时钟较慢（<10MHz）。

- 在从模式下，TXEDGE 位为‘1’时，软件一旦配置 TXREG 写入数据，不等待 SCK 时钟输入的边沿而是立即发送到 MISO 引脚线上；TXEDGE 位为‘0’时，从设备总是等到一个有效时钟边沿才发送数据到 MISO 引脚线上。
- 在主模式下，RXEDGE 位为‘1’时，将后移采样数据的时间点，在传输数据位的尾时钟沿采样数据；RXEDGE 位为‘0’时，在传输数据位的中间点上采样数据。数据帧格式

配置 SPI_CCTL 寄存器中的 LSBFE 位，决定数据输出的优先顺序。LSBFE 位为‘1’会从最低位到最高位的顺序收发数据，为‘0’（缺省）会从最高位到最低位的顺序收发数据。

配置 SPI_CCTL 寄存器的 SPILEN 位，决定数据帧的数据长度。SPILEN 位为‘1’（缺省）则数据帧长是 8 位，为‘0’则数据帧长是 7 位。SPI 的发送和接收都受到数据帧格式配置的控制。

另外，可通过配置 SPI_GCTL.DW8_32=1，和寄存器 SPI_EXTCTL 来实现任意帧长的数据格式（帧长范围在 1~32 位之间）；任意帧长配置时也支持 LSBFE 位的功能（LSB 或 MSB 优先）。

在配合 DMA 进行数据传输时，需要将 DMA 的数据长度配置为 8bit。

13.3.2 SPI 主要特征

- 完全兼容 Motorola 的 SPI 规格
- 在 3 根线上支持全双工同步传输
- 16 位的可配置波特率生成器
- 支持主机模式和从机模式
- 支持一个主机与多个从机通信
- SPI 在主、从机模式下，时钟最快分别可达 PCLK/2、PCLK/4（PCLK 为 APB 时钟）
- 可配置的时钟极性和相位
- 可配置的数据帧长度（固定 8 位或 7 位帧长，1~32 位任意帧长）
- 可配置的数据顺序，MSB 在前或者 LSB 在前（1~32 位任意帧长支持 LSB、8 位数据收发都支持）
- 4 个字节的接收/发送缓冲，同时具备下列中断事件或状态供软件配置使用：
    - 发送缓冲为空
    - 发送缓冲和发送移位寄存器同时为空
    - 发送端下溢
    - 接收到有效字节
    - 接收缓冲上溢

- 接收缓冲满
- 主模式下接收到指定的字节个数

13.3.3 SPI 从模式

SPI 作为从设备时，SCK 引脚输入来自主设备的串行时钟，因此从设备工作时不使用波特率发生器，不需配置寄存器 SPI_SPBRG（从设备下无效）。

13.3.3.1 配置步骤

1. 配置 SPI_GCTL.SPILEN，定义数据帧格式为 7 位或者 8 位；配置 SPI_GCTL.DW8_32 为‘1’，可配置 SPI_EXTCTL 寄存器来定义成任意帧格式（SPILEN 需固定为‘1’）。
2. 配置寄存器 SPI_CCTL 中的 CPOL、CPHA/CPHASEL 位，以确定时序模式。
3. 配置 SPI_CCTL.LSBFE，确定数据帧的收发顺序（LSB 或 MSB 位优先）。
4. 配置寄存器 SPI_GCTL 中的 MODE 位为‘0’（从模式），SPIEN 位为‘1’（SPI 功能使能），并配置 SPI 工作所需的 GPIO 功能引脚。
5. 配置寄存器 SPI_GCTL 中 TXEN、RXEN 位为‘1’，打开发送、接收的许可（发送时需提前写入数据到寄存器 SPI_TXREG），SPI 将在从模式下接收 MOSI 引脚数据，并从 MISO 引脚输出数据。

注意：必须配置主、从设备的时序模式和数据帧收发顺序为一致，以保证数据能正常传输。

13.3.3.2 数据发送过程

写数据到发送数据寄存器 SPI_TXREG 后，整个数据会一起传输到发送缓冲。

当从设备接收到 SCK 引脚上的时钟信号，也同时接收到 MOSI 引脚传来的第一个数据位；从设备利用 SCK 的变化边沿，把发送数据逐位发送到 MISO 引脚上。发送数据的过程符合数据/时钟的相关时序（由 CPOL、CPHA/CPHASEL 位来决定）。

但在高速传输时（配置 SPI_CCTL.TXEDGE=1），数据将不再按照输入的 SCK 时钟边沿而变化，而会提前以内部 PCLK 的时钟边沿把数据送到 MISO 引脚上(此提前量不会早于前一位数据接收采样的 SCK 时钟边沿)。

当数据第一位被发送时，硬件会置位 SPI_INTSTAT.TX_INTF 标志，软件可利用此标志来写 TXREG 以实现数据的连续发送（配置 SPI_INTEN.TX_IEN 位为‘1’来产生 CPU 中断）。

注意：从机时钟信号由主机提供，因此，连续传输的前提必须是主机能提供连续不断的时钟。

13.3.3.3 数据接收过程

从设备接收到 MOSI 引脚输入的一个完整数据时：

- 此数据通过移位寄存器，会在最后一个采样时钟边沿被传输到接收缓冲中；硬件也同时会置位 SPI_INTSTAT.RX_INTF 标志。之后软件通过读 SPI_RXREG，就能从接收缓冲中获取该数据。
- 软件配置 SPI_INTEN.RX_IEN 位为‘1’来打开中断使能，利用 CPU 中断获取接收数据。

13.3.4 SPI 主模式

SPI 作为主设备时，输出串行时钟到 SCK 引脚上，供从设备使用。

13.3.4.1 配置步骤

1. 配置 SPI_SPBRG 寄存器，定义串行时钟波特率。
2. 配置 SPI_CCTL 寄存器中 CPOL、CPHA/CPHASEL 位，确定时序模式。
3. 配置 SPI_CCTL.SPILEN 来定义 8 或 7 位数据帧格式；配置 SPI_GCTL.DW8_32 为‘1’，可配置 SPI_EXTCTL 寄存器来定义成任意帧格式（SPILEN 需固定为‘1’）。
4. 配置 SPI_CCTL.LSBFE 来确定数据收发的顺序（LSB 或 MSB 位优先）。
5. 如果只接收而不发送数据，可配置 SPI_RXDNR 寄存器来定义需要接收的字节数（当接收到指定个数字节后，SCK 时钟输出会结束并保持在 CPOL 位配置的状态上）。
6. 配置寄存器 SPI_GCTL 中的 MODE 位为‘1’（主模式）、SPIEN 位为‘1’（SPI 功能使能），并配置 SPI 工作所需的 GPIO 功能引脚。
7. 配置寄存器 SPI_GCTL 中 TXEN、RXEN 位为‘1’，打开发送、接收的许可（发送时打开 TXEN 后写入数据到寄存器 SPI_TXREG），SPI 将在主模式下输出时钟 SCK 和同步数据 MOSI 到引脚上，并从 MISO 引脚上采样输入数据；NSS 是主设备可选的输出功能。

注意：必须配置主、从设备的时序模式和数据帧收发顺序为一致，以保证数据能正常传输。

13.3.4.2 数据发送过程

配置 TXEN 位为‘1’后，写数据到发送数据寄存器 TXREG，此数据将传输到发送缓冲，主设备开始发送。主设备按照预先配置好的波特率串行输出 SCK 时钟和 MOSI 数据到引脚上，此过程符合数据/时钟的相关时序（由 CPOL、CPHA/CPHASEL 位决定）；而且 LSBFE 位决定了数据串行传输顺序。

当数据第一位被发送时，硬件会置位 SPI_INTSTAT.TX_INTF 标志，软件利用此标志来写 TXREG 以实现数据的连续发送（配置 SPI_INTEN.TX_IEN 位为‘1’来产生 CPU 中断）。

13.3.4.3 数据接收过程

主设备接收到 MISO 引脚输入的一个完整数据时：

- 此数据通过移位寄存器，会在最后一个采样时钟边沿被传输到接收缓冲中；硬件也同时会置位

SPI_INTSTAT.RX_INTF 标志。之后软件通过读 SPI_RXREG，就能从接收缓冲中获取该数据。

- 软件配置 SPI_INTEN.RX_IEN 位为‘1’可打开中断使能，利用 CPU 中断获取接收数据。
- 只接收时，接收完 RXDNR 定义的字节个数后，硬件将置位 SPI_INTSTAT.RXMATCH_INTF 标志，同时主设备不再发送时钟信号，SCK 输出将保持在 CPOL 位配置的状态上（固定高或低电平）。

13.3.5 波特率设置

SCK 引脚输出的时钟频率符合波特率配置，它由内部时钟 PCLK 按照 SPI_SPBRG 寄存器的配置值分频得到。寄存器 SPBREG 控制一个 16 位计数器的计数周期。

按照期望的波特率和 Fpclk（APB 模块 PCLK 时钟频率），使用下表公式可计算出给寄存器 SPBRG 的配置值（下表中的 X），X 在 2~65535 范围之内。

表 13-1 波特率公式

模式	公式
SPI 模式	波特率 = Fpclk/X

13.3.6 中断

13.3.6.1 状态标志

为了软件操作的方便，应用程序可以通过 4 个当前状态标志和 7 个中断状态标志来监控 SPI 总线的状态。

当前状态标志是只读，由硬件自动置位和清除。

中断状态标志会在事件发生时置位，可在中断使能时产生 CPU 中断请求，并由软件清除。

SPI 内部分别有一个 8 字节的发送缓冲和接收缓冲，CPU 可根据 SPI_GCTL 寄存器中 DW8_32 位的配置，每次读写 1 个或 4 个字节。根据 DW8_32 的配置，发送和接收缓冲分别有 1 个字节或者 1 个有效数据的状态标志。

注意：配置 SPI_GCTL.DW8_32=1，收发缓冲至多有两个有效数据；此时，当帧长配置为 8bit 及以下时一个有效数据为 1 个字节；帧长配置在 9~16bit 范围时一个有效数据为 2 个字节；帧长配置在 17~24bit 范围时一个有效数据为 3 个字节；帧长配置在 25~32bit 范围时一个有效数据为 4 个字节。

表 13-2 SPI 状态

分类	状态标志	缓冲和信号状态
中断状态	TX_INTF	发送缓冲为空，根据 DW8_32 配置能完成一次发送数据寄存器 TXREG 的写操作
	RX_INTF	根据 DW8_32 设置，至少有一个有效数据的数据，能完成一次接收数据寄存器 RXREG 的读操作

分类	状态标志	缓冲和信号状态
	UNDERRUN_INTF	发送缓冲空且重复发送
	RXOERR_INTF	接收缓冲非空且被覆盖
	RXMATCH_INTF	非空，指定个数的最后 1 个数据传到接收缓冲中（主模式下有效）
	RXFULL_INTF	接收缓冲满，不能再接收新的数据
	TXEPT_INTF	发送缓冲和发送移位寄存器都为空
当前状态	RXAVL_4BYTE	接收缓冲有超过 4 字节有效数据
	TXFULL	发送缓冲满
	TXEPT	发送缓冲和发送移位寄存器都为空
	RXAVL	接收缓冲非空

13.4 寄存器

表 13-3 SPI 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	SPI_TXREG	发送数据寄存器	0x00000000
0x04	SPI_RXREG	接收数据寄存器	0x00000000
0x08	SPI_CSTAT	当前状态寄存器	0x00000001
0x0C	SPI_INTSTAT	中断状态寄存器	0x00000000
0x10	SPI_INTEN	中断使能寄存器	0x00000000
0x14	SPI_INTCLR	中断清除寄存器	0x00000000
0x18	SPI_GCTL	全局控制寄存器	0x00000004
0x1C	SPI_CCTL	通用控制寄存器	0x00000008
0x20	SPI_SPBRG	波特率发生器	0x00000002
0x24	SPI_RXDNR	接收数据个数寄存器	0x00000001
0x28	SPI_NSSR	从机片选寄存器	0x000000FF
0x2C	SPI_EXTCTL	数据控制寄存器	0x00000008

13.4.1 SPI_TXREG 发送数据寄存器

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

Bit	Field	Description
31：0	TXREG	发送数据寄存器（Transmit data register） 有效数据位由 DW8_32 控制： DW8_32=0 时，只有低 8 位有效 DW8_32=1 时，TXREG[31：0]都有效

13.4.2 SPI_RXREG 接收数据寄存器

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

Bit	Field	Description
31：0	RXREG	接收数据寄存器（Receive data register） 有效数据位由 DW8_32 控制： DW8_32=0 时，只有低 8 位有效 DW8_32=1 时，RXREG[31：0]都有效 注：该寄存器可读不可写。

13.4.3 SPI_CSTAT 当前状态寄存器

偏移地址：0x08

复位值：0x0000 0001

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					RXFADDR			Res.	TXFADDR			RXAVL_4BYTE	TXFULL	RXAVL	TXEPT
					r				r			r	r	r	r

Bit	Field	Description
31：14	Reserved	始终读为 0
10：8	RXFADDR	当前接收缓冲中有效字节个数

Bit	Field	Description
7	Reserved	始终读为 0
6：4	TXFADDR	当前发送缓冲中有效字节个数
3	RXAVL_4BYTE	接收缓冲中有效数据达到 4 个字节标志（Receive available 4 byte data message） 0：接收缓冲中数据小于 4 个字节 1：接收缓冲中有超过 4 个字节
2	TXFULL	发送缓冲满标志位（Transmitter FIFO full status bit） 0：发送缓冲未满 1：发送缓冲满
1	RXAVL	接收有效数据标志位（Receive available byte data message） 当接收缓冲中接收到一个字节数据时置位该位。 0：接收缓冲空 1：接收缓冲非空 注：该位只读，由硬件自动置位和清除。
0	TXEPT	发送端空标志位（Transmitter empty bit） 0：发送缓冲或发送移位寄存器不为空 1：发送缓冲和发送移位寄存器都为空 注：该位只读，由硬件自动置位和清除。

13.4.4 SPI_INTSTAT 中断状态寄存器

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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TXEPT_INTF	RXFULL_INTF	RXMATCH_INTF	RXOERR_INTF	UNDERRUN_INTF	RX_INTF	TX_INTF
									r	r	r	r	r	r	r

Bit	Field	Description
31:7	Reserved	始终读为 0。
6	TXEPT_INTF	发送端空中断标志位 (Transmitter empty interrupt flag bit) 硬件自动置位，写 INTCLR 寄存器 TXEPT_ICLR 位清除。 1：发送端缓冲器和 TX 移位寄存器为空 0：发送端不为空 注意：该位是中断状态信号， TXEPT 是状态信号。

Bit	Field	Description
5	RXFULL_INTF	接收端缓冲器满中断标志位 (RX FIFO full interrupt flag bit) 硬件自动置位，写 INTCLR 寄存器 RXFULL_ICLR 位清除。 1 : RX 缓冲器满 0 : RX 缓冲器未满
4	RXMATCH_ INTF	接收指定字节数中断标志位 (Receive data match the RXDNR number, the receive process will be completed and generate the interrupt) 硬件自动置位，写 INTCLR 寄存器 RXMATCH_ICLR 位清除。 1 : 接收了 RXDNR 寄存器指定的字节数 0 : 未完成 RXDNR 寄存器指定的字节数
3	RXOERR_ INTF	接收端溢出错误中断标志位 (Receive overrun error interrupt flag bit) 硬件自动置位，写 INTCLR 寄存器 RXOERR_ICLR 位清除。 1 : 溢出错误 0 : 没有溢出错误
2	UNDERRUN_ INTF	SPI 从机模式下溢标志位 (SPI underrun interrupt flag bit) 硬件自动置位，写 INTCLR 寄存器 UNDERRUN_ICLR 位清除。 1 : 下溢错误 0 : 没有下溢错误
1	RX_INTF	接收端数据有效中断标志位 (Receive data available interrupt flag bit) 硬件自动置位，写 INTCLR 寄存器 RX_ICLR 位清除。 当接收端缓冲器接收了一个完整字节数据。 1 : 接收端缓冲器有有效字节数据 0 : 接收端缓冲器空
0	TX_INTF	发送缓冲器有效中断标志位 (发送了一个字节的数据)(Transmit FIFO available interrupt flag bit) 硬件自动置位，发送缓冲器不为空自动清零。 1 : 发送端缓冲器有效 0 : 发送端缓冲器无效

13.4.5 SPI_INTEN 中断使能寄存器

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
Reserved															

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

Reserved	TXEPT_IEN	RXFULL_IEN	RXMATCH_IEN	RXOERR_IEN	UNDERRUN_IEN	RX_IEN	TX_IEN
	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31：7	Reserved	保留，始终读为 0
6	TXEPT_IEN	发送端空中断使能位（Transmit empty interrupt enable bit） 0：中断禁止 1：中断使能
5	RXFULL_IEN	接收缓冲满中断使能位（Receive FIFO full interrupt enable bit） 0：中断禁止 1：中断使能
4	RXMATCH_IEN	接收指定字节数中断使能位（Receive data complete interrupt enable bit） 0：中断禁止 1：中断使能
3	RXOERR_IEN	接收端溢出错误中断使能位（Overrun error interrupt enable bit） 0：中断禁止 1：中断使能
2	UNDERRUN_IEN	SPI 从机模式下溢中断使能位（SPI 从机模式）（Transmitterunderrun interrupt enable bit（SPI slave mode only）） 0：中断禁止 1：中断使能
1	RX_IEN	接收端数据中断使能位（Receive FIFO interrupt enable bit） 0：中断禁止 1：中断使能
0	TX_IEN	发送缓冲空中断使能位（Transmit FIFO empty interrupt enable bit） 0：中断禁止 1：中断使能

13.4.6 SPI_INTCLR 中断清除寄存器

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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TXEPT_ICLR	RXFULL_ICLR	RXMATCH_ICLR	RXOERR_ICLR	UNDERRUN_ICLR	RX_ICLR	TX_ICLR

	w	w	w	w	w	w	w

Bit	Field	Description
31：7	Reserved	保留，始终读为 0
6	TXEPT_ICLR	发送端空中断清除位（Transmitter empty interrupt clear bit） 0：写 0 无意义 1：写 1 清除中断
5	RXFULL_ICLR	接收缓冲满中断清除位（Receiver buffer full interrupt clear bit） 0：写 0 无意义 1：写 1 清除中断
4	RXMATCH_ ICLR	接收指定字节数中断清除位（Receive completed interrupt clear bit） 0：写 0 无意义 1：写 1 清除中断
3	RXOERR_ ICLR	接收端溢出错误中断清除位（Overrun error interrupt clear bit） 0：写 0 无意义 1：写 1 清除中断
2	UNDERRUN_ ICLR	SPI 从机模式下溢中断清除位(SPI 从机模式)(Transmitter underrun interrupt clear bit（SPI slave mode only）） 0：写 0 无意义 1：写 1 清除中断
1	RX_ICLR	接收端数据中断清除位（Receive interrupt clear bit） 0：写 0 无意义 1：写 1 清除中断
0	TX_ICLR	发送缓冲空中断清除位（Transmitter FIFO empty interrupt clear bit） 0：写 0 无意义 1：写 1 清除中断

13.4.7 SPI_GCTL 全局控制寄存器

偏移地址：0x18

复位值：0x0000 0004

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			NSSTOG	DW8_32	NSS	DMAMODE	Reserved				RXEN	TXEN	MODE	INTEN	SPIEN
			rw	rw	rw	rw					rw	rw	rw	rw	rw

Bit	Field	Description
31：13	Reserved	保留，始终读为 0
12	NSSTOG	从设备选择信号自动翻转 (Slave select toggle) 1：NSS 信号在传输完每个数据后自动翻转 0：NSS 信号不翻转 注：该位只在主模式下有效
11	DW8_32	发送和接收数据寄存器有效数据选择 (Valid byte or double- word data select signal) 0：只有低 8 位有效 1：32 位数据都有效 注：必须用指定数据格式访问。
10	NSS	硬件或软件控制主模式下的 NSS 输出 (NSS select signal that from software or hardware) 0：由 NSSR 寄存器值控制 1：进行数据传输时硬件自动控制
9	DMAMODE	DMA 方式（DMA Mode selection bit） 0：正常模式 1：开启 DMA 模式
8：5	Reserved	预留，必须保持复位值。
4	RXEN	接收使能位（Receive enable bit） 0：接收禁止。同时可以清空 RX 缓冲 1：接收使能 注：当 SPI 只工作在主机接收模式时，TXEN 必须设置为 0。
3	TXEN	发送使能位（Transmit enable bit） 0：发送禁止。同时可以清空 TX 缓冲 1：发送使能 注：当在主机模式下发送和接收同时发生。
2	MODE	主机模式位（Master mode bit） 0：从机模式（串行时钟来自外部主机） 1：主机模式（由内部 BRG 产生串行时钟）
1	INTEN	SPI 中断使能位 (SPI interrupt enable bit) 1：使能 SPI 中断 0：禁止 SPI 中断
0	SPIEN	SPI 选择位 (SPI select bit) 0 : SPI 禁止 (复位状态) 1 : SPI 使能

