life.augmented

RM0490
参考手册

STM32C0x1 基于 Arm® 内核的 32 位高级 MCU

简介

本参考手册是对 STM32C0x1 微控制器数据手册的补充，提供了应用（特别是软件开发）所需的信息，属于 STM32C0x1 微控制器上提供的功能集的超集。

有关特定 STM32C0x1 器件的功能集、订购信息以及机械和电气特性的信息，请参见其相应的数据手册。

有关 Arm® Cortex®-M0+ 内核的信息，请参见 Cortex®-M0+ 技术参考手册。

相关文档

- 《Cortex®-M0+ 技术参考手册》，可从 *http://infocenter.arm.com* 获取
- PM0223 Cortex®-M0+ 内核编程手册(a)
- STM32C0x1 数据手册(a)
- AN2606 STM32 MCU 自举应用笔记(a)

a. 可从意法半导体网站 *www.st.com* 获取

目录

表格索引

图片索引

1 文档约定

1.1 概述

STM32C0x1 器件搭载 Arm®[a] Cortex®-M0+ 内核。

1.2 寄存器相关缩写词列表

寄存器说明中使用以下缩写词[b]：

读/写 (rw)	软件可以读写该位。
只读 (r)	软件只能读取该位。
只写 (w)	软件只能写入该位。读取该位时将返回复位值。
读取/写入 0 清零 (rc_w0)	软件可以通过读取该位或通过写入 0 将该位清零。写入 1 对该位的值无影响。
读取/写入 1 清零 (rc_w1)	软件可以通过读取该位或通过写入 1 将该位清零。写入 0 对该位的值无影响。
读取/写入清零 (rc_w)	软件可以通过读取该位或通过写入寄存器将该位清零。写入该位的值并不重要。
读取/读取清零 (rc_r)	软件可以读取该位。读取该位时，将自动清零。写入该位对其值无影响。
读取/读取置 1 (rs_r)	软件可以读取该位。读取该位时，将自动置 1。写入该位对其值无影响。
读取/置位 (rs)	软件可以读取该位，也可将其置 1。写入 0 对该位的值无影响。
读/仅可写入一次 (rwo)	软件仅可写入一次该位，但可随时读取该位。只能通过复位将该位返回到复位值。
切换 (t)	软件可以通过写入 1 来切换该位。写入 0 无影响。
只读，写触发 (rt_w1)	软件可以读取该位。写入 1 时，将触发事件，但不会影响该位的值。
保留 (Res.)	保留位，必须保持复位值。

a. Arm 是 Arm Limited（或其子公司）在美国和/或其他地区的注册商标。

b. 这是一个涵盖适用于意法半导体微控制器的所有缩写词的详尽列表，其中一些缩写词在当前文档中可能并未使用。

1.3 词汇表

本节简要介绍本文档中所用首字母缩略词和缩写词的定义：

- **字**：32 位数据。
- **半字**：16 位数据。
- **字节**：8 位数据。
- **SWD-DP（SWD 调试端口）**：SWD-DP 提供基于串行线调试 (SWD) 协议的 2 引脚（时钟和数据）接口。请参见 Cortex®-M0+ 技术参考手册。
- **IAP（在应用中编程）**：IAP 是指可以在用户程序运行期间对微控制器的 Flash 进行重新编程。
- **ICP（在线编程）**：ICP 是指可以在器件安装于用户应用电路板上时使用 SWD 协议或自举程序对微控制器的 Flash 进行编程。
- **选项字节**：存储于 Flash 中的产品配置位。
- **OBL**：选项字节加载器。
- **AHB**：高级高性能总线。
- **APB**：高级外设总线。

1.4 外设可用性

有关各型号产品所支持的外设类型及数量，请参见相关器件数据手册。

2 存储器和总线架构

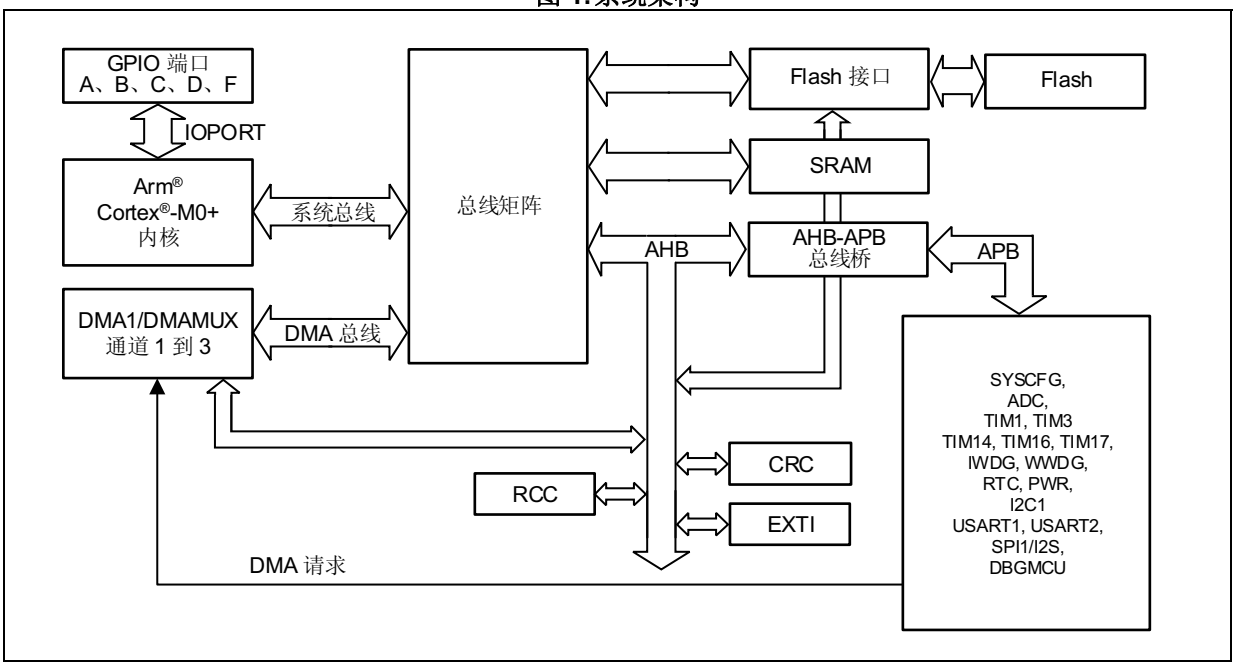
2.1 系统架构

主系统包括：

- 两个主器件：
  - Cortex®-M0+ 内核
  - 通用 DMA
- 三个从器件：
  - 内部 SRAM
  - 内部 Flash
  - 带 AHB-APB 桥的 AHB，用于连接所有 APB 外设

这些外设使用多层 AHB 总线架构互连，如*图 1* 所示。

图 1. 系统架构



系统总线（S 总线）

此总线将 Cortex®-M0+ 内核的系统总线（外设总线）连接到总线矩阵，而总线矩阵管理着内核和 DMA 之间的仲裁。

DMA 总线

此总线用于将 DMA 的 AHB 主接口连接到总线矩阵，而总线矩阵管理 CPU 和 DMA 对 SRAM、Flash 以及 AHB/APB 外设的访问。

总线矩阵

总线矩阵对内核系统总线和 DMA 主控总线之间的访问进行仲裁。仲裁采用循环调度算法。总线矩阵由主控总线（CPU 和 DMA）和被控总线（Flash、SRAM 和 AHB-APB 桥）组成。

AHB 外设通过总线矩阵连接到系统总线，以实现 DMA 访问。

AHB-APB 总线桥 (APB)

AHB-APB 桥可在 AHB 与 APB 总线之间实现完全同步的连接。

有关连接到此总线桥的外设的地址映射，请参见*第 2.2 节：存储器结构*。

每次芯片复位后，所有外设时钟都被关闭（SRAM 和 Flash 除外）。使用外设之前，必须先通过 RCC_AHBENR、RCC_APBENRx 或 RCC_IOPENR 寄存器使能其时钟。

注意： *对 APB 寄存器执行 16 位或 8 位访问时，该访问将转换为 32 位访问：总线桥将 16 位或 8 位数据复制后提供给 32 位向量。*

2.2 存储器结构

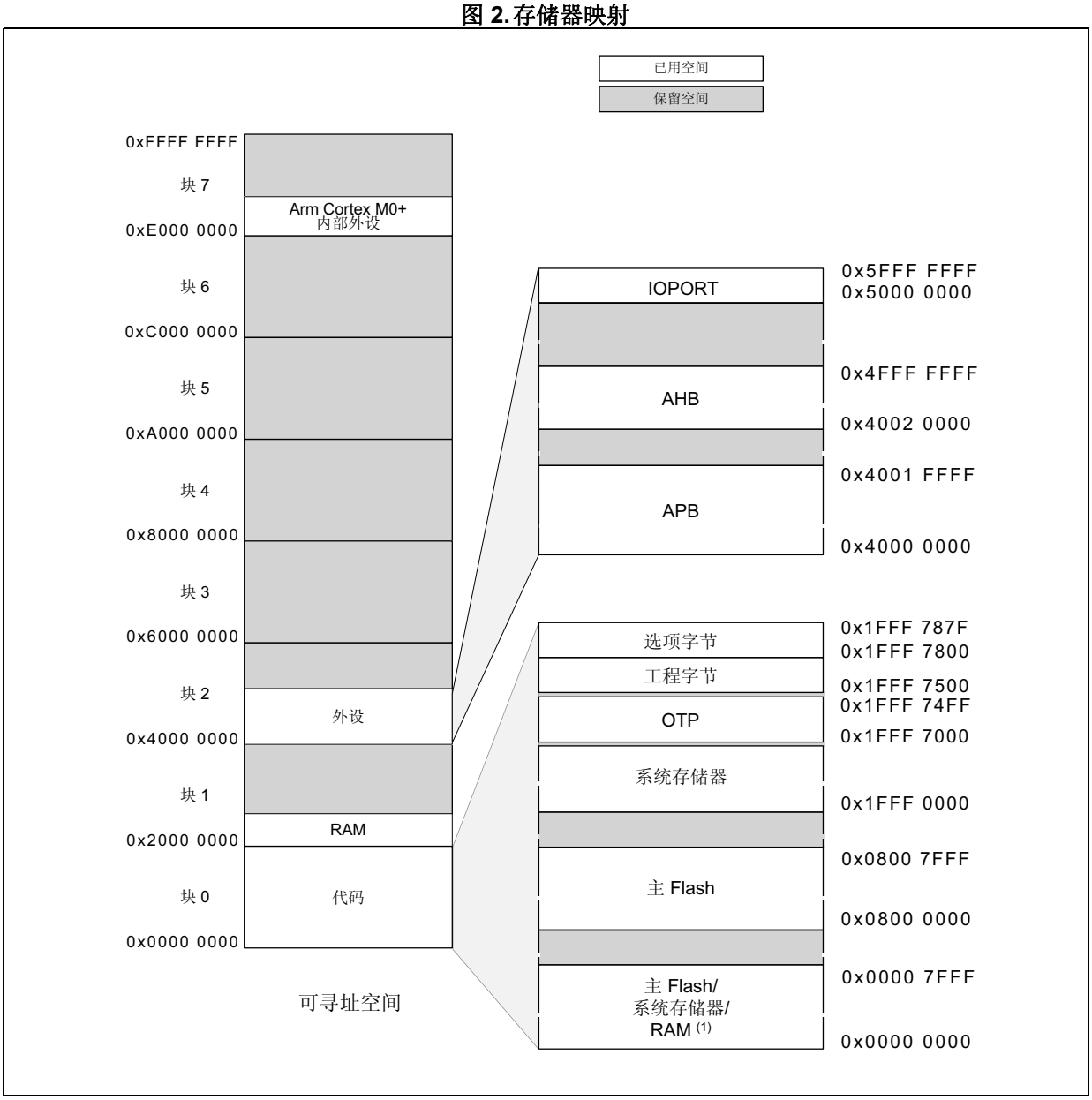
2.2.1 简介

程序存储器、数据存储器、寄存器和 I/O 端口排列在同一个线性（即地址连续）的 4 GB 地址空间内。

各字节按小端格式在存储器中编码。字中编号最低的字节被视为该字的最低有效字节，而编号最高的字节被视为最高有效字节。

2.2.2 存储器映射和寄存器边界地址

图 2.存储器映射



1. 取决于自举配置

未分配给片上存储器和外设的所有存储器映射区域均视为保留区。有关可用存储器和寄存器区域的详细映射，请参见以下各表。

表 1. STM32C0x1 边界地址

类型	边界地址	大小	存储区	寄存器说明
SRAM	0x2000 3000 - 0x3FFF FFFF	~512 MB	保留	-
	0x2000 0000 - 0x2000 2FFF	12 KB	SRAM	-

表 1. STM32C0x1 边界地址（续）

类型	边界地址	大小	存储区	寄存器说明
代码	0x1FFF 7880 - 0x1FFF FFFF	~34 KB	保留	-
	0x1FFF 7800 - 0x1FFF 787F	128 B	选项字节	*第 51 页的第 3.4 节*
	0x1FFF 7500 - 0x1FFF 77FF	768 B	工程字节	-
	0x1FFF 7400- 0x1FFF 74FF	256 B	保留	-
	0x1FFF 7000 - 0x1FFF 73FF	1 KB	OTP	-
	0x1FFF 1800 - 0x1FFF 6FFF	~22 KB	保留	-
	0x1FFF 0000 - 0x1FFF 17FF	6 KB	系统存储器	-
	0x0800 8000 - 0x1FFF D7FF	~383 MB	保留	-
	0x0800 0000 - 0x0800 7FFF	32 KB	主 Flash	*第 44 页的第 3.3.1 节*
	0x0000 8000 - 0x07FF FFFF	~127 MB	保留	-
	0x0000 0000 - 0x000 7FFF	32 KB	主 Flash、系统存储器或 SRAM（取决于 BOOT 配置）	-

表 2. STM32C0x1 外设寄存器边界地址

总线	边界地址	大小	外设	外设寄存器映射
-	0xE000 0000 - 0xE00F FFFF	1MB	Cortex®-M0+ 内部外设	-
IOPORT	0x5000 1800 - 0x5FFF 17FF	~256 MB	保留	-
	0x5000 1400 - 0x5000 17FF	1 KB	GPIOF	*第 149 页的第 6.4.12 节*
	0x5000 1000 - 0x5000 13FF	1 KB	保留	-
	0x5000 0C00 - 0x5000 0FFF	1 KB	GPIOD	*第 149 页的第 6.4.12 节*
	0x5000 0800 - 0x5000 0BFF	1 KB	GPIOC	*第 149 页的第 6.4.12 节*
	0x5000 0400 - 0x5000 07FF	1 KB	GPIOB	*第 149 页的第 6.4.12 节*
	0x5000 0000 - 0x5000 03FF	1 KB	GPIOA	*第 149 页的第 6.4.12 节*

表 2. STM32C0x1 外设寄存器边界地址（续）

总线	边界地址	大小	外设	外设寄存器映射
AHB	0x4002 3400 - 0x4FFF FFFF	~256 MB	保留	-
	0x4002 3000 - 0x4002 33FF	1 KB	CRC	*第 217 页的第 13.4.6 节*
	0x4002 2400 - 0x4002 2FFF	3 KB	保留	-
	0x4002 2000 - 0x4002 23FF	1 KB	FLASH	*第 62 页的第 3.7.4 节*
	0x4002 1C00 - 0x4002 1FFF	3 KB	保留	-
	0x4002 1800 - 0x4002 1BFF	1 KB	EXTI	*第 211 页的第 12.5.9 节*
	0x4002 1400 - 0x4002 17FF	1 KB	保留	-
	0x4002 1000 - 0x4002 13FF	1 KB	RCC	*第 132 页的第 5.4.22 节*
	0x4002 0C00 - 0x4002 0FFF	1 KB	保留	-
	0x4002 0800 - 0x4002 0BFF	1 KB	DMAMUX	*第 196 页的第 10.6.7 节*
	0x4002 0400 - 0x4002 07FF	1 KB	保留	-
	0x4002 0000 - 0x4002 03FF	1 KB	DMA1	*第 183 页的第 9.6.7 节*
APB	0x4001 5C00 - 0x4001 FFFF	32 KB	保留	-
	0x4001 5800 - 0x4001 5BFF	1 KB	DBG	*第 746 页的第 26.10.5 节*
	0x4001 4C00 - 0x4001 57FF	3 KB	保留	-
	0x4001 4800 - 0x4001 4BFF	1 KB	TIM17	*第 497 页的第 18.4.21 节*
	0x4001 4400 - 0x4001 47FF	1 KB	TIM16	*第 497 页的第 18.4.21 节*
	0x4001 4000 - 0x4001 43FF	1 KB	保留	-
	0x4001 3C00 - 0x4001 3FFF	1 KB	保留	-
	0x4001 3800 - 0x4001 3BFF	1 KB	USART1	*第 650 页的第 24.8 节*
	0x4001 3400 - 0x4001 37FF	1 KB	保留	-
APB	0x4001 3000 - 0x4001 33FF	1 KB	SPI1/I2S1	*第 734 页的第 25.9.10 节*
	0x4001 2C00 - 0x4001 2FFF	1 KB	TIM1	*第 360 页的第 15.4.29 节*
	0x4001 2800 - 0x4001 2BFF	1 KB	保留	-
	0x4001 2400 - 0x4001 27FF	1 KB	ADC	*第 270 页的第 14.13 节*
	0x4001 0400 - 0x4001 23FF	8 KB	保留	-
	0x4001 0080 - 0x4001 03FF	1 KB	SYSCFG(ITLINE)(1)	*第 162 页的第 7.1.25 节*
	0x4001 001D - 0x4001 007F		保留	-
	0x4001 0000 - 0x4001 001C		SYSCFG	*第 162 页的第 7.1.25 节*
	0x4000 B400 - 0x4000 FFFF	19 KB	保留	-
	0x4000 B000 - 0x4000 B3FF	1 KB	保留	-
	0x4000 8C00 - 0x4000 AFFF	9 KB	保留	-
	0x4000 8800 - 0x4000 8BFF	1 KB	保留	-
	0x4000 7400 - 0x4000 87FF	5 KB	保留	-
	0x4000 7000 - 0x4000 73FF	1 KB	PWR	*第 93 页的第 4.4.18 节*

表 2. STM32C0x1 外设寄存器边界地址（续）

总线	边界地址	大小	外设	外设寄存器映射
APB	0x4000 6000 - 0x4000 6FFF	4 KB	保留	-
	0x4000 5C00 - 0x4000 5FFF	1 KB	保留	-
	0x4000 5800 - 0x4000 5BFF	1 KB	保留	-
	0x4000 5400 - 0x4000 57FF	1 KB	I2C1	*第 593 页的第 23.7.2 节*
	0x4000 5000 - 0x4000 53FF	1 KB	保留	-
	0x4000 4C00 - 0x4000 4FFF	1 KB	保留	-
	0x4000 4800 - 0x4000 4BFF	1 KB	保留	-
	0x4000 4400 - 0x4000 47FF	1 KB	USART2	*第 681 页的第 24.8.15 节*
	0x4000 4000 - 0x4000 43FF	1 KB	保留	-
	0x4000 3C00 - 0x4000 3FFF	1 KB	保留	-
	0x4000 3800 - 0x4000 3BFF	1 KB	保留	-
	0x4000 3400 - 0x4000 37FF	1 KB	保留	-
	0x4000 3000 - 0x4000 33FF	1 KB	IWDG	*第 507 页的第 20.4.6 节*
	0x4000 2C00 - 0x4000 2FFF	1 KB	WWDG	*第 513 页的第 21.5.4 节*
	0x4000 2800 - 0x4000 2BFF	1 KB	RTC	*第 541 页的第 22.6.18 节*
	0x4000 2400 - 0x4000 27FF	1 KB	保留	-
	0x4000 2000 - 0x4000 23FF	1 KB	TIM14	*第 450 页的第 17.4.13 节*
	0x4000 1800 - 0x4000 1FFF	2 KB	保留	-
	0x4000 1400 - 0x4000 17FF	1 KB	保留	-
	0x4000 1000 - 0x4000 13FF	1 KB	保留	-
	0x4000 0C00 - 0x4000 0FFF	1 KB	保留	-
	0x4000 0800 - 0x4000 0BFF	1 KB	保留	-
	0x4000 0400 - 0x4000 07FF	1 KB	TIM3	*第 427 页的第 16.4.24 节*
	0x4000 0000 - 0x4000 03FF	1 KB	保留	-

1. SYSCFG (ITLINE) 寄存器使用 0x4001 0000 作为参考外设基址。

2.3 嵌入式 SRAM

下表总结了器件上的 SRAM 资源（无论使能还是禁止奇偶校验）。

表 3. SRAM 大小

器件	使能和禁止奇偶校验时的 SRAM (KB)
STM32C011xx	6
STM32C031xx	12

SRAM 可按字节、半字（16 位）或全字（32 位）以最大系统时钟频率访问，没有等待周期，因此 CPU 和 DMA 均可访问。

奇偶校验

用户可以使用用户选项字节中的选项位 RAM_PARITY_CHECK 使能奇偶校验（请参见 *第 3.4 节：FLASH 选项字节*）。

由于 B 类或 SIL 标准等要求提高存储器稳健性，因此数据总线宽度为 36 位，其中有 4 位用于奇偶校验（每字节 1 位）。

在写入 SRAM 时，将计算并保存奇偶校验位。随后读取时会自动对其进行校验。如果某一位失败，则将生成 NMI。

注意： *当使能 SRAM 奇偶校验时，建议在代码开始处使用软件初始化整个 SRAM，以免在读取非初始化位置时出现奇偶校验错误。*

2.4 Flash 概述

Flash 由两个不同的物理区域组成：

- 主 Flash 块。它包含应用程序和用户数据（如有必要）。
- 信息块。它包含三部分：
  - 用于硬件和存储器保护用户配置的选项字节。
  - 包含专有自举程序代码的系统存储器。
  - OTP（一次性可编程）区域
    更多详细信息，请参见*第 3 节：嵌入式 Flash (FLASH)*。

Flash 接口根据 AHB 协议实施指令访问和数据访问。它将实施可加快 CPU 代码执行速度的预取缓冲器，以及执行通过 Flash 寄存器进行控制的 Flash 操作（编程/擦除）所必需的逻辑。

2.5 自举配置

在 STM32C0x1 中，可通过 BOOT0 引脚以及用户选项字节中的自举配置位 nBOOT1、nBOOT_SEL 和 nBOOT0 选择三种不同的自举模式，如下表所示。

表 4. 自举模式

自举模式配置					所选自举区域
BOOT_LOCK	nBOOT1 位	BOOT0 引脚	nBOOT_SEL 位	nBOOT0 位	
0	x	0	0	x	主 Flash
0	1	1	0	x	系统存储器
0	0	1	0	x	嵌入式 SRAM
0	x	x	1	1	主 Flash
0	1	x	1	0	系统存储器
0	0	x	1	0	嵌入式 SRAM
1	x	x	x	x	主 Flash

复位后，在 SYSCLK 的第四个上升沿锁存自举模式配置。用户可以设置与所需自举模式相关的自举模式配置。

退出待机模式后，还可以对自举模式配置进行重新采样。因此，在待机模式中，这些引脚必须保持所需的自举模式配置。该启动延迟结束后，CPU 将从地址 0x0000 0000 获取栈顶值，然后从位于 0x0000 0004 处的自举存储器中存储的地址开始执行代码。

根据所选的自举模式，主 Flash、系统存储器或 SRAM 可如下访问：

- 从主 Flash 自举：主 Flash 在自举存储空间 (0x0000 0000) 中有别名，但也可从它原来的存储空间 (0x0800 0000) 访问。换句话说，Flash 内容可从地址 0x0000 0000 或 0x0800 0000 开始访问。
- 从系统存储器自举：系统存储器在自举存储空间（0x0000 0000）中有别名，但也可从它原来的存储空间 0x1FFF 0000 访问。
- 从嵌入式 SRAM 自举：SRAM 在自举存储空间 (0x0000 0000) 中有别名，但也可从它原来的存储空间 (0x2000 0000) 访问。

BOOT_LOCK 位可用于强制自举主 Flash 中的唯一入口点，而与其他自举模式配置位无关。请参见*第 3.5.6 节：强制从 Flash 自举*。

空检查

内部空检查标志（*FLASH 访问控制寄存器 (FLASH_ACR)* 的 EMPTY 位）旨在实现通过自举程序轻松编程原始器件。当 BOOT0 引脚定义主 Flash 作为目标自举区域时，将使用该标志。当该标志置 1 时，器件被视为空，并且将选择系统存储器（自举程序）代替主 Flash 作为自举区域，以便用户对 Flash 进行编程。

该标志仅会在选项字节加载期间进行更新：当地址 0x0800 0000 的内容读为 0xFFFF FFFF 时，该标志会置 1，否则会清零。这意味着将原始器件设定为在系统复位后执行用户代码之后，需要上电复位或将 FLASH_CR 寄存器中的 OBL_LAUNCH 位置 1 才能清零该标志。此外，还可通过软件直接写入 EMPTY 位。

注意： *如果器件是首次进行编程，但选项字节未重新载入，器件在系统复位后仍将选择系统存储器作为自举区域。*

物理重映射

选择了自举模式后，应用软件可修改代码区域中可访问的存储器。这种修改是通过对 *SYSCFG 配置寄存器 1 (SYSCFG_CFGR1)* 中的 MEM_MODE 位进行编程设定实现的。

嵌入式自举程序

嵌入式自举程序位于系统存储器中，由 ST 在生产阶段编程。它用于通过以下串行接口重新编程 Flash：

- USART1
- I2C1

更多详细信息，请参见器件数据手册和应用笔记 AN2606。

3 嵌入式 Flash (FLASH)

3.1 FLASH 简介

Flash 接口可管理 CPU (Cortex®-M0+) AHB 对 Flash 进行的访问。该接口可针对 Flash 执行擦除和编程操作，并实施读写保护和安全机制。

Flash 接口通过指令预取和缓存机制加速代码执行。

3.2 Flash 主要特性

- 32 KB 的 Flash（主存储器）
- 存储器构成：
  - 主存储器块：32 KB（4 KB x 64 位）
  - 信息块：10 KB（1.28 KB x 64 位）
  - 页大小：2 KB
- 数据读取宽度为 64 位（无 ECC）
- 页擦除 (2 KB) 和批量擦除

Flash 接口特性：

- Flash 读操作
- Flash 编程/擦除操作
- 由选项字节 (RDP) 激活的读保护
- 由选项字节 (WRP) 选择的两个写保护区域
- 由选项字节 (PCROP) 选择的两个专有代码读保护区域
- 受控安全存储区
- Flash 为空检查
- 预取缓冲器
- CPU 指令缓存：两个64位缓存行（16 字节 RAM）
- 选项字节加载器

3.3 FLASH 功能描述

3.3.1 Flash 构成

Flash 为 64 位宽的存储单元，可用于存储代码和数据常量。

Flash 构成如下：

- 主存储器块，其中包含 16 页，每页 8 行（共 2 KB），每行 256 字节
- 信息块，其中包含：
  - 系统存储器，CPU 在系统存储器自举模式下从该存储器自举。该区域为保留区域，其中包含自举程序，用以通过以下接口之一对 Flash 进行重新编程：USART1 和 I2C1。器件在生产线上进行编程并获得保护，以防止误写/误擦除操作。更多详细信息，请参见 *www.st.com* 上提供的 AN2606。

– 1 KB（128 个双字）OTP（一次性可编程）字节，用于存储用户数据。OTP 数据不能擦除，只能执行一次写操作。如果仅一位为 0，则即使值为 0x0000 0000 0000 0000，整个双字（64 位）也无法再写入。当 RDP 级别为 1 且自举源不是主 Flash 区域时，无法读取 OTP 区域。
– 用于用户配置的选项字节。

下表给出了 Flash 到信息块和主存储器区域的映射。

表 5. Flash 构成

区域	地址	大小（字节）	存储器类型
信息块	0x1FFF 7800 - 0x1FFF 7FFF	2 K（仅使用前 128 个字节）	选项字节
	0x1FFF 7400 - 0x1FFF 77FF	1 K	工程字节
	0x1FFF 7000 - 0x1FFF 73FF	1 K	OTP 区域
	0x1FFF 0000 - 0x1FFF 17FF	6 K	系统存储器
主存储器	0x0800 7800 - 0x0800 7FFF	2 K	页 15
	...	...	...
	0x0800 1000 - 0x0800 17FF	2 K	页 2
	0x0800 0800 - 0x0800 0FFF	2 K	页 1
	0x0800 0000 - 0x0800 07FF	2 K	页 0

3.3.2 Flash 空检查

在 OBL 阶段，加载所有选项后，Flash 接口会检查主存储器的第一个存储单元是否已编程。该检查结果与 boot0 和 boot1 信息一起用于确定系统从哪个存储器区域自举。在尚未编写任何用户代码时，它会阻止系统从主 Flash 区域自举。

可从 *FLASH 访问控制寄存器 (FLASH_ACR)* 中的 EMPTY 位读取主 Flash 为空检查状态。可通过软件向 EMPTY 位写入适当值，以修改主 Flash 为空状态。

3.3.3 FLASH 读访问延迟

为了确保从 Flash 正确读取数据，请根据下表设置 *FLASH 访问控制寄存器 (FLASH_ACR)* 寄存器的 LATENCY[2:0] 位域。

表 6. LATENCY[2:0] 设置与 HCLK 频率的关系

HCLK (MHz)	LATENCY[2:0]
≤ 24	000（1 个 HCLK 周期）
≤ 48	001（2 个 HCLK 周期）

上电复位或从待机状态唤醒后，HCLK 时钟频率自动设置为 12 MHz，LATENCY[2:0] 位域自动设置为 000。请参见*第 5.2 节：时钟*。

要更改 HCLK 频率，请遵循以下顺序：

增大 HCLK 频率

1. 根据目标 HCLK 频率设置对应的 LATENCY[2:0] 位域（请参见*表 6*）。
2. 读取 LATENCY[2:0] 位域，直到返回上一步写入的值。
3. 通过 RCC_CFGR 寄存器的 SW[2:0] 位域按需选择系统时钟源。
4. 通过 RCC_CFGR 寄存器的 HPRE[3:0] 位域按需设置 HCLK 时钟预分频器。

可通过读取 RCC_CFGR 寄存器的时钟源状态位域 SWS[2:0] 检查系统有效选择的时钟源。此外，也可通过读取 RCC_CFGR 寄存器的 HPRE[3:0] 位域检查其内容。

减小系统时钟频率

1. 通过 RCC_CFGR 寄存器的 SW[2:0] 位域按需选择系统时钟源。
2. 通过 RCC_CFGR 寄存器的 HPRE[3:0] 位域按需设置 HCLK 时钟预分频器。
3. 读取 RCC_CFGR 寄存器的 SWS[2:0] 位域，直到返回步骤 1 中为 SW[2:0] 设置的值。此外，也可通过读取 RCC_CFGR 寄存器的 HPRE[3:0] 位域检查其内容。
4. 根据目标 HCLK 频率设置对应的 LATENCY[2:0] 位域（请参见*表 6*）。

3.3.4 Flash 加速

指令预取

每个 Flash 读操作可读取 64 位，可以是两条 32 位指令，也可以是四条 16 位指令，具体取决于启动的程序。该 64 位当前指令行保存在当前缓冲区中。因此对于顺序执行的代码，至少需要两个 CPU 周期来执行前一指令行的读取操作。在 CPU 请求当前指令行时，可使用 CPU S 总线的预取操作读取 Flash 中的下一个连续存放的指令行。

可将 *FLASH 访问控制寄存器 (FLASH_ACR)* 的 PRFTEN 位置 1，来使能预取功能。当访问 Flash 至少需要一个等待周期时，该功能非常有用。

处理非顺序执行的代码（有分支）时，指令可能并不存在于当前使用的或预取的指令行中。这种情况下，CPU 等待时间至少等于等待周期数。

如果当前缓冲区中存在循环，则不会执行新的访问。

缓存存储器

为了减少因指令跳转而产生的时间损耗，可将两个缓存行的 64 位指令（16 字节）保存到指令缓存存储器中。可将 *FLASH 访问控制寄存器 (FLASH_ACR)* 的指令缓存使能 (ICEN) 位置 1，来使能这一特性。每当出现指令缺失（即请求的指令未存在于当前使用的指令行、预取指令行或指令缓存存储器中）时，系统会将新读取的行复制到指令缓存存储器中。如果 CPU 请求的指令已存在于指令缓存存储器中，则无需任何延时即可立即获取。指令缓存存储器存满后，可采用 LRU（最近最少使用）策略确定指令缓存存储器中待替换的指令行。该特性非常适用于包含循环的代码。

系统复位后指令缓存存储器使能。

Cortex®-M0+ 不支持数据缓存。

3.3.5 FLASH 编程和擦除操作

器件嵌入式 Flash 可采用在线编程或在应用中编程两种方式。

在线编程 (ICP) 方式适用于更新 Flash 的所有内容，更新时使用 SWD 协议或系统自举程序支持的接口将 CPU 的用户应用程序加载到微控制器。ICP 可实现快速而高效的设计迭代，并且避免了不必要的器件封装处理或插接。

与 ICP 方法相比，**在应用中编程 (IAP)** 可通过微控制器支持的任何通信接口（I/O、UART、I^2C 和 SPI 等）将编程数据下载到存储器。IAP 允许用户在应用程序运行时重新编程 Flash。但是，部分应用程序必须事先通过 ICP 方式编程到 Flash。

如果数据字编程操作和页/存储区擦除操作由于器件复位或断电而中止，则无法保证这些操作成功完成。

对 Flash 执行编程/擦除操作期间，如果尝试读取 Flash，总线会停止工作。在完成编程/擦除操作后，会正确执行读操作。

解锁 Flash

复位后，*FLASH 控制寄存器 (FLASH_CR)* 不允许执行写操作，以防因电气干扰等原因出现对 Flash 的意外操作。上述寄存器的解锁顺序如下：

1. 在 *FLASH 密钥寄存器 (FLASH_KEYR)* 中写入 KEY1 = 0x4567 0123。
2. 在 *FLASH 密钥寄存器 (FLASH_KEYR)* 中写入 KEY2 = 0xCDEF 89AB。