13.4.8 SPI_CCTL 通用控制寄存器

偏移地址：0x1C

复位值：0x0000 0008

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

Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									CPHASEL	TXEDGE	RXEDGE	SPILEN	LSBFE	CPOL	CPHA
									rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:7	Reserved	始终读为 0。
6	CPHASEL	CPHA 极性取反选择 (CPHA polarity invert select) 1 : CPHA 设置取反。 CPHA 为 1 时，第一个数据位采样从第二个时钟边沿开始。 CPHA 为 0 时，第一个数据位采样从第一个时钟边沿开始。 0 : CPHA 设置保持不变。
5	TXEDGE	发送数据相位调整位 (从模式)(Transmit data edge select) 1： 发送数据立即发送到数据总线 可用于高速模式时 (SPBRG = 4)。 0： 发送数据在一个有效时钟边沿后发送到数据总线可用于低速模式时 (SPBRG > 4)。
4	RXEDGE	接收数据采样时钟沿选择位 (主模式)(Receive data edge select) 1： 在传输数据位的尾时钟沿采样数据 (用于高速模式) 0： 在传输数据位的中间采样数据
3	SPILEN	SPI 数据宽度位 (SPI character length bit) 该位在 DW8_32 置位后 (DW8_32=0) 配置后起作用。 1 : 8 位数据 (缺省) 0 : 7 位数据
2	LSBFE	LSBFE：LSB 在前使能位 (LSI first enable bit) 1： 数据传输或接收最低位在前 0： 数据传输或接收最高位在前
1	CPOL	时钟极性标志位 (Clock polarity select bit) 1： 时钟在空闲状态为高电平 (两次传输之间) 0： 时钟在空闲状态为低电平 (两次传输之间)
0	CPHA	时钟相位选择位 (Clock phase select bit) 1： 第一个数据位采样从第一个时钟边沿开始 0： 第一个数据位采样从第二个时钟边沿开始

13.4.9 SPI_SPBRG 波特率发生器

偏移地址：0x20

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPBRG															

rw

Bit	Field	Description
31：16	Reserved	保留，始终读为 0
15：0	SPBRG	SPI 波特率控制寄存器用于产生波特率（SPI baud rate control register for baud rate） 波特率公式： 波特率 = Fpclk / SPBRG （Fpclk 是 APB 时钟频率） 注：不能对该寄存器写值为 0、1。

13.4.10 SPI_RXDNR 接收数据个数寄存器

偏移地址：0x24

复位值：0x0000 0001

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

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

Bit	Field	Description
31：16	Reserved	保留，始终读为 0
15：0	RXDNR	该寄存器用于存储下次接收过程需要接收字节的个数（The register is used to hold a count of to be received bytes in next receive process） 注：该寄存器的值仅在 SPI 为主机接收模式下有效；缺省值是 1。该寄存器值通过软件来改变，且不能对该寄存器写值为 0。

13.4.11 SPI_NSSR 从机片选寄存器

偏移地址：0x28

复位值：0x0000 00FF

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

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

Bit	Field	Description
31：1	Reserved	保留，必须保持复位值。
0	NSS	主模式下片选输出信号。低有效，从模式下该位无效（Chip select output signal in Master mode）。 0：从器件被选中（允许从器件与主模式建立通信） 1：从器件未选中

13.4.12 SPI_EXTCTL 数据控制寄存器

偏移地址：0x2C

复位值：0x0000 0008

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

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

Bit	Field	Description
31：5	Reserved	保留，始终读为 0
4：0	EXTLEN	控制 SPI 数据帧长度的选择 0 0000：32 bit 0 0001：1 bit 0 0010：2 bit 0 0011：3 bit …… 1 1100：28 bit 1 1101：29 bit 1 1110：30 bit 1 1111：31 bit 注：仅当 SPI_GCTL 寄存器 DW8_32 位为‘0’时有效，且 SPI_CCTL 寄存器的 LSBFE 位必须配置为‘1’， SPILEN 位也必须配置为‘1’。

14 I2C 内部集成电路接口

14.1 简介

微控制器通过 I2C 总线接口实现芯片间的串行互联。所有 I2C 总线特定的序列、协议仲裁和时序，都可以通过 I2C 提供的多主功能来控制。

I2C 总线是一种两线串行接口，串行数据线（SDA）和串行时钟（SCL）在连接到总线的器件间传递信息。每个器件都通过一个唯一的地址进行识别，且都可以作为发送或接收器。此外，器件在执行数据传输时也可以被看作是主器件或从器件。主器件是在总线上发起数据传输，并产生允许该传输的时钟信号的器件。此时，任何被寻址的器件都被认为是从器件。

I2C 有两种速率模式可供选择：标准模式（数据传输速率最大为 100Kbps）、快速模式（数据传输速率最大为 400Kbps）。

14.2 主要特征

- I2C 总线协议转换器/并行总线
- 半双工同步操作
- 支持主从模式
- 支持 7 位和 10 位地址格式
- 支持起始（START）、停止（STOP）、重新起始（RESTART）以及应答（ACK）信号的生成和检测
- 支持标准模式（最大 100Kbps）、快速模式（最大 400Kbps）
- 分别有 2 字节的发送和接收缓冲
- 支持过滤毛刺功能
- 支持中断和查询操作
- 支持多个从地址（详细见 I2C_SLVMASK 寄存器描述）

14.3 功能描述

14.3.1 功能框图

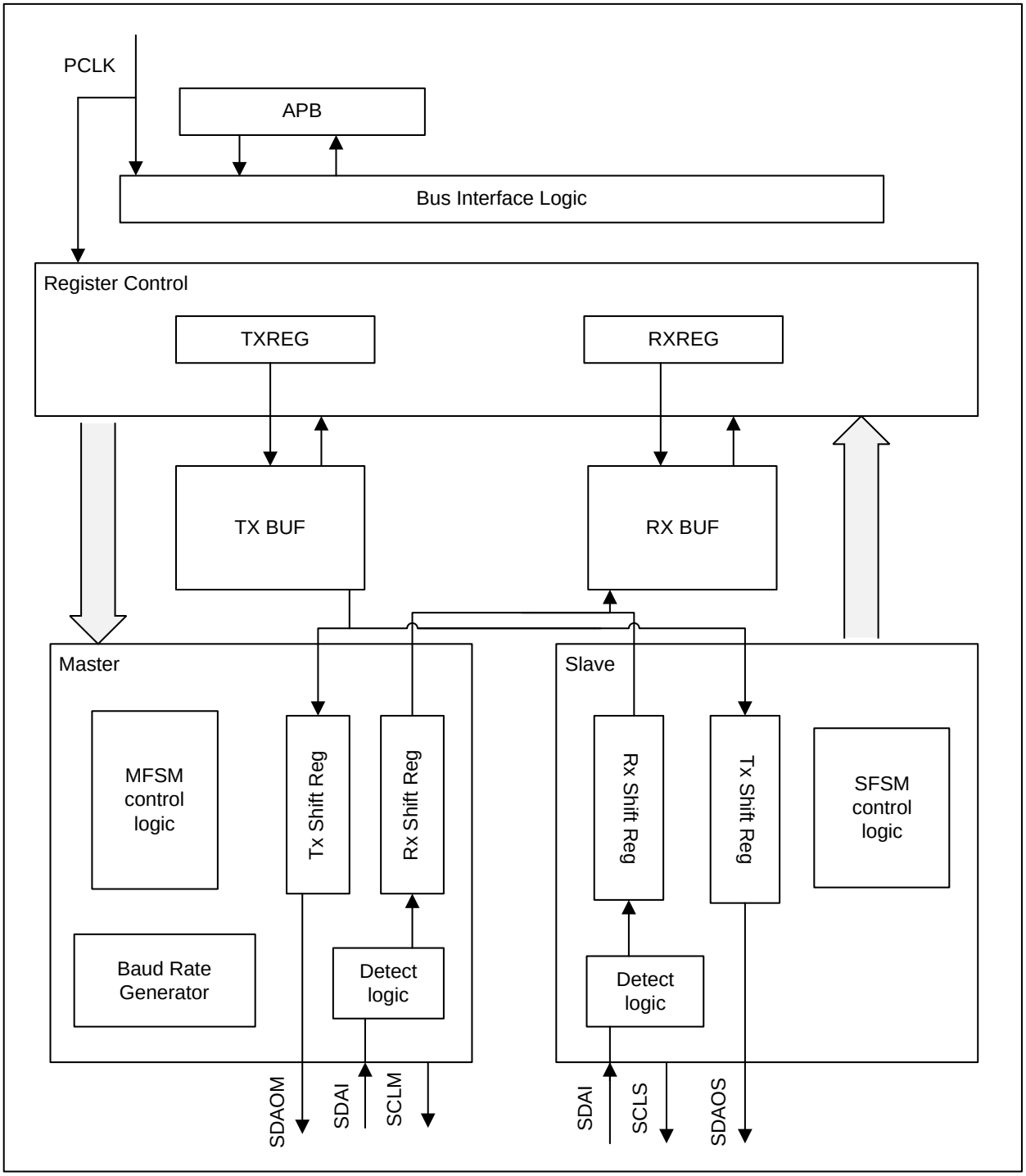

图 14-1 I2C 功能框图

14.3.2 信号描述

表 14-1 引脚定义

引脚名	属性	描述
I2C_SCL	I/O	I2C 时钟
I2C_SDA	I/O	I2C 数据

注：使用时引脚均需配置为开漏模式，配置方法请参考 GPIO 章节。

14.3.3 I2C 协议

14.3.3.1 起始和停止条件

总线处于空闲状态时，SCL 和 SDA 同时被外部上拉电阻拉为高电平。主器件启动数据传输时，必须先产生起始条件。在 SCL 线为高电平时，SDA 线从高电平向低电平切换表示起始条件。主器件结束传输时要发送停止条件。在 SCL 线为高电平时，SDA 线由低电平向高电平切换表示停止条件。下图显示了起始和停止条件的时序。数据传输过程中，当 SCL 为 1 时，SDA 必须保持稳定。

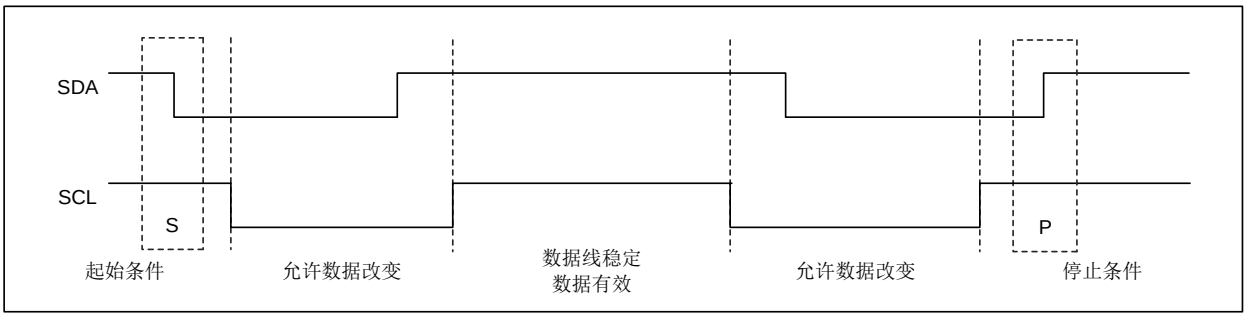

图 14-2 起始和停止条件

14.3.3.2 寻址协议

I2C 有两种地址格式：7 位地址格式和 10 位地址格式。

- 7 位地址格式

下图中，起始条件（S）后发送的第一个字节的前七位（b7：1）为从地址，最低位（b0）为数据方向位。b0 为 0 表示主器件写数据到从器件，b0 为 1 表示主器件从从器件读数据。

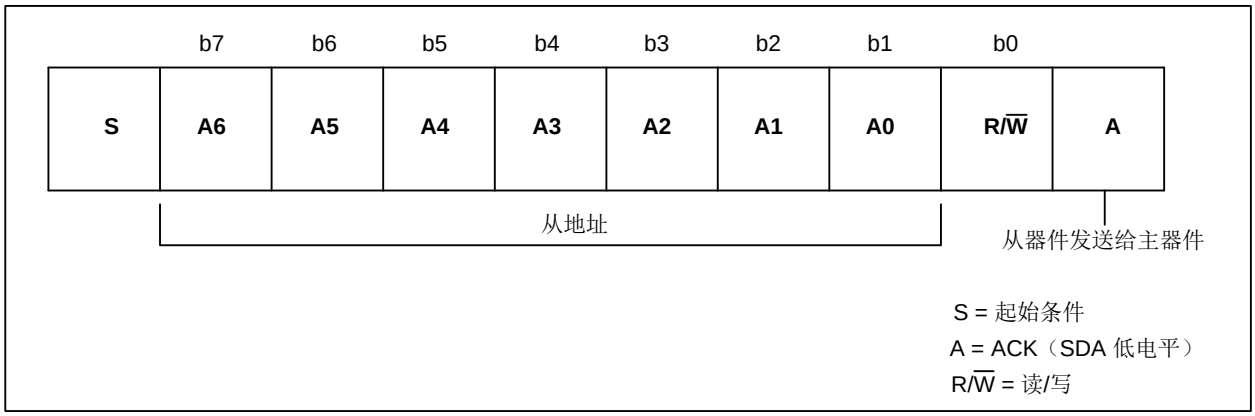

图 14-3 7 位地址格式

- 10 位地址格式

在 10 位地址格式中，需要发送 2 个字节来传输 10 位地址。第一个字节的描述如下：前五位（b7：3）用于通知从器件传输为 10 位地址格式；后两位（b2：1）为从地址的 b9：8；最低位（b0）为数据方向位（R/W）。第二个字节为 10 位地址的低八位。

具体如下图所示:

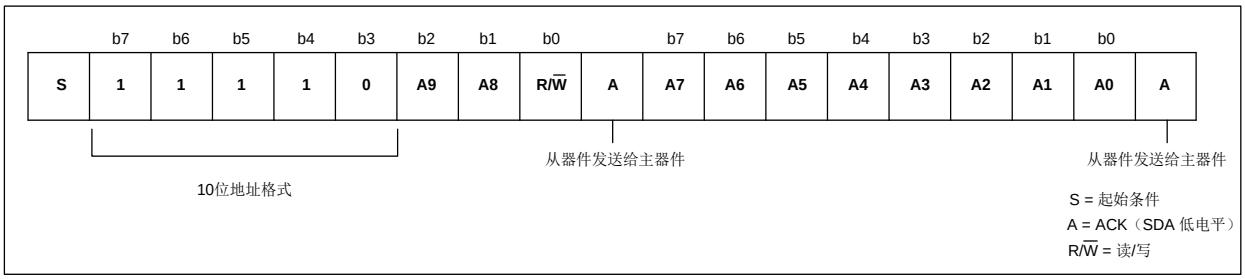

图 14-4 10 位地址格式

下表定义了 I2C 首字节的特殊用途和保留地址。

表 14-2 I2C 首字节

从地址	R/W 位	描述
0000 000	0	广播呼叫地址。I2C 将数据放入接收缓冲，并产生广播呼叫中断
0000 000	1	起始字节
0000 001	x	CBUS 地址。I2C 接口忽略该访问
0000 010	x	保留
0000 011	x	保留
0000 1xx	x	保留
1111 1xx	x	保留
1111 0xx	x	10 位从器件寻址

14.3.3.3 发送和接收协议

主器件可以发起数据传输，作为主发送器或主接收器，向总线发送数据或从总线接收数据。从器件响应主器件的请求，充当从发送器或从接收器。

- 主发送和从接收

所有数据都以字节格式传输，对每次传输的字节数没有限制。主器件发送完地址和 R/W 位或者一个字节的数据到从器件后，从接收器必须产生一个响应信号（ACK）。从接收器不能产生响应信号（ACK）时，主器件将会产生一个停止条件中止传输。从器件不能响应时，必须释放 SDA 为高电平，以便主器件产生停止条件。

当主发送器如下图所示传输数据时，从接收器在接收到的每个字节后产生一个 ACK 来响应主发送器。

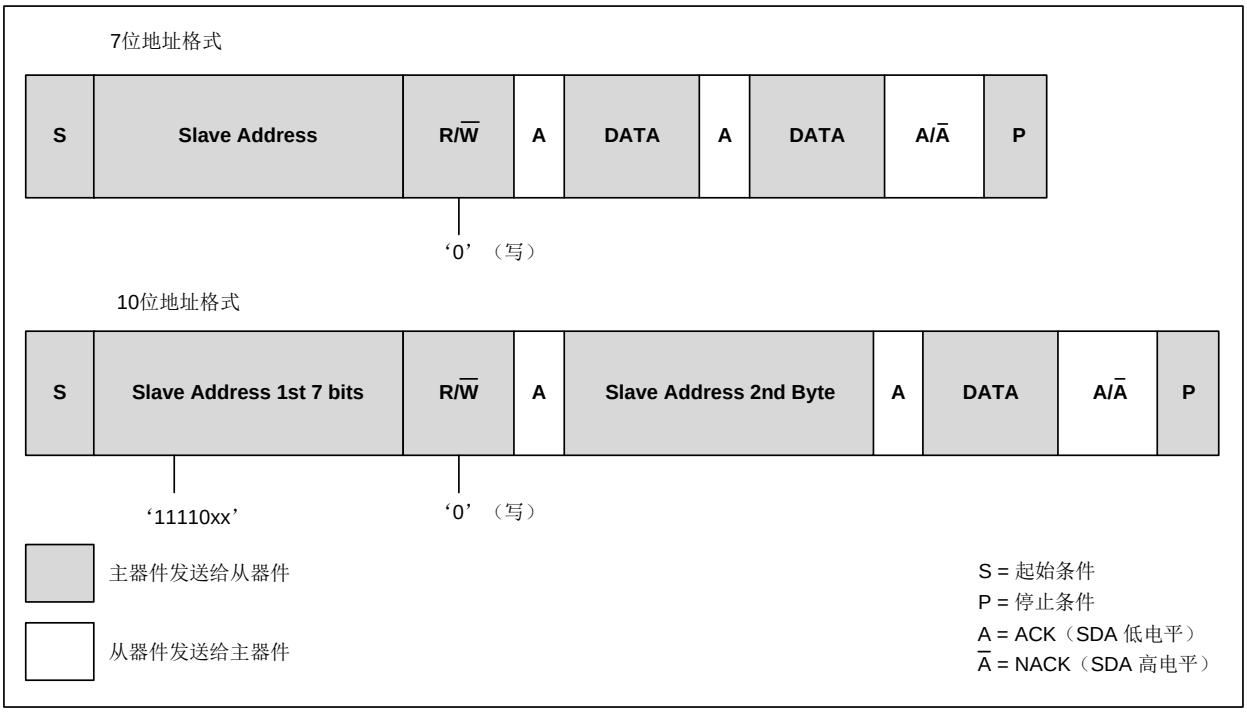

图 14-5 主发送协议

- 主接收和从发送

当主器件如下图所示接收数据时，主器件在每次接收到一个字节数据后响应从发送器，最后一个字节除外。通过这种方式，主接收器能够通知从发送器是否为最后一个字节。从发送器在检测到 NACK 时必须释放 SDA，以便主器件产生停止条件。

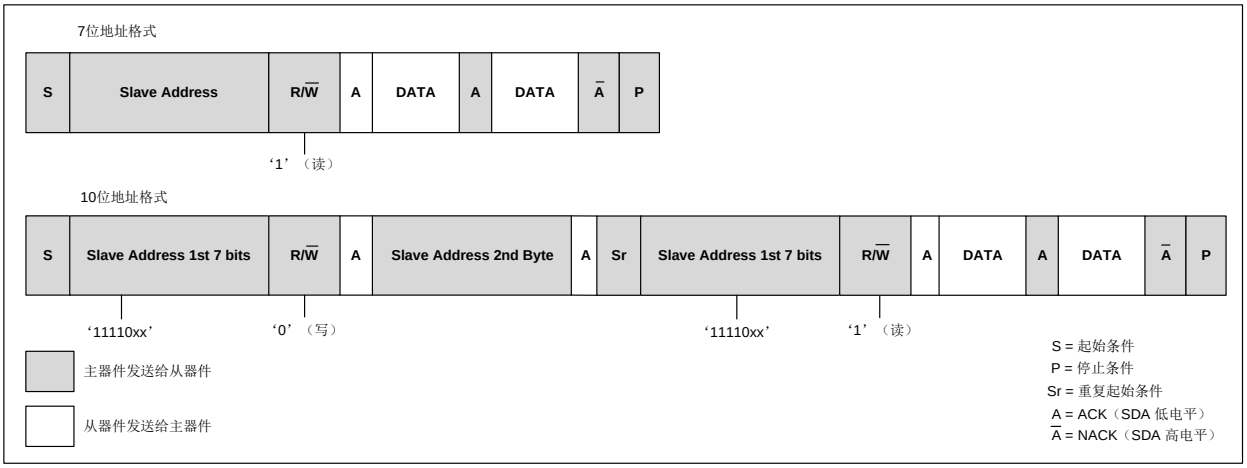

图 14-6 主接收协议

主器件不想因产生停止条件而释放总线时，可以产生一个重复起始条件。重复起始条件与起始条件相同，但重复起始条件在 ACK 后产生。在主模式下，I2C 接口可以使用不同的传输方向与相同的从器件通信。

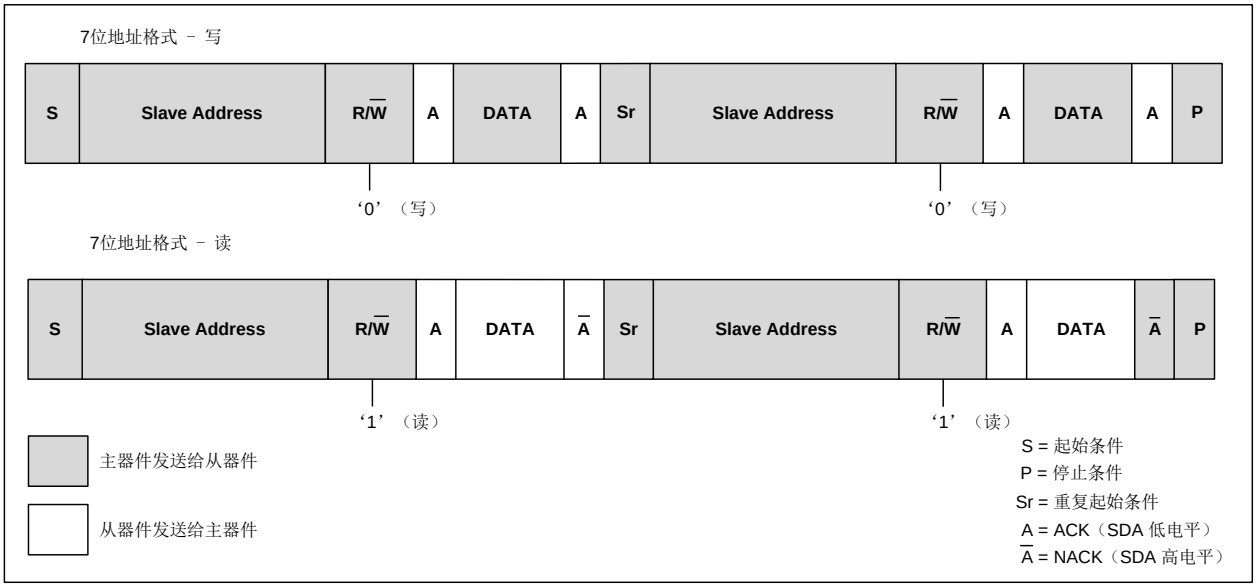

图 14-7 带 RESTART（SR）信号的主发送和接收协议

- 起始字节传输协议

起始字节传输协议用于无专用 I2C 硬件模块的系统。当 I2C 接口模块作为主器件时，在每次传输开始，可以为需要的从器件产生起始字节输出。

如下图所示，该协议由 7 个 0 以及 1 个 1 组成。处理器可以在地址阶段用低速采样来查询总线。一旦检测到 0，处理器可以从低速采样切换到主器件的正常速率。

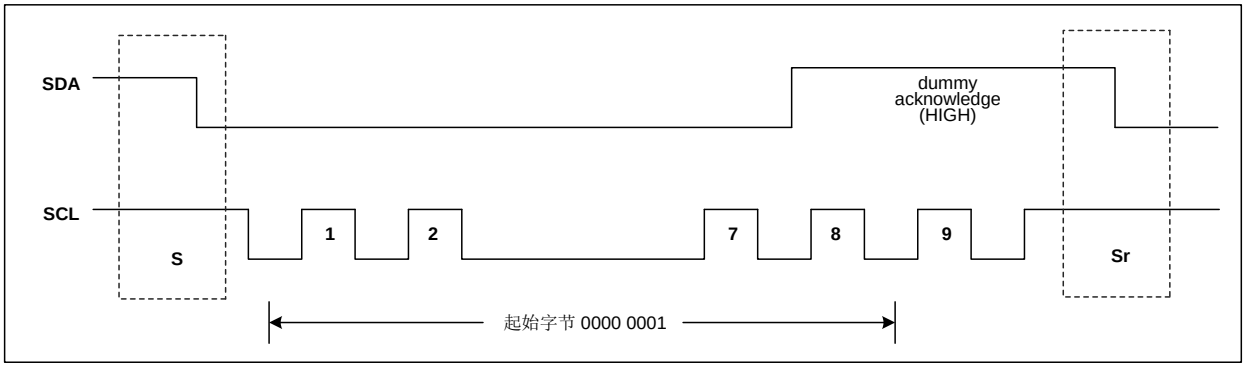

图 14-8 起始字节传输

起始字节程序流程如下:

1） 主器件产生起始条件

2） 主器件发送起始字节（0000 0001）

3） 主器件发送应答相关的时钟脉冲（为了符合总线上的字节处理格式）

4） 没有从器件响应 ACK 信号

5） 主器件产生重复起始条件（RESTART）

I2C 硬件接收器无需响应起始字节，因为这是一个保留地址，且地址会在 RESTART 后复位。

14.3.3.4 发送缓冲管理以及起始、停止和重复起始条件产生

当工作在主机模式，每当 TX FIFO 为空时 I2C 模块就在总线上产生一个停止条件。如果重复起始产生功能使能 (RESTART = 1)，则传输方向从读变为写或者写变为读时产生重复起始条件。如果没有使能重复起始条件，则会在停止条件后产生一个起始条件。

下图显示了 DR 寄存器的位。

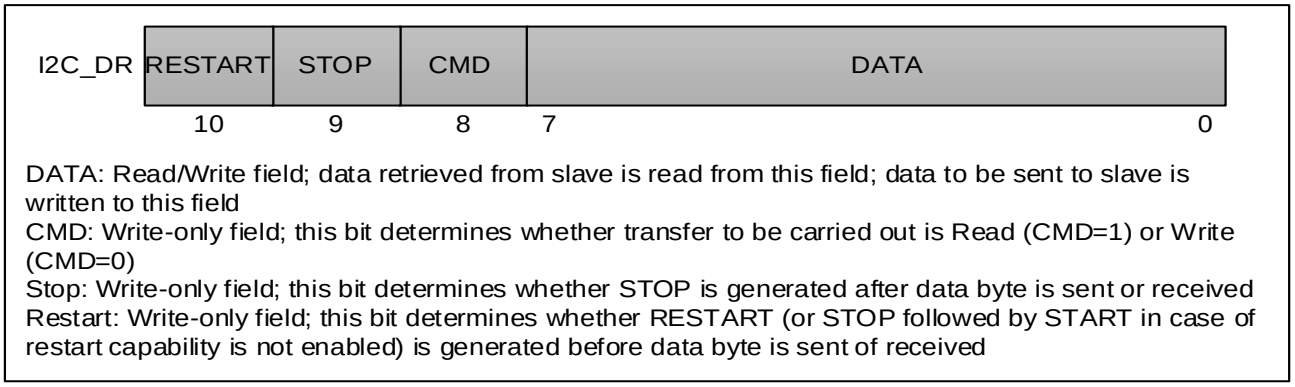

图 14-9 I2C_DR 寄存器

下面的时序图描述了 I2C 模块工作在主发送模式下 TX FIFO 变为空或产生 STOP 时的行为。

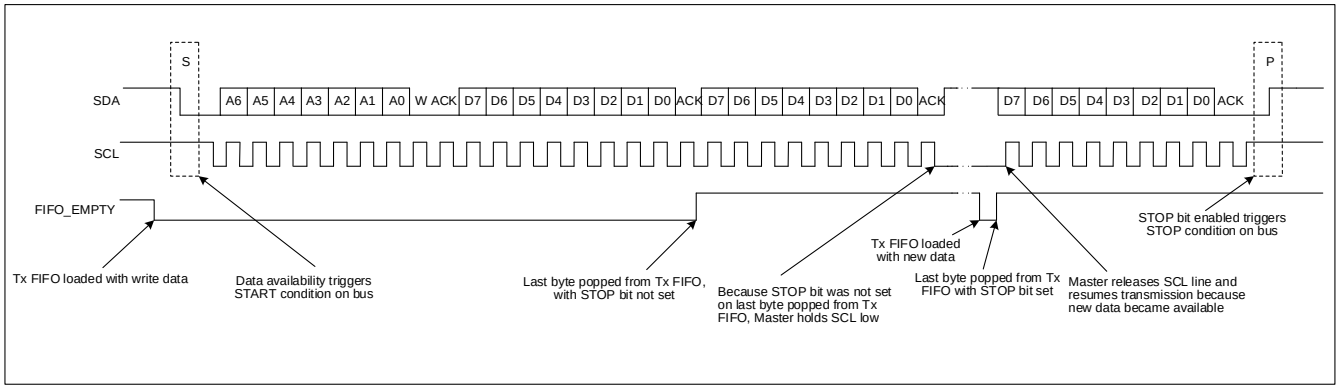

图 14-10 主发送-TX FIFO 为空或产生 STOP

下面的时序图描述了 I2C 模块工作在主接收模式下当 TX FIFO 变为空或产生 STOP 时的行为

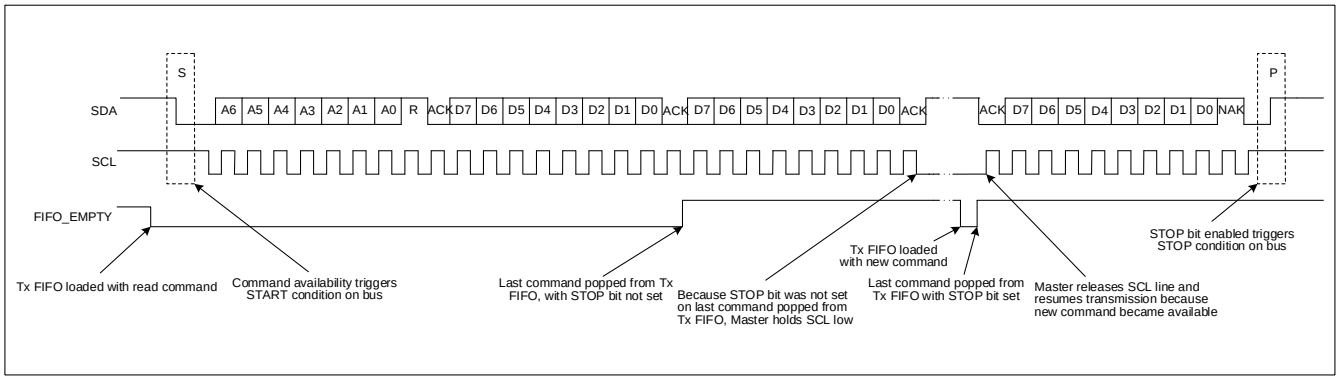

图 14-11 主接收-TX FIFO 为空或产生 STOP

下图描述了 I2C 模块工作在主发送模式下，发生 RESTART（I2C_CR.REPEN 为 1）时的时序。I2C_CR.REPEN 为 0 时，则先生成 STOP 再生成 START，以此代替 RESTART。

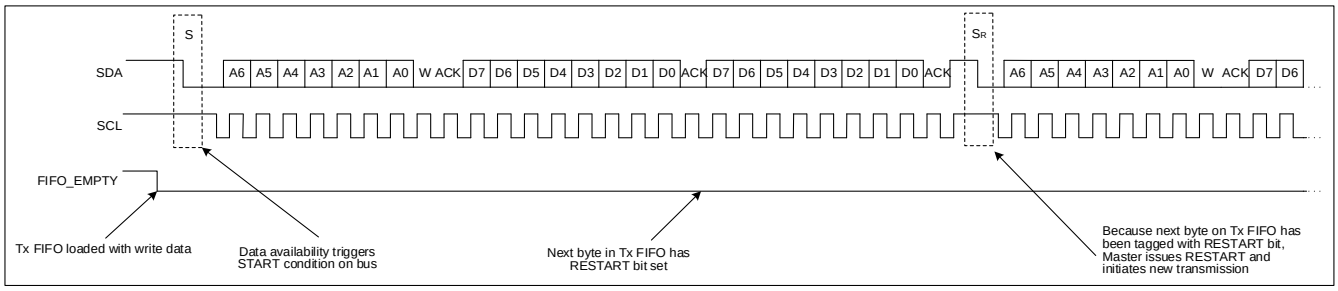

图 14-12 主发送-产生 RESTART

下图描述了 I2C 模块工作在主接收模式下，发生 RESTART 时的时序。

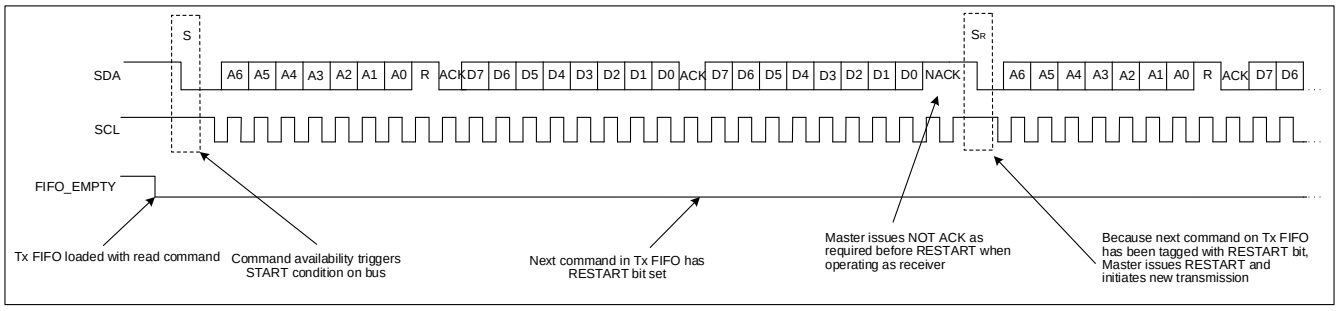

图 14-13 主接收-产生 RESTART

14.3.3.5 仲裁

如果总线上有多个主器件试图通过同时生成 START 条件来控制总线，那么就会有一个仲裁过程（多个主器件同时尝试控制总线，但只允许其中一个控制总线并使报文不被破坏的过程）。一旦其中一个主器件已经控制了总线，那么直到该主器件发送停止条件并将总线释放为空闲状态后，其他主器件才能控制总线。

当 SCL 线为高电平时，仲裁在 SDA 线发生。如果两个或多个主器件尝试发送信息到总线，在其他

主器件都产生 0 的情况下，首先产生 1 的主器件将丢失仲裁。丢失仲裁的主器件可以继续产生时钟脉冲直到字节传输结束。如果每个主器件都尝试寻址相同的器件，仲裁会继续在数据阶段进行。

检测到丢失仲裁后，I2C 接口会停止产生 SCL 信号。

下图显示了两个主器件仲裁的总线时序。

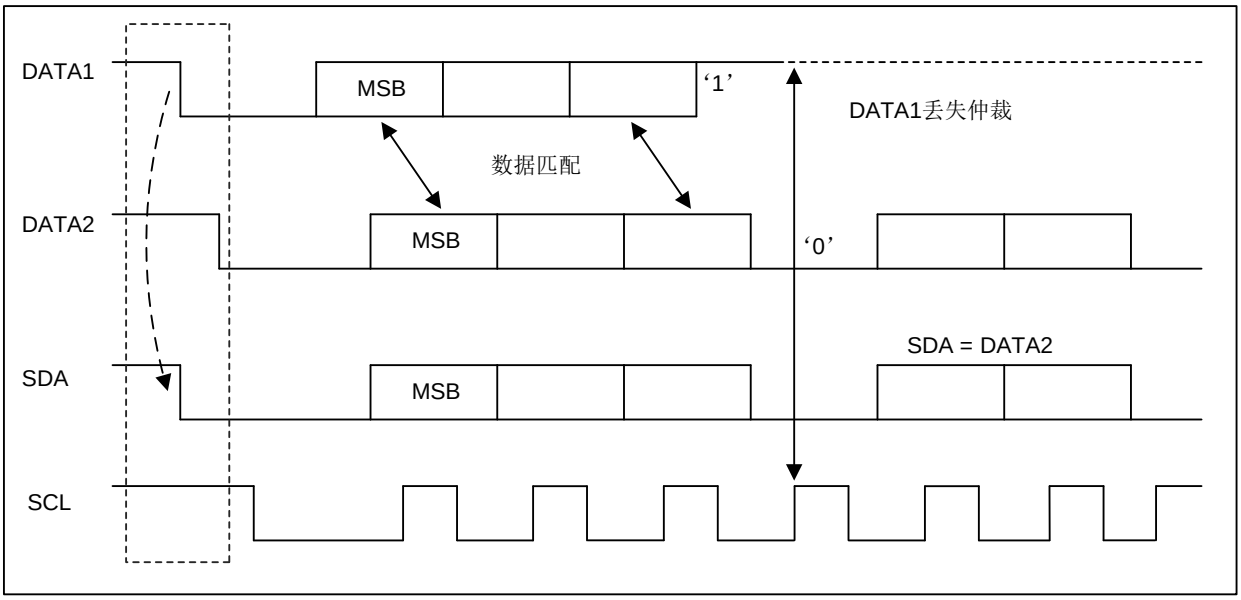

图 14-14 两个主器件仲裁

14.3.3.6 时钟同步

当两个或多个主器件试图同时在总线上传输信息时，必须仲裁和同步 SCL 时钟。所有主器件都各自产生时钟来传输消息。数据只在时钟的高电平有效。时钟同步通过 SCL 信号的线“与”连接进行。当主器件把 SCL 时钟转换为 0 时，主器件会计算 SCL 低电平的时间，在下一个时钟周期开始时把 SCL 转换为 1。但如果另一个主器件把 SCL 保持为 0，那么这个主器件会进入等待状态直到 SCL 变为 1。

之后，所有主器件计算各自的高电平时间，最短高电平时间的主器件把 SCL 转换为 0。接着，所有主器件计算各自的低电平时间，最长低电平时间的主器件强制其他主器件进入等待状态。由此产生一个同步后的 SCL 时钟，如下图所示。