如果操作顺序不正确，会锁定 FLASH_CR 寄存器，下次系统复位才会解锁。如果键值操作顺序不正确，会检测到总线错误并生成硬性故障中断。

也可通过软件将 FLASH_CR 寄存器中的 LOCK 位置 1 来锁定 FLASH_CR 寄存器。

注意： *Flash 状态寄存器 (FLASH_SR) 的 BSY1 位置 1 时，FLASH_CR 寄存器无法写入。在 BSY1 位置 1 时，尝试对该寄存器进行写操作会导致 AHB 总线阻塞，直到 BSY1 位清零。*

3.3.6 Flash 主存储器擦除顺序

Flash 擦除操作可在页级别（页擦除）或在整个存储器（批量擦除）执行。批量擦除不影响信息块（系统 Flash、OTP 和选项字节）。

Flash 页擦除

当某页受 PCROP 或 WRP 保护时，该页将不会被擦除，并且 WRPERR 位将置 1。

表 7. 页擦除概述

SEC_PROT	PCROP	WRP	PCROP_RDP	备注	WRPERR	CPU 总线错误
0	无	无	x	启动页擦除	无	无
	无	有		中止页擦除（不启动页擦除）	有	
	有	无				
	有	有				
1	x			页受保护	无	有

页 (2 KB) 擦除的具体步骤如下：

1. 检查 *Flash 状态寄存器 (FLASH_SR)* 的 BSY1 位，以确认当前未执行任何 Flash 操作。
2. 检查并清零之前的编程所导致的全部错误编程标志。否则，PGSERR 将置 1。
3. 在 *FLASH 控制寄存器 (FLASH_CR)* 中，将 PER 位置 1 并选择要擦除的页 (PNB)。
4. 将 *FLASH 控制寄存器 (FLASH_CR)* 的 STRT 位置 1。
5. 等待 *Flash 状态寄存器 (FLASH_SR)* 的 BSY1 位清零。

注意： *当 STRT 位置 1 时，将自动使能 HSI48 内部振荡器（经过三分频提供 16 MHz 频率）。当 STRT 位清零时，将自动禁止 HSI48 内部振荡器，除非之前已通过 RCC_CR 寄存器的 HSION 位使能。*

Flash 存储区擦除或批量擦除

当使能 PCROP 或 WRP 时，将中止 Flash 批量擦除，不会启动擦除操作，并且 WRPERR 位将置 1。

表 8. 批量擦除概述

SEC_PROT	PCROP	WRP	PCROP_RDP	备注	WRPERR	CPU 总线错误
0	无	无	x	启动存储器擦除	无	无
	无	有		中止擦除（不启动擦除）	有	
	有	无				
	有	有				
1	x			中止擦除（不启动擦除）	无	有

批量擦除的具体步骤如下：

1. 检查 *Flash 状态寄存器 (FLASH_SR)* 的 BSY1 位，以确认当前未执行任何 Flash 操作。
2. 检查并清零之前的编程所导致的全部错误编程标志。否则，PGSERR 将置 1。
3. 将 *FLASH 控制寄存器 (FLASH_CR)* 的 MER1 位置 1。
4. 将 *FLASH 控制寄存器 (FLASH_CR)* 的 STRT 位置 1。
5. 等待 *Flash 状态寄存器 (FLASH_SR)* 的 BSY1 位清零。

注意： *当 STRT 位置 1 时，将自动使能 HSI48 内部振荡器（经过三分频提供 16 MHz 频率）。当 STRT 位清零时，将自动禁止 HSI48 内部振荡器，除非之前已通过 RCC_CR 寄存器的 HSION 位使能。*

3.3.7 FLASH 主存储器编程顺序

Flash 一次编程 64 位。

对于之前已完成编程的地址，不允许再为其编程非零数据。否则，*Flash 状态寄存器 (FLASH_SR)* 的 PROGERR 标志将置 1。

只能编程双字（2 x 32 位数据）。

- 尝试写入字节（8 位）或半字（16 位）将使 *Flash 状态寄存器 (FLASH_SR)* 的 SIZERR 标志置 1。
- 尝试写入与双字地址不对齐的双字时，将使 *Flash 状态寄存器 (FLASH_SR)* 的 PGAERR 标志置 1。

标准编程

标准模式下，Flash 编程顺序如下：

1. 检查 *Flash 状态寄存器 (FLASH_SR)* 的 BSY1 位，以确认当前未执行任何主 Flash 操作。
2. 检查并清零之前的编程所导致的全部错误编程标志。否则，PGSERR 将置 1。
3. 将 *FLASH 控制寄存器 (FLASH_CR)* 的 PG 位置 1。
4. 针对所需存储器地址（主存储器块或 OTP 区域内）执行数据写入操作。只能编程双字（64 位）。
   a) 在与双字匹配的地址中写入第一个字
   b) 写入第二个字。
5. 等待 *Flash 状态寄存器 (FLASH_SR)* 的 BSY1 位清零。
6. 检查 *Flash 状态寄存器 (FLASH_SR)* 中的 EOP 标志是否置 1，如果置 1，则表示已成功执行编程操作，并通过软件将其清零。
7. 如果不再有编程请求，则将 *FLASH 控制寄存器 (FLASH_CR)* 的 PG 位清零。

注意： *Flash 接口接收到适当的序列（双字）后，自动启动编程且 BSY1 位置 1。当 PG 位置 1 时，将自动使能 HSI48 内部振荡器（经过三分频提供 16 MHz 频率）。当 PG 位清零时，将自动禁止 HSI48 内部振荡器，除非之前已通过 RCC_CR 寄存器的 HSION 位使能。*

快速编程

该模式主要用于缩短页编程时间。由于无需在编程之前验证 Flash 存储单元，因此可针对每个双字节省高压的上升和下降时间。

该模式允许编程一行（32 个双字 = 256 字节）。

快速编程过程中，Flash 时钟 (HCLK) 频率必须至少为 8 MHz。

只有主存储器可编程为快速编程模式。

标准模式下，主 Flash 编程顺序如下：

1. 执行批量擦除或页擦除。否则，PGSERR 将置 1。
2. 检查 *Flash 状态寄存器 (FLASH_SR)* 的 BSY1 位，以确认当前未执行任何主 Flash 操作。
3. 检查并清零之前的编程所导致的全部错误编程标志。
4. 将 *FLASH 控制寄存器 (FLASH_CR)* 中的 FSTPG 位置 1。
5. 写入 16 个双字以编程一行（128 字节）。
6. 等待 *Flash 状态寄存器 (FLASH_SR)* 的 BSY1 位清零。
7. 检查 *Flash 状态寄存器 (FLASH_SR)* 中的 EOP 标志是否置 1，如果置 1，则表示已成功执行编程操作，此时用软件将其清零。
8. 如果不再有编程请求，则将 *Flash 状态寄存器 (FLASH_SR)* 的 FSTPG 位清零。

注意： *当正在执行读操作时，如果尝试在快速编程模式下进行写操作，则会中止编程，且不会发出任何系统通知（无错误标志置 1）。*

Flash 接口接收到第一个双字后，编程自动启动。对第一个双字施加高压时，BSY1 位置 1，最后一个双字已编程或发生错误时，该位清零。
当 FSTPG 位置 1 时，将自动使能 HSI48 内部振荡器（经过三分频提供 16 MHz 频率）。当 FSTPG 位清零时，将自动禁止 HSI48 内部振荡器，除非之前已通过 RCC_CR 寄存器的 HSION 位使能。

16 个双字必须连续写入。Flash 上将保持高压以进行所有编程。两个双字写入请求之间的最长时间为编程时间（约为 20 µs）。如果第二个双字在此编程时间过后传到，则快速编程被中断，MISSERR 置 1。

在两次擦除之间，对于一整行，高压持续时间不得超过 8 ms。可通过基于大于等于 8 MHz 的时钟系统连续写入的 16 个双字的序列来保证这一点。设置快速编程后内部超时计数器计数 7 ms，并在超时后会停止编程。此时，FASTERR 位将置 1。

如果发生错误，则高压停止，不会对下一个要编程的双字进行编程。总之，之前所有的双字均已正确编程。

编程错误

可检测到多种错误。若发生错误，Flash 操作（编程或擦除）会中止。

- **PROGERR**：编程错误

  在标准编程过程中：如果要写入的字之前未擦除，则 PROGERR 置 1（要编程的值全为 0 且目标地址位于主存储器中时除外）。

- **SIZERR**：编程大小错误

  在标准编程或快速编程过程中：只能编程双字且只能写入 32 位数据。如果写入一个字节或半字，则 SIZERR 置 1。

- **PGAERR**：编程对齐错误

  如果发生以下事件，PGAERR 位会置 1：

  – 在标准编程过程中：要编程的第一个字与双字地址不匹配，或第二个字不属于同一双字地址。

  – 在快速编程过程中：要编程的数据与之前编程的双字不属于同一行，或者要编程的地址小于或等于之前的地址。

- **PGSERR**：编程顺序错误

  如果发生以下事件，PGSERR 位会置 1：

  – 对于标准编程顺序或快速编程顺序：PG 和 FSTPG 清零时写入数据。

  – 对于标准编程顺序或快速编程顺序：PG 或 FSTPG 置 1 后，MER1 和 PER 不清零。

  – 对于快速编程顺序：将 FSTPG 位置 1 后执行批量擦除操作。

  – 对于批量擦除顺序：MER1 置 1 后，PG、FSTPG 和 PER 不清零。

  – 对于页擦除顺序：PER 置 1 后，PG、FSTPG 和 MER1 不清零。

  – 如果由于之前发生编程错误，导致 PROGERR、SIZERR、PGAERR、WRPERR、MISSERR、FASTERR 或 PGSERR 置 1，则 PGSERR 也会置 1。

- **WRPERR**：写保护错误

  如果发生以下事件，WRPERR 位会置 1：

  – 尝试在写保护区域 (WRP) 或 PCROP 区域执行编程或擦除操作。

  – 尝试在一页或多页受 WRP 或 PCROP 保护时，执行批量擦除操作。

  – 读保护 (RDP) 设为级别 1 时，已连接调试功能或从 SRAM 或系统 Flash 进行自举。

  – 尝试在读保护 (RDP) 设为级别 2 时修改选项字节。

- **MISSERR**：快速编程数据丢失错误

  在快速编程过程中：所有数据必须连续写入。如果之前的数据编程已完成，并且要编程的下一个数据尚未写入，则 MISSERR 置 1。

- **FASTERR**：快速编程错误

  在快速编程过程中：如果发生以下事件，FASTERR 位会置 1：

  – FSTPG 位置 1 的时间超过 8 ms，这会生成超时检测

  – 行快速编程被 MISSERR、PGAERR、WRPERR 或 SIZERR 中断

如果在编程或擦除操作期间出现错误，则 *Flash 状态寄存器 (FLASH_SR)* 的以下错误标志之一将置 1：

- PROGERR、SIZERR、PGAERR、PGSERR、MISSERR（编程错误标志）
- WRPERR（保护错误标志）

这种情况下，*Flash 状态寄存器 (FLASH_SR)* 的操作错误标志 OPERR 置 1，并且如果 *FLASH 控制寄存器 (FLASH_CR)* 的错误中断使能位 ERRIE 置 1，则将产生一个中断。

注意： *如果检测到多个连续错误（例如，在对 Flash 进行 DMA 传输期间），则直到连续写操作请求结束，这些错误标志才会清零。*

编程与缓存

如果 Flash 中的擦除操作也涉及指令缓存中的数据，则用户必须确保在代码执行期间访问这些数据之前先将其重新写入缓存。

注意： *该缓存只有在被禁止 (ICEN = 0) 时才能刷新。*

3.4 FLASH 选项字节

3.4.1 FLASH 选项字节说明

选项字节由最终用户根据具体的应用要求进行配置。以如下配置为例：可在硬件或软件模式下选择看门狗（参见*第 3.4.2 节：FLASH 选项字节编程*）。

一个双字基于选项字节进行拆分，如*表 9* 所示。

表 9. 选项字节格式

63-56	55-48	47-40	39-32	31-24	23-16	15-8	7-0
补码选项字节 3	补码选项字节 2	补码选项字节 1	补码选项字节 0	选项字节 3	选项字节 2	选项字节 1	选项字节 0

表 10 显示了 Flash 信息块中选项字节（仅低位字）的构成。可通过软件从这些 Flash 存储单元或表中引用的相应选项寄存器中读取选项字节。有关选项寄存器位域以及选项字节位域的说明，请参见第 *3.7.6* 节至第 *3.7.13* 节。

表 10. 选项字节的构成

地址[1]	相应的选项寄存器（章节）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		选项字节 3								选项字节 2								选项字节 1								选项字节 0							
0x1FFF7800	FLASH_OPTR *(3.7.6)*	保留		IRHEN	NRST_MODE[1:0]		nBOOT0	nBOOT1	nBOOT_SEL	SECURE_MUXING_EN	RAM_PARITY_CHECK	HSE_NOT_REMAPPED[2]	保留	WWDG_SW	IWDG_STBY	IWDG_STOP	IWDG_SW	nRST_SHDW	nRST_STDBY	nRST_STOP	BORF_LEV		BORR_LEV		BOR_EN	RDP							
	出厂值	X	X	1	1	1	1	1	1	1	1	1	X	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0
0x1FFF7808	FLASH_PCROP1ASR *(3.7.7)*	保留																										PCROP1A_STRT					
	出厂值	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1	1	1

表 10. 选项字节的构成（续）

地址[1]	相应的选项寄存器（章节）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		选项字节 3								选项字节 2								选项字节 1								选项字节 0							
0x1FFF7810	FLASH_PCROP1AER (3.7.8)	PCROP_RDP	保留																									PCROP1A_END					
	出厂值	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0
0x1FFF7818	FLASH_WRP1AR (3.7.9)	保留												WRP1A_END				保留												WRP1A_STRT			
	出厂值	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1
0x1FFF7820	FLASH_WRP1BR (3.7.10)	保留												WRP1B_END				保留												WRP1B_STRT			
	出厂值	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1
0x1FFF7828	FLASH_PCROP1BSR (3.7.11)	保留																										PCROP1B_STRT					
	出厂值	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1	1	1
0x1FFF7830	FLASH_ PCROP1BER (3.7.12)	保留																										PCROP1B_END					
	出厂值	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0
0x1FFF7870	FLASH_SECR (3.7.13)	保留															BOOT_LOCK	保留												SEC_SIZE			
	出厂值	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0

1. 双字地址的高 32 位包含低 32 位的补码数据。
2. 仅与只有 STM32C031xx 提供的 48 引脚封装相关。

3.4.2 FLASH 选项字节编程

复位后，*FLASH 控制寄存器 (FLASH_CR)* 的选项相关位受到写保护。要针对选项字节页执行任何操作，*FLASH 控制寄存器 (FLASH_CR)* 的选项锁定位 OPTLOCK 必须清零。该寄存器的解锁顺序如下：

1. 使用 LOCK 清零序列解锁 FLASH_CR（参见*解锁 Flash*）
2. 在 *FLASH 选项密钥寄存器 (FLASH_OPTKEYR)* 中写入 OPTKEY1 = 0x0819 2A3B
3. 在 *FLASH 选项密钥寄存器 (FLASH_OPTKEYR)* 中写入 OPTKEY2 = 0x4C5D 6E7F

如果操作顺序不正确，会锁定 Flash 选项寄存器，下次系统复位才会解锁。如果键值操作顺序不正确，会检测到总线错误并生成硬性故障中断。

通过软件将 OPTLOCK 位置 1 后，可防止用户选项发生意外的擦除/编程操作。

注意： *如果 LOCK 由软件置 1，则 OPTLOCK 也自动置 1。*

修改用户选项

选项字节与主存储器用户地址的编程方式不同。

要修改用户选项的值，请执行以下步骤：

1. 按照上述清零序列将 OPTLOCK 选项锁定位清零。
2. 在 FLASH 选项寄存器中写入所需值。
3. 检查 *Flash 状态寄存器 (FLASH_SR)* 的 BSY1 位，以确认当前未执行任何 Flash 操作。
4. 将 *FLASH 控制寄存器 (FLASH_CR)* 的选项启动位 OPTSTRT 置 1。
5. 等待 BSY1 位清零。

注意： *先擦除用户选项字节页，然后以 Flash 选项寄存器中包含的值对所有选项字节进行编程，从而自动修改一个选项的值。*

OPTSTRT 位置 1 后，会自动计算补码值并写入补码选项字节。

小心： 在选项字节编程失败时（任何原因所致，例如在选项字节更改序列期间发生断电或复位），选项字节的不匹配值会在复位后加载。这些不匹配值会强制进行安全配置，可能永久锁定器件。为防止发生这种情况，仅限在安全环境中（即安全的电源供应，无挂起的看门狗，干净的复位线）编程选项字节。

选项字节加载

BSY1 位清零后，所有新选项都将更新到 Flash，但不应用到系统。读取选项寄存器仍将返回上次加载的选项字节值，新选项仅在加载后才对系统有影响。

在以下两种情况下执行选项字节加载：

- *FLASH 控制寄存器 (FLASH_CR)* 的 OBL_LAUNCH 位置 1 时
- 电源复位后（BOR 复位或退出待机/关断模式）

选项字节加载器读取选项块并将数据存储到内部选项寄存器。这些内部寄存器配置系统，可由软件读取。将 OBL_LAUNCH 置 1 会产生复位，因此在系统复位下执行选项字节加载。

每个选项位在同一双字中也有各自的补码。选项加载期间，验证选项位及其补码可检查是否已正确进行加载。

选项字节加载期间，选项由双字读取。

如果字与其补码匹配，则将选项字/字节复制到选项寄存器。

如果字与其补码不匹配，则 OPTVERR 状态位置 1。不匹配值被强制放入选项寄存器：

- 对于 USR OPT 选项，除了 BOR_EN 位为 0（BOR 禁用）外，所有选项位的不匹配的值均为 1。
- 对于 WRP 选项，不匹配值为默认值“无保护”。
- 对于 RDP 选项，不匹配值为默认值“级别 1”。
- 对于 PCROP，不匹配值为“所有存储器均受保护”。
- 对于 BOOT_LOCK，不匹配值为“强制从主 Flash 自举”。

系统复位后，选项字节会复制到以下可由软件读写的选项寄存器中：

- FLASH_OPTR
- FLASH_PCROP1xSR（x = A 或 B）
- FLASH_PCROP1xER（x = A 或 B）
- FLASH_WRP1xR（x = A 或 B）
- FLASH_SECR

上述寄存器也用于修改选项。如果这些寄存器未由用户修改，则会反映系统的选项状态。更多详细信息，请参见*修改用户选项*。

3.5 Flash 保护

可对主 Flash 进行保护，使其具有读保护 (RDP)，以防其遭受外部访问。也可对页进行保护，以防因程序指针错乱而发生意外的写操作 (WRP)。写保护 WRP 粒度为 2 KB。除 RDP 和 WRP 外，还可防止 Flash 被第三方读写 (PCROP)。PCROP 粒度（子页大小）为 512 字节。

3.5.1 FLASH 读保护 (RDP)

通过将 RDP 选项字节置 1，然后应用系统复位来重载新的 RDP 选项字节，可激活读保护。读保护用于保护主 Flash、选项字节和 SRAM。

注意： *如果在调试器仍通过 SWD 连接时设置读保护，则应用电源复位，而非系统复位。*

有三种读保护级别：无保护（级别 0）到最大保护或禁止调试（级别 2）。

RDP 选项字节及其补码包含成对值时，Flash 受保护，如*表 11* 所示。

表 11. Flash 读保护状态

RDP 字节值	RDP 补码字节值	读保护级别
0xAA	0x55	级别 0
除组合 [0xAA, 0x55] 和 [0xCC, 0x33] 外的任何值		级别 1（默认值）
0xCC	0x33	级别 2

无论保护级别如何，系统存储器区域均可进行读访问。对于编程/擦除操作，则不可访问。

级别 0：无保护

可对主 Flash 区域执行读取、编程和擦除操作。也可对选项字节执行所有操作。

级别 1：读保护

当 RDP 字节和 RDP 补码字节包含 [0xAA, 0x55] 和 [0xCC, 0x33] 以外的任何值组合时，将设置级别 1 的读保护。RDP 选项字节被擦除时，级别 1 为默认保护级别。

- **用户模式：**在用户模式下执行的代码（从用户 Flash 自举）可对主 Flash 和选项字节执行所有访问。
- **调试模式、从 SRAM 自举模式以及从系统存储器自举模式：**在调试模式下或当代码从 SRAM 或系统存储器自举时，主 Flash 完全不可访问。在上述模式下，对 Flash 进行读访问或写访问会生成总线错误和硬性故障中断。

小心： 如果配置了级别 1 且未定义 PCROP 区域，则必须将 PCROP_RDP 位置 1（RDP 级别从级别 1 降到级别 0 时，进行完全批量擦除）。如果配置了级别 1 且定义了 PCROP 区域，则通过 RDP（而非 PCROP）保护的用户代码必须置于包含 PCROP 保护子页的页之外。

级别 2：禁止调试

在此级别下，可保证保护级别 1。此外，CPU 调试端口、从 RAM 自举（自举 RAM 模式）和从系统存储器自举（自举程序模式）不再可用。在用户执行模式（自举 FLASH 模式）下，允许对主 Flash 执行所有操作。

注意： *在复位时 CPU 调试端口也被禁止。*

注意： *意法半导体无法对设为保护级别 2 的器件做失效分析。*

更改读保护级别

读保护级别可以：

- 从级别 0 更改为级别 1，具体方法是将 RDP 字节的值更改为除 0xCC 外的任意值
- 从级别 0 或级别 1 更改为级别 2，具体方法是将 RDP 字节的值更改为 0xCC
- 从级别 1 更改为级别 0，具体方法是将 RDP 字节的值更改为 0xAA

更改为级别 2 后，无法再更改读保护级别。

FLASH PCROP 区域 A 结束地址寄存器 (FLASH_PCROP1AER) 的 PCROP_RDP 位置 1 时，从级别 1 更改为级别 0 会触发主 Flash 进行完全批量擦除。除 PCROP 保护外的用户选项设置为之前从 FLASH_OPTR 和 FLASH_WRP1xR（x = A 或 B）复制的值。PCROP 被禁止。OTP 区域不受批量擦除操作的影响，同样保持不变。

PCROP_RDP 位清零时，会发生部分批量擦除，仅擦除未与 PCROP 区域重叠的 Flash 页（不包含任何 PCROP 保护子页）。选项字节使用之前的值重新编程。这同样适用于 FLASH_PCROP1xSR 和 FLASH_PCROP1xER 寄存器（x = A 或 B）。

Table 12: RDP 从级别 1 降为级别 0 后的批量擦除

PCROP 区域	PCROP_RDP	批量擦除
无	x	完全
部分 Flash	1	
	0	部分 （未与 PCROP 区域重叠的 Flash 页）
整个 Flash		无

注意： *只有 RDP 从级别 1 降为级别 0 才会触发批量擦除（完全或部分）。RDP 从级别 0 升为级别 1、从级别 1 升为级别 2 或从 0 升为级别 2 不会触发任何批量擦除。*

为验证保护级别更改，必须通过将 FLASH 控制寄存器 (FLASH_CR) 的 OBL_LAUNCH 位置 1 来重载选项字节。

图 3. 更改读保护 (RDP) 级别

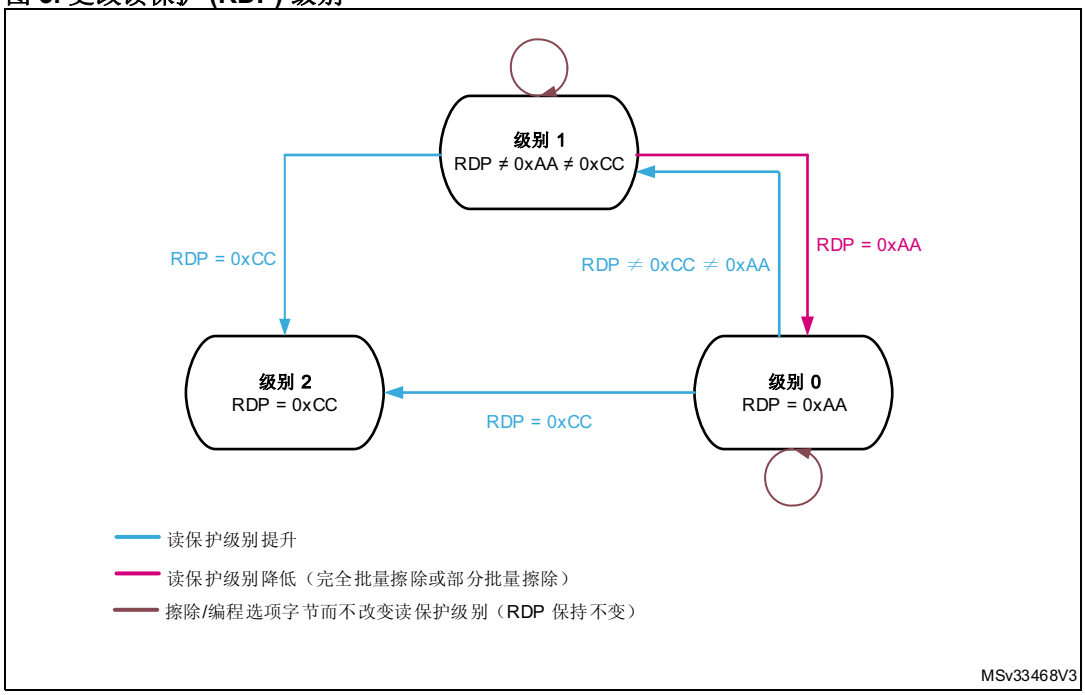


表 13. 访问状态与保护级别和执行模式的关系

区域	保护级别	用户执行（从 Flash 自举）			调试/从 RAM 自举/从加载程序自举		
		读	写	擦除	读	写	擦除
主 Flash	1	可以	可以	可以	不可以	不可以	不可以(3)
	2	可以	可以	可以	N/A(1)	N/A(1)	N/A(1)
系统存储器(2)	1	可以	不可以	不可以	可以	不可以	不可以
	2	可以	不可以	不可以	N/A(1)	N/A(1)	N/A(1)
选项字节	1	可以	可以(3)	可以	可以	可以(3)	可以
	2	可以	不可以	不可以	N/A(1)	N/A(1)	N/A(1)
OTP	1	可以	可以	N/A	不可以	不可以	N/A
	2	可以	可以	N/A	N/A(1)	N/A(1)	N/A(1)

1. 保护级别 2 激活后，将禁止调试端口、从 RAM 自举和从系统存储器自举。
2. 无论保护级别（0、1 或 2）和执行模式如何，都只能对系统存储器进行读访问。
3. 在禁止所有级别保护 (0xAA) 的情况下编程 RDP 选项字节时，会擦除 Flash 主存储器。

3.5.2 FLASH 专有代码读保护 (PCROP)

可防止 Flash 的两个区域遭受第三方的意外读取和/或写入。

受保护区域为仅执行区域：只能由 STM32 CPU 访问（使用指令代码），所有其他访问（DMA、调试和 CPU 数据读取、写入和擦除）被严格禁止。PCROP 区域具有子页（512 字节）粒度。另有一个选项位 (PCROP_RDP) 可用来选择在 RDP 保护从级别 1 变为级别 0 时 PCROP 区域是否被擦除（参见*更改读保护级别*）。

在 Flash 中，每个 PCROP 区域由起始子页偏移和结束子页偏移定义。上述偏移使用 PCROP 地址寄存器 *FLASH PCROP 区域 A 起始地址寄存器 (FLASH_PCROP1ASR)*、*FLASH PCROP 区域 A 结束地址寄存器 (FLASH_PCROP1AER)*、*FLASH PCROP 区域 B 起始地址寄存器 (FLASH_PCROP1BSR)* 和 *FLASH PCROP 区域 B 结束地址寄存器 (FLASH_PCROP1BER)* 的相应位域定义。

PCROP 区域 ***x（A 或 B）定义为从地址：***

Flash 基址 + [PCROP1x_STRT x 0x200]（包含）

到地址：

Flash 基址 + [(PCROP1x_END + 1) x 0x200]（不包含）。

PCROP 区域最小为两个 PCROP 子页（2 x 512 字节）：

PCROP1x_END = PCROP1x_STRT + 1。

当

PCROP1x_END = PCROP1x_STRT 时，

整个 Flash 均受 PCROP 保护。

例如，要对 0x0800 0800 到 0x0800 13FF 的地址区域进行 PCROP 保护，应按如下设置 FLASH_PCROP1xSR 寄存器的 PCROP 起始子页位域和 FLASH_PCROP1xER 寄存器（x = A 或 B）的 PCROP 结束子页位域：

– PCROP1x_STRT = 0x04（PCROP 区域起始地址 0x0800 0800）
– PCROP1x_END = 0x09（PCROP 区域结束地址 0x0800 13FF）

对 PCROP 保护地址进行数据读访问会将 RDERR 标志置 1。

PCROP 保护地址还受写保护。对 PCROP 保护地址进行写访问会将 WRPERR 标志置 1。

PCROP 保护区域还受擦除保护。尝试擦除包含至少一个 PCROP 保护子页的页会失败。此外，如果定义了 PCROP 保护区域，则无法执行软件批量擦除。

只有当 RDP 从级别 1 更改为级别 0 时，PCROP 才会无效。通过修改用户选项来清除 PCROP 或减小 PCROP 保护区域不会对 PCROP 区域产生任何影响。相反，可能会增大 PCROP 保护区域。

选项位 PCROP_RDP 清零时，RDP 从级别 1 更改为级别 0 会触发部分批量擦除，保留与 PCROP 保护区域重叠的 Flash 页的内容。有关详细信息，请参见*更改读保护级别*部分。

Table 14: PCROP 保护

PCROP 寄存器值 （x = A 或 B）	PCROP 保护区域
PCROP1x_STRT = PCROP1x_END	整个 Flash
PCROP1x_STRT > PCROP1x_END	无（不受保护）
PCROP1x_STRT < PCROP1x_END	从 PCROP1x_STRT 到 PCROP1x_END 的子页 （读保护、写保护和擦除保护）； PCROP 区域边界页（擦除保护）。

注意： *PCROP_RDP 清零时，建议在 Flash 页边界上定义 PCROP 区域起始地址和结束地址（2 KB 粒度），或保留和清空 PCROP 区域边界页（PCROP 区域起始地址和结束地址所在的页）上不受 PCROP 保护的存储空间。*

3.5.3 FLASH 写保护 (WRP)

可对 Flash 中的用户区域实施保护，以防其遭受意外的写操作。可定义两个写保护 (WRP) 区域，页面大小为 2 KB。每个区域由与物理 Flash 基址相关的起始页偏移和结束页偏移定义。上述偏移在 *FLASH WRP 区域 A 地址寄存器 (FLASH_WRP1AR)* 和 *FLASH WRP 区域 B 地址寄存器 (FLASH_WRP1BR)* WRP 地址寄存器中定义。

WRP ***x*** *区域（x = A 或 B）定义为从地址*

Flash 基址 + [WRP1x_STRT x 0x0800]（包含）

到地址

Flash 基址 + [(WRP1x_END+1) x 0x0800]（不包含）。

WRP 区域最小为一个 WRP 页 (2 KB)：

WRP1x_END = WRP1x_STRT。

例如，通过从地址 0x0800 1000（包含）到地址 0x0800 3FFF（包含）的 WRP 保护 Flash：

如果选择在 Flash 中自举，则必须对 FLASH_WRP1AR 寄存器进行如下编程：

– WRP1A_STRT = 0x02。
– WRP1A_END = 0x07。

也可使用 FLASH_WRP1BR 中的 WRP1B_STRT 和 WRP1B_END（Flash 中的区域 B）。

WRP 有效时，无法对其执行擦除或编程操作。因此，如果某区域受写保护，则无法执行软件批量擦除操作。

如果尝试对 Flash 的写保护区域执行擦除/编程操作，则 FLASH_SR 寄存器的写保护错误标志 (WRPERR) 将置 1。对以下区域进行写访问时，该标志也会置 1：

– OTP 区域
– Flash 中永远不能执行写操作的部分（例如 ICP）
– PCROP 区域

注意： *选择 Flash 读保护级别（RDP 级别 = 1）后，如果已连接 CPU 调试功能（单线调试）或者正在从 SRAM 或系统 Flash 执行自举代码，则即使未激活 WRP，也无法对存储器执行编程或擦除操作。尝试执行任何相关操作都会产生硬性故障 (BusFault)。*

Table 15: WRP 保护

WRP 寄存器值（x = A 或 B）	WRP 保护区域
WRP1x_STRT = WRP1x_END	页 WRP1x
WRP1x_STRT > WRP1x_END	无（不受保护）
WRP1x_STRT < WRP1x_END	从 WRP1x_STRT 到 WRP1x_END 的页

注意： *为验证 WRP 选项，必须通过将 Flash 控制寄存器中的 OBL_LAUNCH 位置 1 重载选项字节。*

3.5.4 受控安全存储区

受控安全存储区主要用于保护 Flash 的特定区域，以防其遭受意外的访问。系统复位后，受控安全存储区中的代码只能在受控安全区域变为安全状态后执行，并且在下一次系统复位前不会再次执行。这样便可实现安全键值存储或安全自举等软件安全服务。

受控安全存储区位于主 Flash 中，专用于执行可信代码。在非安全状态下，受控安全存储区与主 Flash 的其余部分无异。在安全状态下（FLASH_CR 寄存器的 SEC_PROT 位置 1），对受控安全存储区的任何访问（取、读、编程、擦除）都会被拒绝，并生成一个总线错误。受控安全区域只能由系统复位取消安全状态。

受控安全存储区的大小由 FLASH_SECR 寄存器的 SEC_SIZE[4:0] 位域定义，只能在 RDP 级别 0 下修改。RDP 从级别 1 变为级别 0 后，受控安全存储区的内容会被擦除，即使其与 PCROP 子页重叠时也不例外。

注意： *受控安全存储区的起始地址为 0x0800 0000。激活受控安全存储区之前，在必要时将向量表移出第 0 页。*

注意： *RDP 从级别 1 变为级别 0 且 PCROP_RDP 位清零后，受控安全存储区会被擦除，即使其与 PCROP 子页重叠时也不例外。未与受控安全存储区重叠的 PCROP 子页不会被擦除。请参见表 16。*