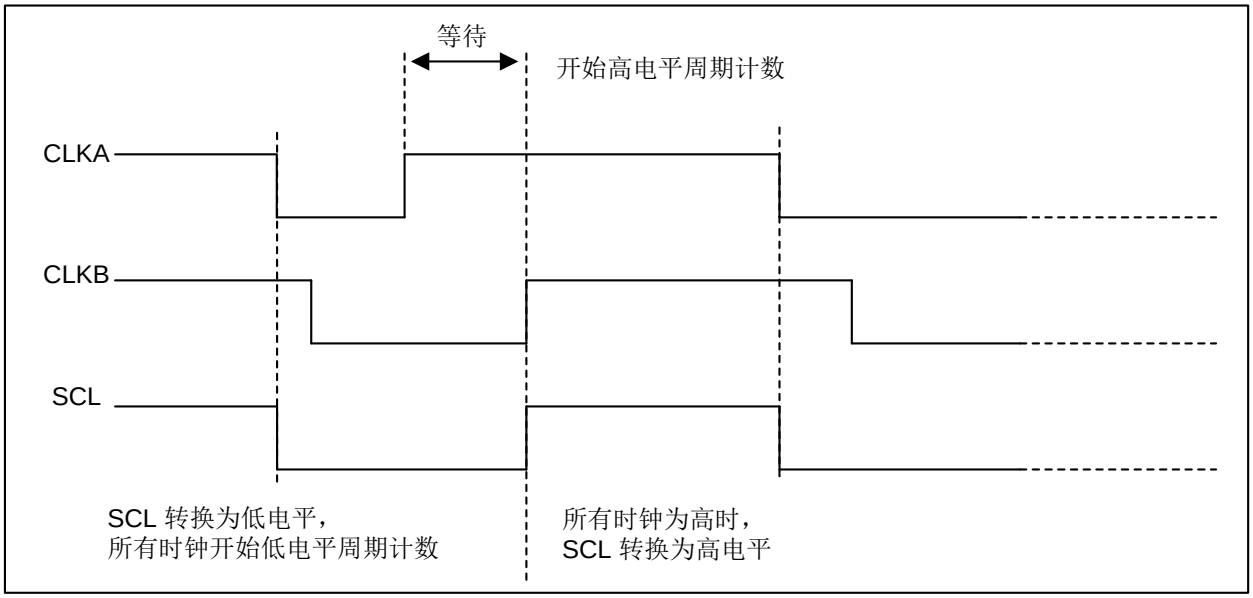

图 14-15 时钟同步（示意图）

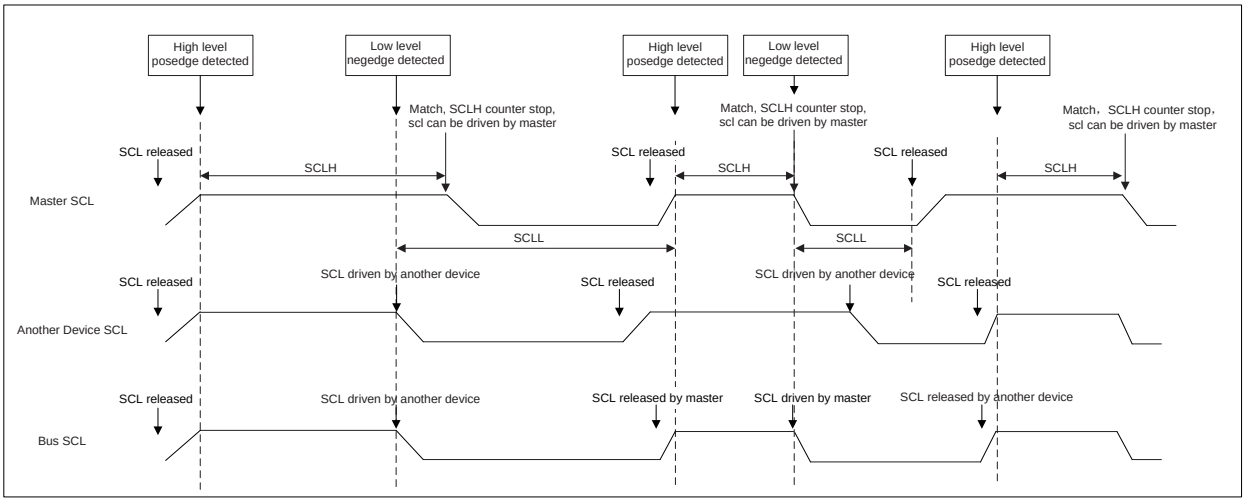

图 14-16 时钟同步（时序图）

14.3.3.7 SCL 配置

I2C 的 SCL 可参考如下配置：

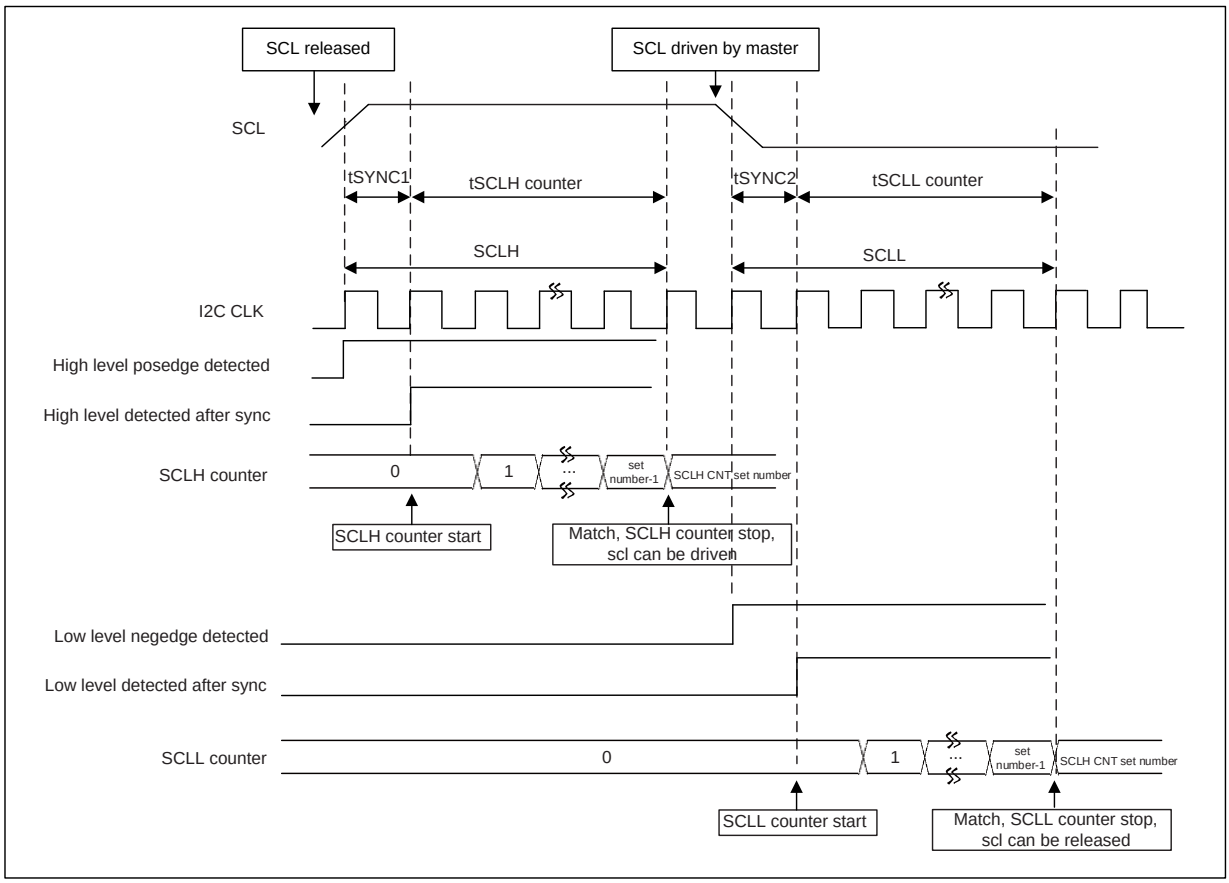

图 14-17 SCL 生成时序

- 标准模式：

SCLH =（SSHR + 12） x I2C CLK + tSYNC1

SCLL =（SSLR + 1） x I2C CLK + tSYNC2

注：tSYNC1 为 0~1 个 I2C CLK，tSYNC2 为 0~1 个 I2C CLK

- 快速模式：

SCLH =（FSHR + 12） x I2C CLK + tSYNC1

SCLL =（FSLR + 1） x I2C CLK + tSYNC2

注：tSYNC1 为 2~3 个 I2C CLK，tSYNC2 为 2~3 个 I2C CLK

14.3.4 工作模式

I2C 接口可以以下述 4 种方式之一运行：

- 从发送模式
- 从接收模式
- 主发送模式

- 主接收模式

注：I2C 接口模块只能工作在主模式或从模式，不能同时工作在两种模式下。因此需要确保寄存器 I2C_CR.DISSLAVE（位 6）和 I2C_CR.MASTER（位 0）不能分别设置为 0 和 1（或者分别设置为 1 和 0）。

14.3.4.1 从模式

- 初始化配置

1） 写 0 到 I2C_ENR.ENABLE（位 0）以禁止 I2C

2） 通过配置 I2C_SAR 寄存器指定从地址（该地址为 I2C 接口所响应的地址）

3） 通过配置 I2C_CR.SLAVE10（位 3）指定地址格式（7 位或 10 位地址）；写 0 到 I2C_CR.DISSLAVE（位 6）和 I2C_CR.MASTER（位 0）

4） 置位 I2C_ENR.ENABLE（位 0），使能 I2C 接口模块

- 从发送的单字节操作

I2C 接口被其他 I2C 主器件寻址并请求数据时，I2C 接口工作在从发送模式，工作流程如下：

1） 其他 I2C 主器件发起 I2C 传输，发送地址与 I2C_SAR 寄存器的从地址匹配

2） I2C 接口响应发送的地址，识别传输方向以表明其工作在从发送模式

3） I2C 接口产生 RD_REQ 中断（I2C_RAWISR 位 5），并将 SCL 线拉低；总线一直处于等待状态直到软件响应。若 RD_REQ 中断被屏蔽（I2C_IMR 位 5 为 0），建议 CPU 定期查询 I2C_RAWISR 寄存器：

a. 读到 I2C_RAWISR.RD_REQ（位 5）为 1 等效于产生了 RD_REQ 中断

b. 软件必须采取行动以满足 I2C 传输的要求

c. 时间间隔通常在 10 个 SCL 时钟周期左右（例如，400kbps 的时间间隔是 25us）

4） 如果在接收到读请求前 TX FIFO 中仍有数据，I2C 接口会产生 TX_ABRT 中断（I2C_RAWISR 位 6），以清空 TX FIFO 中的数据（I2C_CR.SLV_TX_ ABRT_DIS 为 0 时）

因此，在试图写入 TX FIFO 前，软件有必要通过读取 I2C_TX_ABRT 寄存器来清除 TX_ABRT 中断。若 TX_ABRT 中断被屏蔽（I2C_IMR 位 6 为 0），建议 CPU 定期查询 I2C_RAWISR 寄存器。读到 I2C_RAWISR.TX_ABRT（位 6）为 1 等效于产生了 TX_ABRT 中断。

5） 软件写数据到 I2C_DR 寄存器（其中位 8 设置为 0）

6） 软件必须先清除 I2C_RAWISR.RD_REQ（位 5）和 I2C_RAWISR.TX_ABRT（位 6）中断

7） I2C 接口释放 SCL，并发送数据字节

8） 主器件发送重复起始条件控制总线，或发送停止条件释放总线

- 从接收的单字节操作

其他主器件寻址 I2C 接口并发送数据时，I2C 接口工作在从接收模式，工作流程如下：

1） 其他 I2C 主器件发起 I2C 传输，发送地址与 I2C_SAR 寄存器的从地址匹配

2） I2C 接口响应发送的地址，识别传输方向以表明其工作在从接收模式

3） I2C 接口接收主器件发送的数据并将其存储在接收缓冲中

4） I2C 接口产生 RX_FULL 中断（I2C_RAWISR 位 2）

若 RX_FULL 中断被屏蔽（I2C_IMR 位 2 为 0），建议 CPU 定期查询 I2C_SR 寄存器。读到 I2C_SR.RFNE（位 3）为 1 等效于产生了 RX_FULL 中断

5） 软件通过读 I2C_DR 寄存器（位 7:0）来获得接收到的数据

6） 主器件发送重复起始条件控制总线，或发送停止条件释放总线

- 程序流程图

下图为 I2C 接口作为从器件时的程序示例：

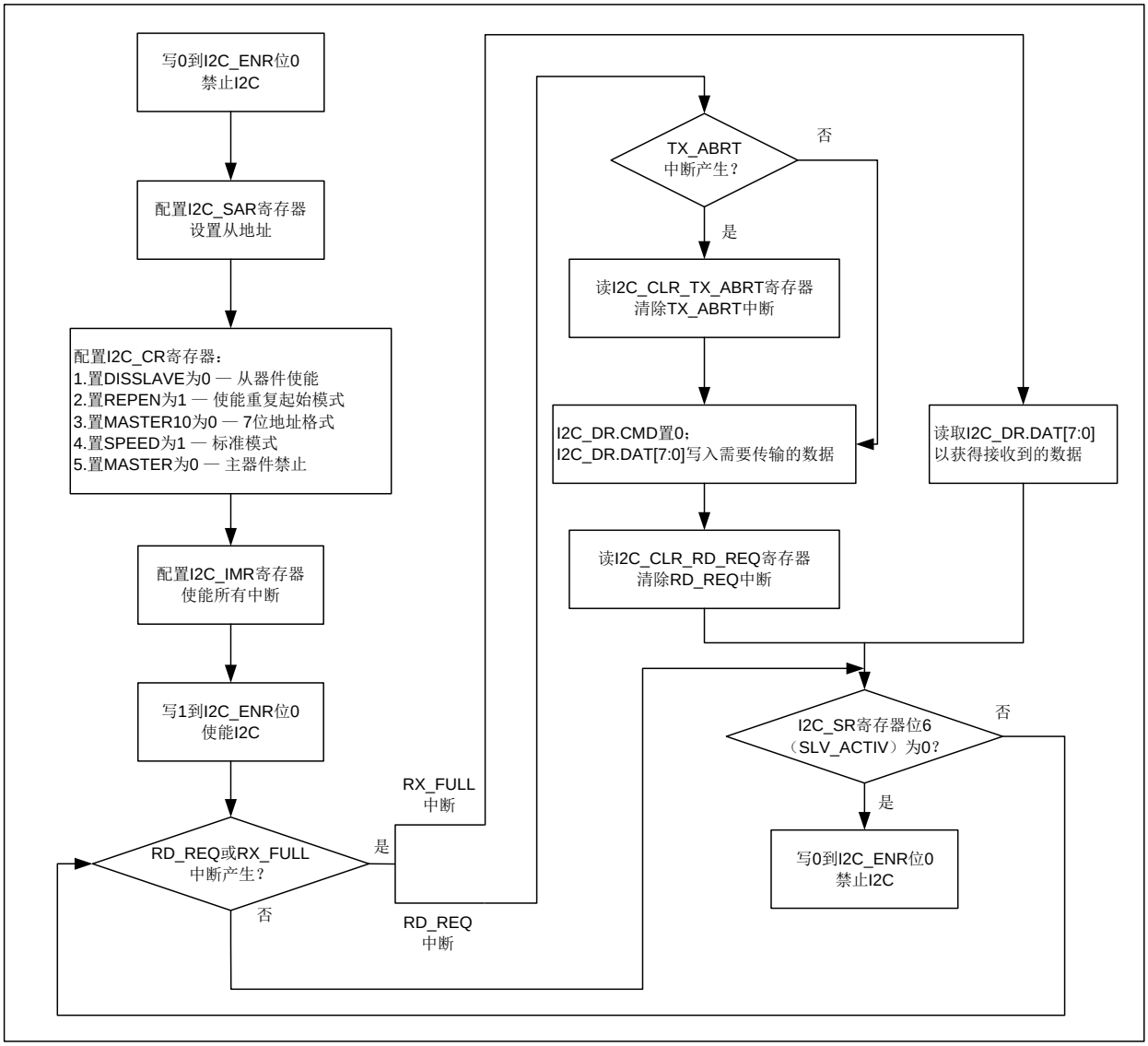

图 14-18 流程图（I2C 接口作为从器件）

- 从器件的块传输操作

标准 I2C 协议中，所有的数据处理都是单字节的处理，程序通过写一个字节到从器件的 TX FIFO 来响应主器件的读请求。当从器件（从发送）接收到主器件（主接收）的读请求（RD_REQ）时，至少应在从发送的 TX FIFO 中放置一个数据。I2C 接口模块可以处理 TX FIFO 中的多个数据，因此，后续的读请求无需再产生中断来获取数据。这极大地减少了因每次数据中断而导致的等待时间。

此模式仅适用于 I2C 接口作为从发送器的情况。如果接收到主器件发送的 ACK，从器件的 TX FIFO 中没有数据，I2C 接口将拉低总线的 SCL 线直到读请求中断（RD_REQ）产生并且 TX FIFO 的数据准备好后才释放 SCL 线。

如果 RX_REQ 中断被屏蔽（I2C_ISR 位 5 为 0），软件可以定期查询 I2C_RAWISR 寄存器。读到 I2C_RAWISR. RX_REQ（位 5）为 1 等效于产生了 RX_REQ 中断。

RD_REQ 中断由读请求产生，和中断一样必须在退出中断服务程序（ISR）时清除。在中断服务程序中（ISR）可以写一个或多个字节的数据到 TX FIFO。在将这些字节传输给主器件的过程中，如果主器件响应了最后一个字节，则说明主器件正在要求更多的数据，那么从器件必须再次产生 RD_REQ 中断请求。

如果主器件要从 I2C 接口接收 n 个字节，但程序写到 TX FIFO 中的数据个数大于 n，那么从器件在发送完要求的 n 个字节数据后，将会清空 TX FIFO 并忽略额外的字节。

14.3.4.2 主模式

- 初始化配置

1） 写 0 到 I2C_ENR.ENALE（位 0）以禁止 I2C

2） 通过配置 I2C_CR.SPEED（位 2:1）指定 I2C 工作的速率模式（标准模式、快速模式）；同时确保 I2C_CR.DISSLAVE（位 6）和 I2C_CR.MASTER（位 0）都为 1

3）将要寻址的 I2C 器件地址写入 I2C_TAR（该寄存器还可以配置为广播呼叫地址或起始字节命令）

4） 置位 I2C_ENR.ENALE（位 0），使能 I2C 接口模块

5） 将传输方向和数据写入 I2C_DR，I2C 接口由此产生起始条件并发送地址字节

如果在使能 I2C 接口前配置了 I2C_DR 寄存器，数据和命令都会丢失，因为在 I2C 接口禁止的情况下缓冲是清空的。

- 主发送和主接收

I2C 接口支持读写的动态切换。发送数据时，写数据到 I2C_DR 的低字节中，配置 I2C_DR.CMD（位 8）为 0 产生写操作。随后的读命令无需配置 I2C_DR 的低字节，只需要配置 I2C_DR.CMD（位 8）为 1 即可。若 TX FIFO 为空，I2C 模块拉低 SCL 直到下个命令写入到 TX FIFO 中。

- 程序流程图

下图为 I2C 接口作为主器件时的程序示例：

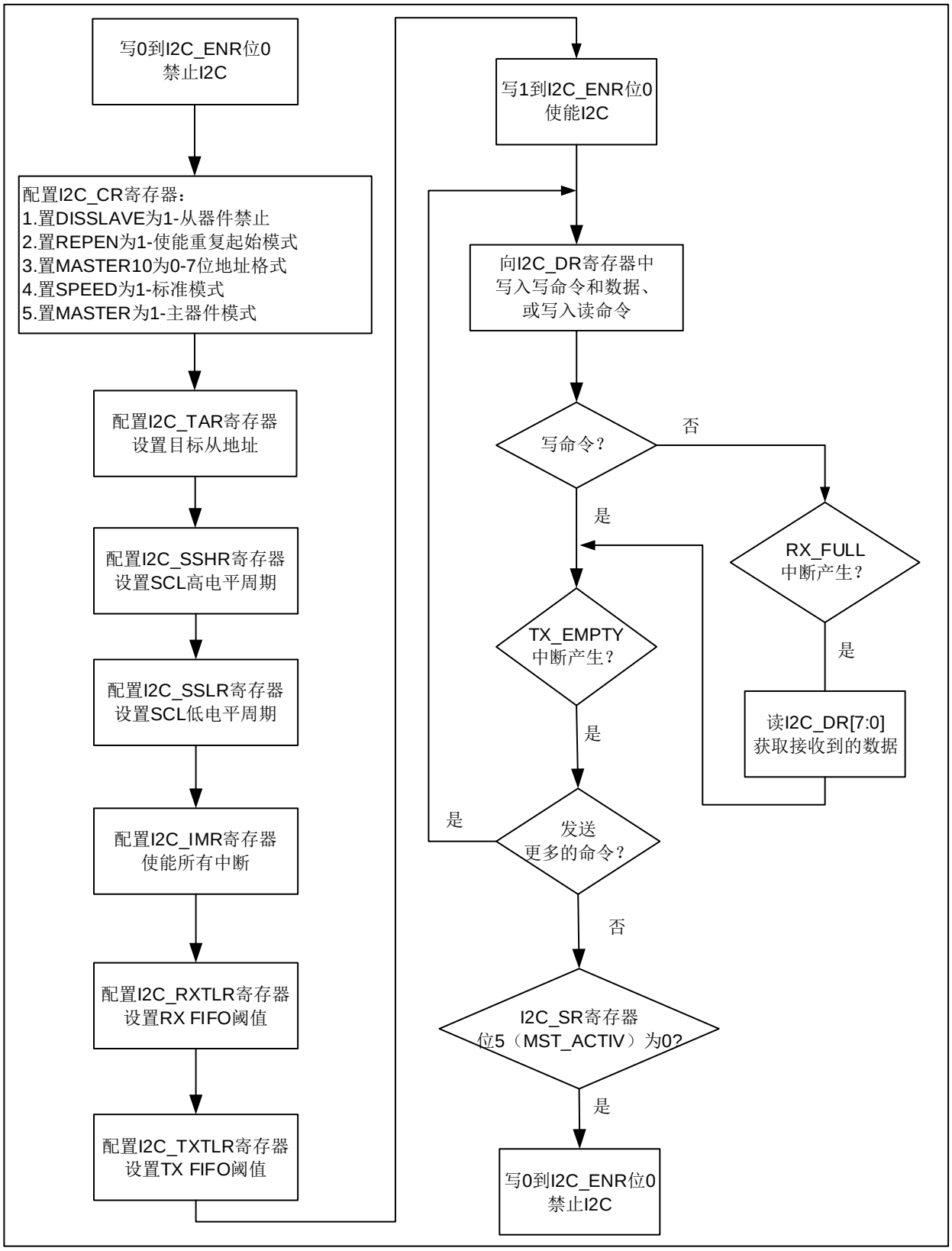

图 14-19 流程图（I2C 接口作为主器件）

14.3.4.3 中止传输

I2C_ENR.ABORT（位 1）允许软件在完成 TX FIFO 中的传输命令之前放弃 I2C 总线。作为 ABORT 请求的响应，I2C 模块发出停止条件到 I2C 总线，同时清空 TX FIFO。中止传输只允许在主模式下执行。

- 程序流程

1） 停止向 TX FIFO（I2C_DR）中写入新命令

14.3.5 中断

下表列出了 I2C 的中断位及其设置和清除方式。部分位由硬件置位并由软件清除；另一部分位由硬件置位和清除。

表 14-3 中断位的置位和清除

中断位	硬件置位/软件清除	硬件置位和清除
GEN_CALL	√	x
START_DET	√	x
STOP_DET	√	x
ACTIVITY	√	x
RX_DONE	√	x
TX_ABRT	√	x
RD_REQ	√	x
TX_EMPTY	x	√
TX_OVER	√	x
RX_FULL	x	√
RX_OVER	√	x
RX_UNDER	√	x

下图描述了中断寄存器中，中断位被硬件置位和软件清除的操作。

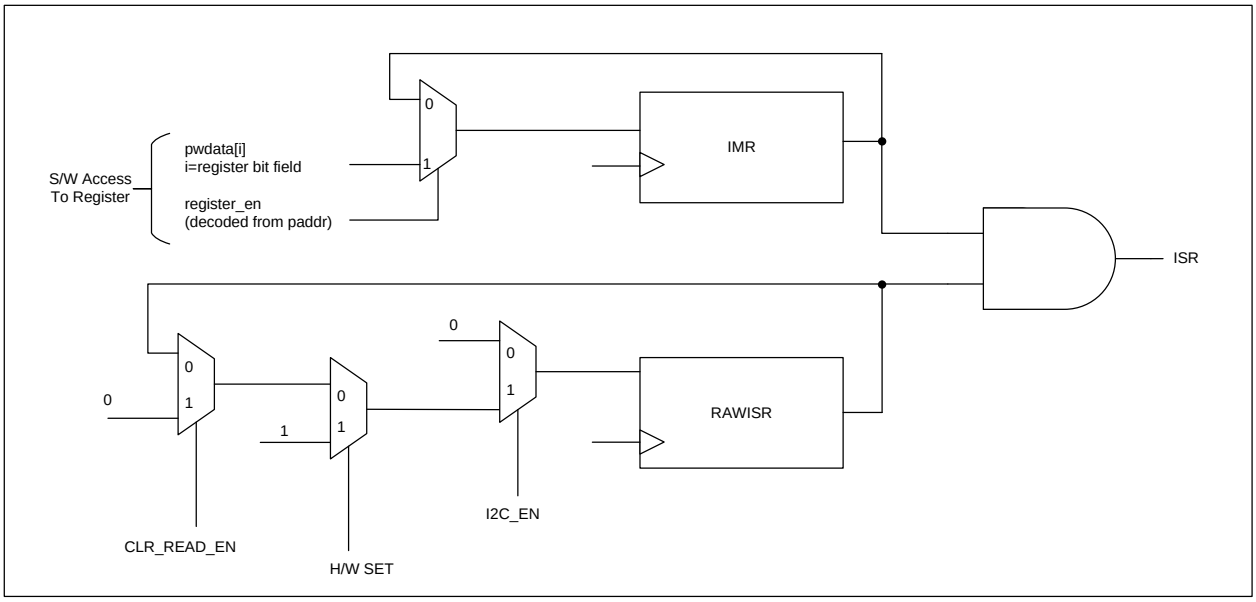

图 14-20 I2C 中断机制

14.4 寄存器

14.4.1 寄存器总览

表 14-4 I2C 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	I2C_CR	控制寄存器	0x0000007F
0x04	I2C_TAR	目标地址寄存器	0x00000055
0x08	I2C_SAR	从地址寄存器	0x00000055
0x10	I2C_DR	数据命令寄存器	0x00000000
0x14	I2C_SSHR	标准模式时钟高电平计数寄存器	0x00000190
0x18	I2C_SSLR	标准模式时钟低电平计数寄存器	0x000001D6
0x1C	I2C_FSHR	快速模式时钟高电平计数寄存器	0x0000003C
0x20	I2C_FSLR	快速模式时钟低电平计数寄存器	0x00000082
0x2C	I2C_ISR	中断状态寄存器	0x00000000
0x30	I2C_IMR	中断屏蔽寄存器	0x000008FF
0x34	I2C_RAWISR	RAW 中断寄存器	0x00000000
0x38	I2C_RXTLR	接收阈值	0x00000000
0x3C	I2C_TXTLR	发送阈值	0x00000000
0x40	I2C_ICR	组合和独立中断清除寄存器	0x00000000
0x44	I2C_RX_UNDER	清除 RX_UNDER 中断寄存器	0x00000000
0x48	I2C_RX_OVER	清除 RX_OVER 中断寄存器	0x00000000
0x4C	I2C_TX_OVER	清除 TX_OVER 中断寄存器	0x00000000
0x50	I2C_RD_REQ	清除 RD_REQ 中断寄存器	0x00000000

Offset	Acronym	Register Name	Reset
0x54	I2C_TX_ABRT	清除 TX_ABRT 中断寄存器	0x00000000
0x58	I2C_RX_DONE	清除 RX_DONE 中断寄存器	0x00000000
0x5C	I2C_ACTIV	清除 ACTIVITY 中断寄存器	0x00000000
0x60	I2C_STOP	清除 STOP_DET 中断寄存器	0x00000000
0x64	I2C_START	清除 START_DET 中断寄存器	0x00000000
0x68	I2C_GC	清除 GEN_CALL 中断寄存器	0x00000000
0x6C	I2C_ENR	使能寄存器	0x00000000
0x70	I2C_SR	状态寄存器	0x00000006
0x74	I2C_TXFLR	发送缓冲级别寄存器	0x00000000
0x78	I2C_RXFLR	接收缓冲级别寄存器	0x00000000
0x7C	I2C_HOLD	SDA 保持时间寄存器	0x00000001
0x94	I2C_SETUP	SDA 建立时间寄存器	0x00000064
0x98	I2C_GCR	广播呼叫 ACK 寄存器	0x00000001
0xB0	I2C_SLVMASK	从地址掩码寄存器	0x000003FF
0xB4	I2C_SLVRCVADDR	从器件接收地址寄存器	0x00000000

14.4.2 I2C_CR 控制寄存器

偏移地址：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.			PAD_SEL	Res.			EMPINT	STOPINT	DISSLAVE	REPEN	MASTER10	SLAVE10	SPEED		MASTER
			rw				rw	rw	rw	rw	rw	rw	rw		rw

Bit	Field	Description
31：13	Reserved	保留，必须保持复位值
12	PAD_SEL	PAD MUX 交换 SCL、SDA 对应 PAD 0：PAD0 ≒ SCL；PAD1 ≒ SDA 1：PAD0 ≒ SDA；PAD1 ≒ SCL
11：9	Reserved	保留，必须保持复位值
8	EMPINT	该位控制 TX_EMPTY 中断的产生，细节请参考 I2C_RAWISR 寄存器

Bit	Field	Description
7	STOPINT	从模式下，是否产生 STOP 中断 0：无论地址是否匹配，都产生 STOP 中断 1：仅在地址匹配时产生 STOP 中断 注：广播地址寻址时，如果该位置位，从器件不产生 STOP 中断。STOP 中断仅在发送地址与从地址（I2C_SAR）匹配时产生。
6	DISSLAVE	该位控制是否禁用 I2C 从器件 0：从器件使能 1：从器件禁止
5	REPEN	I2C 作为主器件时是否发送 RESTART 条件 0：禁止 1：使能 RESTART 条件可以替换为先发送停止条件再发送起始条件。 当 RESTART 被禁止时，I2C 接口作为主器件不能执行以下功能： 发送起始字节 组合格式模式下改变传输方向 10 位地址格式的读操作 执行上述操作会置位 I2C_RAWISR.TX_ABRT（位 6）。
4	MASTER10	I2C 作为主器件时的地址格式 0：7 位地址格式 1：10 位地址格式
3	SLAVE10	I2C 作为从器件时，该位控制是否响应 7 位或 10 位地址 0：7 位寻址。I2C 接口忽略 10 位寻址；对于 7 位寻址，仅比较 I2C_SAR 寄存器的低 7 位 1：10 位寻址。I2C 仅响应 10 位寻址，接收地址与 I2C_SAR 的 10 位比较
2：1	SPEED	该两位控制 I2C 接口工作在主模式时的速率 01：标准模式（最大 100Kbps） 10：快速模式（最大 400Kbps）
0	MASTER	该位控制是否使能主器件 0：主器件禁止 1：主器件使能

I2C_CR.DISSLAVE（位 6）和 I2C_CR.MASTER（位 0）的配置组合如下表所示：

表 14-5 DISSLAVE 和 MASTER 配置

DISSLAVE（I2C_CR[6]）	MASTER（I2C_CR[0]）	状态
0	0	从器件
0	1	禁止配置
1	0	禁止配置
1	1	主器件

14.4.3 I2C_TAR 目标地址寄存器

偏移地址：0x04

复位值：0x0000 0055

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.				SPECIAL	GC	ADDR									
				rw	rw	rw									

Bit	Field	Description
31：12	Reserved	保留，必须保持复位值
11	SPECIAL	该位指示软件执行的是否为特殊命令（广播呼叫或起始字节命令） 0：忽略 GC（位 10），正常使用 ADDR（位 9:0） 1：执行特殊 I2C 命令，细节请参考 GC 位的描述
10	GC	位 11 置位时，该位显示 I2C 执行的是广播呼叫还是起始字节命令 0：广播呼叫 发起广播呼叫后只能执行写操作。I2C 接口一直工作在广播呼叫模式下，直到 SPECIAL（位 11）被清零 1：起始字节命令
9：0	ADDR	主模式时的目标地址 发起广播呼叫时，该字段被忽略。 要产生开始字节命令，CPU 只需要对该字段写一次。

14.4.4 I2C_SAR 从地址寄存器

偏移地址：0x08

复位值：0x0000 0055

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.						ADDR									
						rw									

Bit	Field	Description
31：10	Reserved	保留，必须保持复位值
9：0	ADDR	I2C 接口的从地址。对于 7 位地址格式，ADDR 只有[6：0]有效。

14.4.5 I2C_DR 数据命令寄存器

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.					RESTART	STOP	CMD	DAT							
					w	w	w	rw							

Bit	Field	Description
31：12	Reserved	保留，必须保持复位值
10	RESTART	发送或接收字节前，是否产生 RESTART 0：如果 REPEN 为 1，则仅在前一个命令改变传输方向时才产生 RESTART；如果 REPEN 为 0，则先产生 STOP 再产生 START 1：如果 REPEN 为 1，则在数据接收或发送（根据 CMD 的值）前产生 RESTART，无论前一个命令是否改变数据的传输方向；如果 REPEN 为 0，则先产生 STOP 再产生 START
9	STOP	发送或接收字节后，是否产生 STOP 0：当前字节之后不产生 STOP，无论 TX FIFO 是否为空。如果 TX FIFO 不为空，主器件继续当前传输（根据 CMD 的值发送或接收数据）。如果 TX FIFO 为空，主器件将拉低 SCL 线，挂起总线直至 TX FIFO 收到新命令 1：当前字节之后产生 STOP，无论 TX FIFO 是否为空。如果 TX FIFO 不为空，主机将立即通过发送 START 来尝试发起一次新的传输。
8	CMD	控制在主模式下执行读或写操作 0：写 1：读 向 TX FIFO 中输入命令时，该位用于区分读和写命令。从接收模式下，对该位的写操作被忽略；从发送模式下，写 0 表示发送 I2C_DR 寄存器的数据。
7：0	DAT	I2C 总线待发送或接收到的数据

14.4.6 I2C_SSHR 标准模式时钟高电平计数寄存器

偏移地址：0x14

复位值：0x0000 0190

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
CNT															
rw															

Bit	Field	Description
31：16	Reserved	保留，必须保持复位值
15：0	CNT	标准模式下 SCL 时钟高电平周期 注：该寄存器只可配置在 6 至 65525 之间，这是由于 I2C 接口使用了一个 16 位的计数器，该计数器值等于 CNT+10 时标志 I2C 总线处于空闲状态。

14.4.7 I2C_SSLR 标准模式时钟低电平计数寄存器

偏移地址：0x18

复位值：0x0000 01D6

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
CNT															
rw															

Bit	Field	Description
31：16	Reserved	保留，必须保持复位值
15：0	CNT	标准模式下 SCL 时钟低电平周期，最小值为 8。

14.4.8 I2C_FSHR 快速模式时钟高电平计数寄存器

偏移地址：0x1C

复位值：0x0000 003C

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
CNT															
280															

rw

Bit	Field	Description
31：16	Reserved	保留，必须保持复位值
15：0	CNT	快速模式下 SCL 时钟高电平周期，最小值为 6。

14.4.9 I2C_FSLR 快速模式时钟低电平计数寄存器

偏移地址：0x20

复位值：0x0000 0082

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
CNT															
rw															

Bit	Field	Description
31：16	Reserved	保留，必须保持复位值
15：0	CNT	快速模式下 SCL 时钟低电平周期，最小值为 8。

14.4.10 I2C_ISR 中断状态寄存器

偏移地址：0x2C

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.				GC	START	STOP	ACTIV	RX_DONE	TX_ABRT	RD_REQ	TX_EMPTY	TX_OVER	RX_FULL	RX_OVER	RX_UNDER
				r	r	r	r	r	r	r	r	r	r	r	r

Bit	Field	Description
31：12	Reserved	保留，必须保持复位值
11：0	ISR	每位的具体描述可以参考 I2C_RAWISR 寄存器

14.4.11 I2C_IMR 中断屏蔽寄存器

偏移地址：0x30

复位值：0x0000 08FF

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.				GC	START	STOP	ACTIV	RX_DONE	TX_ABRT	RD_REQ	TX_EMPTY	TX_OVER	RX_FULL	RX_OVER	RX_UNDER
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31：12	Reserved	保留，必须保持复位值
11：0	IMR	每一位屏蔽与 I2C_ISR 寄存器对应的中断状态位

14.4.12 I2C_RAWISR RAW 中断寄存器

偏移地址：0x34

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.				GC	START	STOP	ACTIV	RX_DONE	TX_ABRT	RD_REQ	TX_EMPTY	TX_OVER	RX_FULL	RX_OVER	RX_UNDER
				r	r	r	r	r	r	r	r	r	r	r	r

Bit	Field	Description
31：12	Reserved	保留，必须保持复位值
11	GC	广播呼叫（General call） 接收到广播呼叫地址并响应时置位。 禁止 I2C 接口或当 CPU 读 I2C_GC.GC（位 0）时清零。 I2C 将接收到的数据存储在接收缓冲中。
10	START	起始条件检测 无论 I2C 接口工作在主模式或从模式，一旦检测到 I2C 接口上发生了起始或重复起始条件即置位该位。

Bit	Field	Description
9	STOP	停止条件检测 该位状态取决于 I2C_CR.STOPINT（位 7）的状态： STOPINT = 0： 无论 I2C 接口工作在主模式或从模式，一旦检测到 I2C 接口上发生了停止条件时即置位该位。从模式下，无论从器件是否被寻址都会产生 STOP 中断。 STOPINT = 1： 主模式下（MASTER = 1），该位指示 I2C 接口是否发生停止条件； 从模式下（MASTER = 0），仅当从器件被寻址时产生 STOP 中断。
8	ACTIV	该位用于捕捉 I2C 模块的活动状态。置位后只能由以下四种方式清零： ▪ 禁止 I2C 接口 ▪ 读 I2C_ACTIV 寄存器 ▪ 读 I2C_ICR 寄存器 ▪ 系统复位 即使 I2C 处于空闲状态，该位也仍然保持为高直到被清零。
7	RX_DONE	从发送结束（Transmit done） 当 I2C 作为从发送器时，如果主器件没有响应发送的字节，则该位置位。 该情况发生在传输的最后一个字节则表示传输结束。
6	TX_ABRT	发送中止（Transmit abort） 当 I2C 接口作为发送器，不能发送完缓冲中的数据时置位。 注意：发送中止会将 I2C 接口的接收和发送缓冲清空。发送缓冲会处于刷新状态直到读 I2C_TX_ABRT 寄存器。一旦该读操作执行后，发送缓冲就可以接收 APB 接口上的新数据。
5	RD_REQ	读请求（Read request） 当 I2C 作为从器件，其他主器件试图从 I2C 接口读取数据时置位。 I2C 接口会使总线保持在等待状态（SCL = 0）直到中断被处理。这表明 I2C 接口被其他主器件寻址成功且被要求发送数据。处理器必须响应该中断然后写数据到 I2C_DR 寄存器中。该位在处理器读 I2C_RD_REQ 寄存器后清零。
4	TX_EMPTY	发送缓冲空（Transmit buffer empty） 该位状态取决于 I2C_CR.EMPINT（位 8）的状态： EMPINT = 0，发送缓冲区数据个数小于等于阈值时置位； EMPINT = 1，发送缓冲区数据个数小于等于阈值且前一个发送地址/数据的内部移位寄存器发送结束时置位。 发送缓冲区数据个数大于阈值时由硬件自动清零。
3	TX_OVER	发送缓冲过载（Transmit buffer over） 发送缓冲满时处理器写入新数据导致溢出时置位。
2	RX_FULL	接收缓冲满（Receive buffer full） 接收缓冲区数据大于阈值时置位。 接收缓冲区数据小于等于阈值时由硬件清零。
1	RX_OVER	接收缓冲过载（Receive buffer over） 接收缓冲满时收到新数据时置位。此时 I2C 接口会响应，但新的数据会丢失。

Bit	Field	Description
0	RX_UNDER	接收缓冲欠载（Receive buffer under） RX FIFO 为空时处理器读 I2C_DR 寄存器则该位置位。

14.4.13 I2C_RXTLR 接收阈值

偏移地址：0x38

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.								TL							
								rw							

Bit	Field	Description
31：8	Reserved	保留，必须保持复位值
7：0	TL	接收 FIFO 阈值（Receive FIFO threshold level） 控制 RX_FULL 中断触发。

14.4.14 I2C_TXTLR 发送阈值

偏移地址：0x3C

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.								TL							
								rw							

Bit	Field	Description
31：8	Reserved	保留，必须保持复位值
7：0	TL	发送 FIFO 阈值（Transmit FIFO threshold level） 控制 TX_EMPTY 中断触发。

14.4.15 I2C_ICR 组合和独立中断清除寄存器

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
Res.															ICR
															r

Bit	Field	Description
31：1	Reserved	保留，必须保持复位值
0	ICR	读该寄存器将会清除所有组合中断、独立中断。 该位不清除硬件可自动清除的中断，仅清除软件可清除中断。

14.4.16 I2C_RX_UNDER 清除 RX_UNDER 中断寄存器

偏移地址：0x44

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.															RX_UN
															DER
															r

Bit	Field	Description
31：1	Reserved	保留，必须保持复位值
0	RX_UNDER	读该寄存器清零 RX_UNDER 中断（I2C_RAWISR[0]）

14.4.17 I2C_RX_OVER 清除 RX_OVER 中断寄存器

偏移地址：0x48

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.															RX_OVER
															r

Bit	Field	Description
31：1	Reserved	保留，必须保持复位值
0	RX_OVER	读该寄存器清零 RX_OVER 中断（I2C_RAWISR[1]）

14.4.18 I2C_TX_OVER 清除 TX_OVER 中断寄存器

偏移地址：0x4C

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.															TX_OVER
															r

Bit	Field	Description
31：1	Reserved	保留，必须保持复位值
0	TX_OVER	读该寄存器清零 TX_OVER 中断（I2C_RAWISR[3]）

14.4.19 I2C_RD_REQ 清除 RD_REQ 中断寄存器

偏移地址：0x50

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.															RD_REQ
															r

Bit	Field	Description
31：1	Reserved	保留，必须保持复位值
0	RD_REQ	读该寄存器清零 RD_REQ 中断（I2C_RAWISR[5]）

14.4.20 I2C_TX_ABRT 清除 TX_ABRT 中断寄存器

偏移地址：0x54

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.															TX_ABRT
															r

Bit	Field	Description
31：1	Reserved	保留，必须保持复位值
0	TX_ABRT	读该寄存器清零 TX_ABRT 中断（I2C_RAWISR[6]） 同时将 TX FIFO 从刷新/复位状态中释放，以便接收写入的数据。