表 16. RDP 从级别 1 变为级别 0 时的受控安全存储器擦除

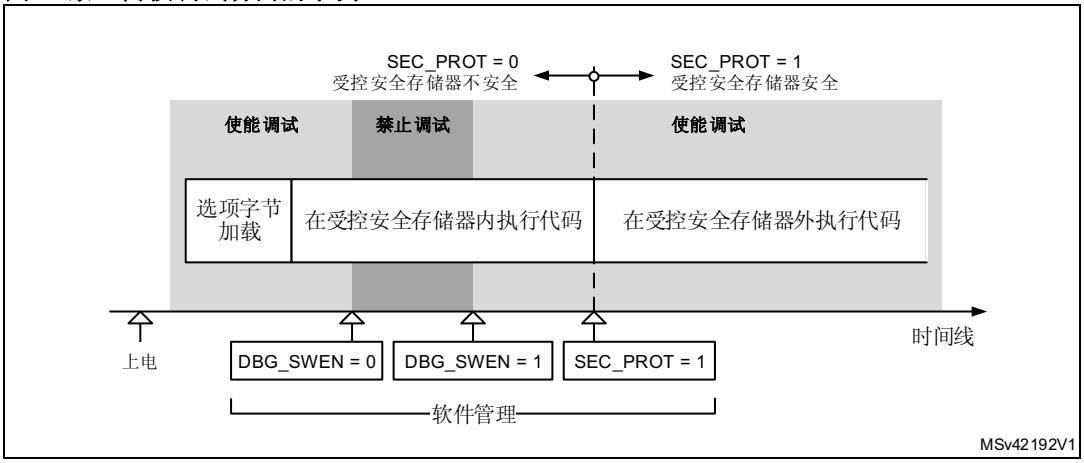
受控安全存储器大小 (SEC_SIZE[4:0])	PCROP_RDP	擦除的页
0	1	所有页（批量擦除）
0	0	除 PCROP 外的所有页
> 0	1	所有页（批量擦除）
> 0	0	受控安全存储区之外除 PCROP 外的所有页

3.5.5 禁止内核调试访问

要在受控安全存储区中执行敏感代码或处理敏感数据，可临时禁止对内核的调试访问。

图 4 给出了管理 DBG_SWEN 和 SEC_PROT 位的示例。

图 4. 禁止内核调试访问的示例



3.5.6 强制从 Flash 自举

要提高安全性和建立信任链，可通过 FLASH_SECR 寄存器的 BOOT_LOCK 选项位强制系统从主 Flash 自举，而与其他启动选项无关。BOOT_LOCK 位随时可以置 1，但只能在以下情况下复位：

- RDP 设置为级别 0，或者
- 当请求级别 0 且执行完全批量擦除时，RDP 设置为级别 1。

3.6 FLASH 中断

表 17. FLASH 中断请求

中断事件	事件标志	事件标志/中断清除方法	中断使能控制位
操作结束	EOP(1)	写入 EOP=1	EOPIE
操作错误	OPERR(2)	写入 OPERR=1	ERRIE
读保护错误	RDERR	写入 RDERR=1	RDERRIE
写保护错误	WRPERR	写入 WRPERR=1	N/A
大小错误	SIZERR	写入 SIZERR=1	N/A
编程顺序错误	PROGERR	写入 PROGERR=1	N/A
编程对齐错误	PGAERR	写入 PGAERR=1	N/A
编程顺序错误	PGSERR	写入 PGSERR=1	N/A
快速编程期间数据丢失错误	MISSERR	写入 MISSERR=1	N/A
快速编程错误	FASTERR	写入 FASTERR=1	N/A

1. 仅当 EOPIE 置 1 后，EOP 才会置 1。
2. 仅当 ERRIE 置 1 后，OPERR 才会置 1。

3.7 FLASH 寄存器

3.7.1 FLASH 访问控制寄存器 (FLASH_ACR)

FLASH access control register

偏移地址：0x000

复位值：0x0004 0600

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

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

位 31:19 保留，必须保持复位值。

位 18 **DBG_SWEN**：调试访问软件使能 (Debug access software enable)
可通过软件使用该位来使能/禁止调试器读访问。
0：禁止调试器
1：使能调试器

位 17 保留，必须保持复位值。

位 16 **EMPTY**：主 Flash 区域为空 (Main flash memory area empty)
该位指示主 Flash 区域的第一个存储单元是已擦除还是具有编程值。
0：主 Flash 区域已编程
1：主 Flash 区域为空
该位可由软件置 1 和复位。

位 15:12 保留，必须保持复位值。

位 11 **ICRST**：CPU 指令缓存复位 (CPU Instruction cache reset)
0：CPU 指令缓存不复位
1：CPU 指令缓存复位
只有在禁止指令缓存后才能写入该位。

位 10 保留，必须保持复位值。

位 9 **ICEN**：CPU 指令缓存使能 (CPU Instruction cache enable)
0：禁止 CPU 指令缓存
1：使能 CPU 指令缓存

位 8 **PRFTEN**：CPU 预取使能 (CPU Prefetch enable)
0：禁止 CPU 预取
1：使能 CPU 预取

位 7:3 保留，必须保持复位值。

位 2:0 **LATENCY[2:0]**：Flash 访问延迟 (Flash memory access latency)
该位域中的值表示访问 Flash 时的 CPU 等待周期数。
000：零个等待周期
001：一个等待周期
其他值：保留
对位域执行新的写操作后，只有读取时返回相同的值，该写操作才生效。

3.7.2 FLASH 密钥寄存器 (FLASH_KEYR)

FLASH key register

偏移地址：0x008

复位值：0x0000 0000

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

位 31:0 **KEY[31:0]**：FLASH 密钥 (FLASH key)

要将 *FLASH 控制寄存器 (FLASH_CR)* 解锁以允许执行编程/擦除操作，必须按顺序写入以下键值：

KEY1：0x4567 0123

KEY2：0xCDEF 89AB

3.7.3 FLASH 选项密钥寄存器 (FLASH_OPTKEYR)

FLASH option key register

偏移地址：0x00C

复位值：0x0000 0000

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

位 31:0 **OPTKEY[31:0]**：选项字节密钥 (Option byte key)

要将 Flash 选项寄存器解锁以允许执行选项字节编程/擦除操作，必须按顺序写入以下键值：

KEY1：0x0819 2A3B

KEY2：0x4C5D 6E7F

3.7.4 Flash 状态寄存器 (FLASH_SR)

FLASH status register

偏移地址：0x010

复位值：0x000X 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFGBSY	Res.	BSY1
													r		r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTV ERR	RD ERR	Res.	Res.	Res.	Res.	FAST ERR	MISS ERR	PGS ERR	SIZ ERR	PGA ERR	WRP ERR	PROG ERR	Res.	OP ERR	EOP
rc_w1	rc_w1					rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1

位 31:19 保留，必须保持复位值。

位 18 **CFGBSY**：编程或擦除配置繁忙 (Programming or erase configuration busy)

该标志由硬件置 1 和清零（当 Flash 编程或擦除操作完成或因错误而结束后，发送第一个字时置 1，发送第二个字时清零）。

置 1 时，*FLASH 控制寄存器 (FLASH_CR)* 中的 PG、PNB[3:0]、PER 和 MER1 位域的编程和擦除设置正在使用（繁忙），不能更改（正在进行编程或擦除操作）。

清零时，*FLASH 控制寄存器 (FLASH_CR)* 中的 PG、PNB[3:0]、PER 和 MER1 位域的编程和擦除设置可以修改。

位 17 保留，必须保持复位值。

位 16 **BSY1**：繁忙 (Busy)

该标志指示 *FLASH 控制寄存器 (FLASH_CR)* 请求的 Flash 操作正在进行。该位在 Flash 操作开始时置 1，在操作结束或出现错误时清零。

位 15 **OPTVERR**：选项和工程位加载有效性错误 (Option and Engineering bits loading validity error)

读取的选项和工程位可能不是用户配置或生产过程中配置的位时，该位由硬件置 1。如果选项和工程位尚未正确加载，则在每次系统复位后，OPTVERR 将再次置 1。加载失败的选项字节将被强制设为安全值，请参见*第 3.4.2 节：FLASH 选项字节编程*。

写入 1 即可将该位清零。

位 14 **RDERR**：PCROP 读取错误 (PCROP read error)

要读取的地址属于 Flash 的读保护区域（PCROP 保护）时，该位由硬件置 1。如果 FLASH_CR 中的 RDERRIE 置 1，将生成中断。

写入 1 即可将该位清零。

位 13:10 保留，必须保持复位值。

位 9 **FASTERR**：快速编程错误 (Fast programming error)

因错误（对齐、大小、写保护或数据丢失）导致快速编程顺序（由 FSTPG 激活）中断时，该位由硬件置 1。同时，相应状态位（PGAERR、SIZERR、WRPERR 或 MISSERR）也会置 1。

写入 1 即可将该位清零。

位 8 **MISSERR**：快速编程数据丢失错误 (Fast programming data miss error)

在快速编程模式下，必须将 16 个双字（128 字节）连续发送到 Flash，并且必须在当前数据完全编程之前将新数据发送到逻辑控制。如果新数据未及时出现，则 MISSERR 由硬件置 1。

写入 1 即可将该位清零。

位 7 **PGSERR**：编程顺序错误 (Programming sequence error)

如果代码在 PG 或 FSTPG 先前未置 1 的情况下对 Flash 执行写访问，则该位由硬件置 1。由于之前的编程错误而导致 PROGERR、SIZERR、PGAERR、WRPERR、MISSERR 或 FASTERR 置 1 时，该位也由硬件置 1。

写入 1 即可将该位清零。

位 6 **SIZERR**：大小错误 (Size error)

在编程或快速编程顺序中，访问大小为字节或半字时，该位由硬件置 1。只允许双字编程（即按字访问）。

写入 1 即可将该位清零。

位 5 **PGAERR**：编程对齐错误 (Programming alignment error)

如果在标准编程期间要编程的数据无法包含在同一双字（64 位）Flash 中，或快速编程时存在页面更改，该位将由硬件置 1。

写入 1 即可将该位清零。

位 4 **WRPERR**：写保护错误 (Write protection error)

如果要擦除/编程的地址属于 Flash 中受写保护（受 WRP、PCROP 或 RDP 等级 1 保护）的区域，则该位由硬件置 1。

写入 1 即可将该位清零。

位 3 **PROGERR**：编程错误 (Programming error)

编程前，要编程的双字地址包含不同于“0xFFFF FFFF”的值时，该位由硬件置 1（要写入的数据为“0x0000 0000”时除外）。

写入 1 即可将该位清零。

位 2 保留，必须保持复位值。

位 1 **OPERR**：操作错误 (Operation error)

当 Flash 操作（编程/擦除）失败时，该位由硬件置 1。

只有在使能错误中断 (ERRIE=1) 后，该位才会置 1。

写入“1”即可将该位清零。

位 0 **EOP**：操作结束 (End of operation)

当成功完成一个或多个 Flash 操作（编程/擦除）时，该位由硬件置 1。

只有在使能操作结束中断 (EOPIE=1) 后，该位才会置 1。

写入 1 即可将该位清零。

3.7.5 FLASH 控制寄存器 (FLASH_CR)

FLASH control register

偏移地址：0x014

复位值：0xC000 0000

访问：当前未执行任何 Flash 操作时无等待周期；按字、半字和字节访问

Flash 状态寄存器 (FLASH_SR) 中的 CFGBSY 置 1 时，无法修改该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OPT LOCK	Res.	SEC_ PROT	OBL_ LAUNCH	RD ERRIE	ERRIE	EOPIE	Res.	Res.	Res.	Res.	Res.	FSTPG	OPT STRT	STRT
rs	rs		rw	rc_w1	rw	rw	rw						rw	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PNB[3:0]				MER1	PER	PG
									rw	rw	rw	rw	rw	rw	rw

位 31 **LOCK**：FLASH_CR 锁定 (FLASH_CR Lock)

该位只置 1。置 1 后，FLASH_CR 寄存器被锁定。当检测到解锁序列时，由硬件将该位清零。

如果解锁操作失败，该位仍保持置 1，直到下一次系统复位。

位 30 **OPTLOCK**：选项锁定 (Option lock)

该位只置 1。置 1 后，FLASH_CR 寄存器中所有与用户选项相关的位以及选项页都被锁定。当检测到解锁序列时，由硬件将该位清零。对 OPTLOCK 位执行解锁序列之前，必须将 LOCK 位清零。

如果解锁操作失败，该位仍保持置 1，直到下一次复位。

位 29 保留，必须保持复位值。

位 28 **SEC_PROT**：受控安全存储区保护使能 (Securable memory area protection enable)
该位使能对受控安全区域的保护，前提是在选项字节中定义了非空受控安全存储区大小 (SEC_SIZE[4:0])。
0：禁止（受控安全区域可访问）
1：使能（受控安全区域不可访问）
该位只能由软件置 1 且只能通过系统复位清零。

位 27 **OBL_LAUNCH**：选项字节加载启动 (Option byte load launch)
该位置 1 时可触发选项字节加载到选项寄存器中。加载完成后，该位自动清零。该位的高电平状态指示选项字节加载操作挂起。
该位不能由软件清零，并且只要 OPTLOCK 置 1 就无法写入。

位 26 **RDERRIE**：PCROP 读取错误中断使能 (PCROP read error interrupt enable)
当 FLASH_SR 寄存器中的 RDERR 标志置 1 后，可通过该位使能中断产生功能。
0：禁止
1：使能

位 25 **ERRIE**：错误中断使能 (Error interrupt enable)
当 FLASH_SR 寄存器中的 OPERR 标志置 1 后，可通过该位使能中断产生功能。
0：禁止
1：使能

位 24 **EOPIE**：操作结束中断使能 (End-of-operation interrupt enable)
当 FLASH_SR 寄存器中的 EOP 标志置 1 后，可通过该位使能中断产生功能。
0：禁止
1：使能

位 23:19 保留，必须保持复位值。

位 18 **FSTPG**：快速编程使能 (Fast programming enable)
0：禁止
1：使能

位 17 **OPTSTRT**：开始修改选项字节 (Start of modification of option bytes)
该位置 1 后可触发选项操作。
该位只能由软件置 1，并在 FLASH_SR 中的 BSY1 位清零后随之清零。

位 16 **STRT**：开始擦除操作 (Start erase operation)
该位置 1 后可触发擦除操作。
该位只能由软件置 1 且只能由硬件清零。当 FLASH_SR 寄存器中的 BSY1 和 BSY2 标志之一切换为 0 时，该位由硬件清零。

位 15:7 保留，必须保持复位值。

位 6:3 **PNB[3:0]**：页编号选择 (Page number selection)
这些位用于选择要擦除的页：
0x0：页 0
0x1：页 1
...
0xF：页 15
注意：不支持主存储器之外的地址所对应的值。

位 2 **MER1**：批量擦除 (Mass erase)
该位置 1 时可触发批量擦除，即擦除所有用户页。

位 1 **PER**：页擦除使能 (Page erase enable)
0：禁止
1：使能

位 0 **PG**：Flash 编程使能 (Flash memory programming enable)
0：禁止
1：使能

3.7.6 FLASH 选项寄存器 (FLASH_OPTR)

FLASH option register

偏移地址：0x020

复位值：0xXXXX XXXX（上电复位信号释放时，Flash 中的值将加载到这些选项位。）

访问：当前未执行任何 Flash 操作时无等待周期；按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	IRHEN	NRST_MODE [1:0]		n BOOT0	n BOOT1	nBOOT _SEL	SECUR E_MUX ING_E N	RAM_P ARITY_ CHECK	HSE_N OT_RE MAPPE D	Res.	WWDG _SW	IWGD _STDBY	IWDG _STOP	IWDG _SW
		rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nRST_ SHDW	nRST_ STDBY	nRST_ STOP	BORF_LEV[1:0]		BORR_LEV[1:0]		BOR_ EN	RDP[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29 **IRHEN**：内部复位保持信号使能位 (Internal reset holder enable bit)
0：内部复位以简单脉冲的形式在 NRST 引脚上传播
1：内部复位将 NRST 引脚驱动为低电平，直到检测到该引脚为低电平

位 28:27 **NRST_MODE[1:0]**：NRST 引脚配置 (NRST pin configuration)
00：保留
01：仅复位输入：NRST 引脚为低电平时会产生系统复位；内部复位不会传播到 NRST 引脚。
10：标准 GPIO：仅支持内部复位
11：双向复位：NRST 引脚配置为复位输入/输出（传统）模式

位 26 **nBOOT0**：nBOOT0 选项位 (nBOOT0 option bit)
0：nBOOT0 = 0
1：nBOOT0 = 1

位 25 **nBOOT1**：自举配置 (Boot configuration)
该位与 BOOT0 引脚或选项位 nBOOT0（具体取决于 nBOOT_SEL 选项位配置）搭配使用，用于选择从主 Flash、SRAM 或系统存储器自举模式。请参见*第 2.5 节：自举配置*。

位 24 **nBOOT_SEL**：BOOT0 信号源选择 (BOOT0 signal source selection)
该选项位用于定义 BOOT0 信号源。
0：BOOT0 引脚（传统模式）
1：nBOOT0 选项位

位 23 **SECURE_MUXING_EN**：多连接安全性 (Multiple-bonding security)

该位可使能自动 I/O 配置，以防止同一引脚上连接的多个 I/O 发生冲突。

0：禁止

1：使能

如果通过软件配置 SYSCFG_CFGR3 寄存器将同一引脚连接的 I/O 之一设置为有效状态，则使能该位将自动强制其他 I/O 设为数字输入模式，而与其软件配置无关。

禁止该位时，SYSCFG_CFGR3 寄存器设置会被忽略，给定引脚上连接的所有 GPIO 均为有效状态，并且可设置为相应 GPIOx_MODER 寄存器指定的模式。用户软件必须确保各 GPIO 之间不存在冲突。

位 22 **RAM_PARITY_CHECK**：SRAM 奇偶校验控制使能/禁止 (SRAM parity check control enable/disable)

0：使能

1：禁止

位 21 **HSE_NOT_REMAPPED**：HSE 重映射使能/禁止 (HSE remapping enable/disable)

该位清零时会将 HSE 时钟源从 PF0-OSC_IN/PF1-OSC_OUT 引脚映射到 PC14-OSCX_IN/PC15-OSCX_OUT。因此，PC14-OSCX_IN/PC15-OSCX_OUT 由 LSE 和 HSE 共用，但这两个时钟源无法同时使用。

0：使能

1：禁止

对于采用少于 48 引脚封装的器件，重映射始终使能（PF0-OSC_IN/PF1-OSC_OUT 不可用），而与该位无关。由于所有 STM32C011xx 封装的引脚数均少于 48 个，因此该位仅适用于 STM32C031xx。

注意：对于采用 48 引脚封装的器件，当 HSE_NOT_REMAPPED 复位时，HSE 无法在旁路模式下使用。更多详细信息，请参见产品勘误表。

位 20 保留，必须保持复位值。

位 19 **WWDG_SW**：窗口看门狗选择 (Window watchdog selection)

0：硬件窗口看门狗

1：软件窗口看门狗

位 18 **IWDG_STDBY**：在待机模式下冻结独立看门狗计数器 (Independent watchdog counter freeze in Standby mode)

0：在待机模式下冻结独立看门狗计数器

1：在待机模式下运行独立看门狗计数器

位 17 **IWDG_STOP**：在停止模式下冻结独立看门狗计数器 (Independent watchdog counter freeze in Stop mode)

0：在停止模式下冻结独立看门狗计数器

1：在停止模式下运行独立看门狗计数器

位 16 **IDWG_SW**：独立看门狗选择 (Independent watchdog selection)

0：硬件独立看门狗

1：软件独立看门狗

位 15 **nRSTS_SHDW**

0：进入关断模式时产生复位

1：进入关断模式时不产生复位

位 14 **nRST_STDBY**

0：进入待机模式时产生复位

1：进入待机模式时不产生复位

位 13 **nRST_STOP**

0：进入停止模式时产生复位

1：进入停止模式时不产生复位

位 12:11 **BORF_LEV[1:0]**：V_{DD} 电源下降时的 BOR 阈值 (BOR threshold at falling V_{DD} supply)
V_{DD} 降至该阈值以下时会激活复位信号。
00：BOR 下降级别 1：阈值约为 2.0 V
01：BOR 下降级别 2：阈值约为 2.2 V
10：BOR 下降级别 3：阈值约为 2.5 V
11：BOR 下降级别 4：阈值约为 2.8 V

位 10:9 **BORR_LEV[1:0]**：V_{DD} 电源上升时的 BOR 阈值 (BOR threshold at rising V_{DD} supply)
V_{DD} 升至该阈值以上时会释放复位信号。
00：BOR 上升级别 1：阈值约为 2.1 V
01：BOR 上升级别 2：阈值约为 2.3 V
10：BOR 上升级别 3：阈值约为 2.6 V
11：BOR 上升级别 4：阈值约为 2.9 V

位 8 **BOR_EN**：欠压复位使能 (Brown out reset enable)
0：禁止可配置欠压复位，通过 POR/PDR 电压定义上电复位
1：使能可配置欠压复位，考虑 BORR_LEV 和 BORF_LEV 的值

位 7:0 **RDP[7:0]**：读保护级别 (Read protection level)
0xAA：级别 0，未激活读保护
0xCC：级别 2，激活芯片读保护
其他值：级别 1，激活存储器读保护

3.7.7 FLASH PCROP 区域 A 起始地址寄存器 (FLASH_PCROP1ASR)

FLASH PCROP area A start address register

偏移地址：0x024

复位值：0b0000 0000 0000 0000 0000 0000 00XX XXXX（上电复位信号释放时，Flash 中的值将加载到这些选项位。）

访问：当前未执行任何 Flash 操作时无等待周期；按字、半字访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1A_STRT[5:0]					
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持清零

位 5:0 **PCROP1A_STRT[5:0]**：PCROP1A 区域起始偏移 (PCROP1A area start offset)
包含 PCROP1A 区域的第一个子页的偏移。
注意：有效位数取决于器件中的 Flash 的大小。

3.7.8 FLASH PCROP 区域 A 结束地址寄存器 (FLASH_PCROP1AER)

FLASH PCROP area A end address register

偏移地址：0x028

复位值：0bX000 0000 0000 0000 0000 0000 00XX XXXX（上电复位信号释放时，Flash 中的值将加载到这些选项位。）

访问：当前未执行任何 Flash 操作时无等待周期；按字、半字访问。可通过字节访问的形式访问 PCROP_RDP 位

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PCROP_RDP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1A_END[5:0]					
										rw	rw	rw	rw	rw	rw

位 31 **PCROP_RDP**：RDP 降级后的 PCROP 区域擦除 (PCROP area erase upon RDP level regression)

该位确定 RDP 从级别 1 降为级别 0 所触发的批量擦除是否擦除了 PCROP 区域（以及全部 PCROP 区域边界页）：

0：未擦除

1：已擦除

软件只能将该位置 1。在 RDP 从级别 1 降为级别 0 后的批量擦除后，该位自动复位。

位 30:6 保留，必须保持清零

位 5:0 **PCROP1A_END[5:0]**：PCROP1A 区域结束偏移 (PCROP1A area end offset)

包含 PCROP1A 区域的最后一个子页的偏移。

注意：有效位数取决于器件中的 Flash 的大小。

3.7.9 FLASH WRP 区域 A 地址寄存器 (FLASH_WRP1AR)

FLASH WRP area A address register

偏移地址：0x02C

复位值：0x000X 000X（上电复位信号释放时，Flash 中的值将加载到这些选项位。）

访问：当前未执行任何 Flash 操作时无等待周期；按字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_END[3:0]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_STRT[3:0]			
												rw	rw	rw	rw

位 31:20 保留，必须保持复位值。

位 19:16 **WRP1A_END[3:0]**：WRP 区域 A 结束偏移 (WRP area A end offset)

该位域包含 WRP 区域 A 的最后一页的偏移。

注意：有效位数取决于器件中的 Flash 的大小。

位 15:4 保留，必须保持复位值。

位 3:0 **WRP1A_STRT[3:0]**：WRP 区域 A 起始偏移 (WRP area A start offset)
该位域包含 WRP 区域 A 的第一页的偏移。
注意：有效位数取决于器件中的 Flash 的大小。

3.7.10 FLASH WRP 区域 B 地址寄存器 (FLASH_WRP1BR)

FLASH WRP area B address register

偏移地址：0x030

复位值：0x000X 000X（上电复位信号释放时，Flash 中的值将加载到这些选项位。）

访问：当前未执行任何 Flash 操作时无等待周期；按字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_END[3:0]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_STRT[3:0]			
												rw	rw	rw	rw

位 31:20 保留，必须保持复位值。

位 19:16 **WRP1B_END[3:0]**：WRP 区域 B 结束偏移 (WRP area B end offset)
该位域包含 WRP 区域 B 的最后一页的偏移。
注意：有效位数取决于器件中的 Flash 的大小。

位 15:4 保留，必须保持复位值。

位 3:0 **WRP1B_STRT[3:0]**：WRP 区域 B 起始偏移 (WRP area B start offset)
该位域包含 WRP 区域 B 的第一页的偏移。
注意：有效位数取决于器件中的 Flash 的大小。

3.7.11 FLASH PCROP 区域 B 起始地址寄存器 (FLASH_PCROP1BSR)

FLASH PCROP area B start address register

偏移地址：0x034

复位值：0b0000 0000 0000 0000 0000 0000 00XX XXXX（上电复位信号释放时，Flash 中的值将加载到这些选项位。）

访问：当前未执行任何 Flash 操作时无等待周期；按字、半字访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1B_STRT[5:0]					
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持清零

位 5:0 **PCROP1B_STRT[5:0]**：PCROP1B 区域起始偏移 (PCROP1B area start offset)
包含 PCROP1B 区域的第一个子页的偏移。
注意：有效位数取决于器件中的 Flash 的大小。

3.7.12 FLASH PCROP 区域 B 结束地址寄存器 (FLASH_PCROP1BER)

FLASH PCROP area B end address register

偏移地址：0x038

复位值：0b0000 0000 0000 0000 0000 0000 00XX XXXX（上电复位信号释放时，Flash 中的值将加载到这些选项位。）

访问：当前未执行任何 Flash 操作时无等待周期；按字、半字访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1B_END[5:0]					
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持清零

位 5:0 **PCROP1B_END[5:0]**：PCROP1B 区域结束偏移 (PCROP1B area end offset)
包含 PCROP1B 区域的最后一个子页的偏移。
注意：有效位数取决于器件中的 Flash 的大小。

3.7.13 FLASH 安全寄存器 (FLASH_SECR)

FLASH security register

偏移地址：0x080

复位值：0b0000 0000 0000 000X 0000 0000 000X XXXX（上电复位信号释放时，Flash 中的值将加载到这些选项位。）

访问：当前未执行任何 Flash 操作时无等待周期；按字、半字访问

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

位 31:17 保留，必须保持复位值

位 16 **BOOT_LOCK**：用于强制从用户区域自举
0：基于引脚/选项位配置自举
1：强制从主 Flash 自举

小心： 如果该位随 RDP 级别 1 置 1，则会禁止调试功能，但出现不良 OBL（不匹配）时除外。

位 15:5 保留，必须保持复位值

位 4:0 **SEC_SIZE[4:0]**：受控安全存储区大小 (Securable memory area size)
包含受控安全 Flash 页数。
注意：有效位数取决于器件中的 Flash 的大小。

3.7.14 FLASH 寄存器映射

表 18. FLASH 寄存器映射与复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	FLASH_ACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_SWEN	Res.	EMPTY	Res.	Res.	Res.	Res.	ICRST	Res.	ICEN	PRFTEN	Res.	Res.	Res.	Res.	Res.	LATENCY [2:0]		
	Reset value														1		X					0		1	0						0	0	0
0x004	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x008	FLASH_KEYR	KEYR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	FLASH_OPT KEYR	OPTKEY[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	FLASH_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFGBSY	Res.	BSY1	OPTVERR	RDERR	Res.	Res.	Res.	Res.	FASTERR	MISERR	PGSERR	SIZERR	PGAERR	WRPERR	PROGERR	Res.	OPERR	EOP
	Reset value														0		0	X	0					0	0	0	0	0	0	0		0	0
0x014	FLASH_CR	LOCK	OPTLOCK	Res.	SEC_PROT	OBL_LAUNCH	RDERRIE	ERRIE	EOPIE	Res.	Res.	Res.	Res.	Res.	FSTPG	OPTSTRT	STRT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PNB[3:0]				MER1	PER	PG
	Reset value	1	1		0	0	0	0	0						0	0	0										0	0	0	0	0	0	0
0x018	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x020	FLASH_OPTR	Res.	Res.	IRHEN	NRST_MODE[1:0]		nBOOT0	nBOOT1	nBOOT_SEL	SECURE_MUXING_EN	RAM_PARITY_CHECK	HSE_NOT_REMAPPED	Res.	WWDG_SW	IWDG_STBY	IWDG_STOP	IWDG_SW	nRST_SHDW	nRST_STDBY	nRST_STOP	BORF_LEV[1:0]		BORR_LEV[1:0]		BOR_EN	RDP[7:0]							
	Reset value			X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x024	FLASH_ PCROP1ASR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1A_STRT [5:0]					
	Reset value																											X	X	X	X	X	X
0x028	FLASH_ PCROP1AER	PCROP_RDP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1A_END[5:0]					
	Reset value	X																										X	X	X	X	X	X
0x02C	FLASH_ WRP1AR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_END [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A _STRT[3:0]			
	Reset value													X	X	X	X													X	X	X	X
0x030	FLASH_ WRP1BR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_END [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B _STRT[3:0]			
	Reset value													X	X	X	X													X	X	X	X
0x034	FLASH_ PCROP1BSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1B_STRT [5:0]					
	Reset value																											X	X	X	X	X	X
0x038	FLASH_ PCROP1BER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1B_END[5:0]					
	Reset value																											X	X	X	X	X	X
0x03C - 0x07F	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 18. FLASH 寄存器映射与复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x080	FLASH_SECR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOOT_LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEC_SIZE[4:0]				
	Reset value																X												X	X	X	X	X

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

4 电源控制 (PWR)

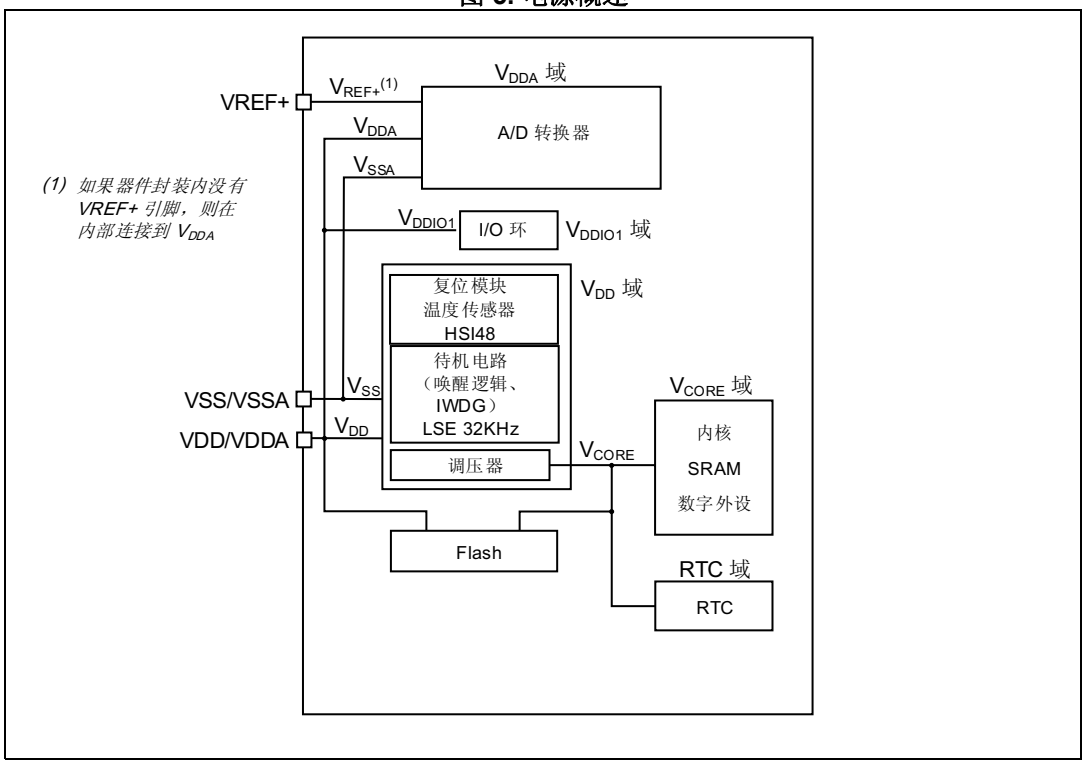
4.1 电源和参考电压

器件通过 VDD/VDDA 引脚供电。在器件内部，V_{DD} 电源域的内部资源以及 Flash 通过 V_{DD} 电源线路供电，ADC 通过 V_{DDA} 电源线路供电，I/O 环通过 V_{DDIO1} 电源线路供电。

调压器通过 V_{CORE} 电源线路为 V_{CORE} 电源域的资源（包括内核、SRAM 和数字外设）、RTC 域以及 Flash 供电。

更多信息，请参见下图以及器件数据手册。

图 5. 电源概述



4.1.1 ADC 参考电压

V_{REF+} 定义了全量程 ADC 输入信号电平。对于采用 VREF+ 输入引脚的封装，V_{REF+} 参考电压由外部提供，这样可以提高 ADC 分辨率（单位步长对应的电压）并降低噪声。关于允许的 V_{REF+} 范围，请参见器件数据手册。