14.4.21 I2C_RX_DONE 清除 RX_DONE 中断寄存器

偏移地址：0x58

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.															RX_DONE
															r

Bit	Field	Description
31：1	Reserved	保留，必须保持复位值
0	RX_DONE	读该寄存器清零 RX_DONE 中断（I2C_RAWISR[7]）

14.4.22 I2C_ACTIV 清除 ACTIVITY 中断寄存器

偏移地址：0x5C

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.															ACTIV
															r

Bit	Field	Description
31：1	Reserved	保留，必须保持复位值
0	ACTIV	I2C 不活动时，读该寄存器清零 ACTIV 中断（I2C_RAWISR[8]） 如果 I2C 仍然活动，那么 ACTIV 中断将继续置位。当 I2C 模块禁止或 I2C 总线不再活动时该位由硬件清零。可以通过读该寄存器得到 I2C_RAWISR.ACTIV（位 8）的状态。

14.4.23 I2C_STOP 清除 STOP_DET 中断寄存器

偏移地址：0x60

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.															STOP
															r

Bit	Field	Description
31：1	Reserved	保留，必须保持复位值
0	STOP	读该寄存器清零 STOP 中断（I2C_RAWISR[9]）

14.4.24 I2C_START 清除 START_DET 中断寄存器

偏移地址：0x64

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.															START
															r

Bit	Field	Description
31：1	Reserved	保留，必须保持复位值
0	START	读该寄存器清零 START 中断（I2C_RAWISR[10]）

14.4.25 I2C_GC 清除 GEN_CALL 中断寄存器

偏移地址：0x68

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.															GC
															r

Bit	Field	Description
31：1	Reserved	保留，必须保持复位值
0	GC	读该寄存器清零 GC 中断（I2C_RAWISR[11]）

14.4.26 I2C_ENR 使能寄存器

偏移地址：0x6C

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.														ABORT	ENABLE
														rw	rw

Bit	Field	Description
31：2	Reserved	保留，必须保持复位值
1	ABORT	I2C 传输中止（I2C transfer abort） 0：中止没有发生或者已经结束 1：中止操作正在进行 I2C 模块作为主器件时，软件可以置位该位以中止 I2C 传输。一旦置位不能立即清除。置位后 I2C 模块控制逻辑会在完成当前传输后产生 STOP 条件并清空发送缓冲，中止操作之后产生 TX_ABRT 中断。该 ABORT 位会在中止操作结束后自动清零。
0	ENABLE	I2C 模块使能 0：禁止 I2C 模块（发送和接收缓冲保持擦除状态） 1：使能 I2C 模块

14.4.27 I2C_SR 状态寄存器

偏移地址：0x70

复位值：0x0000 0006

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.									SLV_ACTIV	MST_ACTIV	RFF	RFNE	TFE	TFNF	ACTIV
									r	r	r	r	r	r	r

Bit	Field	Description
31：7	Reserved	保留，必须保持复位值
6	SLV_ACTIV	从状态机（Slave FSM）活动状态 0：从状态机处于 IDLE 状态，I2C 从器件部分不活动 1：从状态机不处于 IDLE 状态，I2C 从器件部分活动
5	MST_ACTIV	主状态机（Master FSM）活动状态 0：主状态机处于 IDLE 状态，I2C 主器件部分不活动 1：主状态机不处于 IDLE 状态，I2C 主器件部分活动
4	RFF	接收缓冲满 0：接收缓冲未满 1：接收缓冲满
3	RFNE	接收缓冲非空 0：接收缓冲空 1：接收缓冲非空

Bit	Field	Description
2	TFE	发送缓冲空 0：发送缓冲非空 1：发送缓冲空
1	TFNF	发送缓冲未满 0：发送缓冲满 1：发送缓冲未满
0	ACTIV	I2C 活动状态 MST_ACTIV 位与 SLV_ACTIV 位相或的结果。

14.4.28 I2C_TXFLR 发送缓冲级别寄存器

偏移地址：0x74

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.														CNT	
														r	

Bit	Field	Description
31：2	Reserved	保留，必须保持复位值
1：0	CNT	发送缓冲中有效数据的个数（0~2）

14.4.29 I2C_RXFLR 接收缓冲级别寄存器

偏移地址：0x78

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.														CNT	
														r	

Bit	Field	Description
31：2	Reserved	保留，必须保持复位值

Bit	Field	Description
1：0	CNT	接收缓冲中有效数据的个数（0~2）

14.4.30 I2C_HOLD SDA 保持时间寄存器

偏移地址：0x7C

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								RX_HOLD							
								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HOLD															
rw															

Bit	Field	Description
31：24	Reserved	保留，必须保持复位值
23：16	RX_HOLD	I2C 作为接收器时，配置 SDA 内部保持时间（SCL 为高电平时有效），单位为 APB 时钟周期。
15：0	TX_HOLD	I2C 作为发送器时，配置 SDA 保持时间（SCL 从高电平转换为低电平后），单位为 APB 时钟周期。

14.4.31 I2C_SETUP SDA 建立时间寄存器

偏移地址：0x94

复位值：0x0000 0064

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.								CNT							
								rw							

Bit	Field	Description
31：8	Reserved	保留，必须保持复位值
7：0	CNT	SDA 建立时间，最小值为 2。 如果所需延迟时间为 1000ns，当 APB 时钟频率为 10MHz 时，建议将该寄存器设为 11。

14.4.32 I2C_GCR 广播呼叫 ACK 寄存器

偏移地址：0x98

复位值：0x0000 0001

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.															GC
															rw

Bit	Field	Description
31：1	Reserved	保留，必须保持复位值
0	GC	广播呼叫 ACK 0：接收到广播呼叫后不响应（NACK），也不产生中断 1：接收到广播呼叫后响应（ACK）

14.4.33 I2C_SLVMASK 从地址掩码寄存器

偏移地址：0xB0

复位值：0x0000 03FF

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.						MASK									
						rw									

Bit	Field	Description
31：10	Reserved	保留，必须保持复位值
9：0	MASK	从地址掩码 0：I2C_SAR 寄存器的相应位被掩码忽略，不需要比较 1：I2C_SAR 寄存器的相应位需要比较

14.4.34 I2C_SLVRCVADDR 从器件接收地址寄存器

偏移地址：0xB4

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.						ADDR									
						r									

Bit	Field	Description
31：10	Reserved	保留，必须保持复位值
9：0	ADDR	从器件实际接收到的地址

15 USART 通用同步异步收发器

15.1 介绍

本通用同步/异步收发器（USART）可以灵活地与外部设备进行全双工数据交换。通过内置波特率（包含整数及小数设定）发生器，USART 可以支持宽范围的波特率。

USART 支持异步模式（UART）、同步模式，其中 UART 支持单线半双工通信。

15.2 USART 特性

- 支持全双工异步通信，全双工时钟同步通信
- 波特率发生器（包含整数及分数配置）
  - 可编程波特率，供发送器和接收器使用（最小分频系数为 1）
- 独立的发送和接收缓冲寄存器，且发送器和接收器可单独使能。
- 支持 LSB、MSB 收发模式
- 数据位长度可编程（ 8 或 9 位）
- 停止位可配置（1/2 位）
- 校验位功能可配置（奇校验、偶校验，无校验）
- 支持空闲帧的产生（ TE 使能时自动输出）和接收检测
- 支持信号收发引脚的互换，接收和发送信号的电平取反
- 支持下面中断源：
  - 发送端数据寄存器为空（TXE）
  - 发送完成（TC）
  - 接收端数据有效（RXNE）
  - 接收缓冲器溢出（OVR）
  - 接收空闲帧完成（IDLE）
  - 奇偶校验错误（PE）
  - 噪声标志（NF）和帧错误（FE）

15.3 USART 功能描述

15.3.1 功能框图

USART 的功能框图可参考如下，可分为寄存器相关的控制单元、收发数据控制器、时钟控制器以及引脚控制逻辑单元。

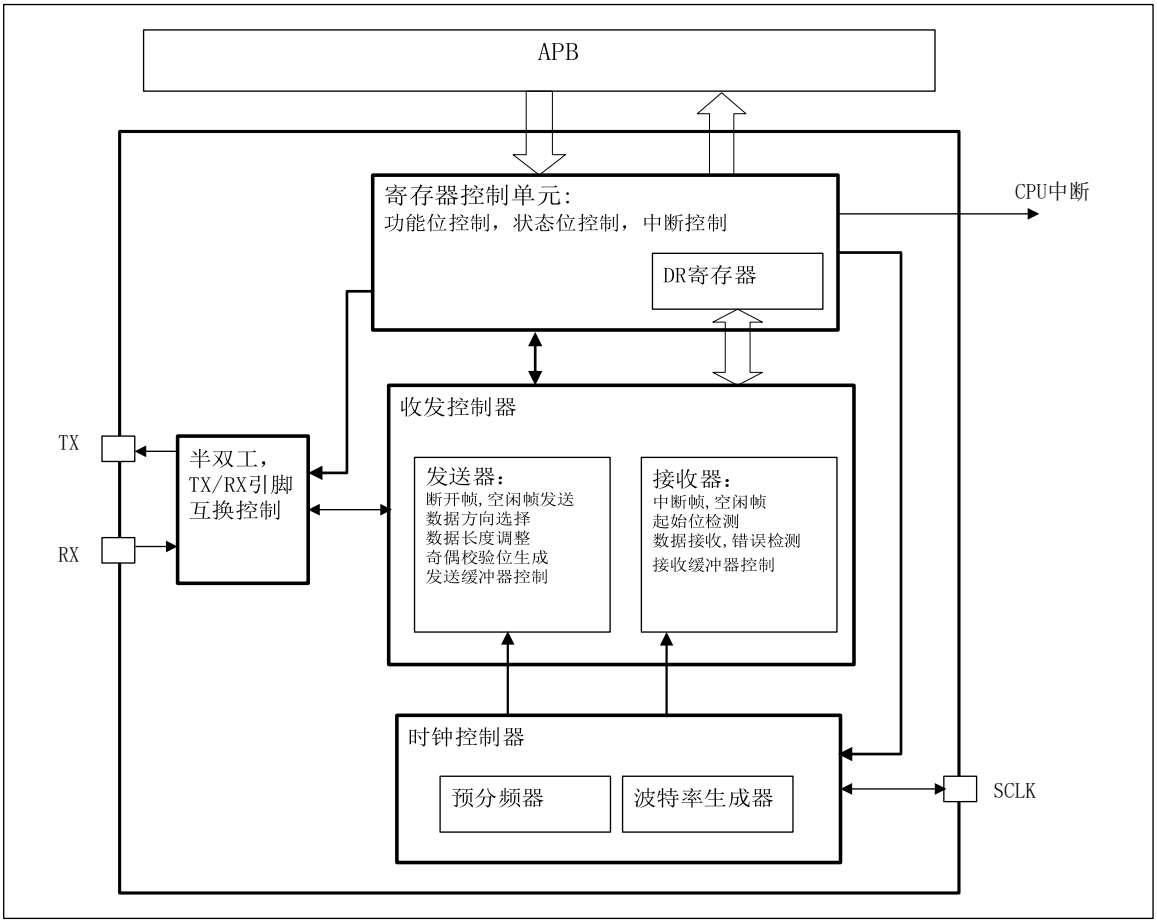

图 15-1 UART 功能框图

15.3.2 信号描述

信号名	类型	描述
USART_SCLK	输出或输入	同步模式下的输入或输出时钟引脚、
USART_TX	输出或输入	发送数据引脚，或半双工收发数据引脚
USART_RX	输入	接收数据引脚（全双工时）

15.3.3 功能概述

在全双工通信的情况下，至少需要分配两个脚给 USART：接收数据输入（RX）和发送数据输出（TX）。

RX：外部串行数据通过该引脚，传送给 USART 接收器。对于传输过程中产生的噪音，可以使用过采样的技术将其与数据区分并剔除，得到原本的数据。

TX：USART 发送器内部产生的串行数据通过该引脚发送输出。当发送器被使能，并且无数据发送时，TX 引脚输出高电平。

空闲状态为总线在开始发送或者开始接收前的初始状态。

起始位为一位，用‘0’表示。

在 USART 通信中，一个数据（8 或 9 位）的发送和接收顺序可配置为从最低位到最高位（LSB），或最高位到最低位（MSB）。

停止位用‘1’表示一帧的结束，位数可配置为 1、2 位。

通过编程波特率寄存器 USART_BRR 来控制内部的波特率发生器以得到期望的数据传输波特率（参考 1.3.5 章节的波特率计算公式）。

通过配置 USART_CR2 寄存器的 SWAP 位，可以交换接收和发送端的引脚。

通过配置 USART_CR2 寄存器的 RXTOG/TXTOG 位，可以将接收/发送端的电平信号取反（包含起始位、停止位）。

USART 模块还支持同步模式（不同于 UART），此时需要下列引脚：

- SCLK 发送器时钟输出或时钟输入：该引脚用于同步模式，同步模式下，支持时钟的输入和输出功能，且时钟极性和相位可通过软件配置。

15.3.4 特性描述

通过配置 USART_CR1.DL 位，可调整字符位长度为 8 或 9 位。发送器会在发送起始位时拉低 TX 引脚，在发送停止位时拉高 TX 引脚。

包括停止位在内，一个完全由‘1’组成的完整数据帧，定义为一个空闲帧。下一个数据帧的起始位跟在空闲帧之后。

包括停止位在内，一个完全由‘0’组成的完整数据帧，定义为一个断开帧。在断开帧结束后，发送器会再发送一个‘1’的停止位，使得下一帧的起始位能够被识别到（产生下降沿被接收端检测到）。

波特率发生器产生的时钟经过发送器或接收器的使能位置位控制之后，供给发送器或接收器使用。

下图为数据帧格式，断开帧，以及空闲帧的样例。

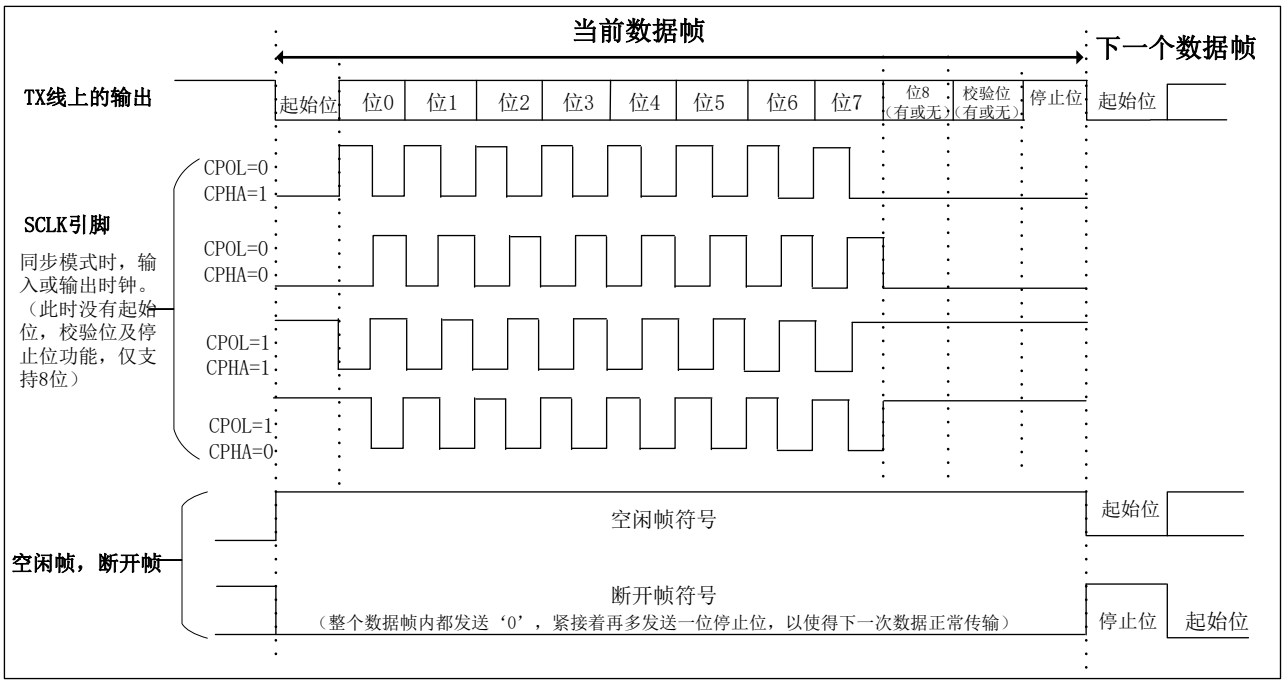

图 15-2 UART 数据帧类型示意图

15.3.5 波特率发生器

各通信模式下，可按照以下公式配置波特率。

- 异步（UART）模式：

通信波特率为：

$$f_{baudrate} = \frac{\mathrm{PCLK}}{N\times\ (\mathrm{MFD} + \mathrm{FFD}/N)}$$

误差 E（%）为：

$$E(\%) = \{\frac{\mathrm{PCLK}}{f_{baudrate} \times N\times\ (\mathrm{MFD} + \mathrm{FFD/N})} - 1\}\times 100$$

上述公式中，PCLK 为内部时钟源的频率；MFD、FFD 为 波特率寄存器 USART_BRR 中的整数和小数分频；N=8× (2－OVER8)，其中 OVER8 为控制寄存器 USART_CR1 中的过采样模式选择。当 OVER8=1（8 倍过采样），FFD[3:0] 只使用低 3 位，用户应配置 FFD[3] 位为 1’h0。

- 同步模式（小数分频配置无效，软件配置 DIV_fraction[3:0] 位为 4’h0）：

通信波特率为：

$$f_{baudrate} = \frac{\mathrm{PCLK}}{4\times\ \mathrm{MFD}}$$

上述公式中，PCLK 为内部时钟源的频率；MFD 为 波特率寄存器 USART_BRR 中整数分频。同步模式下小数分频（FFD）无效，用户应配置 FFD[3:0] 位为 4’h0。

15.3.6 采样

UART 内置检测电路检测一帧数据的开始，并对 RX 引脚进行采样，UART 采用 8 或 16 倍数据波特率的时钟采样 RX 引脚的数据。

可配置 USART_CR1.OVER8 位，来选择 USART 采用 16 倍或 8 倍数据波特率的时钟进行 RX 引脚的数据采样。

当选择 8 倍过采样 （OVER8=1） 时，可获得更高的速度（高达 $f_{PCLK}/8$），但接收器对时钟偏差的最大容差将会降低。

15.3.7 奇偶校验控制

配置 USART_CR1 寄存器中的 PCE 位为‘1’来使能奇偶校验控制，PS 位用来选择奇校验或偶校验。

偶校验：校验位加上数据中‘1’的总数为偶数。

奇校验：校验位加上数据中‘1’的总数为奇数。

奇偶校验控制有效时：

- 发送器会自动生成一个奇偶校验位，并在停止位前输出。
- 接收器会对奇偶校验位进行检测并判断该位是否错误。如果奇偶校验位错误则硬件自动置位 USART_SR.PE 标志，但当前接收数据仍然会从移位寄存器传送到 USART_DR 寄存器。

15.3.8 发送器

配置 USART_CR1.TE 位为‘1’，使能发送器，数据会串行输出到 TX 引脚上。发送数据寄存器 USART_DR 和内部的发送移位寄存器组成双缓冲器结构，可以连续发送数据。其中 UART 模式可以通过设置 USART_CR1.DL 位来选择长度（8 或 9 位）。

15.3.8.1 字符发送

在 USART 发送期间，数据从 USART_DR 寄存器写入，通过发送移位寄存器将数据帧字节以最低位到最高位（USART_CR1.MLS=0），或最高位到最低位（USART_CR1.MLS=1）的顺序在 TX 引脚上输出。

发送数据顺序：1 位起始位，字符，1 位奇偶校验位（有或无），停止位。

可通过寄存器 USART_CR2.STOP[1:0] 来配置停止位的个数（可配置为 1 或 2 位停止位）。

当前数据传输未完成前不能清零 USART_CR1.TE 位，否则波特率发生器会停止产生时钟，导致该数据后部分丢失。

15.3.8.2 发送断开帧

配置 USART_CR1.SBK 位为‘1’ 即可发送断开帧。如果在数据传输过程中配置 SBK=1，则当前数据发送完成后才会输出断开帧到 TX 引脚上。

断开帧发送完成时，硬件自动清零 SBK 位，并发送一位高电平的停止位（以保证相连的下一帧数据的起始位被检测到）。

断开帧的长度取决于数据帧长度（CR2.DL），奇偶校验使能位（CR1.PS），以及停止位（CR2.STOP）。例如，无奇偶校验且停止位为 1 位时，CR2.DL=0 则断开帧为 10 位连续的‘0’，CR2.DL=1 则断开帧为 11 位连续的‘0’。

15.3.8.3 发送配置步骤

可参考下面的步骤，来配置 USART 进行数据帧的发送：

1. 配置 USART 所需要使用的引脚功能。
2. 使能 USART（USART_CR1.UE=1）。
3. 配置 USART_BRR 寄存器。
4. 根据传输数据帧等需求配置 USART_CR1，USART_CR2，USART_CR3 寄存器。
5. 使能发送器（USART_CR1.TE=1），如果需要使用发送数据寄存器空中断，则设置 USART_CR1.TXEIEN=1。
6. 等待发送数据寄存器空，写通信数据到 USART_DR，数据传输到发送移位寄存器，发送开始。
7. 如果需要连续发送数据时，重复步骤 6。
8. 通过确认 USART_SR.TC 位确认发送是否完成。如果配置 TCIEN=1，则最后一帧数据发送结束后，产生发送完成中断。

注：USART 的发送器支持两种中断，即发送数据寄存器空中断 TXE 和发送完成中断 TC，可通过 USART_SR 寄存器中的状态位查询。配置 TXEIEN=1，当 USART_DR 寄存器的值传送到发送移位寄存器时产生 TXE 中断。配置 TCIEN=1，当发送数据的最后一位时 USART_DR.DR 寄存器没有写入更新数据则产生 TC 中断。

15.3.9 接收器

数据寄存器 USART_DR 寄存器和内部的接收移位寄存器组成双缓冲器结构，可以连续接收数据。

其中 UART 模式可以通过设置 USART_CR1.DL 位来选择数据长度，8 位或 9 位。

接收器使能位 USART_CR.RE 置’1’并检测到开始位后，RX 管脚上数据接收到接收移位寄存器；当接收到一帧数据，数据从接收移位寄存器传送到数据寄存器 USART_DR 中，同时，状态标志 RXNE 将置’1’。配置 RXNEIEN=1 则许可该中断请求。

CPU 利用该请求读取接收数据时，一次请求只能读取一次数据。

接收数据的顺序为：开始位->数据位（MSB/LSB）->校验位（有或者无）->停止位。

15.3.9.1 接收断开帧

USART 接收器识别到一个断开帧时，会置位 USART_SR.FE 标志（等同于在停止位接收到‘0’）。

15.3.9.2 接收空闲帧

UART 正常工作时，接收器接收到了一个空闲帧会置位 USART_SR.IDLE 标志。

配置 IDLEIEN=1 来许可 IDLE 中断请求。

15.3.9.3 接收配置步骤

可参考下面的步骤，来配置 USART 进行数据帧的接收：

1. 设定 UART 所需要使用的功能引脚。
2. 使能 USART（USART_CR1.UE=1）。
3. 配置 USART_BRR 寄存器。
4. 根据数据帧等需求配置来设定 USART_CR1，USART_CR2，USART_CR3 寄存器。
5. USART_BRR 寄存器设定通信波特率（时钟源为外部时钟源时不需要设定）。
6. 使能接收器（USART_CR1.RE=1），如果需要使用接收中断，则设置 USART_CR1. RXNEIEN=1。
7. 当检测到开始位后，接收器将数据接收到接收移位寄存器，并检查校验位和停止位。一共有三种错误标志状态：PE，FE，ORE。当没有错误发生时，接收到的数据从接收移位寄存器传送到 USART_DR 寄存器，并置 RXNE 标志位为‘1’。
8. 可通过 RXNE 中断读取接收到的数据，重复步骤 7 即可连续接收数据。
9. 在接收期间如果检测到接收错误时，对应的错误标志将被置位。

注：为了防止溢出错误，在下一字符接收结束前，RXNE 位必须被清零（软件读取数据寄存器 USART_DR）。当发生 PE，FE，ORE 中任何一种接收错误都不能再进行数据的接收，但可以通过将所有的错误标志清零来重启数据接收。

- 发生上溢错误时接收到的数据丢失，ORE 状态位置‘1’，但 RXNE 中断不发生。
- 发生奇偶校验错误时接收到的数据传送给 USART_DR，PE 状态位置‘1’， 但 RXNE 中断不发生。
- 发生帧错误时接收到的数据传送给 USART_DR，FE 状态位置‘1’， 但 RXNE 中断不发生。

15.3.10 同步模式

通过配置 USART_CR1.SAS 位为‘1’来使能同步模式（时钟引脚功能将同时有效）。

在同步模式下， USART_CR3.HDSEL 位应配置为‘0’。

同步模式支持主模式和从模式：主模式时使用内部波特率生成器生成的时钟，同时输出时钟；从模式时由 SCLK 引脚输入时钟。USART 在同步模式下，能与 SPI 实现数据通信（此时，用户应配置 SPI 与 USART 的时钟极性、时钟相位为一致）。

15.3.10.1 时钟描述

配置 USART_CR2.CLKEN 位为‘1’来使能时钟引脚功能，同时根据 USART_CR3.CKINE 位配置来选择使用内部波特率时钟或从 SCLK 引脚输入时钟，以进行数据通信。

当选择内部波特率时钟时，可通过 SCLK 引脚输出同步时钟。

1 帧数据的收发包含 8 个时钟脉冲。

当 RE 和 TE 都为‘0’，时钟输出会停止，并固定在 USART_CR2.CPOL 配置的电平。

通过配置 USART_CR2.CPOL 位选择时钟极性；通过配置 USART_CR2.CPHA 位选择外部时钟相位。

15.3.10.2 时钟同步功能描述

SCLK 引脚作为发送器的时钟输出时，仅在数据段输出时钟，一帧数据输出 8 个时钟脉冲，最后一位发送完后，通信线保持最后一位的值，时钟输出固定在高电平或低电平（由 CPOL 位决定）。

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 SCLK 变化边沿上采样（上升或下降沿，取决于 CPOL 和 CPHA 位配置情况），而不会进行任何过采样。此时必须确保足够的建立时间和保持时间，以符合时序要求（类同于 SPI 协议）。

内部时钟源时，内部波特率生成器生成的波特率计算公式为：

$$f_{baudrate} = \frac{\mathrm{PCLK}}{4 \times \mathrm{MFD}}$$

其中通信波特率的单为 MBps；PCLK 为内部时钟源的频率；MFD 为 波特率寄存器 USART_BRR 中整数分频（注意，在同步模式下应配置 MFD ≥ 2，且小数分频 FFD 无效，用户应配置 FFD[3:0] 位为 4’h0）。

使用内部时钟源且 MFD=2 时，同步模式的最高波特率为 PCLK/8（MBps）。

外部时钟源时，要求外部输入时钟的最大频率为 PCLK/8（MHz），此时最高波特率也为 PCLK/8（MBps）。

15.3.11 单线半双工通信

配置 USART_CR3.HDSEL 位为‘1’，进入单线半双工模式。

单线半双工模式下的芯片内部逻辑会将 TX 与 RX 互连，同时：

- RX 引脚悬空，不参与传输。传输时 USART 的 TX 直接连接另一个 USART 的 TX。
- 在传输数据时，TX 一直被占用，直到停止位发送完成。
- 在没有传输数据时，TX 处于被释放状态。因此，它在空闲状态的或接收状态时表现为一个标准 I/O 口；TX 对应 I/O 在不被 USART 驱动时，必须配置成悬空输入（或开漏的输出高）。

除了单线引脚的配置外，其余配置和正常传输时一致。

在没有通信前，两个 USART 的 RXEN 都开启，处于等待接收状态，当需要通信时，两个 USART 要约定好谁来发送，发送方的 USART_CR1 寄存器中 RE 关闭， TE 使能。如果两边 UART 都试图发送数据，将产生发送冲突（硬件不会阻碍 USART 的发送：当发送使能位 TE 开启，只要写 USART_DR，TX 就会发送数据）。

15.3.12 中断

本 USART 模块可以支持下面的中断源：

表 15-1 UART 中断请求

中断事件	中断状态位	使能位	UART	同步模式
发送数据寄存器为空	TXE	TXEIEN	√	√
发送完成	TC	TCIEN	√	√
接收数据寄存器满	RXNE	RXNEIEN	√	√
检测到空闲线路	IDLE	IDLEIEN	√	-
奇偶校验错误	PE	PEIEN	√	-
噪声标志	NF	ERRIEN	√	-
上溢错误	ORE	ERRIEN	√	√
帧错误	FE	ERRIEN	√	-

注：“√”表示使用该中断。“-”表示不使用该中断。

15.4 寄存器

15.4.1 寄存器总览

表 15-2 USART 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	USART_SR	状态寄存器	0x0000_00C0
0x04	USART_DR	数据收发寄存器	0x0000_01FF

Offset	Acronym	Register Name	Reset
0x08	USART_BRR	波特率控制寄存器	0x0000_0000
0x0C	USART_CR1	控制寄存器 1	0x0000_0000
0x10	USART_CR2	控制寄存器 2	0x0000_0000
0x14	USART_CR3	控制寄存器 3	0x0000_6000

15.4.2 USART_SR 状态寄存器

偏移地址：0x00

复位值：0x0000 00C0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
								r	r	rc_w0	r	r	r	r	r

Bit	Field	Description
31:8	Reserved	保留，必须保持复位值
7	TXE	发送数据寄存器为空标志 （Transmit data register Empty） 0：发送数据寄存器非空（数据未传送到移位寄存器） 1：发送数据寄存器为空（数据已送到移位寄存器） TXE 位由硬件自动置‘1’及清‘0’，数据未传送到移位寄存器时硬件自动清零该位（写入 DR 寄存器时）；当 TE=0 或数据由 DR 传送至移位寄存器时硬件自动置位该位。
6	TC	发送完成标志（Transmission complete） 0：发送未完成 1：发送完成 TC 清零的条件： 在 TE=1 时向数据寄存器写入发送数据。 TC 的置位条件： TE=0 或数据帧最后一位送出时没有写 USART_DR 来更新发送数据寄存器。
5	RXNE *注 1	接收数据寄存器不为空标志（Read data register not empty） 0：未接收到有效数据 1：接收到有效数据 注：RXNE 位由硬件置‘1’及清‘0’，用户也可以通过向该位写入‘0’来清零。接收到有效数据时硬件自动将 RXNE 置‘1’，读取接收数据后硬件自动将 RXNE 清零。
4	IDLE *注 2	空闲帧检测标志 （IDLE frame detected） 0：未检测到空闲帧 1：检测到空闲帧 UART 检测到空闲帧时，该位由硬件自动置‘1’。

Bit	Field	Description
3	ORE *注 2	接收溢出错误标志 （Overrun error） 0：无接收溢出错误 1：检测到接收溢出错误 注：在 RXNE=1（已有可读数据），又接收到一帧新的数据，该位将由硬件自动置‘1’。
2	NF *注 2	噪声检测标志 （Noise detected flag） 0：未检测到噪声 1：检测到噪声 注：当在接收的信号线上检测到噪声时，该位由硬件自动置‘1’。
1	FE *注 2	接收帧错误标志（Framing error） 0：无接收帧错误 1：发生接收帧错误 该位由硬件自动置‘1’，置位条件： 异步模式（UART）时，接收数据帧的停止位为低电平。 注：FE=1 时，接收的数据会从移位寄存器传送到数据寄存器，但不产生 RXNE 中断请求，同时将停止后续的数据接收动作。
0	PE *注 2	奇偶校验错误标志 （Parity error） 0：无奇偶校验错误 1：奇偶校验错误 在接收数据时检测到了奇偶校验错误，该位由硬件自动置‘1’。 注：PE=1 时，接收的数据会从移位寄存器传送到数据寄存器，但不产生 RXNE 中断请求，同时将停止后续的数据接收动作。

*注 1：通过软件写‘0’可将该位清零。

*注 2：通过软件序列可将该位清零（读取状态寄存器，然后对 USART_DR 数据寄存器执行读访问）。

15.4.3 USART_DR 数据寄存器

偏移地址：0x04

复位值：0x0000 01FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							DR[8:0]								
							rw								

Bit	Field	Description
31:9	Reserved	保留，必须保持复位值

Bit	Field	Description
8:0	DR[8:0]	发送/接收数据寄存器 包含接收到的数据字符，或要发送的数据字符，取决于所执行的操作是“读取”操作还是“写入”操作：读取时，表示接收到的数据；写入时，表示准备要发送的数据。 注：最高位 DR[8] 只在异步模式（UART）且数据长度为 9 位（DL=1）时有效。

15.4.4 USART_BRR 波特率寄存器

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

Bit	Field	Description
31:16	Reserved	保留，必须保持复位值
19:4	MFD [15:0]	波特率整数分频（Mantissa frequency division of baudrate） 这 16 位用于定义 USART 波特率的整数分频。 使能发送或接收（TE 或 RE 设定为 1）之前，用户应根据波特率需求配置该整数分频值。 注：需配置 MFD≥1。
3:0	FFD[3:0]	波特率小数分频（Fraction frequency division of baudrate） 这 4 位用于定义 USART 波特率的小数分频。 使能发送或接收（TE 或 RE 配置为‘1’）之前，用户应根据波特率需求配置该小数分频值。 注：配置 FFD[3:0]=4’h0 则小数分频无效。异步模式（UART）且 USART_CR1.OVER8=1 时，最高位 FFD[3] 无效，用户应配置 FFD[3]=0。在同步模式时，小数分频无效，用户应配置 FFD[3:0]=4’h0。

15.4.5 USART_CR1 控制寄存器 1

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
Reserved														SAS	MLS

														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	Res.	UE	DL	Res.	PCE	PS	PEIEN	TXEIEN	TCIEN	RXNEIEN	IDLEIEN	TE	RE	Res.	SBK
rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

Bit	Field	Description
31:18	Reserved	保留，必须保持复位值
17	SAS	通信模式选择位（synchronous/asynchronous mode selection） 0：异步模式（UART）（asynchronous） 1：同步模式（synchronous） 注：用户应在 TE=0 和 RE=0 时配置此位。
16	MLS	MSB/LSB 方式选择位 0：LSB 方式 1：MSB 方式 注：用户应在 TE=0 和 RE=0 时配置此位。
15	OVER8	UART 过采样模式 （Oversampling mode） 0：16 倍过采样 1：8 倍过采样 注：用户应在 TE=0 和 RE=0 时配置此位。
14	Reserved	保留，必须保持复位值
13	UE	USART 使能 （USART enable） 0：禁止 USART 预分频器和引脚输出 1：使能 USART 该位清零后，USART 预分频器和输出将停止，并会结束当前传输以降低功耗。 注：该位由用户置‘1’和清零。
12	DL	数据长度 （Data length） 0：8 位 1：9 位 注：用户应在 TE=0 和 RE=0 时配置此位。
11	Reserved	保留，必须保持复位值
10	PCE	奇偶校验控制使能 （Parity control enable） 0：禁止奇偶校验功能 1：使能奇偶校验功能 注：该位由用户置‘1’和清零。同步模式时应配置 PCE=0（复位值）。
9	PS	校验模式 0：偶校验 1：奇校验 注：该位由用户置‘1’和 清零，只在 PCE=1 时有效。

Bit	Field	Description
8	PEIEN	PE 中断使能 （PE interrupt enable） 0：禁止 PE 中断请求 1：允许 PE 中断请求 注：该位由用户置‘1’和清零。
7	TXEIEN	TXE 中断使能 （TXE interrupt enable） 0：禁止 TXE 中断请求 1：允许 TXE 中断请求 注：该位由用户置‘1’和清零。
6	TCIEN	传送完成中断使能 （Transmission complete Interrupt enable） 0：禁止 TC 中断请求 1：允许 TC 中断请求 注：该位由用户置‘1’和清零。
5	RXNEIEN	RXNE 中断使能 （RXNE interrupt enable） 0：禁止 RXNE 中断请求 1：允许 RXNE 中断请求 注：该位由用户置‘1’和清零。
4	IDLEIEN	IDLE 中断使能 （IDLE interrupt enable） 0：禁止 IDLE 中断请求 1：允许 IDLE 中断请求 注：该位由用户置‘1’和清零。
3	TE	发送器使能 （Transmitter enable） 0：禁止发送器 1：使能发送器 注：该位由用户置‘1’和清零。在同步模式下，如需同时收发，用户必须同时配置 TE 和 RE 位，以保证时钟与数据收发的时序正常。
2	RE	接收器使能 （Receiver enable） 0：禁止接收器 1：使能接收器 注：该位由用户置‘1’和清零。在同步模式下，如需同时收发，用户必须同时配置 RE 和 TE 位，以保证时钟与数据收发的时序正常。
1	Reserved	保留，必须保持复位值
0	SBK	发送断路 （Send break） 0：不发送断开帧 1：发送断开帧 该位用于发送断开帧，可由用户置‘1’，并在发送完断开帧后由硬件自动清零。

15.4.6 USART_CR2 控制寄存器 2

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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	Res.	STOP[1:0]		Res.	CPOL	CPHA	Res.								
rw		rw			rw	rw									

Bit	Field	Description
31:15	Reserved	保留，必须保持复位值
15	SWAP	输入与输出引脚交换 0：IO 引脚功能不交换 1：IO 引脚功能的输入与输出交换 注：SWAP 置位后，GPIOx_CRL 寄存器的 MODE 需要更改，如：原输入模式变为输出模式。
14	Reserved	保留，必须保持复位值
13:12	STOP[1:0]	停止位 （STOP bit） UART 模式： 00：1 个停止位 10：2 个停止位 01：保留 11：保留
11	Reserved	保留，必须保持复位值
10	CPOL	时钟极性 （Clock POLarity） 0：空闲时，时钟为低电平。 1：空闲时，时钟为高电平。 注：该位与 CPHA 位结合使用可获得所需的时钟/数据关系（仅在同步时钟模式下有效）。
9	CPHA	时钟相位（Clock PHAse） 0：在时钟第一个变化沿捕获数据 1：在时钟第二个变化沿捕获数据 注：该位与 CPOL 位结合使用可获得所需的时钟/数据关系（仅在同步时钟模式下有效）。
8:0	Reserved	保留，必须保持复位值

15.4.7 USART_CR3 控制寄存器 3

偏移地址：0x14

复位值：0x0000 6000

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

Res.		TXTOG	RXTOG	Reserved											CKINE
		rw	rw												rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				ONEBIT	Res.							HDSEL	Res.		ERRIEN
				rw								rw			rw

Bit	Field	Description
31:30	Reserved	保留，必须保持复位值
29	TXTOG	发送取反位 0：发送取反功能无效 1：发送信号电平取反
28	RXTOG	接收取反 0：接收取反功能无效 1：接收信号电平取反
27:17	Reserved	保留，必须保持复位值
16	CKINE	同步模式下时钟输入许可 此位用于控制时钟是否由外部输入（CR1.SAS 位为‘1’时该位配置有效）： 0：时钟不由外部输入 1：时钟由外部输入 软件应在 TE=0 和 RE=0 时配置此位。
15:12	Reserved	保留，必须保持复位值
11	ONEBIT	UART 单次采样方式使能（One sample bit method enable） 0：三次采样（多数判决） 1：单次采样 注：用户应在 TE=0 和 RE=0 时配置此位。当选择一个采样位方法后，噪声检测标志（USART_SR.NF）失效。
10:4	Reserved	保留，必须保持复位值
3	HDSEL	单线半双工选择 （Half-duplex selection） 0：全双工模式 1：半双工模式
2:1	Reserved	保留，必须保持复位值
0	ERRIEN	错误中断使能 （Error interrupt enable） 0：禁止错误中断请求 1：允许错误中断请求 错误中断包括 FE，ORE，NF 三种。 注：当对 USART 进行 DR 的读写时，如果配置了 ERRIEN=1，则许可向 CPU 发出通信异常的中断请求。

16 SYSCFG 系统控制器

16.1 简介

该芯片具有一组系统配置寄存器。这些寄存器的主要功能如下：

- 管理连接到 GPIO 口的外部中断（引脚配置）
- 重映射存储器到代码起始区域
- 部分外设的系统级配置

16.2 寄存器

16.2.1 寄存器总览

表 16-1 SYSCFG 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	SYSCFG_CFGR	SYSCFG 配置寄存器	0x00000000
0x08	SYSCFG_EXTICR1	SYSCFG 外部中断配置寄存器 1	0x00000000
0x0C	SYSCFG_EXTICR2	SYSCFG 外部中断配置寄存器 2	0x00000000
0x10	SYSCFG_EXTICR3	SYSCFG 外部中断配置寄存器 3	0x00000000
0x14	SYSCFG_EXTICR4	SYSCFG 外部中断配置寄存器 4	0x00000000
0x18	SYSCFG_PADHYS	SYSCFG PAD 配置寄存器	0x00000000

16.2.2 SYSCFG_CFGR 配置寄存器

具有两个可配置内存起始 0x0000 0000 地址的控制位。

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.														MEM_MODE	
														rw	