4.1.2 调压器

使能调压器后将为 V_{CORE} 域提供 V_{CORE} 电源电压。

调压器在器件复位后即会使能，并且只要器件处于运行、睡眠或停止模式，调压器就会一直保持使能状态。

在待机和关断模式下，调压器处于禁止状态，不会为 V_{CORE} 域供电。因此，SRAM 和寄存器的内容会丢失。

4.2 电源监控器

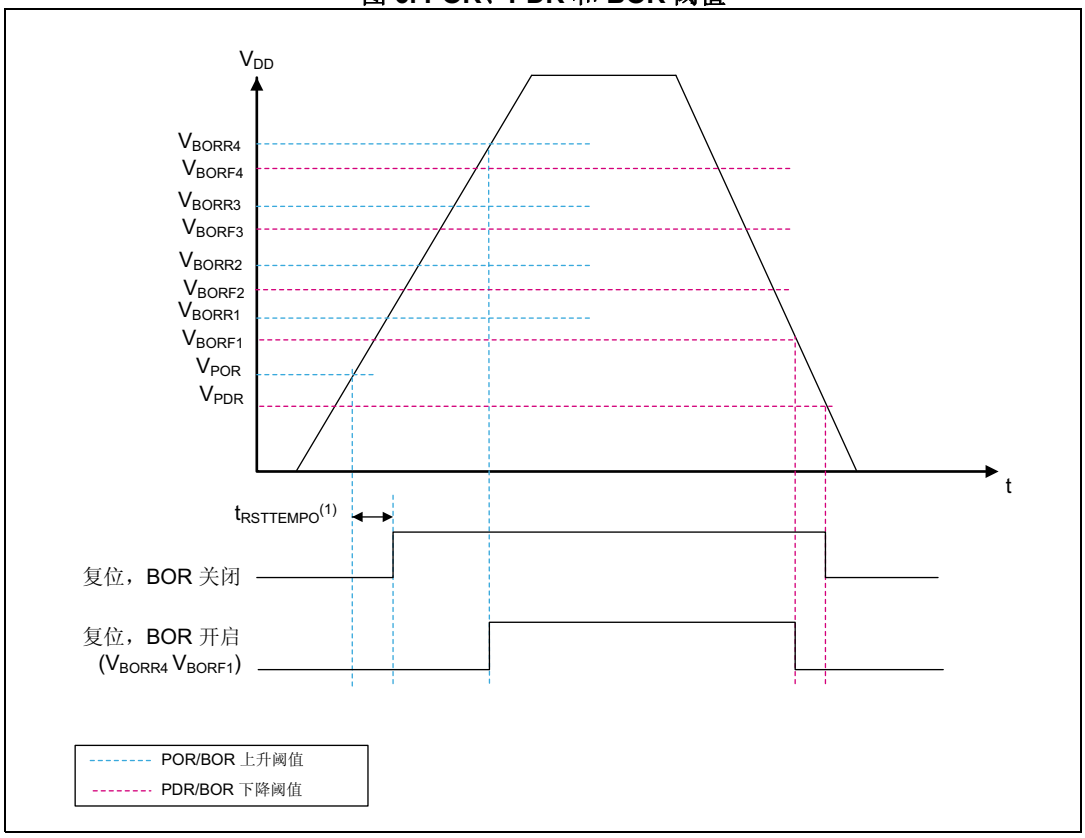
4.2.1 上电复位 (POR)/掉电复位 (PDR)/欠压复位 (BOR)

该器件具有一个集成的上电复位 (POR)/掉电复位 (PDR)，外加欠压复位 (BOR) 电路。POR/PDR 在所有电源模式下都处于工作状态。BOR 只能通过设置选项字节来使能或禁止，并且在关断模式下不可用。

使能 BOR 后，可以通过选项字节来选择四种 BOR 级别，并且可以单独配置上升阈值和下降阈值。上电期间，BOR 将使器件保持复位状态，直到 V_{DD} 电源电压达到指定的 BOR 上升阈值 (V_{BORRx})。达到时，器件将退出复位状态，随后系统便可启动。在掉电期间，当 V_{DD} 降至选定的 BOR 下降阈值 (V_{BORFx}) 以下时，器件将再次进入复位状态。

警告： BOR 下降阈值 (V_{BORFx}) 不得高于 BOR 上升阈值 (V_{BORRx})。

图 6. POR、PDR 和 BOR 阈值



1. 复位持续时间 $t_{RSTTEMPO}$ 从 V_{DD} 超过 V_{POR} 阈值时开始，与 BOR 选项位的配置无关。

有关欠压复位阈值的更多详细信息，请参见数据手册的电气特性部分。

4.3 工作模式

器件具有一种完全工作模式（称为运行模式），以及多种可大幅节能的低功耗模式。

系统复位或上电复位后，器件首先进入运行模式。

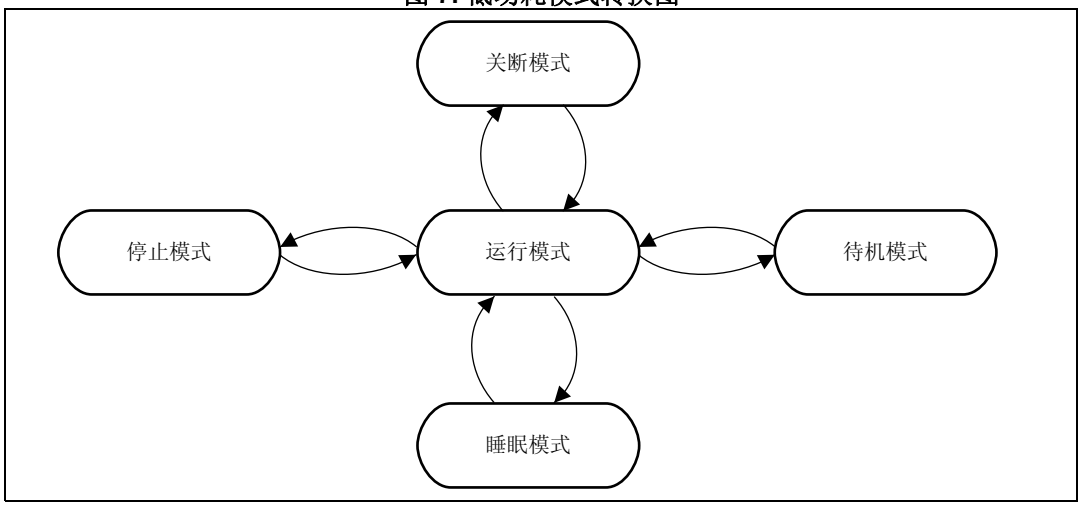
当不需要 CPU 执行指令（例如，等待外部事件）时，可以选择进入睡眠、停止、待机和关断等低功耗模式来降低功耗。

不同的低功耗模式在功耗、启动速度和唤醒能力方面提供不同的权衡。在睡眠模式下，唤醒速度最快，但节能效果最差。在关断模式下，功耗最低，但唤醒速度较慢，而且不支持电源监测功能和 BOR/PDR。在待机模式下，LSI 振荡器、IWDG 和电压监测功能可保持工作状态，但功耗略高。在停止模式下，低速时钟和外设以及 GPIO 可保持工作状态。

有关各种工作模式的详细信息，请参见*表 19：不同工作模式下使能的器件资源*和*表 21：退出低功耗模式概述*。

器件可以在运行模式与任何低功耗模式之间自由转换。低功耗模式之间无法转换。请参见下图。

图 7. 低功耗模式转换图



下表概述了各种工作模式下可用的器件资源，以及它们能否将器件从低功耗模式唤醒并进入运行模式。表格下方的脚注给出了补充信息。

表 19. 不同工作模式下使能的器件资源

功能	工作模式(1)							
	运行	睡眠	停止(2)		待机(2)		关断(2)	
CPU	Y	-	-	-	-	-	-	-
Flash	Y	Y	A(3)	-	-	-	-	-
SRAM	Y	Y	U	-	-	-	-	-
V_{CORE} 供电(4)	Y	Y	Y	-	-	-	-	-
BOR/POR/PDR	O	O	U	U	U	U	-(5)	-
NRST	Y	Y	Y	Y	Y	Y	Y	Y
DMA/DMAMUX	O	U	-	-	-	-	-	-

表 19. 不同工作模式下使能的器件资源（续）

功能	工作模式[1]							
	运行	睡眠	停止[2]		待机[2]		关断[2]	
HSI48	Y	U[6]	-[6]	-	-	-	-	-
HSE	O	U	-	-	-	-	-	-
LSI	O	U	U	-	U	-	-	-
LSE	O	U	U	-	-	-	-	-
CSS	O	U	-	-	-	-	-	-
LSE 上的 CSS	O	U	U	O	-	-	-	-
RTC/自动唤醒	O	U	U	O	-	-	-	-
USART1	O	U[7]	U[7]	O[7]	-	-	-	-
USART2	O	U[7]	-	-	-	-	-	-
I2C1	O	U[8]	U[8]	O[8]	-	-	-	-
SPI1/I2S1	O	U	-	-	-	-	-	-
ADC	O	U	-[9]	-	-	-	-	-
温度传感器	O	U	-[9]	-	-	-	-	-
TIMx	O	U	-	-	-	-	-	-
IWDG	O	U	U	O	U	O	-	-
WWDG	O	U	-	-	-	-	-	-
SysTick 定时器	O	U	-	-	-	-	-	-
CRC	O	U	-	-	-	-	-	-
GPIO	O	U	U	O	(10)	(11)	(10)	(11)
各个外设时钟	O	A[12]	A[12]	-	-	-	-	-

1. **Y** =（是）：该资源在复位后以及从停止、待机和关机模式唤醒后使能；**O** =（可选）：该资源默认禁止，可通过软件使能；**U** =（不变）：该资源保持在进入低功耗模式前的工作状态；**A** =（自动）：该资源可在转换到低功耗模式后自动禁止/掉电；**-** = 该资源不可用/没有唤醒能力（灰色列）。
2. 灰色列表示资源将器件从低功耗模式唤醒的能力。
3. 可自动掉电。
4. 不可用（禁止调压器）时，SRAM 和寄存器的内容会丢失。
5. 禁止电源电压监测，并且电源电压下降时不能保证产品特性。
6. 当 HSI48 关闭时，具有唤醒能力的外设（如 I2C1）可将 HSI48 振荡器暂时唤醒以评估器件是否满足唤醒条件（如 I²C 地址匹配），满足则唤醒 HSI48，不满足则关闭 HSI48。在低功耗模式期间，当 HSI48 由具有唤醒能力的外设重启后，HSI48 只会为该外设提供时钟。
7. USART 接收功能在睡眠模式和停止模式下正常工作。该功能会在检测到 START 条件、发生地址匹配或接收到帧事件时产生唤醒中断。当发生地址匹配或接收到帧事件时，该外设可重启 HSI48 振荡器以评估器件是否满足唤醒条件。
8. I²C 地址检测功能在睡眠模式和停止模式下正常工作。该功能会在发生地址匹配时产生唤醒中断。当发生地址匹配事件时，该外设可重启 HSI48 振荡器以评估器件是否满足唤醒条件。
9. 除非在进入停止模式前禁止，否则会一直消耗空闲电流。
10. 在待机和关断模式下，I/O 可保持内部上拉、内部下拉或浮空配置。请参见 PUCRx 和 PDCRx 寄存器以及 PWR_CR3 寄存器中的 APC 位。退出待机模式时会保持该配置，但退出关断模式时会丢失该配置。
11. WKUPx I/O（最多五个，具体取决于封装）,具备将器件从待机/关断模式唤醒的能力。
12. 通过 RCC_AHBSMENR 和 RCC_APBSMENRx 寄存器，软件可以将各个外设时钟配置为在进入睡眠和停止模式后自动禁止。

4.3.1 运行模式下的节能操作

在运行模式下，可以通过多种操作来降低功耗，具体包括选择一个频率较低的系统时钟、降低系统时钟的频率、禁止未使用的外设并/或停止其时钟（外设时钟门控）。

降低系统时钟速度

通过预分频器寄存器控制预分频器，可以降低 SYSCLK、HCLK 和 PCLK 时钟的频率。这些寄存器设置同样适用于睡眠模式。

有关详细信息，请参见*第 5.4.3 节：RCC 时钟配置寄存器 (RCC_CFGR)*。

外设时钟门控

可随时停止（时钟门控）各外设和存储器的 HCLK 和 PCLK 以降低功耗。

RCC_AHBENR 和 RCC_APBENRx 寄存器用于使能或禁止各个外设时钟（时钟门控）。

4.3.2 低功耗模式

器件有以下低功耗模式：

- **睡眠模式**
  CPU 时钟关闭，包括 Cortex®-M0+ 内核外设（例如 NVIC 和 SysTick）在内的所有外设都可以运行，并在发生中断或事件时唤醒 CPU。
  要进一步降低睡眠模式的功耗，可在执行 WFI 或 WFE 指令之前禁止外设时钟。该操作也可以通过配置 RCC_AHBSMENR 和 RCC_APBSMENRx 寄存器来自动完成。
  如果 HSI48 在进入睡眠模式前处于禁止状态，则可以通过具备唤醒能力的外设（支持 HSI48）重启它。
- **停止模式**
  SRAM 和寄存器内容将保留。HSE 和 HSI48 停止工作。可以通过具备唤醒能力的外设（支持 HSI48）重启 HSI48。
  LSI 和 LSE 可以保持运行。
  RTC 可以保持有效（带 RTC 的停止模式，不带 RTC 的停止模式）。
  发生退出停止模式的事件时，会使能 HSI48 振荡器并选择 HSISYS 作为系统时钟。
- **待机模式**
  V_{CORE} 域掉电，SRAM 的内容以及除 *PWR 控制寄存器 3 (PWR_CR3)* 和 *PWR 备份 x 寄存器 (PWR_BKPxR)* 之外的所有寄存器的内容会丢失。
  V_{CORE} 域中的所有时钟都会停止，HSI48 和 HSE 振荡器处于禁止状态。IWDG 和 LSI 振荡器可以保持运行。
  发生退出待机模式的事件时，会使能 HSI48 振荡器，选择 HSISYS 作为系统时钟并将其预分频器的分频系数设置为 4 (HSIDIV[2:0] = 010)。
- **关断模式**
  V_{CORE} 域掉电，SRAM 和寄存器的内容会丢失。
  所有振荡器都处于禁止状态。
  发生退出关断模式的事件时，会使能 HSI48 振荡器，选择 HSISYS 作为系统时钟并将其预分频器的分频系数设置为 4 (HSIDIV[2:0] = 010)。
  在该模式下，会禁止电源电压监测，并且电源电压下降时不能保证产品特性。

低功耗模式调试

默认情况下，进入停止模式、关断模式或待机模式后不再为内核提供时钟，因此会立即断开调试连接。

不过，如果使用 DBGMCU_CR 寄存器中的特定设置，即使在低功耗模式下也可以调试软件。有关详细信息，请参见*第 26.9.1 节：对低功耗模式的调试支持*。

进入和退出低功耗模式

通过软件控制以下各项来进入、选择和退出低功耗模式：

- Cortex®-M0+ 系统控制寄存器的 SEVONPEND、SLEEPDEEP 和 SLEEPONEXIT 位
- *PWR 控制寄存器 1 (PWR_CR1)* 的 LPMS[2:0] 位域
- WFI（等待中断）和 WFE（等待事件）指令（进入低功耗模式的刺激源）
- NVIC、EXTI 和外设挂起中断与事件标志
- 配置“从 ISR（中断服务程序）返回”作为进入低功耗模式的刺激源

进入低功耗模式

出现以下任何一个刺激源时，器件将按条件进入低功耗模式：

- WFI（等待中断）指令
- WFE（等待事件）指令
- 从 ISR 返回（当 SLEEPONEXIT 位为高电平时）

在满足退出低功耗模式的条件时出现的进入低功耗模式的刺激源将被忽略（取消进入低功耗模式），程序继续执行。

对于除睡眠模式之外的低功耗模式，如果当前存在 Flash 或 APB 访问，则会延迟进入这些低功耗模式，即等到上述访问终止后再进入低功耗模式，而不会取消。

使用 SLEEPDEEP 和 LPMS[2:0] 位域选择要进入的低功耗模式。

下表（包括表格下方的脚注）详细列出了进入低功耗模式的条件。表格行中的全部条件必须同时满足，才能进入对应的低功耗模式。

表 20. 进入低功耗模式概述

低功耗模式	进入低功耗模式的刺激源	条件(1)							
		SLEEPDEEP	SLEEPONEXIT	LPMS[2:0]	中断挂起(2)	事件挂起(3)	WUFx 位置 1	Flash 访问	APB 访问
睡眠	WFI	0	-	-	Aa	-	-	-	-
	WFE	0	-	-	-	Aa	-	-	-
	从 ISR 返回	0	1	-	Aa	-	-	-	-
停止	WFI	1	-	000	Aa	-	-	Ad	Ad
	WFE	1	-	000	-	Aa	-	Ad	Ad
	从 ISR 返回	1	1	000	Aa	-	-	Ad	Ad

表 20. 进入低功耗模式概述（续）

低功耗模式	进入低功耗模式的刺激源	条件(1)							
		SLEEPDEEP	SLEEPONEXIT	LPMS[2:0]	中断挂起(2)	事件挂起(3)	WUFx 位置 1	Flash 访问	APB 访问
待机	WFI	1	-	011	Aa	-	Aa	Ad	Ad
	WFE	1	-	011	-	Aa	Aa	Ad	Ad
	从 ISR 返回	1	1	011	Aa	-	Aa	Ad	Ad
关断	WFI	1	-	1XX	Aa	-	Aa	Ad	Ad
	WFE	1	-	1XX	-	Aa	Aa	Ad	Ad
	从 ISR 返回	1	1	1XX	Aa	-	Aa	Ad	Ad

1. “**-**” = 无关或不适用。“**X**” = 无关（位值）。“**Aa**” = 不存在/无；如果存在，则取消进入低功耗模式。“**Ad**” = 不存在/无；只要存在，就延迟进入低功耗模式。
2. 任何 EXTI 线中断标志（在 *EXTI 上升沿挂起寄存器 1 (EXTI_RPR1)* 和 *EXTI 下降沿挂起寄存器 1 (EXTI_FPR1)* 中）或任何外设唤醒中断标志置 1。
3. 任何使能的 EXTI 事件。对于睡眠模式，还包括与外设中使能的中断相关联的任何外设事件。

退出低功耗模式

当器件在 NRST 引脚上检测到外部复位时，会立即退出任何低功耗模式。

此外，器件还会在 BOR/PDR 和 IWDG 复位后退出睡眠模式、停止模式和待机模式，以及在检测到 LSE 上的 CSS 后立即退出睡眠模式和停止模式。

有关退出所有其他低功耗模式的条件，请参见下表（包括表格下方的脚注）。表格行中的全部条件必须同时满足，才能退出对应的低功耗模式。

表 21. 退出低功耗模式概述

低功耗模式	进入低功耗模式的刺激源	条件(1)						唤醒延迟	退出后		
		SEVONPEND	外设事件/中断	EXTI 中断(2)	EXTI 事件(3)	NVIC IRQ 中断(4)	WUFx 位		时钟	HSIDIV [2:0](5)	SBF(6)
睡眠	WFI 或从 ISR 返回	-	E	-	-	R	-	无	与进入前一样	与进入前一样	0
		-	E	R	-	E	-				
	WFE	0	E	-	-	R	-				
		0	E	R	-	E	-				
		0	E	-	R	-	-				
		1	E	-	-	R	-				
		1	E	R	-	-	-				
		1	E	-	R	-	-				
停止	WFI 或从 ISR 返回	-	E	R	-	E	-	HSI48 起振/Flash 启动(7)	HSISYS	与进入前一样	0
	WFE	0	E	R	-	E	-				
		0	E	-	R	-	-				
		1	E	R	-	-	-				
		1	E	-	R	-	-				
待机	-	-	-	-	-	-	R	复位阶段	HSISYS	010	1
关断	-	-	-	-	-	-	R	复位阶段	HSISYS	010	0

1. “**-**”= 无关或不适用，“**E**”= 使能，“**R**”= 使能并置 1。
2. 在 *EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR1)* 中设为未屏蔽状态的任何 EXTI 线。
3. 在 *EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR1)* 中设为未屏蔽状态的任何 EXTI 线。
4. *表 40：向量表* 中已通过 NVIC_ISER 寄存器激活（请参见产品编程手册）的任何 IRQ 中断。
5. *RCC 时钟控制寄存器 (RCC_CR)* 的位域，用于控制 HSISYS 预分频。值 HSIDIV[2:0] = 010 对应于四分频。
6. *PWR 状态寄存器 1 (PWR_SR1)* 中的标志。
7. HSI48 振荡器起振时间与 Flash 启动时间的较长者（从停止模式唤醒）。

注意： *对于任何 NVIC IRQ 中断、EXTI 中断或 EXTI 事件，上表均假定分别通过 NVIC ISER、EXTI_IMR1 或 EXTI_EMR1 寄存器来激活。对于任何中断或事件，还必须在相应的外设中激活。*

某些外设会将中断请求同时发送给 NVIC 和 EXTI，并且经过配置后可以同时生成 NVIC IRQ 和 EXTI 中断。某些其他外设只会将中断信号发送给 NVIC，因此只能生成 NVIC 中断。EXTI 外设无法将系统从停止模式唤醒。

当系统从睡眠模式和停止模式唤醒后，必须将与唤醒中断/事件相关联的挂起位清零。此外，还可能需要清零相应外设中的中断标志。

从待机或关断模式唤醒后，程序将按照复位（启动引脚采样、选项字节加载和复位向量已获取等）后的方式重新执行。

从停止模式自动唤醒

RTC 可以定期将器件从停止模式唤醒，无需任何外部刺激源。为此，需通过 *RCC 控制/状态寄存器 1 (RCC_CSR1)* 的 RTCSEL[1:0] 位域选择 LSI 或 LSE 作为 RTC 时钟源。

LSI 振荡器不使用外部石英晶体时，可降低系统成本，但精度较低。LSE 振荡器使用外部石英晶体时，可提高精度，但会产生额外的成本。

要通过 RTC 闹钟从停止模式唤醒：

- 将 EXTI 线 19 配置为上升沿有效。
- 配置 RTC 以生成唤醒事件。

4.4 PWR 寄存器

外设寄存器可按半字（16 位）或字（32 位）访问。

4.4.1 PWR 控制寄存器 1 (PWR_CR1)

PWR control register 1

从待机模式唤醒后，此寄存器会复位。

偏移地址：0x00

复位值：0x0000 0208

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

位 31:6 保留，必须保持复位值。

位 5 **FPD_SLP**：睡眠模式期间 Flash 掉电

该位确定当器件进入睡眠模式时，Flash 进入掉电模式还是保持空闲模式。

0：Flash 空闲

1：Flash 掉电

位 4 保留，必须保持复位值。

位 3 **FPD_STOP**：停止模式期间 Flash 掉电
该位确定当器件进入停止模式时，Flash 进入掉电模式还是保持空闲模式。
0：Flash 空闲
1：Flash 掉电

位 2:0 **LPMS[2:0]**：低功耗模式选择
当 CPU 进入深睡眠模式时，使用这些位选择进入的低功耗模式。
000：停止模式
001：保留
010：保留
011：待机模式
1XX：关断模式

4.4.2 PWR 控制寄存器 3 (PWR_CR3)

PWR control register 3

该寄存器在退出待机模式时不会复位，而是根据 *RCC APB 外设复位寄存器 1 (RCC_APBRSTR1)* 的 PWRRST 位执行复位。

偏移地址：0x08

复位值：0x0000 8000

访问：与标准 APB 访问相比，需要三个（写访问）或两个（读访问）额外的 APB 时钟周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIWUL	Res.	Res.	Res.	Res.	APC	Res.	Res.	Res.	Res.	EWUP6	Res.	EWUP4	EWUP3	EWUP2	EWUP1
rw					rw					rw		rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **EIWUL**：使能内部唤醒线 (Enable internal wakeup line)
置 1 时，内部唤醒线上的上升沿会触发唤醒事件。
0：禁止
1：使能

位 14:11 保留，必须保持复位值。

位 10 **APC**：应用上拉和下拉配置 (Apply pull-up and pull-down configuration)
该位确定是否应用 PWR_PUCRx 和 PWR_PDCRx 寄存器中定义的 I/O 上拉和下拉配置。
0：不应用
1：应用

位 9:6 保留，必须保持复位值。

位 5 **EWUP6**：使能 WKUP6 唤醒引脚 (Enable WKUP6 wakeup pin)
该位置 1 会使能 WKUP6 外部唤醒引脚，并在出现上升沿或下降沿时触发唤醒事件。通过 PWR_CR4 寄存器中的 WP6 位配置有效边沿。

位 4 保留，必须保持复位值。

位 3 **EWUP4**：使能 WKUP4 唤醒引脚 (Enable WKUP4 wakeup pin)
该位置 1 会使能 WKUP4 外部唤醒引脚，并在出现上升沿或下降沿时触发唤醒事件。通过 PWR_CR4 寄存器中的 WP4 位配置有效边沿。

位 2 **EWUP3**：使能 WKUP3 唤醒引脚 (Enable WKUP3 wakeup pin)
该位置 1 会使能 WKUP3 外部唤醒引脚，并在出现上升沿或下降沿时触发唤醒事件。通过 PWR_CR4 寄存器中的 WP3 位配置有效边沿。

位 1 **EWUP2**：使能 WKUP2 唤醒引脚 (Enable WKUP2 wakeup pin)
该位置 1 会使能 WKUP2 外部唤醒引脚，并在出现上升沿或下降沿时触发唤醒事件。通过 PWR_CR4 寄存器中的 WP2 位配置有效边沿。

位 0 **EWUP1**：使能 WKUP1 唤醒引脚 (Enable WKUP1 wakeup pin)
该位置 1 会使能 WKUP1 外部唤醒引脚，并在出现上升沿或下降沿时触发唤醒事件。通过 PWR_CR4 寄存器中的 WP1 位配置有效边沿。

4.4.3 PWR 控制寄存器 4 (PWR_CR4)

PWR control register 4

该寄存器在退出待机模式时不会复位，而是根据 *RCC APB 外设复位寄存器 1 (RCC_APBRSTR1)* 的 PWRRST 位执行复位。

偏移地址：0x0C

复位值：0x0000 0000

访问：与标准 APB 访问相比，需要三个（写访问）或两个（读访问）额外的 APB 时钟周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WP6	Res.	WP4	WP3	WP2	WP1
										rw		rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5 **WP6**：WKUP6 唤醒引脚极性 (WKUP6 wakeup pin polarity)
用于产生唤醒条件的 WKUP6 外部唤醒信号极性（电平或边沿）：
0：高电平或上升沿
1：低电平或下降沿

位 4 保留，必须保持复位值。

位 3 **WP4**：WKUP4 唤醒引脚极性 (WKUP4 wakeup pin polarity)
用于产生唤醒条件的 WKUP4 外部唤醒信号极性（电平或边沿）：
0：高电平或上升沿
1：低电平或下降沿

位 2 **WP3**：WKUP3 唤醒引脚极性 (WKUP3 wakeup pin polarity)
用于产生唤醒条件的 WKUP3 外部唤醒信号极性（电平或边沿）：
0：高电平或上升沿
1：低电平或下降沿

位 1 **WP2**：WKUP2 唤醒引脚极性 (WKUP2 wakeup pin polarity)
用于产生唤醒条件的 WKUP2 外部唤醒信号极性（电平或边沿）：
0：高电平或上升沿
1：低电平或下降沿

位 0 **WP1**：WKUP1 唤醒引脚极性 (WKUP1 wakeup pin polarity)
用于产生唤醒条件的 WKUP1 外部唤醒信号极性（电平或边沿）：
0：高电平或上升沿
1：低电平或下降沿

4.4.4 PWR 状态寄存器 1 (PWR_SR1)

PWR status register 1

该寄存器在退出待机模式时不会复位，而是根据 *RCC APB 外设复位寄存器 1 (RCC_APBRSTR1)* 的 PWRRST 位执行复位。

偏移地址：0x10

复位值：0x0000 0000

访问：与标准 APB 访问相比，需要两个额外的 APB 时钟周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WUFI	Res.	Res.	Res.	Res.	Res.	Res.	SBF	Res.	Res.	WUF6	Res.	WUF4	WUF3	WUF2	WUF1
r							r			r		r	r	r	r

位 31:16 保留，必须保持复位值。

位 15 **WUFI**：内部唤醒标志 (Wakeup flag internal)
在内部唤醒线上检测到唤醒条件时，该位置 1。清除所有内部唤醒源时，该位清零。

位 14:9 保留，必须保持复位值。

位 8 **SBF**：待机标志 (Standby flag)
当器件进入待机模式时，该位由硬件置 1，通过将 PWR_SCR 寄存器的 CSBF 位置 1 或通过上电复位，该位清零。该位不通过系统复位清零。
0：器件未进入待机模式
1：器件进入待机模式

位 7:6 保留，必须保持复位值。

位 5 **WUF6**：唤醒标志 6 (Wakeup flag 6)
当在 WKUP6 唤醒引脚上检测到唤醒事件时，该位置 1。通过向 PWR_SCR 寄存器中的 CWUF6 写入 1 进行清零。

位 4 保留，必须保持复位值。

位 3 **WUF4**：唤醒标志 4 (Wakeup flag 4)
当在 WKUP4 唤醒引脚上检测到唤醒事件时，该位置 1。通过向 PWR_SCR 寄存器中的 CWUF4 写入 1 进行清零。

位 2 **WUF3**：唤醒标志 3 (Wakeup flag 3)

当在 WKUP3 唤醒引脚上检测到唤醒事件时，该位置 1。通过将 PWR_SCR 寄存器中的 CWUF3 位置 1 进行清零。

位 1 **WUF2**：唤醒标志 2 (Wakeup flag 2)

当在 WKUP2 唤醒引脚上检测到唤醒事件时，该位置 1。通过将 PWR_SCR 寄存器中的 CWUF2 位置 1 进行清零。

位 0 **WUF1**：唤醒标志 1 (Wakeup flag 1)

当在 WKUP1 唤醒引脚上检测到唤醒事件时，该位置 1。通过将 PWR_SCR 寄存器中的 CWUF1 位置 1 进行清零。

4.4.5 PWR 状态寄存器 2 (PWR_SR2)

PWR status register 2

退出待机/关断模式后，此寄存器会复位。

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

位 31:8 保留，必须保持复位值。

位 7 **FLASH_RDY**：Flash 就绪标志 (Flash ready flag)

硬件将该位置 1 时表示，从掉电模式唤醒后，Flash 已做好访问准备。要使 Flash 进入掉电模式，应将 FPD_SLP 或 FPD_STP 位置 1。

0：Flash 处于掉电模式

1：Flash 已做好访问准备

注意：如果系统从 SRAM 自举，用户应用程序必须等待至 FLASH_RDY 位置 1 后再跳转到 Flash。

位 6:0 保留，必须保持复位值。

4.4.6 PWR 状态清零寄存器 (PWR_SCR)

PWR status clear register

偏移地址：0x18

复位值：0x0000 0000

访问：与标准 APB 访问相比，需要三个额外的 APB 时钟周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSBF	Res.	Res.	CWUF6	Res.	CWUF4	CWUF3	CWUF2	CWUF1
							w			w		w	w	w	w

位 31:9 保留，必须保持复位值。

位 8 **CSBF**：将待机标志清零 (Clear standby flag)
将该位置 1 会清零 PWR_SR1 寄存器中的 SBF 标志。

位 7:6 保留，必须保持复位值。

位 5 **CWUF6**：将唤醒标志 6 清零 (Clear wakeup flag 6)
将该位置 1 会清零 PWR_SR1 寄存器中的 WUF6 标志。

位 4 保留，必须保持复位值。

位 3 **CWUF4**：将唤醒标志 4 清零 (Clear wakeup flag 4)
将该位置 1 会清零 PWR_SR1 寄存器中的 WUF4 标志。

位 2 **CWUF3**：将唤醒标志 3 清零 (Clear wakeup flag 3)
将该位置 1 会清零 PWR_SR1 寄存器中的 WUF3 标志。

位 1 **CWUF2**：将唤醒标志 2 清零 (Clear wakeup flag 2)
将该位置 1 会清零 PWR_SR1 寄存器中的 WUF2 标志。

位 0 **CWUF1**：将唤醒标志 1 清零 (Clear wakeup flag 1)
将该位置 1 会清零 PWR_SR1 寄存器中的 WUF1 标志。

4.4.7 PWR 端口 A 上拉控制寄存器 (PWR_PUCRA)

PWR Port A pull-up control register

该寄存器在退出待机模式时不会复位，而是根据 *RCC APB 外设复位寄存器 1 (RCC_APBRSTR1)* 的 PWRRST 位执行复位。

偏移地址：0x20

复位值：0x0000 0000

访问：与标准 APB 访问相比，需要三个（写访问）或两个（读访问）额外的 APB 时钟周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PUi**：端口 A 上拉位 i (Port A pull-up bit i)（i = 15 到 0）
将 PUi 位置 1 时，如果相应的 PDi 位为零且 PWR_CR3 寄存器的 APC 位置 1，则会激活 PA[i] I/O 上的上拉器件。

4.4.8 PWR 端口 A 下拉控制寄存器 (PWR_PDCRA)

PWR Port A pull-down control register

该寄存器在退出待机模式时不会复位，而是根据 *RCC APB 外设复位寄存器 1 (RCC_APBRSTR1)* 的 PWRRST 位执行复位。

偏移地址：0x24

复位值：0x0000 0000

访问：与标准 APB 访问相比，需要三个（写访问）或两个（读访问）额外的 APB 时钟周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PDi**：端口 A 下拉位 i (Port A pull-down bit i)（i = 15 到 0）

将 PDi 位置 1 时，如果 PWR_CR3 寄存器的 APC 位置 1，则会激活 PA[i] I/O 上的下拉器件。

4.4.9 PWR 端口 B 上拉控制寄存器 (PWR_PUCRB)

PWR Port B pull-up control register

该寄存器在退出待机模式时不会复位，而是根据 *RCC APB 外设复位寄存器 1 (RCC_APBRSTR1)* 的 PWRRST 位执行复位。

偏移地址：0x28

复位值：0x0000 0000

访问：与标准 APB 访问相比，需要三个（写访问）或两个（读访问）额外的 APB 时钟周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PUi**：端口 B 上拉位 i (Port B pull-up bit i)（i = 15 到 0）

将 PUi 位置 1 时，如果相应的 PDi 位为零且 PWR_CR3 寄存器的 APC 位置 1，则会激活 PB[i] I/O 上的上拉器件。

注意：STM32C011xx 只提供 PU7 和 PU6

4.4.10 PWR 端口 B 下拉控制寄存器 (PWR_PDCRB)

PWR Port B pull-down control register

该寄存器在退出待机模式时不会复位，而是根据 *RCC APB 外设复位寄存器 1 (RCC_APBRSTR1)* 的 PWRRST 位执行复位。

偏移地址：0x2C

复位值：0x0000 0000

访问：与标准 APB 访问相比，需要三个（写访问）或两个（读访问）额外的 APB 时钟周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PDi**：端口 B 下拉位 i (Port B pull-down bit i)（i = 15 到 0）

将 PDi 位置 1 时，如果 PWR_CR3 寄存器的 APC 位置 1，则会激活 PB[i] I/O 上的下拉器件。

注意：STM32C011xx 只提供 PD7 和 PD6

4.4.11 PWR 端口 C 上拉控制寄存器 (PWR_PUCRC)

PWR Port C pull-up control register

该寄存器在退出待机模式时不会复位，而是根据 *RCC APB 外设复位寄存器 1 (RCC_APBRSTR1)* 的 PWRRST 位执行复位。

偏移地址：0x30

复位值：0x0000 0000

访问：与标准 APB 访问相比，需要三个（写访问）或两个（读访问）额外的 APB 时钟周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PU15	PU14	PU13	Res.	Res.	Res.	Res.	Res.	PU7	PU6	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw						rw	rw						

位 31:16、12:8、5:0 保留，必须保持复位值。

位 15:13、7:6 **PUi**：端口 C 上拉位 i (Port C pull-up bit i)（i = 15 到 13、7 到 6）

将 PUi 位置 1 时，如果相应的 PDi 位为零且 PWR_CR3 寄存器的 APC 位置 1，则会激活 PC[i] I/O 上的上拉器件。

注意：STM32C011xx 只提供 PU15 和 PU14

4.4.12 PWR 端口 C 下拉控制寄存器 (PWR_PDCRC)

PWR Port C pull-down control register

该寄存器在退出待机模式时不会复位，而是根据 *RCC APB 外设复位寄存器 1 (RCC_APBRSTR1)* 的 PWRRST 位执行复位。

偏移地址：0x34

复位值：0x0000 0000

访问：与标准 APB 访问相比，需要三个（写访问）或两个（读访问）额外的 APB 时钟周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD15	PD14	PD13	Res.	Res.	Res.	Res.	Res.	PD7	PD6	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw						rw	rw						

位 31:16、12:8、5:0 保留，必须保持复位值。

位 15:13、7:6 **PDi**：端口 C 下拉位 i (Port C pull-down bit i)（i = 15、14、13、7、6）

将 PDi 位置 1 时，如果 PWR_CR3 寄存器的 APC 位置 1，则会激活 PC[i] I/O 上的下拉器件。

注意：STM32C011xx 只提供 PD15 和 PD14。

4.4.13 PWR 端口 D 上拉控制寄存器 (PWR_PUCRD)

PWR Port D pull-up control register

该寄存器在退出待机模式时不会复位，而是根据 *RCC APB 外设复位寄存器 1 (RCC_APBRSTR1)* 的 PWRRST 位执行复位。

偏移地址：0x38

复位值：0x0000 0000

访问：与标准 APB 访问相比，需要三个（写访问）或两个（读访问）额外的 APB 时钟周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU3	PU2	PU1	PU0
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **PUi**：端口 D 上拉位 i (Port D pull-up bit i)（i = 3 到 0）

将 PUi 位置 1 时，如果相应的 PDi 位为零且 PWR_CR3 寄存器的 APC 位置 1，则会激活 PD[i] I/O 上的上拉器件。

注意：STM32C011xx 不提供。

4.4.14 PWR 端口 D 下拉控制寄存器 (PWR_PDCRD)

PWR Port D pull-down control register

该寄存器在退出待机模式时不会复位，而是根据 *RCC APB 外设复位寄存器 1 (RCC_APBRSTR1)* 的 PWRRST 位执行复位。

偏移地址：0x3C

复位值：0x0000 0000

访问：与标准 APB 访问相比，需要三个（写访问）或两个（读访问）额外的 APB 时钟周期。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD3	PD2	PD1	PD0
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **PDi**：端口 D 下拉位 i (Port D pull-down bit i)（i = 3 到 0）

将 PDi 位置 1 时，如果 PWR_CR3 寄存器的 APC 位置 1，则会激活 PD[i] I/O 上的下拉器件。

注意：STM32C011xx 不提供。

4.4.15 PWR 端口 F 上拉控制寄存器 (PWR_PUCRF)

PWR Port F pull-up control register

该寄存器在退出待机模式时不会复位，而是根据 *RCC APB 外设复位寄存器 1 (RCC_APBRSTR1)* 的 PWRRST 位执行复位。

偏移地址：0x48

复位值：0x0000 0000

访问：与标准 APB 访问相比，需要三个（写访问）或两个（读访问）额外的 APB 时钟周期。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU2	PU1	PU0
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **PUi**：端口 F 上拉位 i (Port F pull-up bit i)（i = 2 到 0）

将 PUi 位置 1 时，如果相应的 PDi 位为零且 PWR_CR3 寄存器的 APC 位置 1，则会激活 PF[i] I/O 上的上拉器件。

注意：STM32C011xx 只提供 PU2。

4.4.16 PWR 端口 F 下拉控制寄存器 (PWR_PDCRF)

PWR Port F pull-down control register

该寄存器在退出待机模式时不会复位，而是根据 *RCC APB 外设复位寄存器 1 (RCC_APBRSTR1)* 的 PWRRST 位执行复位。

偏移地址：0x4C

复位值：0x0000 0000

访问：与标准 APB 访问相比，需要三个（写访问）或两个（读访问）额外的 APB 时钟周期。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD2	PD1	PD0
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **PDi**：端口 F 下拉位 i (Port F pull-down bit i)（i = 2 到 0）

将 PDi 位置 1 时，如果 PWR_CR3 寄存器的 APC 位置 1，则会激活 PF[i] I/O 上的下拉器件。

注意：STM32C011xx 只提供 PD2。

4.4.17 PWR 备份 x 寄存器 (PWR_BKPxR)

PWR backup x register

该寄存器在退出待机模式时不会复位，而是根据 *RCC APB 外设复位寄存器 1 (RCC_APBRSTR1)* 的 PWRRST 位执行复位。

偏移地址：0x070 + 0x04 * x（x = 0 到 3）

复位值：0x0000 0000

访问：与标准 APB 访问相比，需要三个（写访问）或两个（读访问）额外的 APB 时钟周期。

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

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

位 31:16 保留，必须保持复位值。

位 15:0 **BKP[15:0]**：备份位域

该位域保留器件处于待机模式时的信息。

4.4.18 PWR 寄存器映射

表 22. PWR 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	PWR_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FPD_SLP	Res.	FPD_STOP	LPMS [2:0]		
	复位值																											0		1	0	0	0
0x004	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x008	PWR_CR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EIWUL	Res.	Res.	Res.	Res.	APC	Res.	Res.	Res.	Res.	EWUP6	Res.	EWUP4	EWUP3	EWUP2	EWUP1
	复位值																	1					0					0		0	0	0	0
0x00C	PWR_CR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WP6	Res.	WP4	WP3	WP2	WP1
	复位值																											0		0	0	0	0
0x010	PWR_SR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUFI	Res.	Res.	Res.	Res.	Res.	Res.	SBF	Res.	Res.	WUF6	Res.	WUF4	WUF3	WUF2	WUF1
	复位值																	0							0			0		0	0	0	0
0x014	PWR_SR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLASH_RDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	复位值																									0							
0x018	PWR_SCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSBF	Res.	Res.	CWUF6	Res.	CWUF4	CWUF3	CWUF2	CWUF1
	复位值																								0			0		0	0	0	0
0x020	PWR_PUCRA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	PWR_PDCRA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x028	PWR_PUCRB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x02C	PWR_PDCRB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x030	PWR_PUCRC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU15	PU14	PU13	Res.	Res.	Res.	Res.	Res.	PU7	PU6	Res.	Res.	Res.	Res.	Res.	Res.
	复位值																	0	0	0						0	0						
0x034	PWR_PDCRC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD15	PD14	PD13	Res.	Res.	Res.	Res.	Res.	PD7	PD6	Res.	Res.	Res.	Res.	Res.	Res.
	复位值																	0	0	0						0	0						
0x038	PWR_PUCRD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU3	PU2	PU1	PU0
	复位值																													0	0	0	0

表 22. PWR 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x03C	PWR_PDCRD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD3	PD2	PD1	PD0
	复位值																													0	0	0	0
0x040-044	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x048	PWR_PUCRF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU2	PU1	PU0
	复位值																														0	0	0
0x04C	PWR_PDCRF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD2	PD1	PD0
	复位值																														0	0	0
0x050-06C	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x070	PWR_BKP0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKP[15:0]															
	复位值																	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
0x074	PWR_BKP1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKP[15:0]															
	复位值																	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
0x078	PWR_BKP2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKP[15:0]															
	复位值																	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
0x07C	PWR_BKP3R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKP[15:0]															
	复位值																	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

有关寄存器边界地址的信息，请参见*第 38 页的第 2.2.2 节*。

5 复位和时钟控制 (RCC)

5.1 复位

共有三种类型的复位，分别为系统复位、电源复位和 RTC 域复位。

5.1.1 电源复位

只要发生以下事件之一，就会产生电源复位：

- 上电复位 (POR) 或欠压复位 (BOR)
- 退出待机模式
- 退出关断模式

上电复位和欠压复位会将所有寄存器设置为其复位值。

退出待机模式时，V_{CORE} 域的所有寄存器都设置为其复位值。V_{CORE} 域外的寄存器（WKUP、IWDG 以及待机/关断模式控制）不受影响。

退出关断模式时，会产生欠压复位，将所有寄存器复位。

5.1.2 系统复位

除了 RCC 控制/状态寄存器 2 (RCC_CSR2) 中的复位标志和 RTC 域中的寄存器外，系统复位会将其他全部寄存器都设置为其复位值。

只要发生以下事件之一，就会产生系统复位：

- NRST 引脚低电平（外部复位）
- 窗口看门狗事件（WWDG 复位）
- 独立看门狗事件（IWDG 复位）
- 软件复位（SW 复位）（请参见*软件复位*）
- 低功耗模式安全复位（请参见*低功耗模式安全复位*）
- 选项字节加载器复位（请参见*选项字节加载器复位*）
- 上电复位

可通过查看 RCC_CSR 寄存器中的复位标志确定复位源（请参见*第 5.4.21 节：RCC 控制/状态寄存器 2 (RCC_CSR2)*）。

NRST 引脚（外部复位）：

可通过特定选项位将 NRST 引脚配置为以下工作模式：

- **复位输入/输出**（器件交付时的默认配置）

  该引脚上的有效复位信号传播到内部逻辑，并且每个内部复位源连接到一个脉冲发生器，而脉冲发生器的输出用于驱动该引脚。GPIO 功能 (PF2) 不可用。脉冲发生器可确保每个内部复位源输出到 NRST 引脚上的复位脉冲都至少持续 20 µs。可以使用内部复位保持信号选项（在 *FLASH 选项寄存器 (FLASH_OPTR)* 中使能时）来确保该引脚拉低，直到其电压满足 V_{IL} 阈值为止。该功能可用于在线路面临很大容性负载时，通过外部元件检测内部复位源。

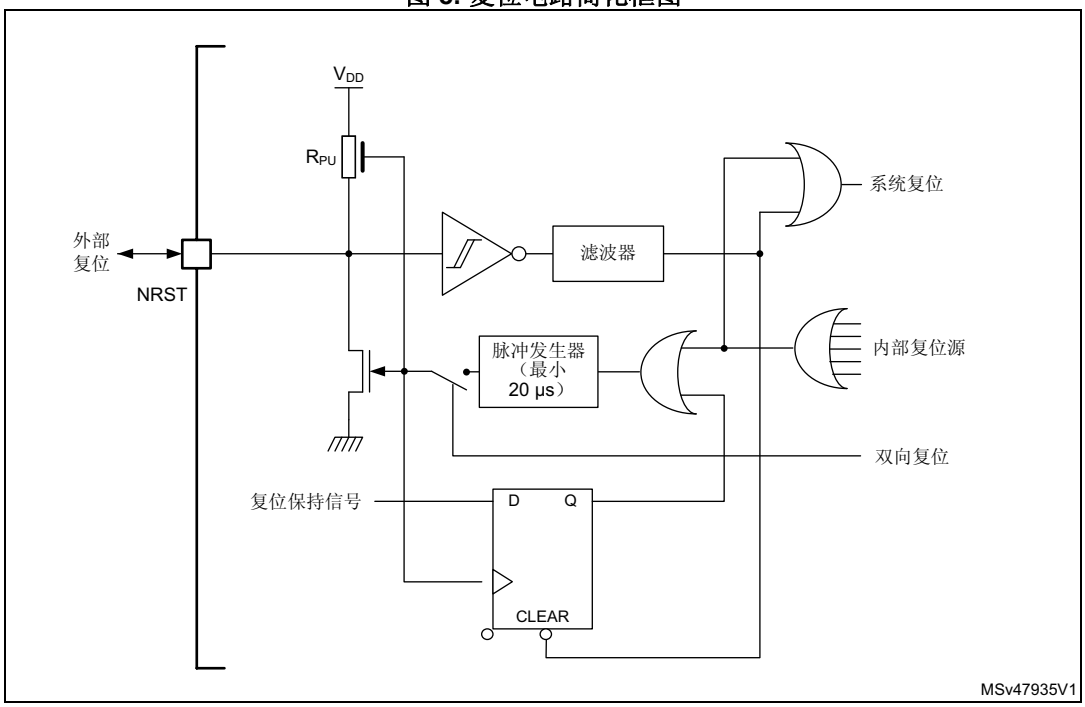
- **复位输入**

  在该模式下，NRST 引脚上的任何有效复位信号都将传播到器件内部逻辑，但器件内部产生的复位在该引脚上不可见。在该配置下，GPIO 功能 (PF2) 不可用。

- **GPIO**

  在该模式下，该引脚可以用作 PF2 标准 GPIO。该引脚的复位功能不可用。复位只能由器件内部复位源实现且不会传播到该引脚。

图 8. 复位电路简化框图



小心： 上电复位或从关断模式唤醒后，在 $t_{RSTTEMPO}$ 时间（请参见数据手册）结束后的第四个时钟周期内，系统会将 NRST 引脚配置为复位输入/输出并驱动为低电平，直到加载选项字节时将该引脚重新配置为预期模式为止。

软件复位

必须将 Cortex®-M0+ 应用中断和复位控制寄存器中的 SYSRESETREQ 位置 1 才能强制器件进行软件复位（请参见编程手册 PM0223）。

低功耗模式安全复位

为了防止关键应用错误地进入低功耗模式，提供了低功耗模式安全复位。如果在选项字节中使能，则在下列情况下会产生这种复位：

- **进入待机模式**

  可通过清零用户选项字节中的 nRST_STDBY 位来使能此类复位。使能后，只要成功执行进入待机模式序列，器件就将复位，而非进入待机模式。

- **进入停止模式**

  可通过清零用户选项字节中的 nRST_STOP 位来使能此类复位。使能后，只要成功执行进入停止模式序列，器件就将复位，而非进入停止模式。

- **进入关断模式**

  可通过复位用户选项字节中的 nRST_SHDW 位来使能此类复位。使能后，只要成功执行进入关断模式序列，器件就将复位，而非进入关断模式。

有关用户选项字节的详细信息，请参见*第 3.4.1 节：FLASH 选项字节说明*。

选项字节加载器复位

当 FLASH_CR 寄存器中的 OBL_LAUNCH 位置 1 时，将产生选项字节加载器复位。该位用来通过软件启动选项字节加载。

5.1.3 RTC 域复位

RTC 域具有两个特殊复位。

只要发生以下事件之一，就会产生 RTC 域复位：

- **软件复位**，通过将 *RCC 控制/状态寄存器 1 (RCC_CSR1)* 的 RTCRST 位置 1 触发。
- **V_{DD} 上电**。

RTC 域复位仅影响 LSE 振荡器、RTC 和 RCC 控制/状态寄存器 1。

5.2 时钟

器件提供以下时钟源来生成主时钟：

- **HSI48 RC**——高速全集成 RC 振荡器，可生成 **HSI48** 时钟 (48 MHz)
- **HSE OSC**——具有外部晶振/陶瓷谐振器或外部时钟源的高速振荡器，可生成 **HSE** 时钟（4 MHz 到 48 MHz）
- **LSI RC**——低速全集成 RC 振荡器，可生成 **LSI** 时钟（约 32 kHz）
- **LSE OSC**——具有外部晶振/陶瓷谐振器或外部时钟源的低速振荡器，可生成 **LSE** 时钟（精确的 32.768 kHz 或高达 1 MHz 的外部时钟）
- **I2S_CKIN**——I2S1 外设的直接时钟输入引脚

对于每个振荡器来说，在未使用时都可单独打开或者关闭，以降低功耗。有关功能的更多详细信息，请参见本节中的各个小节。有关内部和外部时钟源的电气特性，请参见器件数据手册。

器件通过对主时钟分频来生成次级时钟：

- **HSISYS**——通过对 HSI48 分频生成的时钟，分频系数可编程，从 1 到 128
- **SYSCLK**——可选择 LSE、LSI、HSE 和 HSISYS 时钟中的一个
- **HSIKER**——通过对 HSI48 分频生成的时钟，分频系数可编程，从 1 到 8
- **HCLK**——通过对 SYSCLK 分频生成的时钟，分频系数可编程，从 1 到 512
- **HCLK8**——通过对 HCLK 进行八分频生成的时钟
- **PCLK**——通过对 HCLK 分频生成的时钟，分频系数可编程，从 1 到 16
- **TIMPCLK**——由 PCLK 提供的时钟，APB 预分频器分频系数设置为 1 时以 PCLK 频率运行，其他设置则以两倍 PCLK 频率运行

此外，还有更多的次级时钟通过对 HSE、HSI48 和 HCLK 时钟进行固定分频来生成。

从复位中启动后，HSISYS 用作系统时钟源，并进行四分频（生成 12 MHz 频率）。

HCLK 时钟和 PCLK 时钟分别用于 AHB 和 APB 域的时钟，其允许的最大频率为 48 MHz。

除了以下外设之外，其他外设都连接到总线以使用总线上的时钟（AHB 使用 HCLK，APB 使用 PCLK）：

- **TIMx**
  - APB 预分频器分频系数设置为 1 时，TIMPCLK 以 PCLK 频率运行，其他设置则以两倍 PCLK 频率运行
- **USART1**，可选时钟源如下：
  - SYSCLK（系统时钟）
  - HSIKER
  - LSE
  - PCLK（APB 时钟）

  仅当时钟为 HSI48 或 LSE 时，支持从停止模式唤醒。
- **ADC**，可选时钟源如下：
  - SYSCLK（系统时钟）
  - HSIKER
- **I2Cx**，可选时钟源如下：
  - SYSCLK（系统时钟）
  - HSIKER
  - PCLK（APB 时钟）

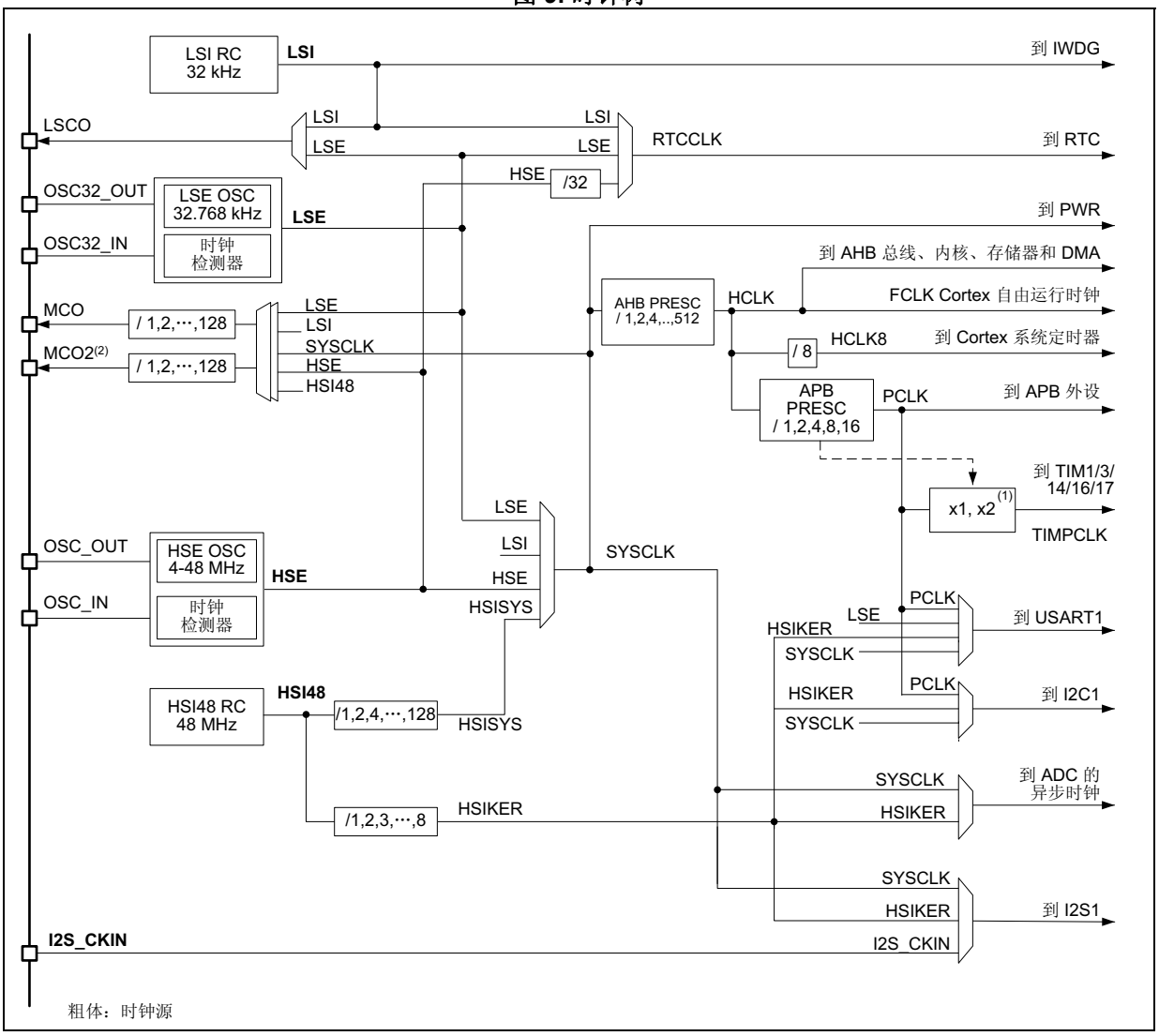
  仅当时钟为 HSI48 时，支持从停止模式唤醒。
- **I2Sx**，可选时钟源如下：
  - SYSCLK（系统时钟）
  - HSIKER
  - I2S_CKIN 引脚
- **RTC**，可选时钟源如下：
  - LSE
  - LSI
  - HSE 时钟 32 分频

  仅当时钟为 LSI 或 LSE 时，支持停止模式的功能（包括唤醒）。
- **IWDG**，始终使用 LSI 时钟提供时钟。
- **SysTick**（Cortex® 内核系统定时器），可选时钟源如下：
  - HCLK（AHB 时钟）
  - HCLK 时钟 8 分频

  通过 SysTick 控制和状态寄存器进行选择。

HCLK 用作 Cortex®-M0+ 的自由运行时钟 (FCLK)。更多详细信息，请参见编程手册 PM0223。

图 9. 时钟树



1. APB 预分频器分频系数设置为 1 时，TIMPCLK 以 PCLK 频率运行，其他设置则以两倍 PCLK 频率运行

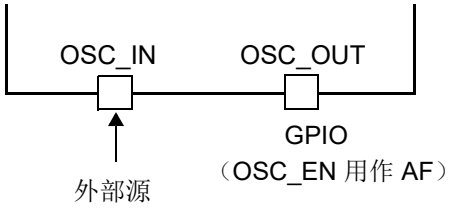
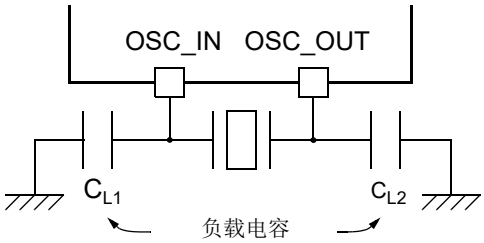
5.2.1 HSE 时钟

高速外部时钟信号 (HSE) 有两个时钟源：

- HSE 外部晶振/陶瓷谐振器
- HSE 用户外部时钟

谐振器和负载电容必须尽可能地靠近振荡器的引脚，以尽量减小输出失真和起振稳定时间。负载电容值必须根据所选振荡器的不同做适当调整。

图 10. HSE/LSE 时钟源

时钟源	硬件配置
外部时钟	
晶振/陶瓷谐振器	

外部晶振/陶瓷谐振器（HSE 晶振）

4 到 48 MHz 外部振荡器的优点是精度非常高。

相关的硬件配置如*图10*所示。更多详细信息，请参见*数据手册*的电气特性部分。

RCC 时钟控制寄存器 (RCC_CR) 中的 HSERDY 标志指示 HSE 振荡器是否稳定。在启动时，硬件将该位置 1 后，该时钟才可以使用。如在 *RCC 时钟中断使能寄存器 (RCC_CIER)* 中使能中断，则可产生中断。

HSE 晶振可通过 *RCC 时钟控制寄存器 (RCC_CR)* 中的 HSEON 位打开或关闭。

外部源（HSE 旁路）

在该模式下，必须提供外部时钟源。最高频率不超过 48 MHz。通过将 *RCC 时钟控制寄存器 (RCC_CR)* 中的 HSEBYP 和 HSEON 位置 1 来选择该模式。如果器件上提供 OSC_IN 和 OSC_OUT 引脚，必须使用外部时钟信号（方波、正弦波或三角波——请参见数据手册）来驱动 OSC_IN 引脚（请参见*图10*）。OSC_OUT 引脚可以用作 GPIO。

OSC_OUT 引脚既可以用作 GPIO，也可以配置为 OSC_EN 复用功能，以便向外部时钟合成器提供使能信号。这样可以在器件进入低功耗模式时停止外部时钟源。

注意： *有关引脚可用性的详细信息，请参见相应器件数据手册中的引脚排列部分。*

为最大程度地降低功耗，建议使用方波信号。

5.2.2 HSI48 时钟

HSI48 时钟信号是从 48 MHz 内部 RC 振荡器生成的。

HSI48 RC 振荡器的优点是成本较低（无需使用外部元件）。它还比 HSE 晶振具有更快的启动时间。但是，即使经过校准后，其精度也不及使用参考频率的振荡器，例如石英晶振或陶瓷谐振器。

可选择通过 HSI48 生成的 HSISYS 时钟作为从停止模式唤醒后的系统时钟。请参见*第 5.3 节：低功耗模式*。它还可作为备份时钟源（辅助时钟）使用，以防 HSE 晶振发生故障。请参见*第 5.2.6 节：时钟安全系统 (CSS)*。

校准

因为生产工艺不同，不同芯片的 RC 振荡器频率也不同。为了补偿这种变化，每个器件出厂时都会校准为在 T_A=25°C 时达到 1 % 的精度。

复位后，工厂校准值将加载到 *RCC 内部时钟源校准寄存器 (RCC_ICSCR)* 的 HSICAL[7:0] 位中。

应用中的电压或温度变化可能影响 RC 振荡器的 HSI48 频率。可以使用 *RCC 内部时钟源校准寄存器 (RCC_ICSCR)* 中的 HSITRIM[6:0] 位对该频率进行微调。

有关如何测量 HSI48 频率变化的更多详细信息，请参见*第 5.2.13 节：基于 TIM14/TIM16/TIM17 的内部/外部时钟测量*。

RCC 时钟控制寄存器 (RCC_CR) 中的 HSIRDY 标志指示 HSI48 RC 是否稳定。在启动时，硬件将该位置 1 后，HSI48 才可以使用。

HSI48 RC 可通过 *RCC 时钟控制寄存器 (RCC_CR)* 中的 HSION 位打开和关闭。

HSI48 信号还可作为备份时钟源（辅助时钟）使用，以防 HSE 晶振发生故障。请参见*第 102 页的第 5.2.6 节：时钟安全系统 (CSS)*。

5.2.3 LSE 时钟

LSE 晶振是 32.768 kHz 晶振或陶瓷谐振器，可作为实时时钟 (RTC) 的时钟源来提供时钟/日历或其他定时功能，具有功耗低且精度高的优点。

LSE 晶振通过 *RCC 控制/状态寄存器 1 (RCC_CSR1)* 中的 LSEON 位打开和关闭。通过 *RCC 控制/状态寄存器 1 (RCC_CSR1)* 的 LSEDRV 位，可在运行时更改晶振驱动强度，以实现稳健性、短启动时间和低功耗之间的最佳平衡。当 LSE 为 ON 时，LSE 驱动可降低为更低的驱动能力（LSEDRV 清零）。但是，当选择了 LSEDRV，如果 LSEON 置 1 则不能提高驱动能力。

RCC 控制/状态寄存器 1 (RCC_CSR1) 的 LSERDY 标志指示 LSE 晶振是否稳定。在启动时，硬件将该位置 1 后，LSE 晶振输出时钟信号才可以使用。如在 *RCC 时钟中断使能寄存器 (RCC_CIER)* 中使能中断，则可产生中断。

外部源（LSE 旁路）

在该模式下，必须提供外部时钟源。最高频率不超过 1 MHz。通过将 *睡眠/停止模式下的 RCC AHB 外设时钟使能寄存器 (RCC_AHBSMENR)* 中的 LSEBYP 和 LSEON 位置 1 来选择该模式。必须使用外部时钟信号（方波、正弦波或三角波——请参见数据手册）来驱动 OSCX_IN 引脚，同时 OSCX_OUT 引脚可用作 GPIO。请参见*图10*。

5.2.4 LSI 时钟

LSI RC 可作为低功耗时钟源在停止和待机模式下保持运行，供独立看门狗 (IWDG) 和 RTC 使用。时钟频率为 32 kHz。有关详细信息，请参见数据手册的电气特性部分。

LSI RC 可通过 *RCC 控制/状态寄存器 2 (RCC_CSR2)* 中的 LSION 位打开或关闭。

RCC 控制/状态寄存器 2 (RCC_CSR2) 中的 LSIRDY 标志指示 LSI 振荡器是否稳定。在启动时，硬件将该位置 1 后，该时钟才可以使用。如在 *RCC 时钟中断使能寄存器 (RCC_CIER)* 中使能中断，则可产生中断。

5.2.5 系统时钟 (SYSCLK) 选择

可选择以下时钟之一作为系统时钟 (SYSCLK)：

- LSI
- LSE
- HSISYS
- HSE

系统时钟最大频率为48 MHz。系统复位后，选择通过 HSI48 振荡器生成的 HSISYS 时钟作为系统时钟。在使用某一时钟源来作为系统时钟时，该时钟源无法停止。

只有在目标时钟源已就绪时（时钟在启动延迟后稳定时），才可从一个时钟源切换到另一个。如果选择尚未就绪的时钟源，则切换在该时钟源就绪时才会进行。*RCC 内部时钟源校准寄存器 (RCC_ICSCR)* 中的状态位指示哪个（些）时钟已就绪，以及当前哪个时钟正充当系统时钟。

5.2.6 时钟安全系统 (CSS)

时钟安全系统可通过软件激活。激活后，时钟监测器将在 HSE 振荡器启动延迟后使能，并在该振荡器停止时被关闭。

如果检测到 HSE 时钟发生故障：

- HSE 振荡器将自动禁止
- 一个时钟故障事件将被发送到 TIM1、TIM16 和 TIM17 定时器的刹车输入
- 将产生 CSSI（时钟安全系统中断）

CSSI 与 Cortex®-M0+ NMI（不可屏蔽中断）异常向量相链接。这样软件便可获悉 HSE 时钟故障，从而执行救援操作。

注意： *如果 CSS 使能并且 HSE 时钟发生故障，则会产生 CSSI，并会自动产生 NMI。NMI 将无限期执行，除非将 CSS 中断挂起位清零。因此，NMI ISR 有必要通过将 RCC 时钟中断清零寄存器 (RCC_CICR) 中的 CSSC 位置 1 来清除 CSSI。*

如果直接或间接选择 HSE 作为系统时钟，则检测到 HSE 时钟故障时，系统时钟将自动切换到 HSISYS，并禁止 HSE 振荡器。

5.2.7 LSE 时钟的时钟安全系统 (LSECSS)

可通过将 *RCC 控制/状态寄存器 1 (RCC_CSR1)* 中的 LSECSSON 位置 1 来激活 LSE 上的时钟安全系统。只能通过硬件复位、RTC 软件复位、或在检测到 LSE 时钟故障后将该位清零。LSE 和 LSI 使能（LSEON 和 LSION 使能）并就绪（LSERDY 和 LSIRDY 标志由硬件置 1）后，以及通过 RTCSEL 选择 RTC 时钟后，必须写入 LSECSSON。

LSECSS 可在除待机和关断外的所有模式下工作，并且在发生系统复位（上电复位除外）时也能保持工作状态。如果在 LSE 振荡器上检测到故障，则不会再向 RTC 提供 LSE 时钟，但其寄存器不受影响。

注意： *如果 LSECSS 使能并且 LSE 时钟发生故障，则会产生 LSECSSI，并会自动产生 NMI。NMI 将无限期执行，除非将 LSECSS 中断挂起位清零。因此，NMI ISR 有必要通过将 RCC 时钟中断清零寄存器 (RCC_CICR) 中的 LSECSSC 位置 1 来清除 LSECSSI。*

如果使用 LSE 作为系统时钟，则检测到 LSE 时钟故障时，系统时钟将自动切换到 LSI。在低功耗模式下，发生 LSE 时钟故障时将产生唤醒信号。之后，必须将 RCC 寄存器中的中断标志清零。

软件随后**必须**禁止 LSECSSON 位，停止出现故障的 32 kHz 振荡器（通过清零 LSEON），并更改 RTC 时钟源（无时钟、LSI 或 HSE，通过 RTCSEL），或者采取任何适当措施来确保应用的安全。

LSE 振荡器的频率必须超过 30 kHz，以免发生误检。

5.2.8 ADC 时钟

ADC 时钟（有关最大频率，请参见器件数据手册）由系统时钟 SYSCLK 或内核时钟输出 HSIKER 提供（请参见 *RCC 外设独立时钟配置寄存器 (RCC_CCIPR)* 的 ADCSEL[1:0] 位域）。可通过配置 ADC_CCR 寄存器的 PRESC[3:0] 位域来使用以下分频系数对 ADC 时钟进行预分频：1、2、4、6、8、10、12、16、32、64、128 或 256。该时钟与 APB 时钟异步。或者，ADC 时钟也可由 ADC 总线接口的 APB 时钟（同步时钟）除以通过 ADC_CFGR2 寄存器的 CKMODE[1:0] 位域设置的可编程系数（1、2 或 4）来提供。

5.2.9 RTC 时钟

RTCCLK 时钟源可以是 HSE/32、LSE 或 LSI 时钟。它是通过编程 *RCC 控制/状态寄存器 1 (RCC_CSR1)* 中的 RTCSEL[1:0] 位来选择的。所做的选择只能通过复位 RTC 域的方式修改。配置系统时，必须始终使 PCLK 频率大于或等于 RTCCLK 频率，以便 RTC 正常工作。