Bit	Field	Description
31：2	Reserved	保留，始终读为 0

Bit	Field	Description
1：0	MEM_MODE	存储映射选择位（Memory selection bit） 由软件设置和清除这些位。它控制存储器内部映射到地址 0x0000 0000。 x0：主闪存存储器映射到 0x0000 0000 01：系统闪存映射到 0x0000 0000 11：嵌入式 RAM 映射到 0x0000 0000

16.2.3 SYSCFG_EXTICR1 外部中断配置寄存器 1

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				EXTI2				EXTI1				EXTI0			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	保留，必须保持复位值
15:0	EXTIx	EXTIx 配置（x=0…3）（EXTIx configuration） 选择 EXTIx 外部中断的输入源。 0000： PA[x]管脚 0001： PB[x] 管脚

16.2.4 SYSCFG_EXTICR2 外部中断配置寄存器 2

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				EXTI6				EXTI5				EXTI4			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	保留，必须保持复位值

Bit	Field	Description
15:0	EXTIx	EXTIx 配置（x=4…7）（EXTIx configuration） 选择 EXTIx 外部中断的输入源。 0000： PA[x]管脚 0001： PB[x]管脚

16.2.5 SYSCFG_EXTICR3 外部中断配置寄存器 3

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
EXTI11				EXTI10				EXTI9				EXTI8			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	保留，必须保持复位值
15:0	EXTIx	EXTIx 配置（x=8…11）（EXTIx configuration） 选择 EXTIx 外部中断的输入源。 0000： PA[x] 管脚 0001： PB[x] 管脚

16.2.6 SYSCFG_EXTICR4 外部中断配置寄存器 4

偏移地址：0x014

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
EXTI15				EXTI14				EXTI13				EXTI12			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	保留，必须保持复位值
15:0	EXTIx	EXTIx 配置（x=12…15）（EXTIx configuration） 选择 EXTIx 外部中断的输入源。 0000： PA[x] 管脚 0001： PB[x] 管脚

16.2.7 PAD 配置寄存器（SYSCFG_PADHYS）

偏移地址：0x018

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															I2C1_mode_sel
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															

Bit	Field	Description
31:17	Reserved	保留，必须保持复位值
16	I2C1_MODE_SEL	I2C1 端口模式选择位 0：开漏模式 1：推挽模式 注： I2C 作为从机时不建议使用推挽模式； I2C 作为主机使用推挽模式时，进行通信的从机无法拉低 SCL
15:0	Reserved	保留，必须保持复位值

17 Device Electronic Signature 器件电子签名

17.1 简介

器件电子签名存放在闪存存储器模块的系统存储区域，可以通过 SWD 或者 CPU 读取。它所包含的芯片识别信息在出厂时编写，用户固件或者外部设备可以读取电子签名，用以自动匹配不同配置的微控制器。

产品唯一的身份标识非常合适：

- 用来作为序列号
- 用来作为密码，在编写闪存时，将此唯一标识与软件加解密算法结合使用，提高代码在闪存存储器的安全性
- 用来激活带安全机制的自举过程
- 96 位的产品唯一身份标识所提供的参考号码对任意一个系列微控制器，在任何情况下都是唯一的。用户在何种情况下，都不能修改这个身份标识。这个 96 位的产品唯一身份标识，按照用户不用的用法，可以以字节 (8 位) 为单位读取，也可以以半字 (16 位) 或者全字 (32 位) 读取。

17.2 寄存器描述

基地址：0x1FFF F7E8

表 17-1 存储器容量寄存器概览

Offset	Acronym	Register Name	Reset
0x00	UID1	唯一标识码 1	0xXXXXXXXX
0x04	UID2	唯一标识码 2	0xXXXXXXXX
0x08	UID3	唯一标识码 3	0xXXXXXXXX

17.2.1 UID1 唯一标识码

偏移地址：0x00

复位值：其值在出厂时编写

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID （31：16）															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID （15：0）															

r															

Bit	Field	Description
31：0	U_ID（31：0）	U_ID：唯一身份标志 31：0 位（31：0 unique ID bits）

17.2.2 UID2 唯一标识码

偏移地址：0x04

复位值：其值在出厂时编写

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID （63：48）															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID （47：32）															
r															

Bit	Field	Description
31：0	U_ID （63：32）	U_ID：唯一身份标志 63： 32 位 （63：32 unique ID bits）

17.2.3 UID3 唯一标识码

偏移地址：0x08

复位值：其值在出厂时编写

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID （95：80）															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID （79：64）															
r															

Bit	Field	Description
31：0	U_ID（95：64）	U_ID：唯一身份标志 95 ：64 位 （95：64 unique ID bits）

18 DBG 调试支持

18.1 介绍

芯片内核包含硬件调试模块，主要用于功能的调试。当内核在取指（指令断点）或访问数据（数据断点）时，硬件调试模块可以控制内核停止，此时用户可以查询内核的内部状态和系统的外部状态。查询完成后，内核可以继续执行当前程序。

当芯片与调试器连接开始调试时，调试器自动调用内核的调试模块进行调试操作。

18.2 功能描述

18.2.1 功能框图

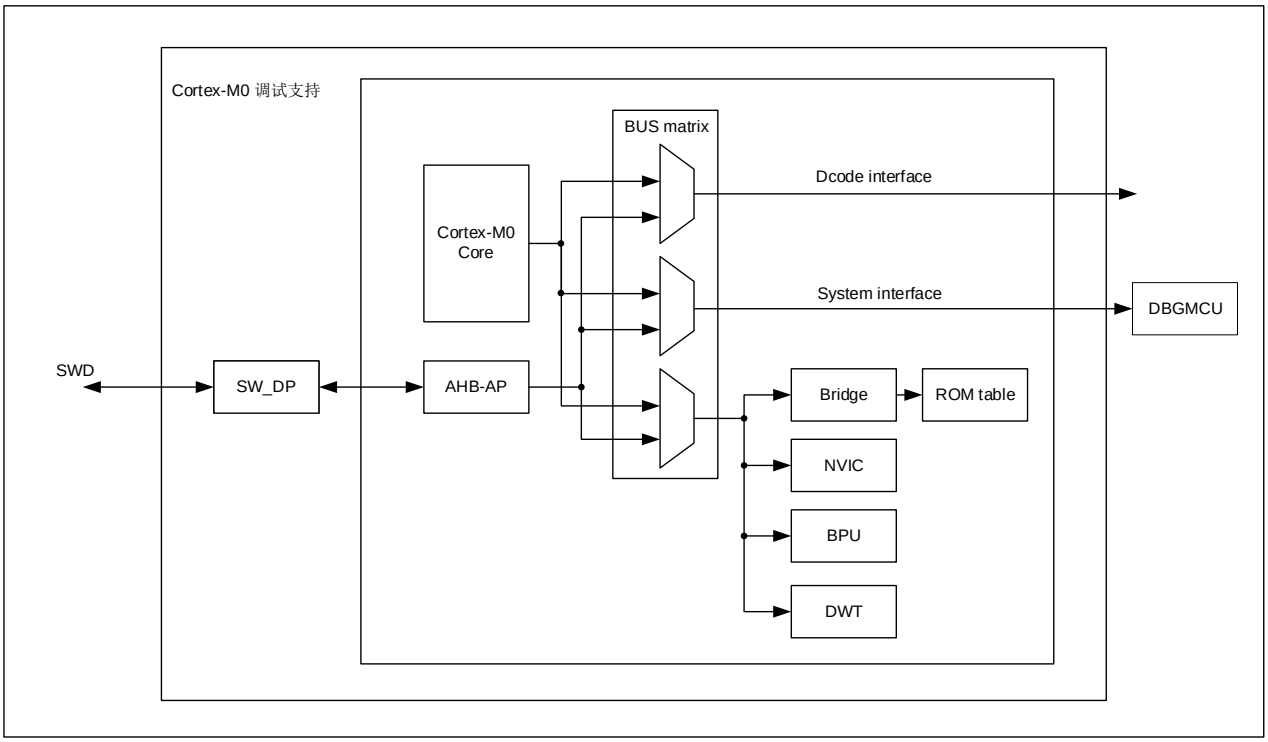

图 18-1 调试功能框图

Cortex-M0 内核含有调试单元，该单元由以下部分组成：

SWDP: SW 调试端口

BPU: 断点调试单元

DWT: 数据观察点和跟踪

18.2.2 SWD 内部上拉与下拉

SWD 引脚输入直接控制调试模式，不能悬空。为了保证 I/O 电平可控， SWD 引脚上内嵌了上拉和下拉电阻。

- SWDIO：内部上拉
- SWCLK：内部下拉

软件可以把这些 I/O 口作为普通的 I/O 口使用，此时默认的上下拉功能关闭，参考通用端口 GPIO 章节。

18.2.3 SWD 调试端口脚

该芯片的 2 个普通 I/O 口可用作 SWD-DP 接口引脚， 不同封装都支持 SWD 调试端口。

表 18-1 SWD 调试端口管脚

SWD-DP 端口引脚名称	SW 调试接口		引脚分配
	类型	调试功能	
SWDIO	输入/输出	串行数据输入/输出	PA13
SWCLK	输入	串行时钟	PA14

18.3 ID 代码

在芯片内部有多个 ID 编码，如下表：

表 18-2 ID 编码

ID 名	芯片
DEV_ID	0x4C505F00
CPU TAP SW ID	0x0BB11477

18.3.1 微控制器设备 ID 编码

微控制器内部包含设备 ID 编码，此 ID 定义了微控制器的硅片版本，并且映射到外部 APB 总线上。通过用户代码与调试接口均能够获取此 ID 编码。

18.3.2 Cortex JEDEC-106 ID 编码

微控制器有一个 JEDEC-106ID 编码。它位于映射到内部 PPB 总线地址为 0xE00FF000_0xE00FFFFF 的 4KB ROM 表中。

18.4 SW 调试端口

18.4.1 SW 协议介绍

此同步串行协议使用 2 个引脚：主机到目标的时钟信号（SWCLK）与双向数据信号（SWDIO）。

作为双向数据线，SWDIO 引脚内嵌了上拉电阻，无需额外的外接电阻。

数据从低位开始传输，允许读写寄存器组 DPACC 与 APACC。

根据协议，当 SWDIO 改变方向时，同时需要插入一个转换时间（默认一个 Bit 时间，具体可以通过 SWCLK 调整），这段间内，任何设备不能驱动此信号线。

18.4.2 SW 协议序列

一次序列包含三个阶段：

- 主机发送 8 位请求包；
- 目标发送 3 位确认应答；
- 根据配置方向，主机或目标发送 33 位（包含一位校验位）数据；

表 18-3 8bit 请求包

比特位	名称	描述
0	起始	必须为 1
1	APnDP	0：访问 DP 1：访问 AP
2	RnW	0：写请求 1：读请求
4: 3	A[3 : 2]	DP 或 AP 寄存器的地址
5	Parity	前面比特位的校验位
6	Stop	0
7	Park	不能由主机驱动，由于有上拉，目标永远读为 1

注：每一个请求包紧跟一个 Bit 转换时间。更多关于 DAPCC 与 APACC 寄存器的信息，查看 ARM 相关的 CPU 技术参考手册。

表 18-4 3bit 应答包

比特位	名称	描述
2: 0	ACK	001：失败 010：等待 100：成功

注：当应答信号（ACK）处于上表的情况之一时，应答位后有一个转换时间。

表 18-5 33bit 数据包

比特位	名称	描述
31: 0	WDATA/RDATA	写或读的数据
32	Parity	32 位数据的奇偶校验位

注：读数据位结束后等待一个转换时间。

18.4.3 SW-DP 状态机 (Reset, Idle states, ID code)

SW-DP 状态机通过内部的 ID 码识别 SW_DP，遵守 JEP-106 标准,具体信息请参考 ARM 相关手册。直到调试器读取 ID 之前，SW-DP 的状态机不会工作。

- 当出现了上电复位，或 DP 从 JTAG 切换到 SWD 后，或超过 50 个周期的高电平时，SW-DP 状态机将处于复位状态；
- 如果 RESET 状态之后出现了至少 2 个周期的低电平，状态机会切换到 IDLE 状态；
- 状态机开始处于复位态，工作时必须先切换到 IDLE 态，先执行读 DP-SW ID 寄存器的操作。否则，调试器无法进行其它正常的传输，会出现 ACK Fault；

18.4.4 DP 和 AP 读 / 写访问

- 对 DP 的读操作没有延时：调试器将直接获得数据 （如果 ACK 返回成功状态），或者处于等待状态（如果 ACK 返回等待状态）；
- 对 AP 的读操作具有延时。这意味着前一次读操作的结果只能在下一次操作时获得。如果下一次的操作不是对 AP 的访问，则必须读 DP-RDBUFF 寄存器来获得上一次读操作的结果；
- DP-CTRL/STAT 寄存器的 READOK 标志位会在每次 AP 读操作和 RDBUFF 读操作后更新，以通知调试器 AP 的读操作是否成功；
- SW-DP 具有写缓冲区（DP 和 AP 都有写缓冲），这使得其它传输进行时，仍然可以接受写操作。如果写缓冲区满，调试器将获得一个等待的 ACK 响应。读 IDCODE 寄存器，读 CTRL/STAT 寄存器和写 ABORT 寄存器操作在写缓冲区满时仍被接受；
- 由于 SWCLK 和 HCLK 的异步性，需要在写操作后（在奇偶校验位后）插入 2 个额外的 SWCLK 周期，以确保内部写操作正确完成。这两个额外的时钟周期需要在 IDLE 状态下插入。这个操作步骤在写 CTRL/STAT 寄存器以提出一个上电请求时尤其重要，否则下一个操作（在内核上电后才有效的操作）会立即执行，这将会导致操作失败；

18.4.5 SW-DP 寄存器

当 APnDP=0 时，可以访问以下这些寄存器。

表 18-6 SW-DP 寄存器

A[3:2]	读 / 写	SELECT 寄存器的 CTRLSEL 位	寄存器	描述
00	读		IDCODE	固定为 0x0BB11477（用于识别 SW-DP）。
00	写		ABORT	
01	读/写	0	DP-CTRL /STAT	请求一个系统或调试的上电操作； 配置 AP 访问的操作式； 控制比较，校验操作； 读取一些状态位（溢出，上电响应）。
01	读/写	1	WIRE CONTROL	配置串行通信物理层协议（如转换时间长度等）。
10	读		READ RESEND	允许从一个错误的调试传输中恢复数据而不用重复最初的 AP 传输。
10	写		SELECT	选择当前的访问端口和有效的 4 字长寄存器窗口。
11	读/写		READ BUFFER	这个寄存器会从 AP 捕获上一次读操作的数据结果，因此可以获得数据而不必再启动一个新的 AP 传输。

18.4.6 SW-AP 寄存器

当 APnDP=1 时，可以访问 AP 寄存器的访问地址由以下两部分组成：

- A[3：2]的值
- DP SELECT 寄存器的当前值

18.5 MCU 调试模块 (DBGMCU)

MCU 调试模块提供以下调试器协助功能：

- 支持低功耗模式
- 断点时定时器与看门狗的时钟控制

18.5.1 低功耗模式的调试支持

MCU 具有多种低功耗模式，能够关闭 CPU 时钟，降低 CPU 的功耗，通过执行 WFE 或 WFI 指令进入低功耗模式。 CPU Free-Run 时钟 FCLK 与 AHB 总线时钟 HCLK 对于调试操作时必须的，不能关闭，同时 MCU 可以通过配置一些寄存器来改变低功耗模式特性，从而支持在低功耗模式下调试代码，具体的配置如下。

- 当进入睡眠模式时，为了能够提供 HCLK 同 FCLK 相同的时钟，调试器必须先置位 DBG_CR 寄存器的 DBG_SLEEP 位。
- 当进入停机模式时，必须先配置 DBG_STOP 位，该操作会激活内部振荡器，从而为 FCLK 与 HCLK 提供时钟。

18.5.2 支持定时器、看门狗

当产生断点时，根据定时器和看门狗的应用不同来选择计数器的工作模式；

- 计数器可以选择继续计数，通常应用在输出 PWM 波控制电机
- 计数器可以选择停止计数，通常应用在看门狗计数

18.6 寄存器

18.6.1 寄存器总览

表 18-7 DBG 寄存器概览

Offset	Acronym	Register Name	Reset
0x00	DBG_IDCODE	DBG ID 编码寄存器	0x4C505F00
0x04	DBG_CR	DBG 控制寄存器	0x00000000

18.6.2 DBG_IDCODE ID 编码寄存器

地址：0x40013400（只支持 32 位访问，只读）

复位值：0x4C505F00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DEV_ID															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEV_ID															
r															

Bit	Field	Description
31:0	DEV_ID	设备识别编码 (Device Identifier) 只读寄存器，始终读为复位值

18.6.3 DBG_CR 控制寄存器

地址：0x40013404 （只支持 32 位访问）

复位值：0x0000 0000（POR 复位，不被系统复位所复位)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.													DBG_TIM14_STOP rw	Res.	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBG_TIM3_PWM_OFF rw	Res.	DBG_TIM1_PWM_OFF rw	DBG_TIM3_STOP rw	Res.	DBG_TIM1_STOP rw	Res.	DBG_IWDG_STOP rw	Res.				DBG_STOP_FOR_LDO rw	Res.	DBG_STOP rw	DBG_SLEEP rw

Bit	Field	Description
31:19	Reserved	保留，必须保持复位值
18	DBG_TIM14_STOP	当内核进入调试状态时计数器 14 停止工作（TIM14 Counter Stopped When Core is Halted） 0：选中定时器的计数器仍然正常工作 1：选中定时器的计数器停止工作
17:16	Reserved	保留，必须保持复位值
15	DBG_ TIM3_PWM_OFF	TIM3 在调试模式下 PWM 输出全部为 0 0: TIM3 保持正常输出 1: TIM3 在 debug halt 时输出全部为 0
14	Reserved	保留，必须保持复位值
13	DBG_TIM1_ PWM_OFF	TIM1 在调试模式下 PWM 输出全部为 0 0: TIM1 保持正常输出 1: TIM1 在 debug halt 时输出全部为 0
12	DBG_TIM3_STOP	当内核进入调试状态时计数器 3 停止工作（TIM3 Counter Stopped When Core is Halted） 0：选中定时器的计数器仍然正常工作 1：选中定时器的计数器停止工作
11	Reserved	保留，必须保持复位值
10	DBG_TIM1_STOP	当内核进入调试状态时计数器 1 停止工作（TIM1 Counter Stopped When Core is Halted） 0：选中定时器的计数器仍然正常工作 1：选中定时器的计数器停止工作
9	Reserved	保留，必须保持复位值
8	DBG_IWDG_STOP	当内核进入调试状态时看门狗停止工作（Debug independent watchdog stopped when core is halted） 0：看门狗计数器仍然正常工作 1：看门狗计数器停止工作

Bit	Field	Description
7:4	Reserved	保留，必须保持复位值
3	DBG_STOP_FOR_LDO	调试停机模式 (Debug Stop mode) STOP LDO
2	Reserved	保留，必须保持复位值
1	DBG_STOP	调试停机模式 (Debug Stop mode) 0:在停机模式时，时钟控制器禁止一切时钟（包括 HCLK 和 FCLK）。当从 STOP 模式退出时，时钟配置与复位之后的配置一致。如果需要重新使能 PLL, HSE 等时钟，必须通过软件重新配置 1：在停机模式时，FCLK 与 HCLK 开启，由 HSI 提供时钟。当从 STOP 模式退出时，如果需要重新使能 PLL, HSE 等时钟，必须通过软件重新配置
0	DBG_SLEEP	调试睡眠模式（Debug Sleep mode） 0：在睡眠模式时，时钟 FCLK 开启，FCLK 保持默认配置的系统时钟，HCLK 则关闭。睡眠模式不会复位配置好的时钟系统，因此退出睡眠模式时，软件不需重新配置系统时钟 1：在睡眠模式时，FCLK 和 HCLK 时钟开启都保持由原先配置好的系统时钟

19 修订记录

表 19-1 修订记录表

修订时间	修订版本	修订内容
2023/08/18	V1.0	PWR 章节：电压调节器描述修改；低功耗模式描述修改；停机模式描述修改；深度停机模式描述修改； RCC 章节：APB1 外设复位寄存器描述修改；APB1 外设时钟使能寄存器描述修改； GPIO 章节：OSC_IN/OSC_OUT 描述删除；GPIO 寄存器描述修改；端口复用高位寄存器修改 TIM1 章节：修改高级控制定时器框图；修改 9.3.13 章节名，描述修改；删除原 9.3.14 章节；比较模式寄存器 1 描述修改；比较模式寄存器 2 描述修改；比较使能寄存器描述修改；比较模式寄存器 3 描述修改；比较寄存器 5 描述修改 SYSFIG 章节：PAD 配置寄存器描述修改 USART 章节：USART 简介修改 DEVICE 章节：Cortex JEDEC-106 ID 代码描述修改
2023/03/06	V0.1	初版发布