如果 V_{DD} 电源掉电或者内部调压器关闭（切断 V_{CORE} 域的供电），则 RTC 不工作。

如果 RTC 时钟为 LSE 或 LSI，则 RTC 在系统复位后仍可获得时钟并保持正常工作。

5.2.10 定时器时钟

定时器时钟 TIMPCLK 由 PCLK（用于 APB）提供，具体如下：

1. 如果 APB 预分频系数设置为 1，则 TIMPCLK 频率等于 PCLK 频率。
2. 其他情况下，TIMPCLK 频率设置为两倍 PCLK 频率。

5.2.11 看门狗时钟

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

5.2.12 时钟输出功能

MCO 和 MCO2

MCO 引脚和 MCO2 引脚相互独立，其可选的输出时钟如下：

- LSI
- LSE
- SYSCLK
- HSI48
- HSE

MCO 和 MCO2 的复用器分别由 *RCC 时钟配置寄存器 (RCC_CFGR)* 的 MCOSEL[2:0] 和 MCO2SEL[2:0] 位域控制。其输出由 *RCC 时钟配置寄存器 (RCC_CFGR)* 的 MCOPRE[2:0] 和 MCO2PRE[2:0] 位域设置的系数进一步分频。

LSCO

LSCO 引脚可用于输出低速时钟：

- LSI
- LSE

该选择由 *RCC 控制/状态寄存器 1 (RCC_CSR1)* 的 LSCOSEL 位控制，并通过 LSCOEN 位使能。必须在复用功能模式下对相应 GPIO 端口的配置寄存器进行编程。

该功能在停止模式下仍可用。

5.2.13 基于 TIM14/TIM16/TIM17 的内部/外部时钟测量

所有板上时钟源的频率都可通过对 TIM14、TIM16 和 TIM17 通道 1 的输入捕获进行间接测量，如*图11*、*图12* 和*图13* 所示。

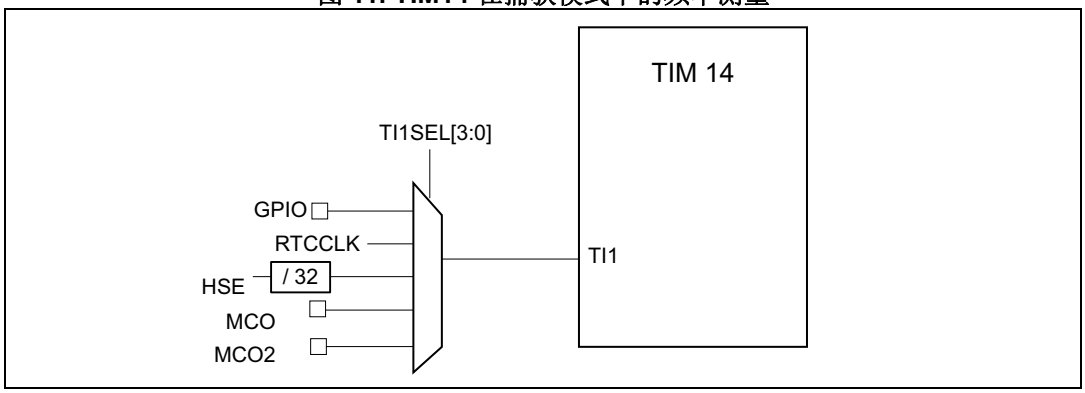
TIM14

通过设置 TIM14_TISEL 寄存器的 TI1SEL[3:0] 位域，可以为 TIM14 的输入捕获通道 1 选择以下时钟：

- GPIO（请参见器件数据手册中的复用功能映射）
- RTC 时钟 (RTCCLK)
- HSE 时钟 32 分频
- MCO（MCU 时钟输出）
- MCO2（MCU 时钟输出）

MCO 和 MCO2 分别由时钟配置寄存器 (RCC_CFGR) 的 MCOSEL[2:0] 和 MCO2SEL[2:0] 位域控制。可以为 MCO 和 MCO2 引脚选择所有时钟源。

图 11. TIM14 在捕获模式下的频率测量



TIM16

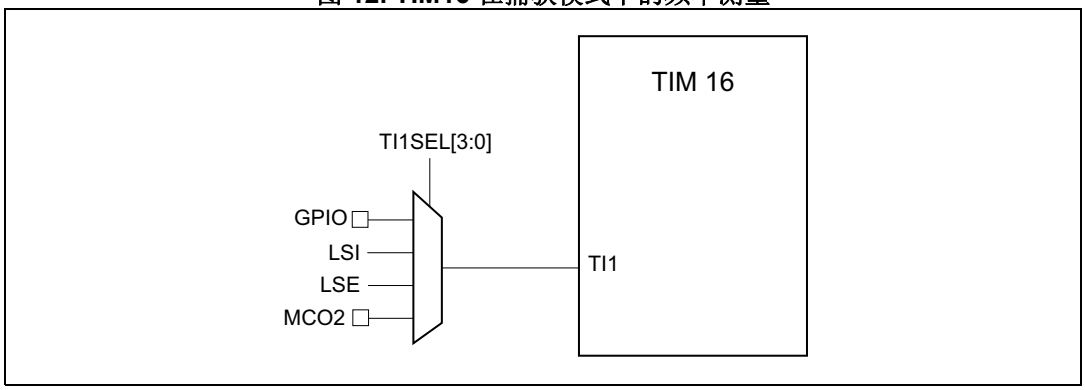
通过设置 TIM16_TISEL 寄存器的 TI1SEL[3:0] 位域，可以为 TIM16 的输入捕获通道 1 选择以下时钟：

- GPIO（请参见器件数据手册中的复用功能映射）
- LSI 时钟
- LSE 时钟
- MCO2

MCO2 由时钟配置寄存器

(RCC_CFGR) 的 MCO2SEL[2:0] 位域控制。可以为 MCO2 引脚选择所有时钟源。

图 12. TIM16 在捕获模式下的频率测量



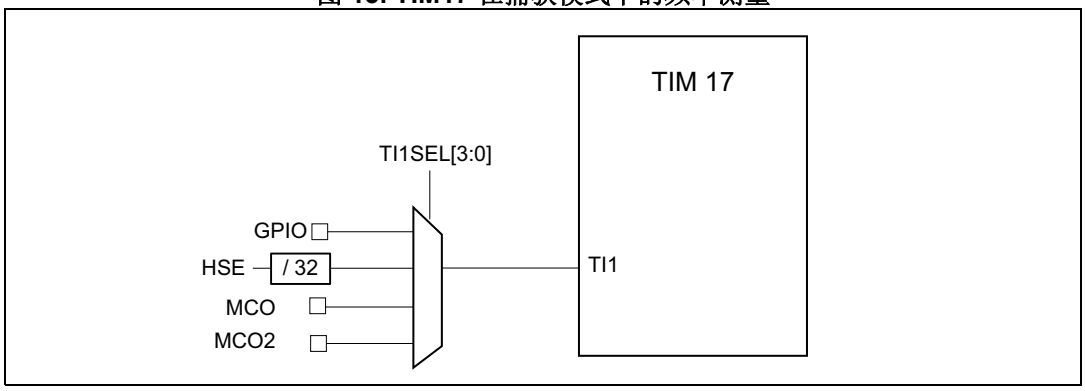
TIM17

通过设置 TIM17_TISEL 寄存器的 TI1SEL[3:0] 位域，可以为 TIM17 的输入捕获通道 1 选择以下时钟：

- GPIO（请参见器件数据手册中的复用功能映射）。
- HSE 时钟 32 分频
- MCO（MCU 时钟输出）
- MCO2（MCU 时钟输出）

MCO 和 MCO2 分别由时钟配置寄存器 (RCC_CFGR) 的 MCOSEL[2:0] 和 MCO2SEL[2:0] 位域控制。可以为 MCO 和 MCO2 引脚选择所有时钟源。

图 13. TIM17 在捕获模式下的频率测量



HSI48 振荡器校准

对于 TIM14、TIM16 和 TIM17，将 LSE 连接到通道 1 输入捕获的主要目的是精确测量被选作系统时钟的 HSISYS（由 HSI48 提供）。通过对 LSE 时钟（时间参考）连续边沿之间的 HSISYS 时钟脉冲进行计数，可以测量 HSISYS（和 HSI48）的时钟周期。此类测量可确定 HSI48 振荡器频率，精度几乎和与 LSE 振荡器配合使用的 32.768 kHz 石英晶振的精度相同（通常为几十 ppm）。随后可对 HSI48 振荡器进行微调，以补偿因生产、工艺、温度和/或电压变化而产生的与目标频率之间的偏差。

HSI48 振荡器设有针对此目的的专用校准位，且支持用户访问。

其基本原理是基于相对的测量（例如，HSISYS/LSE 比）：因此，测量精度与两个时钟源之比紧密相关。增大比值可提高测量精度。

用作时间参考的 HSE 时钟（32 分频）由 HSE 振荡器生成，是另一种达到出色 HSI48 频率测量精度的次优方法。建议在没有 LSE 时钟的情况下使用。

为了进一步提高 HSI48 振荡器校准的精度，建议采用以下一种或多种措施来提高频率测量精度：

- 将 HSISYS 分频系数设置为 1，使 HSISYS 频率等于 HSI48 频率
- 对多次连续测量的结果求平均值
- 使用定时器的输入捕获预分频器（最多每 8 个周期捕获一次）

LSI 振荡器频率测量

LSI 振荡器频率测量的原理与 HSI48 振荡器校准的原理相同。必须将 TIM16 通道 1 输入捕获用于 LSI 时钟，并选择 HSE 作为系统时钟源。通过 TIM16 计数的 LSI 信号连续边沿之间的 HSE 时钟脉冲数量即表示 LSI 时钟周期。

5.2.14 外设时钟使能寄存器

每个外设时钟均可由 RCC_AHBENR 或 RCC_APBENRx 寄存器的相应使能位使能。

当外设时钟未激活时，不支持外设寄存器进行读写访问。

小心： 使能位的同步机制可为外设产生无干扰时钟。使能位置 1 后，在时钟激活前存在时长为 2 个时钟周期的延迟，必须在软件中将这段延迟考虑在内。

5.3 低功耗模式

- 可通过软件禁止 AHB 和 APB 外设时钟，包括 DMA 时钟。
- 在睡眠模式下，CPU 时钟停止工作。在睡眠模式下，可通过软件停止存储器接口时钟（Flash 和 SRAM 接口）。当连接到 AHB-APB 总线桥时钟的所有外设时钟均被禁止时，睡眠模式期间将通过硬件禁止 AHB-APB 总线桥时钟。
- 在停止模式下，将停止 V_{CORE} 域中的所有时钟，并禁止 HSI48 和 HSE 振荡器。

  即使 MCU 处于停止模式（如果选择 HSI48 作为其中一个外设的时钟源），USART1 和 I2C1 外设也可使能 HSI48 振荡器。

  当系统处于停止模式（如果选择 LSE 作为该外设的时钟源）并使能 LSE 振荡器（LSEON 置 1）时，USART1 外设也可使用 LSE 振荡器提供的时钟工作。在这种情况下，LSE 振荡器在器件进入停止模式时始终保持激活（这些外设无法开启 LSE 振荡器）。
- 在待机和关断模式下，将停止 V_{CORE} 域中的所有时钟，并禁止 HSI48 和 HSE 振荡器。

通过将 DBGMCU_CR 寄存器中的 DBG_STOP 或 DBG_STANDBY 位置 1，可以覆盖 CPU 的深度睡眠模式以进行调试。

当退出停止模式时，HSISYS 自动变为系统时钟。

当退出待机和关断模式时，HSISYS（频率等于 HSI48/4）自动变为系统时钟。从待机和关断模式唤醒时，用户微调将丢失。

如果正在执行 Flash 编程操作，则将延迟到 Flash 接口访问结束后再进入停止、待机和关断模式。如果正在访问 APB 域，则将延迟到 APB 访问结束后再进入停止、待机和关断模式。

5.4 RCC 寄存器

除非另外说明，否则 RCC 寄存器支持按字、半字和字节访问，没有任何等待周期。

5.4.1 RCC 时钟控制寄存器 (RCC_CR)

RCC clock control register

偏移地址：0x00

上电复位值：0x0000 0500

其他类型的复位：与上电复位相同，但 HSEBYP 位保留之前的值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSS ON	HSE BYP	HSE RDY	HSE ON
												rs	rw	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	HSIDIV[2:0]			HSI RDY	HSI KERON	HSION	HSIKERDIV[2:0]			Res.	Res.	Res.	Res.	Res.
		rw	rw	rw	r	rw	rw	rw	rw	rw					

位 31:20 保留，必须保持复位值。

位 19 **CSSON**：时钟安全系统使能 (Clock security system enable)

由软件置 1，用于使能时钟安全系统。当该位置 1 时，时钟检测器将在 HSE 振荡器就绪时由硬件使能，并在检出 HSE 时钟故障时由硬件禁止。复位后，该位由硬件清零。

0：禁止

1：使能

位 18 **HSEBYP**：HSE 晶振旁路 (HSE crystal oscillator bypass)

由软件置 1 和清零。

当该位置 1 时，将旁路内部 HSE 振荡器以使用外部时钟。外部时钟随后必须通过将 HSEON 位置 1 的方式使能。只有禁止 HSE 振荡器时，对该位的写访问才有效。

0：无旁路

1：旁路

位 17 **HSERDY**：HSE 时钟就绪标志 (HSE clock ready flag)

由硬件置 1，用以指示 HSE 振荡器已稳定，可供使用。

0：未就绪

1：就绪

注意：在将 HSEON 清零后，HSERDY 将在六个 HSE 时钟周期后转为低电平。

位 16 **HSEON**：HSE 时钟使能 (HSE clock enable)

由软件置 1 和清零。

由硬件清零，用于在进入停止、待机或关断模式时停止 HSE 振荡器。如果 HSE 振荡器直接或间接用作系统时钟，则该位不可清零。

0：禁止

1：使能

位 15:14 保留，必须保持复位值。

位 13:11 **HSIDIV[2:0]**：HSI48 时钟分频系数 (HSI48 clock division factor)

该位域由软件控制，用于设置 HSI48 时钟分频器的分频系数以生成 HSISYS 时钟：

000：1

001：2

010：4（复位值）

011：8

100：16

101：32

110：64

111：128

位 10 **HSIRDY**：HSI48 时钟就绪标志 (HSI48 clock ready flag)

当 HSI48 振荡器通过 HSION 使能后可供使用（稳定）时，由硬件置 1。

0：未就绪

1：就绪

注意：在将 HSION 位清零后，HSIRDY 将在六个 HSI48 时钟周期后转为低电平。

位 9 **HSIKERON**：始终为外设内核使能 HSI48 (HSI48 always-enable for peripheral kernels)

由软件置 1 和清零。

将该位置 1 可在运行模式和停止模式下激活 HSI48 振荡器，而与 HSION 位的状态无关。只有使用 HSI48 作为内核时钟的 USART1、USART2 和 I2C1 外设才能使用 HSI48 时钟。

0：HSI48 振荡器使能取决于 HSION 位

1：HSI48 振荡器在运行模式和停止模式下激活

注意：使 HSI48 在停止模式下保持激活可提高串行接口通信速度，因为 HSI48 时钟在退出停止模式后会立即就绪。

位 8 **HSION**：HSI48 时钟使能 (HSI48 clock enable)

由软件和硬件置 1 和清零，硬件优先。

只要器件处于低功耗模式，就由硬件保持低电平。

只要系统由 HSI48 生成的时钟驱动，就由硬件保持高电平。其中包括退出低功耗模式，以及被选作系统时钟源的 HSE 振荡器时钟发生故障后系统时钟切换回 HSI48。

0：禁止

1：使能

位 7:5 **HSIKERDIV[2:0]**：HSI48 内核时钟分频系数 (HSI48 kernel clock division factor)
该位域由软件控制，用于设置内核时钟分频器的分频系数以生成 HSIKER 时钟：
000：1
001：2
010：3（复位值）
011：4
100：5
101：6
110：7
111：8

位 4:0 保留，必须保持复位值。

5.4.2 RCC 内部时钟源校准寄存器 (RCC_ICSCR)

RCC internal clock source calibration register

偏移地址：0x04

复位值：0x0000 40XX

复位的 X 半字节因器件而异，因为这些半字节取决于器件出厂校准。

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

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

位 31:15 保留，必须保持复位值。

位 14:8 **HSITRIM[6:0]**：HSI48 时钟微调 (HSI48 clock trimming)
该位域的值影响 HSICAL[7:0] 位域的值。
用户可使用该位域来微调 HSI48 时钟频率。
当该位域保持其复位值时，器件数据手册中所述的 HSI48 频率精度适用。

位 7:0 **HSICAL[7:0]**：HSI48 时钟校准 (HSI48 clock calibration)
该位域直接作用于 HSI48 时钟频率。其值为内部出厂编程值与 HSITRIM[6:0] 位域值之和。出厂时，内部值设置为将 HSI48 时钟频率校准为 48 MHz（而 HSITRIM[6:0] 保持其复位值）。有关 HSI48 校准精度和频率微调粒度的信息，请参见器件数据手册。
注意：微调效果在 64 的 HSICAL[7:0] 倍处呈现不连续性。

5.4.3 RCC 时钟配置寄存器 (RCC_CFGR)

RCC clock configuration register

在时钟源切换后访问该寄存器时插入一个或两个等待周期，在更新 APB 或 AHB 预分频值后插入 0 到 15 个等待周期。

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.	MCOPRE[2:0]			Res.	MCOSEL[2:0]			Res.	MCO2PRE[2:0]			Res.	MCO2SEL[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PPRE[2:0]			HPRE[3:0]				Res.	Res.	SWS[2:0]			SW[2:0]		
	rw	rw	rw	rw	rw	rw	rw			r	r	r	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:28 **MCOPRE[2:0]**：微控制器时钟输出预分频器 (Microcontroller clock output prescaler)

该位域由软件控制，用于设置发送到 MCO 输出的时钟的分频系数，具体如下：

000：1

001：2

010：4

...

111：128

强烈建议在 MCO 输出使能之前设置该位域。

位 27 保留，必须保持复位值。

位 26:24 **MCOSEL[2:0]**：微控制器时钟输出时钟选择器 (Microcontroller clock output clock selector)

该位域由软件控制，用于设置 MCO 输出的时钟选择器，具体如下：

000：无时钟

001：SYSCLK

010：保留

011：HSI48

100：HSE

101：保留

110：LSI

111：LSE

注意：该时钟输出在启动时或 MCO 时钟源切换期间可能包含一些被截断的周期。

位 23 保留，必须保持复位值。

位 22:20 **MCO2PRE[2:0]**：微控制器时钟输出 2 预分频器 (Microcontroller clock output 2 prescaler)

该位域由软件控制，用于设置发送到 MCO2 输出的时钟的分频系数，具体如下：

000：1

001：2

010：4

...

111：128

强烈建议在 MCO2 输出使能之前设置该位域。

位 19 保留，必须保持复位值。

位 18:16 **MCO2SEL[2:0]**：微控制器时钟输出 2 时钟选择器 (Microcontroller clock output 2 clock selector)

该位域由软件控制，用于设置 MCO2 输出的时钟选择器，具体如下：

000：无时钟
001：SYSCLK
010：保留
011：HSI48
100：HSE
101：保留
110：LSI
111：LSE

注意：该时钟输出在启动时或 MCO2 时钟源切换期间可能包含一些被截断的周期。

位 15 保留，必须保持复位值。

位 14:12 **PPRE[2:0]**：APB 预分频器 (APB prescaler)

该位域由软件控制，用于设置 HCLK 时钟的分频系数以生成 PCLK 时钟，具体如下：

0xx：1
100：2
101：4
110：8
111：16

位 11:8 **HPRE[3:0]**：AHB 预分频器 (AHB prescaler)

该位域由软件控制，用于设置 SYSCLK 时钟的分频系数以生成 HCLK 时钟，具体如下：

0xxx：1
1000：2
1001：4
1010：8
1011：16
1100：64
1101：128
1110：256
1111：512

位 7:6 保留，必须保持复位值。

位 5:3 **SWS[2:0]**：系统时钟切换状态 (System clock switch status)

该位域由硬件控制，用于指示用作系统时钟的时钟源：

000：HSISYS
001：HSE
010：保留
011：LSI
100：LSE
其他值：保留

位 2:0 **SW[2:0]**：系统时钟切换 (System clock switch)

该位域由软件和硬件控制，用于选择 SYSCLK 的时钟，具体如下：

000：HSISYS
001：HSE
010：保留
011：LSI
100：LSE
其他值：保留

当 MCU 退出停止模式、待机模式或关断模式，或者当设置为 001（选择 HSE）且检测到 HSE 振荡器故障时，硬件会将该位域强制设置为 000（选择 HSISYS）。

5.4.4 RCC 时钟中断使能寄存器 (RCC_CIER)

RCC clock interrupt enable register

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSE RDYIE	HSI RDYIE	Res.	LSE RDYIE	LSI RDYIE
											rw	rw		rw	rw

位 31:5 保留，必须保持复位值。

位 4 **HSERDYIE**：HSE 就绪中断使能 (HSE ready interrupt enable)
由软件置 1 和清零，用于使能/禁止由 HSE 振荡器稳定所引起的中断：
0：禁止
1：使能

位 3 **HSIRDYIE**：HSI48 就绪中断使能 (HSI48 ready interrupt enable)
由软件置 1 和清零，用于使能/禁止由 HSI48 振荡器稳定所引起的中断：
0：禁止
1：使能

位 2 保留，必须保持复位值。

位 1 **LSERDYIE**：LSE 就绪中断使能 (LSE ready interrupt enable)
由软件置 1 和清零，用于使能/禁止由 LSE 振荡器稳定所引起的中断：
0：禁止
1：使能

位 0 **LSIRDYIE**：LSI 就绪中断使能 (LSI ready interrupt enable)
由软件置 1 和清零，用于使能/禁止由 LSI 振荡器稳定所引起的中断：
0：禁止
1：使能

5.4.5 RCC 时钟中断标志寄存器 (RCC_CIFR)

RCC clock interrupt flag register

偏移地址：0x1C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	LSE CSSF	CSSF	Res.	Res.	Res.	HSE RDYF	HSI RDYF	Res.	LSE RDYF	LSI RDYF
						r	r				r	r		r	r

位 31:10 保留，必须保持复位值。

位 9 **LSECSSF**：LSE 时钟安全系统中断标志 (LSE clock security system interrupt flag)
该标志指示 LSE 时钟故障后的挂起中断。
当在 LSE 振荡器中检出故障时由硬件置 1。
由软件将 LSECSSC 位置 1 的方式清零。
0：中断未挂起
1：中断挂起

位 8 **CSSF**：HSE 时钟安全系统中断标志 (HSE clock security system interrupt flag)
该标志指示 HSE 时钟故障后的挂起中断。
当在 HSE 振荡器中检出故障时由硬件置 1。
CSSC 位置 1 时由软件清零。
0：中断未挂起
1：中断挂起

位 7:5 保留，必须保持复位值。

位 4 **HSERDYF**：HSE 就绪中断标志 (HSE ready interrupt flag)
该标志指示 HSE 时钟就绪后的挂起中断。
当 HSE 时钟稳定且 HSERDYIE 置 1 时由硬件置 1。
HSERDYC 位置 1 时由软件清零。
0：中断未挂起
1：中断挂起

位 3 **HSIRDYF**：HSI48 就绪中断标志 (HSI48 ready interrupt flag)
该标志指示 HSI48 时钟就绪后的挂起中断。
当 HSI48 时钟稳定且 HSIRDYIE 置 1 时由硬件置 1，以对 HSION 置 1 作出响应（请参见 *RCC 时钟控制寄存器 (RCC_CR)*）。当 HSION 未置 1，但外设通过时钟请求使能 HSI48 振荡器时，该位不会置 1，也不会生成中断。
HSIRDYC 位置 1 时由软件清零。
0：中断未挂起
1：中断挂起

位 2 保留，必须保持复位值。

位 1 **LSERDYF**：LSE 就绪中断标志 (LSE ready interrupt flag)
该标志指示 LSE 时钟就绪后的挂起中断。
当 LSE 时钟稳定且 LSERDYDIE 置 1 时由硬件置 1。
LSERDYC 位置 1 时由软件清零。
0：中断未挂起
1：中断挂起

位 0 **LSIRDYF**：LSI 就绪中断标志 (LSI ready interrupt flag)
该标志指示 LSE 时钟就绪后的挂起中断。
当 LSI 时钟稳定且 LSIRDYDIE 置 1 时由硬件置 1。
LSIRDYC 位置 1 时由软件清零。
0：中断未挂起
1：中断挂起

5.4.6 RCC 时钟中断清零寄存器 (RCC_CICR)

RCC clock interrupt clear register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	LSE CSSC	CSSC	Res.	Res.	Res.	HSE RDYC	HSI RDYC	Res.	LSE RDYC	LSI RDYC
						w	w				w	w		w	w

位 31:10 保留，必须保持复位值。

位 9 **LSECSSC**：LSE 时钟安全系统中断清零 (LSE Clock security system interrupt clear)
该位由软件置 1，用于将 LSECSSF 标志清零。
0：无影响
1：将 LSECSSF 标志清零

位 8 **CSSC**：时钟安全系统中断清零 (Clock security system interrupt clear)
该位由软件置 1，用于将 HSECSSF 标志清零。
0：无影响
1：将 CSSF 标志清零

位 7:5 保留，必须保持复位值。

位 4 **HSERDYC**：HSE 就绪中断清零 (HSE ready interrupt clear)
该位由软件置 1，用于将 HSERDYF 标志清零。
0：无影响
1：将 HSERDYF 标志清零

位 3 **HSIRDYC**：HSI48 就绪中断清零 (HSI48 ready interrupt clear)
该位由软件置 1，用于将 HSIRDYF 标志清零。
0：无影响
1：将 HSIRDYF 标志清零

位 2 保留，必须保持复位值。

位 1 **LSERDYC**：LSE 就绪中断清零 (LSE ready interrupt clear)
该位由软件置 1，用于将 LSERDYF 标志清零。
0：无影响
1：将 LSERDYF 标志清零

位 0 **LSIRDYC**：LSI 就绪中断清零 (LSI ready interrupt clear)
该位由软件置 1，用于将 LSIRDYF 标志清零。
0：无影响
1：将 LSIRDYF 标志清零

5.4.7 RCC I/O 端口复位寄存器 (RCC_IOPRSTR)

RCC I/O port reset register

偏移地址：0x24

复位值：0x0000 0000

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

位 31:6 保留，必须保持复位值。

位 5 **GPIOFRST**：I/O 端口 F 复位 (I/O port F reset)
该位由软件置 1 和清零。
0：无影响
1：复位 I/O 端口 F

位 4 保留，必须保持复位值。

位 3 **GPIODRST**：I/O 端口 D 复位 (I/O port D reset)
该位由软件置 1 和清零。
0：无影响
1：复位 I/O 端口 D

位 2 **GPIOCRST**：I/O 端口 C 复位 (I/O port C reset)
该位由软件置 1 和清零。
0：无影响
1：复位 I/O 端口 C

位 1 **GPIOBRST**：I/O 端口 B 复位 (I/O port B reset)
该位由软件置 1 和清零。
0：无影响
1：复位 I/O 端口 B

位 0 **GPIOARST**：I/O 端口 A 复位 (I/O port A reset)
该位由软件置 1 和清零。
0：无影响
1：复位 I/O 端口 A

5.4.8 RCC AHB 外设复位寄存器 (RCC_AHBRSTR)

RCC AHB peripheral reset register

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRC RST	Res.	Res.	Res.	FLASH RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA1 RST
			rw				rw								rw

位 31:13 保留，必须保持复位值。

位 12 **CRCRST**：CRC 复位 (CRC reset)
由软件置 1 和清零。
0：无影响
1：复位 CRC

位 11:9 保留，必须保持复位值。

位 8 **FLASHRST**：Flash 接口复位 (Flash memory interface reset)
由软件置 1 和清零。
0：无影响
1：复位 Flash 接口
该位只能在 Flash 处于掉电模式时置 1。

位 7:1 保留，必须保持复位值。

位 0 **DMA1RST**：DMA1 和 DMAMUX 复位 (DMA1 and DMAMUX reset)
由软件置 1 和清零。
0：无影响
1：复位 DMA1 和 DMAMUX

5.4.9 RCC APB 外设复位寄存器 1 (RCC_APBRSTR1)

RCC APB peripheral reset register 1

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.	Res.	Res.	PWR RST	DBG RST	Res.	Res.	Res.	Res.	Res.	I2C1 RST	Res.	Res.	Res.	USART2 RST	Res.
			rw	rw						rw				rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM3 RST	Res.
														rw	

位 31:29 保留，必须保持复位值。

位 28 **PWRRST**：电源接口复位 (Power interface reset)
由软件置 1 和清零。
0：无影响
1：复位 PWR

位 27 **DBGRST**：调试支持复位 (Debug support reset)
由软件置 1 和清零。
0：无影响
1：复位 DBG

位 26:22 保留，必须保持复位值。

位 21 **I2C1RST**：I2C1 复位 (I2C1 reset)
由软件置 1 和清零。
0：无影响
1：复位 I2C1

位 20:18 保留，必须保持复位值。

位 17 **USART2RST**：USART2 复位 (USART2 reset)
由软件置 1 和清零。
0：无影响
1：复位 USART2

位 16:2 保留，必须保持复位值。

位 1 **TIM3RST**：TIM3 定时器复位 (TIM3 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM3

位 0 保留，必须保持复位值。

5.4.10 RCC APB 外设复位寄存器 2 (RCC_APBRSTR2)

RCC APB peripheral reset register 2

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC RST	Res.	TIM17 RST	TIM16 RST	Res.
											rw		rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM14 RST	USART1 RST	Res.	SPI1 RST	TIM1 RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYS CFG RST
rw	rw		rw	rw											rw

位 31:21 保留，必须保持复位值。

位 20 **ADCRST**：ADC 复位 (ADC reset)
由软件置 1 和清零。
0：无影响
1：复位 ADC

位 19 保留，必须保持复位值。

位 18 **TIM17RST**：TIM16 定时器复位 (TIM16 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM17 定时器

位 17 **TIM16RST**：TIM16 定时器复位 (TIM16 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM16 定时器

位 16 保留，必须保持复位值。

位 15 **TIM14RST**：TIM14 定时器复位 (TIM14 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM14 定时器

位 14 **USART1RST**：USART1 复位 (USART1 reset)
由软件置 1 和清零。
0：无影响
1：复位 USART1

位 13 保留，必须保持复位值。

位 12 **SPI1RST**：SPI1 复位 (SPI1 reset)
由软件置 1 和清零。
0：无影响
1：复位 SPI1

位 11 **TIM1RST**：TIM1 定时器复位 (TIM1 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM1 定时器

位 10:1 保留，必须保持复位值。

位 0 **SYSCFGRST**：SYSCFG 复位 (SYSCFG reset)
由软件置 1 和清零。
0：无影响
1：复位 SYSCFG

5.4.11 RCC I/O 端口时钟使能寄存器 (RCC_IOPENR)

RCC I/O port clock enable register

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOFEN	Res.	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
										rw		rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5 **GPIOFEN**：I/O 端口 F 时钟使能 (I/O port F clock enable)
该位由软件置 1 和清零。
0：禁止
1：使能

位 4 保留，必须保持复位值。

位 3 **GPIODEN**：I/O 端口 D 时钟使能 (I/O port D clock enable)
该位由软件置 1 和清零。
0：禁止
1：使能

位 2 **GPIOCEN**：I/O 端口 C 时钟使能 (I/O port C clock enable)
该位由软件置 1 和清零。
0：禁止
1：使能

位 1 **GPIOBEN**：I/O 端口 B 时钟使能 (I/O port B clock enable)
该位由软件置 1 和清零。
0：禁止
1：使能

位 0 **GPIOAEN**：I/O 端口 A 时钟使能 (I/O port A clock enable)
该位由软件置 1 和清零。
0：禁止
1：使能

5.4.12 RCC AHB 外设时钟使能寄存器 (RCC_AHBENR)

RCC AHB peripheral clock enable register

偏移地址：0x38

复位值：0x0000 0100

该寄存器单独使能 AHB 外设的时钟。在睡眠模式和停止模式下，通过该寄存器使能的时钟仅在 RCC_AHBSMENR 寄存器的相应位也置 1 时才提供给外设。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRC EN	Res.	Res.	Res.	FLASH EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA1 EN
			rw				rw								rw

位 31:16 保留，必须保持复位值。

位 15:13 保留，必须保持复位值。

位 12 **CRCEN**：CRC 时钟使能 (CRC clock enable)
由软件置 1 和清零。
0：禁止
1：使能

位 11:9 保留，必须保持复位值。

位 8 **FLASHEN**：Flash 接口时钟使能 (Flash memory interface clock enable)
由软件置 1 和清零。
0：禁止
1：使能
该位只能在 Flash 处于掉电模式时清零。

位 7:1 保留，必须保持复位值。

位 0 **DMA1EN**：DMA1 和 DMAMUX 时钟使能 (DMA1 and DMAMUX clock enable)
由软件置 1 和清零。
0：禁止
1：使能
只要至少一个 DMA 外设使能，DMAMUX 便会使能。

5.4.13 RCC APB 外设时钟使能寄存器 1 (RCC_APBENR1)

RCC APB peripheral clock enable register 1

偏移地址：0x3C

复位值：0x0000 0000

该寄存器单独使能 APB 外设的时钟。在睡眠模式和停止模式下，通过该寄存器使能的时钟仅在 RCC_APBSMENR1 寄存器的相应位也置 1 时才提供给外设。

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

位 31:29 保留，必须保持复位值。

位 28 **PWREN**：电源接口时钟使能 (Power interface clock enable)
由软件置 1 和清零。
0：禁止
1：使能

位 27 **DBGEN**：调试支持时钟使能 (Debug support clock enable)
由软件置 1 和清零。
0：禁止
1：使能

位 26:22 保留，必须保持复位值。

位 21 **I2C1EN**：I2C1 时钟使能 (I2C1 clock enable)
由软件置 1 和清零。
0：禁止
1：使能

位 20:18 保留，必须保持复位值。

位 17 **USART2EN**：USART2 时钟使能 (USART2 clock enable)
由软件置 1 和清零。
0：禁止
1：使能

位 16:12 保留，必须保持复位值。

位 11 **WWDGEN**：WWDG 时钟使能 (WWDG clock enable)
由软件置 1，用于使能窗口看门狗时钟。由硬件系统复位清零
0：禁止
1：使能
如果 WWDG_SW 选项位为 0，该位也可由硬件置 1。

位 10 **RTCAPBEN**：RTC APB 时钟使能 (RTC APB clock enable)
由软件置 1 和清零。
0：禁止
1：使能

位 9:2 保留，必须保持复位值。

位 1 **TIM3EN**：TIM3 定时器时钟使能 (TIM3 timer clock enable)
由软件置 1 和清零。
0：禁止
1：使能

位 0 保留，必须保持复位值。

5.4.14 RCC APB 外设时钟使能寄存器 2 (RCC_APBENR2)

RCC APB peripheral clock enable register 2

偏移地址：0x40

复位值：0x0000 0000

该寄存器单独使能 APB 外设的时钟。在睡眠模式和停止模式下，通过该寄存器使能的时钟仅在 RCC_APBSMENR2 寄存器的相应位也置 1 时才提供给外设。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC EN	Res.	TIM17 EN	TIM16 EN	Res.
											rw		rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM14 EN	USART1 EN	Res.	SPI1 EN	TIM1 EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYS CFG EN
rw	rw		rw	rw											rw

位 31:21 保留，必须保持复位值。

位 20 **ADCEN**：ADC 时钟使能 (ADC clock enable)
由软件置 1 和清零。
0：禁止
1：使能

位 19 保留，必须保持复位值。

位 18 **TIM17EN**：TIM16 定时器时钟使能 (TIM16 timer clock enable)
由软件置 1 和清零。
0：禁止
1：使能

位 17 **TIM16EN**：TIM16 定时器时钟使能 (TIM16 timer clock enable)
由软件置 1 和清零。
0：禁止
1：使能

位 16 保留，必须保持复位值。

位 15 **TIM14EN**：TIM14 定时器时钟使能 (TIM14 timer clock enable)
由软件置 1 和清零。
0：禁止
1：使能

位 14 **USART1EN**：USART1 时钟使能 (USART1 clock enable)
由软件置 1 和清零。
0：禁止
1：使能

位 13 保留，必须保持复位值。

位 12 **SPI1EN**：SPI1 时钟使能 (SPI1 clock enable)
由软件置 1 和清零。
0：禁止
1：使能

位 11 **TIM1EN**：TIM1 定时器时钟使能 (TIM1 timer clock enable)
由软件置 1 和清零。
0：禁止
1：使能

位 10:1 保留，必须保持复位值。

位 0 **SYSCFGEN**：SYSCFG 时钟使能 (SYSCFG clock enable)
由软件置 1 和清零。
0：禁止
1：使能

5.4.15 睡眠模式下的 RCC I/O 端口时钟使能寄存器 (RCC_IOPSMENR)

RCC I/O port in Sleep mode clock enable register

偏移地址：0x44

复位值：0x0000 002F

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

位 31:6 保留，必须保持复位值。

位 5 **GPIOFSMEN**：睡眠模式期间的 I/O 端口 F 时钟使能 (I/O port F clock enable during Sleep mode)

由软件置 1 和清零。

0：禁止

1：使能

位 4 保留，必须保持复位值。

位 3 **GPIODSMEN**：睡眠模式期间的 I/O 端口 D 时钟使能 (I/O port D clock enable during Sleep mode)

由软件置 1 和清零。

0：禁止

1：使能

位 2 **GPIOCSMEN**：睡眠模式期间的 I/O 端口 C 时钟使能 (I/O port C clock enable during Sleep mode)

由软件置 1 和清零。

0：禁止

1：使能

位 1 **GPIOBSMEN**：睡眠模式期间的 I/O 端口 B 时钟使能 (I/O port B clock enable during Sleep mode)

由软件置 1 和清零。

0：禁止

1：使能

位 0 **GPIOASMEN**：睡眠模式期间的 I/O 端口 A 时钟使能 (I/O port A clock enable during Sleep mode)

由软件置 1 和清零。

0：禁止

1：使能

5.4.16 睡眠/停止模式下的 RCC AHB 外设时钟使能寄存器 (RCC_AHBSMENR)

RCC AHB peripheral clock enable in Sleep/Stop mode register

偏移地址：0x48

复位值：0x0000 1301

该寄存器可单独编程在器件进入睡眠模式或停止模式后禁止哪些 AHB 外设时钟（位清零）。当该寄存器的某个位置 1（使能）时，将根据 RCC_AHBENR 寄存器的设置在睡眠模式或停止模式下提供相应的外设时钟。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRC SMEN	Res.	Res.	SRAM SMEN	FLASH SMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA1 SMEN
			rw			rw	rw								rw

位 31:13 保留，必须保持复位值。

位 12 **CRCSMEN**：睡眠模式期间的 CRC 时钟使能 (CRC clock enable during Sleep mode)
由软件置 1 和清零。
0：禁止
1：使能

位 11:10 保留，必须保持复位值。

位 9 **SRAMSMEN**：睡眠模式期间的 SRAM 时钟使能 (SRAM clock enable during Sleep mode)
由软件置 1 和清零。
0：禁止
1：使能

位 8 **FLASHSMEN**：睡眠模式期间的 Flash 接口时钟使能 (Flash memory interface clock enable during Sleep mode)
由软件置 1 和清零。
0：禁止
1：使能
仅当 Flash 处于掉电模式时，才可激活该位。

位 7:1 保留，必须保持复位值。

位 0 **DMA1SMEN**：睡眠模式期间的 DMA1 和 DMAMUX 时钟使能 (DMA1 and DMAMUX clock enable during Sleep mode)
由软件置 1 和清零。
0：禁止
1：使能
只要至少一个 DMA 外设在睡眠模式下使能了时钟，睡眠模式期间就会使能 DMAMUX 时钟。

5.4.17 睡眠/停止模式下的 RCC APB 外设时钟使能寄存器 1 (RCC_APBSMENR1)

RCC APB peripheral clock enable in Sleep/Stop mode register 1

偏移地址：0x4C

复位值：0x18220C02：

该寄存器可单独编程在器件进入睡眠模式或停止模式后禁止哪些 APB 外设时钟（位清零）。当该寄存器的某个位置 1（使能）时，将根据 RCC_APBENR1 寄存器的设置在睡眠模式或停止模式下提供相应的外设时钟。

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

位 31:29 保留，必须保持复位值。

位 28 **PWRSMEN**：睡眠模式期间的电源接口时钟使能 (Power interface clock enable during Sleep mode)

由软件置 1 和清零。
0：禁止
1：使能

位 27 **DBGSMEN**：睡眠模式期间的调试支持时钟使能 (Debug support clock enable during Sleep mode)

由软件置 1 和清零。
0：禁止
1：使能

位 26:22 保留，必须保持复位值。

位 21 **I2C1SMEN**：睡眠和停止模式期间的 I2C1 时钟使能 (I2C1 clock enable during Sleep and Stop modes)

由软件置 1 和清零。
0：禁止
1：使能

位 20:18 保留，必须保持复位值。

位 17 **USART2SMEN**：睡眠和停止模式期间的 USART2 时钟使能 (USART2 clock enable during Sleep and Stop modes)

由软件置 1 和清零。
0：禁止
1：使能

位 16:12 保留，必须保持复位值。

位 11 **WWDGSMEN**：睡眠和停止模式期间的 WWDG 时钟使能 (WWDG clock enable during Sleep and Stop modes)

由软件置 1 和清零。
0：禁止
1：使能

位 10 **RTCAPBSMEN**：睡眠模式期间的 RTC APB 时钟使能 (RTC APB clock enable during Sleep mode)

由软件置 1 和清零。

0：禁止

1：使能

位 9:2 保留，必须保持复位值。

位 1 **TIM3SMEN**：睡眠模式期间的 TIM3 定时器时钟使能 (TIM3 timer clock enable during Sleep mode)

由软件置 1 和清零。

0：禁止

1：使能

位 0 保留，必须保持复位值。

5.4.18 睡眠/停止模式下的 RCC APB 外设时钟使能寄存器 2 (RCC_APBSMENR2)

RCC APB peripheral clock enable in Sleep/Stop mode register 2

偏移地址：0x50

复位值：0x0017 D801

该寄存器可单独编程在器件进入睡眠模式或停止模式后禁止哪些 APB 外设时钟（位清零）。当该寄存器的某个位置 1（使能）时，将根据 RCC_APBENR2 寄存器的设置在睡眠模式或停止模式下提供相应的外设时钟。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC SMEN	Res.	TIM17 SMEN	TIM16 SMEN	Res.
											rw		rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM14 SMEN	USART1 SMEN	Res.	SPI1 SMEN	TIM1 SMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYS CFG SMEN
rw	rw		rw	rw											rw

位 31:21 保留，必须保持复位值。

位 20 **ADCSMEN**：睡眠模式期间的 ADC 时钟使能 (ADC clock enable during Sleep mode)

由软件置 1 和清零。

0：禁止

1：使能

位 19 保留，必须保持复位值。

位 18 **TIM17SMEN**：睡眠模式期间的 TIM17 定时器时钟使能 (TIM17 timer clock enable during Sleep mode)

由软件置 1 和清零。

0：禁止

1：使能

位 17 **TIM16SMEN**：睡眠模式期间的 TIM16 定时器时钟使能 (TIM16 timer clock enable during Sleep mode)

由软件置 1 和清零。

0：禁止

1：使能

位 16 保留，必须保持复位值。

位 15 **TIM14SMEN**：睡眠模式期间的 TIM14 定时器时钟使能 (TIM14 timer clock enable during Sleep mode)

由软件置 1 和清零。
0：禁止
1：使能

位 14 **USART1SMEN**：睡眠和停止模式期间的 USART1 时钟使能 (USART1 clock enable during Sleep and Stop modes)

由软件置 1 和清零。
0：禁止
1：使能

位 13 保留，必须保持复位值。

位 12 **SPI1SMEN**：睡眠模式期间的 SPI1 时钟使能 (SPI1 clock enable during Sleep mode)

由软件置 1 和清零。
0：禁止
1：使能

位 11 **TIM1SMEN**：睡眠模式期间的 TIM1 定时器时钟使能 (TIM1 timer clock enable during Sleep mode)

由软件置 1 和清零。
0：禁止
1：使能

位 10:1 保留，必须保持复位值。

位 0 **SYSCFGSMEN**：睡眠和停止模式期间的 SYSCFG 时钟使能 (WWDG clock enable during Sleep and Stop modes)

由软件置 1 和清零。
0：禁止
1：使能

5.4.19 RCC 外设独立时钟配置寄存器 (RCC_CCIPR)

RCC peripherals independent clock configuration register

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
ADCSEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2S1SEL[1:0]		I2C1SEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USART1SEL [1:0]	
rw	rw	rw	rw											rw	rw

位 31:30 **ADCSEL[1:0]**：ADC 时钟源选择 (ADCs clock source selection)
该位域由软件控制，用于选择 ADC 的异步时钟源：
00：系统时钟
01：保留
10：HSIKER
11：保留

位 29:16 保留，必须保持复位值。

位 15:14 **I2S1SEL[1:0]**：I2S1 时钟源选择 (RTC clock source selection)
该位域由软件控制，用于选择 I2S1 时钟源，具体如下：
00：SYSCLK
01：保留
10：HSIKER
11：I2S_CKIN

位 13:12 **I2C1SEL[1:0]**：I2C1 时钟源选择 (RTC clock source selection)
该位域由软件控制，用于选择 I2C1 时钟源，具体如下：
00：PCLK
01：SYSCLK
10：HSIKER
11：保留

位 11:2 保留，必须保持复位值。

位 1:0 **USART1SEL[1:0]**：USART1 时钟源选择 (RTC clock source selection)
该位域由软件控制，用于选择 USART1 时钟源，具体如下：
00：PCLK
01：SYSCLK
10：HSIKER
11：LSE

5.4.20 RCC 控制/状态寄存器 1 (RCC_CSR1)

RCC control/status register 1

连续访问该寄存器时，最多插入三个等待周期。

除了 LSCOSEL、LSCOEN 和 RTCRST 位仅在 RTC 域上电复位后复位之外，寄存器的其他位都仅在 RTC 域复位（请参见*第 5.1.3 节：RTC 域复位*）后复位。内部复位或外部复位对这些位不会有任何影响。

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.	Res.	Res.	Res.	Res.	Res.	LSCOSEL	LSCOEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTCRST
						rw	rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	Res.	Res.	Res.	Res.	Res.	RTCSEL[1:0]		Res.	LSECSSD	LSECSSON	Res.	LSEDRV	LSEBYP	LSERDY	LSEON
rw						rw	rw		r	rw		rw	rw	r	rw

位 31:26 保留，必须保持复位值。

位 25 **LSCOSEL**：低速时钟输出选择 (Low-speed clock output selection)
由软件置 1 和清零，用于选择低速输出时钟：
0：LSI
1：LSE

位 24 **LSCOEN**：低速时钟输出 (LSCO) 使能 (Low-speed clock output (LSCO) enable)
由软件置 1 和清零。
0：禁止
1：使能

位 23:17 保留，必须保持复位值。

位 16 **RTCRST**：RTC 域软件复位 (RTC domain software reset)
由软件置 1 和清零，用于复位 RTC 域：
0：无影响
1：复位

位 15 **RTCEN**：RTC 时钟使能 (RTC clock enable)
由软件置 1 和清零。该位用于使能 RTC 和 TAMP 的时钟。
0：禁止
1：使能

位 14:10 保留，必须保持复位值。

位 9:8 **RTCSEL[1:0]**：RTC 时钟源选择 (RTC clock source selection)
由软件置 1，用于选择 RTC 的时钟源，具体如下：
00：无时钟
01：LSE
10：LSI
11：HSE 32 分频
选择 RTC 时钟源后，除非 RTC 域复位或在 LSE 上检测到故障（LSECSSD 置 1），否则不可再将其更改。可使用 RTCRST 位将该位域复位为 00。

位 7 保留，必须保持复位值。

位 6 **LSECSSD**：LSE 上的 CSS 故障检测 (CSS on LSE failure Detection)
由硬件置 1，用于指示外部 32 kHz 振荡器 (LSE) 上的时钟安全系统何时检测到故障：
0：未检测到故障
1：检测到故障

位 5 **LSECSSON**：LSE 上的 CSS 使能 (CSS on LSE enable)
由软件置 1，用于使能 LSE (32 kHz) 振荡器上的时钟安全系统，具体如下：
0：禁止
1：使能
在 LSE 振荡器使能（已使能 LSEON 位）和就绪（由硬件将 LSERDY 标志置 1）后以及在选择 RTCSEL 位后，LSECSSON 必须使能。
该位一旦使能便不能禁止，但检测到 LSE 故障 (LSECSSD =1) 后除外。此时，软件**必须**禁止 LSECSSON 位。

位 4 保留，必须保持复位值。

位 3 **LSEDRV**：LSE 振荡器驱动能力 (LSE oscillator drive capability)

由软件置 1，用于选择 LSE 振荡器的驱动能力，具体如下：

0：中高驱动能力

1：高驱动能力

在 LSE 振荡器处于 Xtal 模式（而非旁路模式）时适用。

位 2 **LSEBYP**：LSE 振荡器旁路 (LSE oscillator bypass)

由软件置 1 和清零，用于旁路 LSE 振荡器（在调试模式下）。

0：不旁路

1：旁路

只有在禁止外部 32 kHz 振荡器（LSEON=0 且 LSERDY=0）后才能写入该位。

位 1 **LSERDY**：LSE 振荡器就绪 (LSE oscillator ready)

由硬件置 1 和清零，用于指示外部 32 kHz 振荡器已就绪（稳定）：

0：未就绪

1：就绪

在 LSEON 位被清零后，LSERDY 将在 6 个外部低速振荡器时钟周期后转为低电平。

位 0 **LSEON**：LSE 振荡器使能 (LSE oscillator enable)

由软件置 1 和清零，用于使能 LSE 振荡器：

0：禁止

1：使能

5.4.21 RCC 控制/状态寄存器 2 (RCC_CSR2)

RCC control/status register 2

连续访问该寄存器时，最多插入三个等待周期。该寄存器在系统复位后复位，但有些复位标志仅在上电复位后复位。

偏移地址：0x60

复位值：0xXX00 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWR RSTF	WWDG RSTF	IWDG RSTF	SFT RSTF	PWR RSTF	PIN RSTF	OBL RSTF	Res.	RMVF	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r	r	r		rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSI RDY	LSION
														r	rw

位 31 **LPWRRSTF**：低功耗复位标志 (Low-power reset flag)

由于进入非法停止、待机或关断模式而发生复位时，由硬件置 1。

通过将 RMVF 位置 1 来清零。

0：未发生非法模式复位

1：发生了非法模式复位

只有 nRST_STOP、nRST_STDBY 或 nRST_SHDW 选项位清零时，该标志才起作用。

位 30 **WWDGRSTF**：窗口看门狗复位标志 (Window watchdog reset flag)

发生窗口看门狗复位时，由硬件置 1。

通过将 RMVF 位置 1 来清零。

0：未发生窗口看门狗复位

1：发生窗口看门狗复位

位 29 **IWDGRSTF**：独立窗口看门狗复位标志 (Independent window watchdog reset flag)
发生独立看门狗复位时，由硬件置 1。
通过将 RMVF 位置 1 来清零。
0：未发生独立看门狗复位
1：发生了独立看门狗复位

位 28 **SFTRSTF**：软件复位标志 (Software reset flag)
发生软件复位时，由硬件置 1。
通过将 RMVF 位置 1 来清零。
0：未发生软件复位
1：发生软件复位

位 27 **PWRRSTF**：BOR 或 POR/PDR 标志 (BOR or POR/PDR flag)
发生 BOR 或 POR/PDR 时，由硬件置 1。
通过将 RMVF 位置 1 来清零。
0：未发生 BOR 或 POR
1：发生了 BOR 或 POR

位 26 **PINRSTF**：引脚复位标志 (Pin reset flag)
发生来自 NRST 引脚的复位时，由硬件置 1。
通过将 RMVF 位置 1 来清零。
0：未发生来自 NRST 引脚的复位
1：发生来自 NRST 引脚的复位

位 25 **OBLRSTF**：选项字节加载器复位标志 (Option byte loader reset flag)
发生来自选项字节加载的复位时，由硬件置 1。
通过将 RMVF 位置 1 来清零。
0：未发生来自选项字节加载的复位
1：发生了来自选项字节加载的复位

位 24 保留，必须保持复位值。

位 23 **RMVF**：清除复位标志 (Remove reset flags)
由软件置 1，用于将复位标志清零。
0：无影响
1：清零复位标志

位 22:2 保留，必须保持复位值。

位 1 **LSIRDY**：LSI 振荡器就绪 (LSI oscillator ready)
由硬件置 1 和清零，用于指示 LSI 振荡器已就绪（稳定）：
0：未就绪
1：就绪
在将 LSION 位清零后，LSIRDY 将在 3 个 LSI 振荡器时钟周期后转为低电平。如果存在 LSE 上的时钟安全系统、独立看门狗或 RTC 发出的 LSI 请求，即使 LSION = 0，该位也可置 1。

位 0 **LSION**：LSI 振荡器使能 (LSI oscillator enable)
由软件置 1 和清零，用于使能/禁止 LSI 振荡器：
0：禁止
1：使能

5.4.22 RCC 寄存器映射

下表列出了 RCC 寄存器映射和复位值。

表 23. RCC 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	RCC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSSON	HSEBYP	HSERDY	HSEON	Res.	Res.	HSIDIV[2:0]			HSIRDY	HSIKERON	HSION	HSIKERDIV[2:0]		Res.	Res.	Res.	Res.	Res.	Res.
	Reset value													0	0	0	0			0	0	0	1	0	1	0	0	0					
0x04	RCC_ICSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSITRIM[6:0]							HSICAL[7:0]							
	Reset value																		1	0	0	0	0	0	0	X	X	X	X	X	X	X	X
0x08	RCC_CFGR	Res.	MCOPRE[3:0]			Res.	MCOSEL[2:0]			Res.	MCO2PRE[3:0]			Res.	MCO2SEL[2:0]			Res.	PPRE[2:0]			HPRE[3:0]				Res.	Res.	SWS[2:0]			SW[2:0]		
	Reset value		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0	0	0	0	0			0	0	0	0	0	0
0x0C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x10	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x14	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x18	RCC_CIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSERDYIE	HSIRDYIE	Res.	LSERDYIE	LSIRDYIE
	Reset value																												0	0		0	0
0x1C	RCC_CIFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSECSSF	CSSF	Res.	Res.	Res.	HSERDYF	HSIRDYF	Res.	LSERDYF	LSIRDYF
	Reset value																							0	0				0	0		0	0
0x20	RCC_CICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSECSSC	CSSC	Res.	Res.	Res.	HSERDYC	HSIRDYC	Res.	LSERDYC	LSIRDYC
	Reset value																							0	0				0	0		0	0
0x24	RCC_ IOPRSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOFRST	Res.	GPIODRST	GPIOCRST	GPIOBRST	GPIOARST
	Reset value																											0		0	0	0	0
0x28	RCC_ AHBRSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCRST	Res.	Res.	Res.	FLASHRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA1RST
	Reset value																				0				0								0

表 23. RCC 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x2C	RCC_ APBRSTR1	Res.	Res.	Res.	PWRRST	DBGRST	Res.	Res.	Res.	Res.	Res.	I2C1RST	Res.	Res.	Res.	USART2RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM3RST	Res.
	Reset value				0	0						0				0																0	
0x30	RCC_ APBRSTR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADCRST	Res.	TIM17RST	TIM16RST	Res.	TIM14RST	USART1RST	Res.	SPI1RST	TIM1RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGRST
	Reset value												0		0	0		0	0		0	0												0
0x34	RCC_ IOPENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOFEN	Res.	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
	Reset value																											0		0	0	0	0
0x38	RCC_ AHBENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCEN	Res.	Res.	Res.	FLASHEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA1EN
	Reset value																				0				1								0
0x3C	RCC_ APBENR1	Res.	Res.	Res.	PWREN	DBGEN	Res.	Res.	Res.	Res.	Res.	I2C1EN	Res.	Res.	Res.	USART2EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM3EN	Res.
	Reset value				0	0						0				0																0	
0x40	RCC_ APBENR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADCEN	Res.	TIM17EN	TIM16EN	Res.	TIM14EN	USART1EN	Res.	SPI1EN	TIM1EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGEN
	Reset value												0		0	0		0	0		0	0												0
0x44	RCC_ IOPSMENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOFSMEN	Res.	GPIODSMEN	GPIOCSMEN	GPIOBSMEN	GPIOASMEN
	Reset value																											1		1	1	1	1
0x48	RCC_ AHBSMENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCSMEN	Res.	Res.	SRAMSMEN	FLASHSMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA1SMEN
	Reset value																				1			1	1								1
0x4C	RCC_ APBSMENR1	Res.	Res.	Res.	PWRSMEN	DBGSMEN	Res.	Res.	Res.	Res.	Res.	I2C1SMEN	Res.	Res.	Res.	USART2SMEN	Res.	Res.	Res.	Res.	Res.	WWDGSMEN	RTCAPBSMEN.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM3SMEN	Res.
	Reset value				1	1						1				1																1	

表 23. RCC 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x50	RCC_ APBSMENR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADCSMEN	Res.	TIM17SMEN	TIM16SMEN	Res.	TIM14SMEN	USART1SMEN	Res.	SPI1SMEN	TIM1SMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGSMEN
	Reset value												1		1	1		1	1		1	1											1
0x54	RCC_CCIPR	ADCSEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2S1SEL[1:0]		I2C1SEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USART1SEL[1:0]	
	Reset value	0	0															0	0	0	0											0	0
0x58	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x5C	RCC_CSR1	Res.	Res.	Res.	Res.	Res.	Res.	LSCOSEL	LSCOEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTCRST	RTCEN	Res.	Res.	Res.	Res.	Res.	RTC SEL[1:0]		Res.	LSECSSD	LSECSSON	Res.	LSEDRV	LSEBYP	LSERDY	LSEON
	Reset value							0	0								0	0						0	0		0	0		0	0	0	0
0x60	RCC_CSR2	LPWRRSTF	WWDGRSTF	IWDGRSTF	SFTRSTF	PWRRSTF	PINRSTF	OBLRSTF	Res.	RMVF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSIRDY	LSION
	Reset value	0	0	0	0	0	0	0		0																						0	0

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

6 通用 I/O (GPIO)

6.1 简介

每个通用 I/O 端口包括 4 个 32 位配置寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR 和 GPIOx_PUPDR）、2 个 32 位数据寄存器（GPIOx_IDR 和 GPIOx_ODR）和 1 个 32 位置位/复位寄存器 (GPIOx_BSRR)。此外，所有 GPIO 都包括 1 个 32 位锁定寄存器 (GPIOx_LCKR) 和 2 个 32 位复用功能选择寄存器（GPIOx_AFRH 和 GPIOx_AFRL）。

6.2 GPIO 主要特性

- 输出状态：推挽或开漏 + 上拉/下拉
- 从输出数据寄存器 (GPIOx_ODR) 或外设（复用功能输出）输出数据
- 可为每个 I/O 选择不同的速度
- 输入状态：浮空、上拉/下拉、I/O 模拟模式
- 将数据输入到输入数据寄存器 (GPIOx_IDR) 或外设（复用功能输入）
- 置位和复位寄存器 (GPIOx_BSRR)，对 GPIOx_ODR 具有按位写权限
- 锁定机制 (GPIOx_LCKR)，可冻结 I/O 端口配置
- 模拟功能
- 复用功能选择寄存器（一个 I/O 最多可具有 16 个复用功能）
- 快速翻转，每次翻转最快只需要两个时钟周期
- 引脚复用非常灵活，允许将 I/O 引脚用作 GPIO 或多种外设功能中的一种

6.3 GPIO 功能描述

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

- 输入浮空
- 输入上拉
- 输入下拉
- 模拟
- 具有上拉或下拉功能的开漏输出
- 具有上拉或下拉功能的推挽输出
- 具有上拉或下拉功能的复用功能推挽
- 具有上拉或下拉功能的复用功能开漏

每个 I/O 端口位均可自由编程，但 I/O 端口寄存器必须按 32 位字、半字或字节进行访问。GPIOx_BSRR 寄存器和 GPIOx_BRR 寄存器旨在实现对 GPIOx_ODR 寄存器进行原子读取/修改访问。这样便可确保在读取和修改访问之间发生中断请求也不会有问题。

图 14 显示了标准 I/O 端口位的基本结构。*表 24* 给出了可能的端口位配置方案。

图 14. I/O 端口位的基本结构

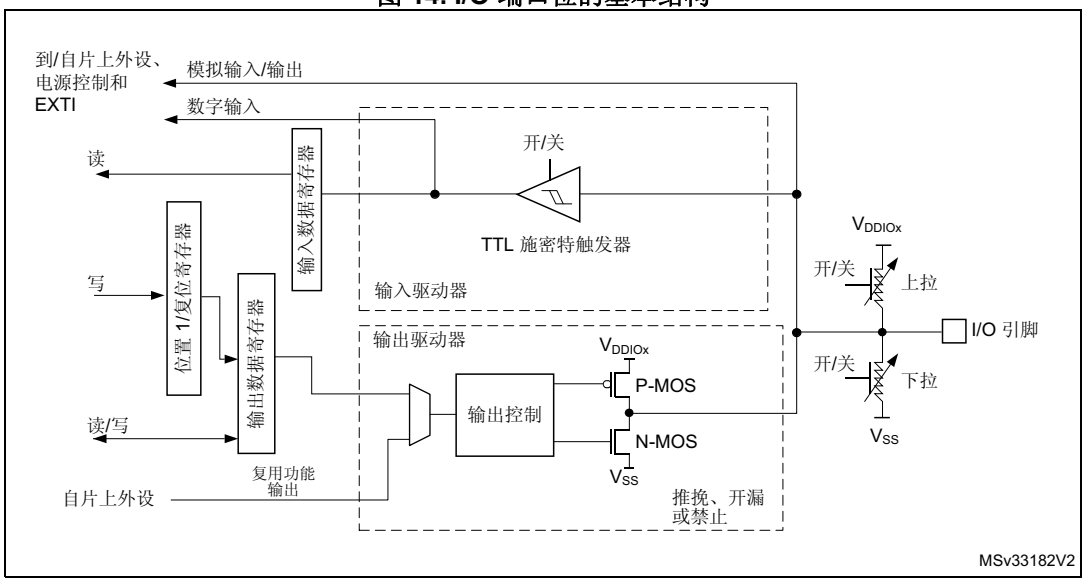


表 24. 端口位配置表(1)

MODE(i) [1:0]	OTYPE(i)	OSPEED(i) [1:0]	PUPD(i) [1:0]		I/O 配置	
01	0	SPEED [1:0]	0	0	GP 输出	PP
	0		0	1	GP 输出	PP + PU
	0		1	0	GP 输出	PP + PD
	0		1	1	保留	
	1		0	0	GP 输出	OD
	1		0	1	GP 输出	OD + PU
	1		1	0	GP 输出	OD + PD
	1		1	1	保留（GP 输出 OD）	
10	0	SPEED [1:0]	0	0	AF	PP
	0		0	1	AF	PP + PU
	0		1	0	AF	PP + PD
	0		1	1	保留	
	1		0	0	AF	OD
	1		0	1	AF	OD + PU
	1		1	0	AF	OD + PD
	1		1	1	保留	

表 24. 端口位配置表(1)（续）

MODE(i) [1:0]	OTYPE(i)	OSPEED(i) [1:0]		PUPD(i) [1:0]		I/O 配置	
00	x	x	x	0	0	输入	浮空
	x	x	x	0	1	输入	PU
	x	x	x	1	0	输入	PD
	x	x	x	1	1	保留（输入浮空）	
11	x	x	x	0	0	输入/输出	模拟
	x	x	x	0	1	保留	
	x	x	x	1	0		
	x	x	x	1	1		

1. GP = 通用、PP = 推挽、PU = 上拉、PD = 下拉、OD = 开漏、AF = 复用功能。

6.3.1 通用 I/O (GPIO)

在复位期间及复位刚刚完成后，复用功能尚未激活，大多数 I/O 端口被配置为模拟模式。

复位后，调试引脚处于复用功能上拉/下拉状态：

- PA14：SWCLK 处于下拉状态
- PA13：SWDIO 处于上拉状态

注意： *PA14 与 BOOT0 功能共用。使用时需小心谨慎，因为调试器件可能会篡改 BOOT0 引脚值。*

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

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

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

GPIO 引脚支持以下工作模式：

- **GPIO**：输出、输入或模拟 I/O，具体取决于 GPIOx_MODER 寄存器设置
- **复用功能**

  具有调试复用功能的 GPIO 在复位后设为复用功能模式。
- **附加功能**

  仅限部分 GPIO 引脚，附加功能模式通过相应功能模块（例如 ADC、DAC、RTC、RCC 和 PWR）的控制寄存器来设置，与 GPIOx_MODER 寄存器设置无关。

  当某个 I/O 设为附加功能模式时，建议在 GPIOx_MODER 寄存器中将其相应的 GPIO 复用器设为模拟模式。

6.3.2 I/O 引脚复用功能复用器和映射

每个连接到器件引脚作为复用功能的功能模块信号都会在内部连接到多个 GPIO 引脚。每个 GPIO 引脚都有一个复用器，每个复用器有 16 个位置（AF0 到 AF15），通过 GPIOx_AFRL 和 GPIOx_AFRH 寄存器进行控制，每次最多可以在 16 个复用功能中选择一个。针对 GPIO 引脚选择的复用功能通过GPIO 模式复用器（由 GPIOx_MODER 寄存器控制 ）物理连接到引脚。

复位后，每个 GPIO 上的复用功能复用器都位于 AF0 位置。

这种灵活性可简化 PCB 布线，即使是低引脚数的器件在经过灵活配置后也可以满足应用要求。

有关将复用功能信号映射到 GPIO 复用功能复用器的详细信息，请参见器件数据手册。

6.3.3 I/O 端口控制寄存器

每个 GPIO 端口有 4 个 32 位存储器映射的控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR），可配置多达 16 个 I/O。GPIOx_MODER 寄存器用于选择 I/O 模式（输入、输出、AF、模拟）。GPIOx_OTYPER 和 GPIOx_OSPEEDR 寄存器用于选择输出类型（推挽或开漏）和速度。无论采用哪种 I/O 方向，GPIOx_PUPDR 寄存器都用于选择上拉/下拉。

6.3.4 I/O 端口数据寄存器

每个 GPIO 都具有 2 个 16 位存储器映射的数据寄存器：输入和输出数据寄存器（GPIOx_IDR 和 GPIOx_ODR）。GPIOx_ODR 用于存储待输出数据，可对其进行读/写访问。通过 I/O 输入的数据存储到输入数据寄存器 (GPIOx_IDR) 中，它是一个只读寄存器。

有关寄存器说明的详细信息，请参见*第 6.4.5 节：GPIO 端口输入数据寄存器 (GPIOx_IDR)（x = A、B、C、D 和 F）*和*第 6.4.6 节：GPIO 端口输出数据寄存器 (GPIOx_ODR)（x = A、B、C、D 和 F）*。

6.3.5 I/O 数据位操作

置 1 复位寄存器 (GPIOx_BSRR) 是一个 32 位寄存器，它允许应用程序在输出数据寄存器 (GPIOx_ODR) 中对各个单独的数据位执行置 1 和复位操作。置位复位寄存器的大小是 GPIOx_ODR 的二倍。

GPIOx_ODR 中的每个数据位对应于 GPIOx_BSRR 中的两个控制位：BS(i) 和 BR(i)。当写入 1 时，BS(i) 位会将对应的 ODR(i) 位**置 1**。当写入 1 时，BR(i) 位会**复位** ODR(i) 对应的位。

在 GPIOx_BSRR 中向任何位写入 0 都不会对 GPIOx_ODR 中的对应位产生任何影响。如果在 GPIOx_BSRR 中同时尝试对某个位执行置位和复位操作，则置位操作优先。

使用 GPIOx_BSRR 寄存器更改 GPIOx_ODR 中各个位的值是一个“单次”操作，不会锁定 GPIOx_ODR 位。随时都可以直接访问 GPIOx_ODR 位。GPIOx_BSRR 寄存器提供了一种执行原子按位处理的方法。

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

6.3.6 GPIO 锁定机制

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

要对 GPIOx_LCKR 寄存器执行写操作，必须应用特定的写/读序列。当正确的 LOCK 序列应用到此寄存器的第 16 位后，会使用 LCKR[15:0] 的值来锁定 I/O 的配置（在写序列期间，LCKR[15:0] 的值必须相同）。将 LOCK 序列应用到某个端口位后，在执行下一次 MCU 复位或外设复位之前，将无法对该端口位的值进行修改。每个 GPIOx_LCKR 位都会冻结控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_AFRH）中的对应位。

LOCK 序列（参见*第 6.4.8 节：GPIO 端口配置锁定寄存器 (GPIOx_LCKR)（x = A、B、C、D 和 F）*）只能通过对 GPIOx_LCKR 寄存器进行字（32 位长）访问的方式来执行，因为 GPIOx_LCKR 的第 16 位必须与 [15:0] 位同时置位。

有关详细信息，请参见*第 6.4.8 节：GPIO 端口配置锁定寄存器 (GPIOx_LCKR)（x = A、B、C、D 和 F）*中的 LCKR 寄存器说明。

6.3.7 I/O 复用功能输入/输出

当某个 I/O 引脚处于复用功能模式时，所选的复用功能将确定该引脚是用作输入还是输出。

上拉/下拉和输出速度设置（分别通过 GPIOx_OTYPER、GPIOx_PUPDR 和 GPIOx_OSPEEDER 寄存器来实现）保持有效。

6.3.8 外部中断线/唤醒线

所有端口都具有外部中断功能。要使用外部中断线，不得将给定的引脚配置为模拟模式，也不得将其用作振荡器引脚，以便输入触发器保持使能状态。

请参见*第 12 节：扩展中断和事件控制器 (EXTI)*。

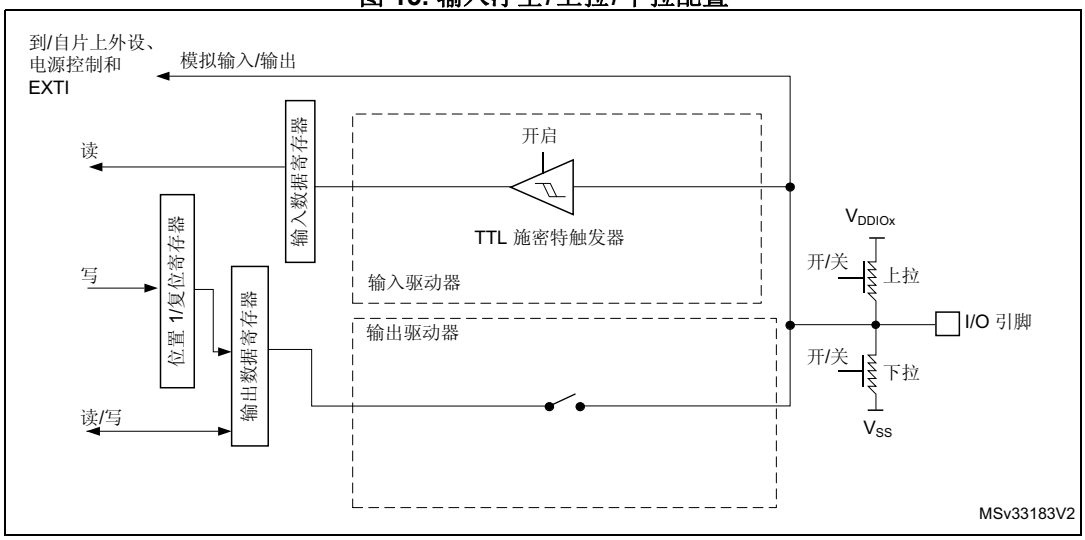
6.3.9 输入配置

对 I/O 端口进行编程作为输入时：

- 输出缓冲器被禁止
- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开上拉和下拉电阻
- 输入数据寄存器每隔 1 个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读访问可获取 I/O 状态

图 15 说明了 I/O 端口位的输入配置。

图 15. 输入浮空/上拉/下拉配置



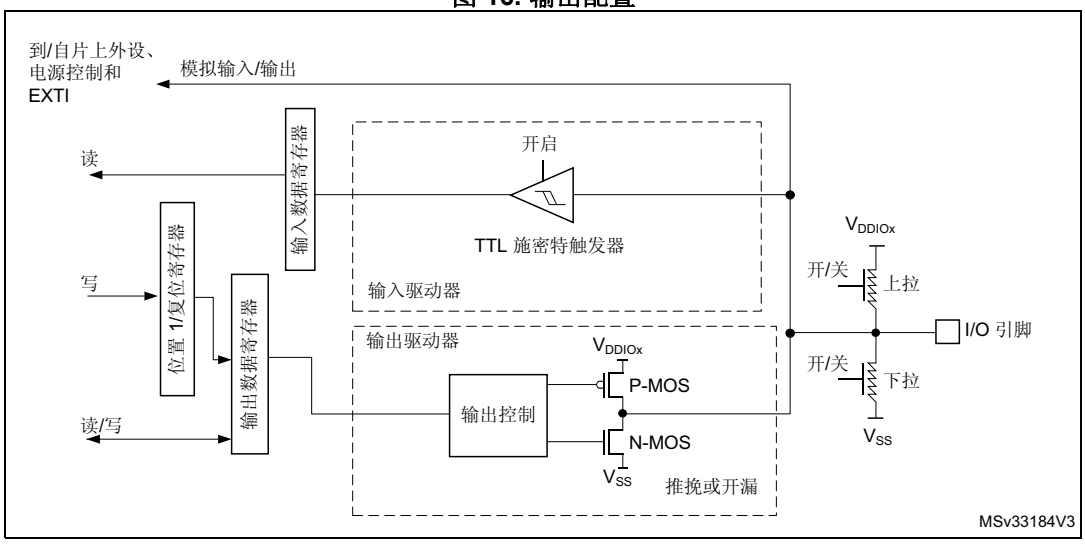
6.3.10 输出配置

对 I/O 端口进行编程作为输出时：

- 输出缓冲器被打开：
  - 开漏模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”会使端口保持高组态 (Hi-Z)（P-MOS 始终不激活）。
  - 推挽模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”可激活 P-MOS
- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开上拉和下拉电阻
- 输入数据寄存器每隔 1 个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读访问可获取 I/O 状态
- 对输出数据寄存器的读访问可获取最后的写入值

图 16 说明了 I/O 端口位的输出配置。

图 16. 输出配置



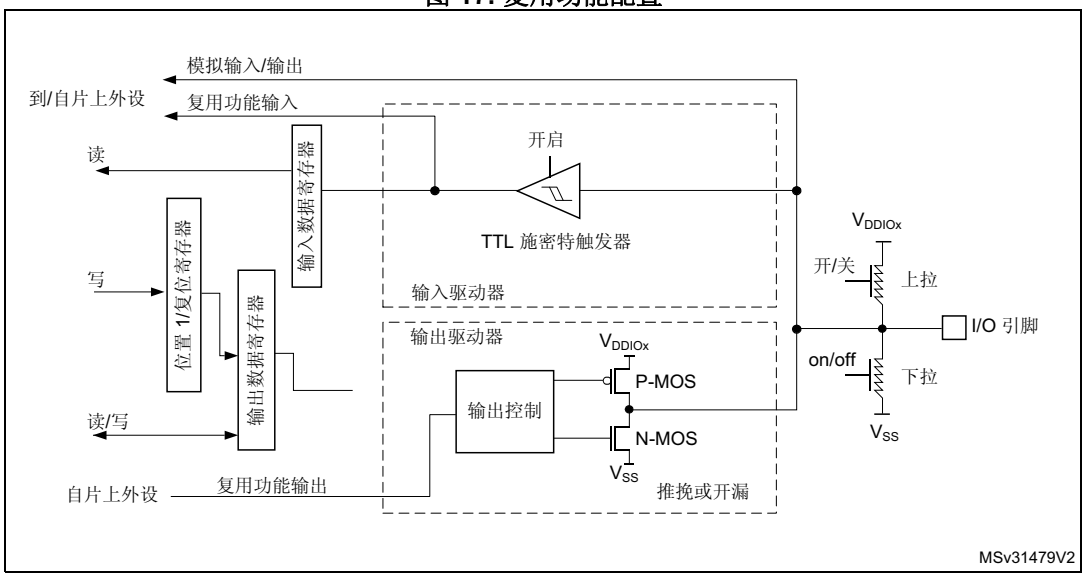
6.3.11 复用功能配置

对 I/O 端口进行编程作为复用功能时：

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

图 17 说明了 I/O 端口位的复用功能配置。

图 17. 复用功能配置



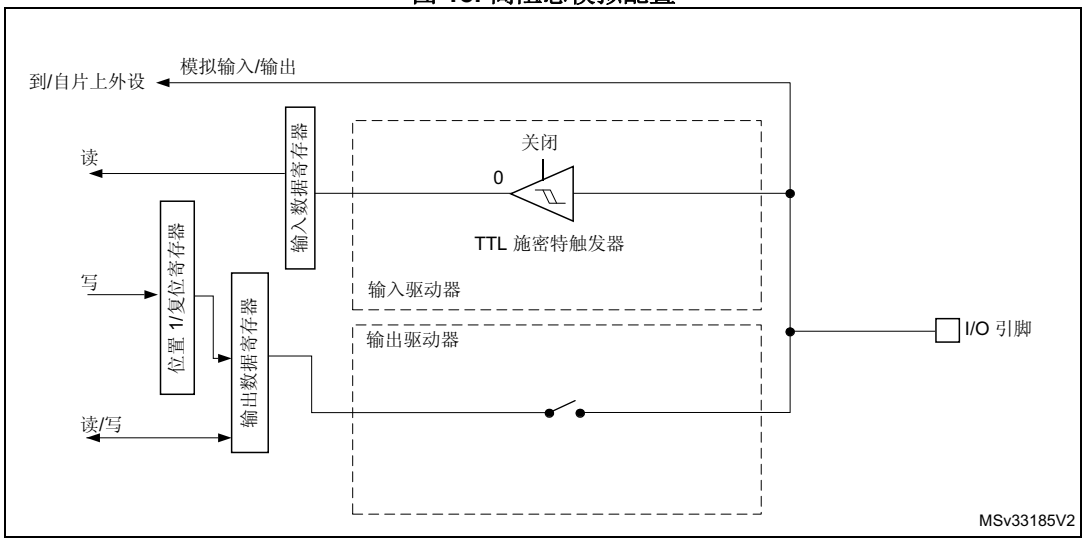
6.3.12 模拟配置

对 I/O 端口进行编程作为模拟配置时：

- 输出缓冲器被禁止
- 施密特触发器输入停用，I/O 引脚的每个模拟输入的功耗变为零。施密特触发器的输出被强制处理为恒定值 (0)。
- 弱上拉和下拉电阻被硬件关闭
- 对输入数据寄存器的读访问值为“0”

图 18 说明了 I/O 端口位的高阻态模拟配置。

图 18. 高阻态模拟配置



6.3.13 将 HSE 或 LSE 振荡器引脚用作 GPIO

当 HSE 或 LSE 振荡器关闭（复位后的默认状态）时，可将相关的振荡器引脚用作 GPIO。

当 HSE 或 LSE 振荡器打开（通过设置 RCC_CSR 寄存器的 HSEON 或 LSEON 位）时，振荡器会控制与其相关联的引脚，这些引脚的 GPIO 配置不起作用。

当 HSE 或 LSE 工作在 BYPASS 模式时，其输入引脚用作外部时钟输入，其输出引脚可用作 GPIO。

对于采用 48 引脚封装的器件，HSE 和 LSE 振荡器都有独立的输入和输出引脚（请参见 FLASH 选项字节的 HSE_NOT_REMAPPED 位）。对于采用少于 48 引脚封装的器件，HSE 和 LSE 振荡器共用一个输入引脚 OSCX_IN 和一个输出引脚 OSCX_OUT，因此每次只能使用其中一个振荡器（另一个振荡器必须禁止）。

6.3.14 小型封装相关调整

SO8 封装的引脚数量有限，因此需要将多个 GPIO 连接到 I/O 引脚。使用 SYSCFG_CFGR3 寄存器可以选择激活的具体引脚，从而防止发生冲突。

6.3.15 GPIO 模式下的复位引脚 (PF2)

PF2 引脚可配置为复位 I/O 或 GPIO。

要将 PF2 配置为 GPIO（输入、输出、AF 或模拟 I/O），应将 FLASH 选项字节的 NOT_RESET_INPUT_ONLY 位置 1、NOT_GPIO_MODE_ONLY 位清零。新设置只有在复位后发生选项字节加载 (OBL) 事件时才会生效。在释放复位之前，PF2 始终用作复位 I/O。

当 PF2 用作 GPIO 时，只能通过器件内部的复位源之一触发复位，无法输出复位信号。

有关复位功能的更多信息，请参见 *RCC* 部分。

6.3.16 GPIO 模式下的 Boot0 引脚 (PA14)

PA14 引脚可配置为 Boot0 输入或 GPIO。

要将 PA14 配置为 GPIO，应将 FLASH 选项字节的 USE_BOOT0_OPT 位置 1。随后，PA14 默认配置为串行线调试 (SWD) 接口时钟输入 (SWCLK)。

有关自举配置的更多信息，请参见*自举配置*部分。

6.4 GPIO 寄存器

本节对 GPIO 寄存器进行了详细介绍。

有关寄存器位、寄存器偏移地址和复位值的汇总，请参见*表 25*。

可按字、半字或字节模式写入外设寄存器。

只有 STM32C03xx 产品提供端口 D。

6.4.1 GPIO 端口模式寄存器 (GPIOx_MODER)（x = A、B、C、D 和 F）

GPIO port mode register

偏移地址：0x00

复位值：0xEBFF FFFF（端口 A）

复位值：0xFFFF FFFF（端口 A 以外的端口）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **MODEy[1:0]**：I/O y 的端口 x 配置 (Port x configuration for I/O y)（y = 15 到 0）

这些位通过软件写入，用于将 I/O 设为四种工作模式之一。

00：输入

01：输出

10：复用功能

11：模拟

6.4.2 GPIO 端口输出类型寄存器 (GPIOx_OTYPER)（x = A、B、C、D 和 F）

GPIO port output type register

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **OTy**：I/O y 的端口 x 配置 (Port x configuration for I/O y)（y = 15 到 0）

这些位通过软件写入，用于配置 I/O 输出类型。

0：输出推挽（复位状态）

1：输出开漏

6.4.3 GPIO 端口输出速度寄存器 (GPIOx_OSPEEDR)（x = A、B、C、D 和 F）

GPIO port output speed register

偏移地址：0x08

复位值：0x0C00 0000（端口 A）

复位值：0x0000 0000（端口 A 以外的端口）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEED15 [1:0]		OSPEED14 [1:0]		OSPEED13 [1:0]		OSPEED12 [1:0]		OSPEED11 [1:0]		OSPEED10 [1:0]		OSPEED9 [1:0]		OSPEED8 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEED7 [1:0]		OSPEED6 [1:0]		OSPEED5 [1:0]		OSPEED4 [1:0]		OSPEED3 [1:0]		OSPEED2 [1:0]		OSPEED1 [1:0]		OSPEED0 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **OSPEEDy[1:0]**：I/O y 的端口 x 配置 (Port x configuration for I/O y)（y = 15 到 0）

这些位通过软件写入，用于配置 I/O 输出速度。

00：超低速
01：低速
10：高速
11：超高速

注意：关于每种速度的频率规范以及电源和负载条件，请参见器件数据手册。
FT_c GPIO 无法设为高速。

6.4.4 GPIO 端口上拉/下拉寄存器 (GPIOx_PUPDR)（x = A、B、C、D 和 F）

GPIO port pull-up/pull-down register

偏移地址：0x0C

复位值：0x2400 0000（端口 A）

复位值：0x0000 0000（端口 A 以外的端口）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **PUPDy[1:0]**：端口 x 配置 I/O y (Port x configuration I/O y)（y = 15 到 0）

这些位通过软件写入，用于配置 I/O 上拉或下拉。

00：无上拉或下拉
01：上拉
10：下拉
11：保留

6.4.5 GPIO 端口输入数据寄存器 (GPIOx_IDR)（x = A、B、C、D 和 F）

GPIO port input data register

偏移地址：0x10

复位值：0x0000 XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **IDy**：端口 x 输入数据 I/O y (Port x input data I/O y)（y = 15 到 0）
这些位为只读。它们包含相应 I/O 端口的输入值。

6.4.6 GPIO 端口输出数据寄存器 (GPIOx_ODR)（x = A、B、C、D 和 F）

GPIO port output data register

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **ODy**：端口输出数据 I/O y (Port output data I/O y)（y = 15 到 0）
这些位可通过软件读取和写入。
注意：对于原子位置 1/ 复位，通过写入 GPIOx_BSRR 寄存器（x = A、B、C、D 和 F），可分别对 OD 位进行置 1 和/或复位。

6.4.7 GPIO 端口位置 1/复位寄存器 (GPIOx_BSRR)（x = A、B、C、D 和 F）

GPIO port bit set/reset register

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
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:16 **BRy**：端口 x 复位 I/O y (Port x reset I/O y)（y = 15 到 0）
这些位为只写。读操作始终返回 0x0000。
0：不会对相应的 ODRx 位执行任何操作
1：对相应的 ODRx 位进行复位
注意：如果同时将 BSx 和 BRx 置 1，则 BSx 的优先级更高。

位 15:0 **BSy**：端口 x 置 1 I/O y (Port x set I/O y)（y = 15 到 0）
这些位为只写。读操作始终返回 0x0000。
0：不会对相应的 ODRx 位执行任何操作
1：对相应的 ODRx 位进行置位

6.4.8 GPIO 端口配置锁定寄存器 (GPIOx_LCKR)（x = A、B、C、D 和 F）

GPIO port configuration lock register

当正确的写序列应用到第 16 位 (LCKK) 时，此寄存器将用于锁定端口位的配置。位 [15:0] 的值用于锁定 GPIO 的配置。在写序列期间，不能更改 LCKR[15:0] 的值。将 LOCK 序列应用到某个端口位后，在执行下一次 MCU 复位或外设复位之前，将无法对该端口位的值进行修改。

注意：可使用特定的写序列对 GPIOx_LCKR 寄存器执行写操作。在此锁定序列期间只允许使用字访问（32 位长）。

每个锁定位冻结一个特定的配置寄存器（控制寄存器和复用功能寄存器）。

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **LCKK**：锁定键

可随时读取该位。可使用锁定键写序列对其进行修改。

0：端口配置锁定键未激活。

1：端口配置锁定键已激活。直到下一次 MCU 复位或外设复位时，才锁定 GPIOx_LCKR 寄存器。

锁定键写序列：

WR LCKR[16] = '1' + LCKR[15:0]

WR LCKR[16] = '0' + LCKR[15:0]

WR LCKR[16] = '1' + LCKR[15:0]

RD LCKR

RD LCKR[16] = '1'（此读操作为可选操作，但它可确认锁定已激活）

注意：在锁定键写序列期间，不能更改 LCK[15:0] 的值。

锁定序列中的任何错误都将中止锁定操作。

在任一端口位上的第一个锁定序列之后，对 LCKK 位的任何读访问都将返回“1”，直到下一次 MCU 复位或外设复位为止。

位 15:0 **LCK[15:0]**：端口 x 锁定 I/O 引脚 y (Port x lock I/O pin y)（y = 15 到 0）

这些位都是读/写位，但只能在 LCKK 位等于“0”时执行写操作。

0：端口配置未锁定

1：端口配置已锁定

6.4.9 GPIO 复用功能低位寄存器 (GPIOx_AFRL)（x = A、B、C、D 和 F）

GPIO alternate function low register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL7[3:0]				AFSEL6[3:0]				AFSEL5[3:0]				AFSEL4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL3[3:0]				AFSEL2[3:0]				AFSEL1[3:0]				AFSEL0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **AFSELy[3:0]**：端口 x 引脚 y 的复用功能选择 (Alternate function selection for port x pin y)（y = 0 到 7）

这些位通过软件写入，用于配置复用功能 I/O。

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

6.4.10 GPIO 复用功能高位寄存器 (GPIOx_AFRH)（x = A、B、C、D 和 F）

GPIO alternate function high register

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL15[3:0]				AFSEL14[3:0]				AFSEL13[3:0]				AFSEL12[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL11[3:0]				AFSEL10[3:0]				AFSEL9[3:0]				AFSEL8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **AFSELy[3:0]**：端口 x I/O y 的复用功能选择 (Alternate function selection for port x, I/O y)（y = 8 到 15）

这些位通过软件写入，用于配置复用功能 I/O。

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 GPIO 端口位复位寄存器 (GPIOx_BRR)（x = A、B、C、D 和 F）

GPIO port bit reset register

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:16 保留，必须保持复位值。

位 15:0 **BRy**：端口 x 复位 I/O y (Port x reset I/O y)（y = 15 到 0）

这些位为只写。读操作始终返回 0x0000。

0：不会对相应的 ODx 位执行任何操作

1：复位相应的 ODx 位

6.4.12 GPIO 寄存器映射

下表提供了 GPIO 寄存器映射和复位值。

表 25. GPIO 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**GPIOx_MODER** （x = A、B、C、D 和 F）	MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]		MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	复位值（端口 A）	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	复位值 （端口 A 以外的端口）	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x04	**GPIOx_OTYPER** （x = A、B、C、D 和 F）	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	**GPIOx_OSPEEDR** （x = A、B、C、D 和 F）	OSPEED15[1:0]		OSPEED14[1:0]		OSPEED13[1:0]		OSPEED12[1:0]		OSPEED11[1:0]		OSPEED10[1:0]		OSPEED9[1:0]		OSPEED8[1:0]		OSPEED7[1:0]		OSPEED6[1:0]		OSPEED5[1:0]		OSPEED4[1:0]		OSPEED3[1:0]		OSPEED2[1:0]		OSPEED1[1:0]		OSPEED0[1:0]	
	复位值（端口 A）	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	复位值 （端口 A 以外的端口）	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**GPIOx_PUPDR** （x = A、B、C、D 和 F）	PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]		PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
	复位值（端口 A）	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	复位值 （端口 A 以外的端口）	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**GPIOx_IDR** （x = A、B、C、D 和 F）	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
	复位值																	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x14	**GPIOx_ODR** （x = A、B、C、D 和 F）	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	**GPIOx_BSRR** （x = A、B、C、D 和 F）	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0	BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**GPIOx_LCKR** （x = A、B、C、D 和 F）	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK	LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
	复位值																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**GPIOx_AFRL** （x = A、B、C、D 和 F）	AFSEL7 [3:0]				AFSEL6 [3:0]				AFSEL5 [3:0]				AFSEL4 [3:0]				AFSEL3 [3:0]				AFSEL2 [3:0]				AFSEL1 [3:0]				AFSEL0 [3:0]			
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	**GPIOx_AFRH** （x = A、B、C、D 和 F）	AFSEL15 [3:0]				AFSEL14 [3:0]				AFSEL13 [3:0]				AFSEL12 [3:0]				AFSEL11 [3:0]				AFSEL10 [3:0]				AFSEL9 [3:0]				AFSEL8 [3:0]			
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**GPIOx_BRR** （x = A、B、C、D 和 F）	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

7 系统配置控制器 (SYSCFG)

器件配有一组配置寄存器。系统配置控制器的主要用途如下：

- 在一些 I/O 端口上使能/禁止 I^2C 超快速模式
- 配置 IR 调制信号及其输出极性
- 重映射一些 I/O 端口
- 重映射位于代码区域开头的存储空间
- 标记来自每条中断线的挂起中断
- 管理稳健性功能

7.1 SYSCFG 寄存器

7.1.1 SYSCFG 配置寄存器 1 (SYSCFG_CFGR1)

该寄存器用于对存储器和 DMA 请求重映射进行特定配置，以及控制特殊 I/O 功能。
使用两个位来配置可在地址 0x0000 0000 访问的存储器类型。软件通过这两位来选择物理映射，从而旁路硬件 BOOT 选择。复位后，这两位的值由实际自举模式配置决定。
激活 FM+ 模式时会忽略 I/O 的速度配置 (GPIOx OSPEEDR)。请确保只有在通过 GPIOx_AFRH 或 GPIOx_AFRL 寄存器将 AFO 选择设为 I2C 后，才能将相应的 x_FMP 位置 1。

偏移地址：0x00

复位值：0x0000 000X（X 是实际自举模式配置所选的存储器模式）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C_PC14_FMP(1)	I2C_PA10_FMP	I2C_PA9_FMP	Res.	I2C1_FMP	I2C_PB9_FMP(1)	I2C_PB8_FMP(1)	I2C_PB7_FMP	I2C_PB6_FMP
							rw	rw	rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IR_MOD[1:0]		IR_POL	PA12_RMP	PA11_RMP	Res.	MEM_MODE[1:0]	
								rw	rw	rw	rw	rw		rw	rw

1. 只有在集成相应外设或功能的器件上才有意义，在其他器件上则保留。

位 31:25 保留，必须保持复位值。

位 24 **I2C_PC14_FMP**：PC14 的超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) enable for PC14)
该位由软件置 1 和清零。它用于在 PC14 I/O 端口上使能 I^2C FM+ 驱动能力。
0：禁止
1：使能
当该位处于禁止状态时，可以通过其中一个 I2Cx_FMP 位在该 I/O 端口上使能 I^2C FM+ 驱动能力。使能 I^2C FM+ 时将忽略速度控制。

位 23 **I2C_PA10_FMP**：PA10 的超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) enable for PA10)
该位由软件置 1 和清零。它用于在 PA10 I/O 端口上使能 I^2C FM+ 驱动能力。
0：禁止
1：使能
当该位处于禁止状态时，可以通过其中一个 I2Cx_FMP 位在该 I/O 端口上使能 I^2C FM+ 驱动能力。使能 I^2C FM+ 时将忽略速度控制。

位 22 **I2C_PA9_FMP**：PA9 的超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) enable for PA9)
该位由软件置 1 和清零。它用于在 PA9 I/O 端口上使能 I^2C FM+ 驱动能力。
0：禁止
1：使能
当该位处于禁止状态时，可以通过其中一个 I2Cx_FMP 位在该 I/O 端口上使能 I^2C FM+ 驱动能力。使能 I^2C FM+ 时将忽略速度控制。

位 21 保留，必须保持复位值。

位 20 **I2C1_FMP**：I2C1 的超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) enable for I2C1)
该位由软件置 1 和清零。它用于在通过 GPIOx_AFR 寄存器配置为 I2C1 的 I/O 端口上使能 I^2C FM+ 驱动能力。
0：禁止
1：使能
当该位处于禁止状态时，可以通过相应的 I2Cx_FMP 位在配置为 I2C1 的 I/O 端口上使能 I^2C FM+ 驱动能力。使能 I^2C FM+ 时将忽略速度控制。

位 19 **I2C_PB9_FMP**：PB9 的超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) enable for PB9)
该位由软件置 1 和清零。它用于在 PB9 I/O 端口上使能 I^2C FM+ 驱动能力。
0：禁止
1：使能
当该位处于禁止状态时，可以通过其中一个 I2Cx_FMP 位在该 I/O 端口上使能 I^2C FM+ 驱动能力。使能 I^2C FM+ 时将忽略速度控制。

位 18 **I2C_PB8_FMP**：PB8 的超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) enable for PB8)
该位由软件置 1 和清零。它用于在 PB8 I/O 端口上使能 I^2C FM+ 驱动能力。
0：禁止
1：使能
当该位处于禁止状态时，可以通过其中一个 I2Cx_FMP 位在该 I/O 端口上使能 I^2C FM+ 驱动能力。使能 I^2C FM+ 时将忽略速度控制。

位 17 **I2C_PB7_FMP**：PB7 的超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) enable for PB7)
该位由软件置 1 和清零。它用于在 PB7 I/O 端口上使能 I^2C FM+ 驱动能力。
0：禁止
1：使能
当该位处于禁止状态时，可以通过其中一个 I2Cx_FMP 位在该 I/O 端口上使能 I^2C FM+ 驱动能力。使能 I^2C FM+ 时将忽略速度控制。

位 16 **I2C_PB6_FMP**：PB6 的超快速模式 (FM+) 使能 (Fast Mode Plus (FM+) enable for PB6)
该位由软件置 1 和清零。它用于在 PB6 I/O 端口上使能 I^2C FM+ 驱动能力。
0：禁止
1：使能
当该位处于禁止状态时，可以通过其中一个 I2Cx_FMP 位在该 I/O 端口上使能 I^2C FM+ 驱动能力。使能 I^2C FM+ 时将忽略速度控制。

位 15:8 保留，必须保持复位值。

位 7:6 **IR_MOD[1:0]**：IR 调制包络信号选择 (IR Modulation Envelope signal selection)
该位域用于选择 IR 调制包络的信号：
00：TIM16
01：USART1
10：USART2
11：保留

位 5 **IR_POL**：IR 输出极性选择 (IR output polarity selection)
0：IRTIM 的输出 (IR_OUT) 非反相
1：IRTIM 的输出 (IR_OUT) 反相

位 4 **PA12_RMP**：PA12 引脚重映射 (PA12 pin remapping)
该位由软件置 1 和清零。置 1 时，它会将 PA12 引脚重新映射为 PA10 GPIO 端口，而非 PA12 GPIO 端口。
0：无重映射 (PA12)
1：重映射 (PA10)
注意：如果 SYSCFG_CFGR3 寄存器的 PINMUX4[1:0] 位域为 00，PA12_RMP 必须保持为 0，以防止由于两个不同输出电平的 GPIO 输出连接到同一引脚而产生冲突。

位 3 **PA11_RMP**：PA11 引脚重映射 (PA11 pin remapping)
该位由软件置 1 和清零。置 1 时，它会将 PA11 引脚重新映射为 PA9 GPIO 端口，而非 PA11 GPIO 端口。
0：无重映射 (PA11)
1：重映射 (PA9)
注意：如果 SYSCFG_CFGR3 寄存器的 PINMUX2[1:0] 位域为 00，PA11_RMP 必须保持为 0，以防止由于两个不同输出电平的 GPIO 输出连接到同一引脚而产生冲突。

位 2 保留，必须保持复位值。

位 1:0 **MEM_MODE[1:0]**：存储器映射选择位 (Memory mapping selection bits)
该位域由软件控制，用于选择哪块存储区域在内部被映射到地址 0x0000 0000。其复位值由自举模式配置确定。更多详细信息，请参见*第 2.5 节：自举配置*。
x0：主 Flash
01：系统 Flash
11：嵌入式 SRAM

7.1.2 SYSCFG 配置寄存器 2 (SYSCFG_CFGR2)

SYSCFG configuration register 2

偏移地址：0x18

系统复位值：0x0000 0000

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

位 31:1 保留，必须保持复位值。

位 0 **LOCKUP_LOCK**：Cortex®-M0+ LOCKUP 使能 (Cortex®-M0+ LOCKUP enable)
该位由软件置 1，通过系统复位清零。置 1 时，它可以使能 Cortex®-M0+ LOCKUP (HardFault) 输出与 TIM1/16/17 刹车输入的连接。
0：禁止
1：使能

7.1.3 SYSCFG 配置寄存器 3 (SYSCFG_CFGR3)

SYSCFG configuration register 3

当有多个 GPIO 在内部连接到同一个引脚时，该寄存器允许指定其中一个 GPIO 保持由其对应的 GPIOx_MODER 寄存器指定的设置。其他 GPIO 则被强制进入数字输入模式，而无论其对应的 GPIOx_MODER 寄存器设置如何。
只有在 *FLASH 选项寄存器 (FLASH_OPTR)* 的 SECURE_MUXING_EN 位置 1（默认）时，该寄存器才有效。
当 SECURE_MUXING_EN 清零时，SYSCFG_CFGR3 将不起作用。连接到同一引脚的所有 GPIO 都保持其各自对应的 GPIOx_MODER 寄存器中设定的配置。用户软件必须确保各 GPIO 之间不存在冲突。

偏移地址：0x3C

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PINMUX5[1:0]		PINMUX4[1:0]		PINMUX3[1:0]		PINMUX2[1:0]		PINMUX1[1:0]		PINMUX0[1:0]	
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:10 **PINMUX5[1:0]**：引脚 GPIO 复用器 5 (Pin GPIO multiplexer 5)
该位由软件置 1，通过系统复位清零。它用于将 GPIO 分配到引脚。
条件：STM32C011x——GPIO 分配到 WLCSP12 引脚 F1
00：PA3
01：PA4
10：PA5
11：PA6

位 9:8 **PINMUX4[1:0]**：引脚 GPIO 复用器 4 (Pin GPIO multiplexer 4)
该位由软件置 1，通过系统复位清零。它用于将 GPIO 分配到引脚。
条件：STM32C011x——GPIO 分配到 WLCSP12 引脚 E2
00：PA7
01：PA12
1x：保留
注意：SYSCFG_CFGR1 寄存器的 PA12_RMP 位的优先级高于通过该位域所做的选择。更多详细信息，请参见 SYSCFG_CFGR1 寄存器的说明。

位 7:6 **PINMUX3[1:0]**：引脚 GPIO 复用器 3 (Pin GPIO multiplexer 3)
该位由软件置 1，通过系统复位清零。它用于将 GPIO 分配到引脚。
条件：STM32C011x——GPIO 分配到 SO8 引脚 8
00：PA14
01：PB6
10：PC15
11：保留

位 5:4 **PINMUX2[1:0]**：引脚 GPIO 复用器 2 (Pin GPIO multiplexer 2)
该位由软件置 1，通过系统复位清零。它用于将 GPIO 分配到引脚。
条件：STM32C011x——GPIO 分配到 SO8 引脚 5
00：PA8
01：PA11
1x：保留
注意：SYSCFG_CFGR1 寄存器的 PA11_RMP 位的优先级高于通过该位域所做的选择。更多详细信息，请参见 SYSCFG_CFGR1 寄存器的说明。

位 3:2 **PINMUX1[1:0]**：引脚 GPIO 复用器 1 (Pin GPIO multiplexer 1)
该位由软件置 1，通过系统复位清零。它用于将 GPIO 分配到引脚。
条件：STM32C011x——GPIO 分配到 SO8 引脚 4
00：PF2
01：PA0
10：PA1
11：PA2

位 1:0 **PINMUX0[1:0]**：引脚 GPIO 复用器 0 (Pin GPIO multiplexer 0)
该位由软件置 1，通过系统复位清零。它用于将 GPIO 分配到引脚。
条件：STM32C011x——GPIO 分配到 SO8 引脚 1
00：PB7
01：PC14
1x：保留

7.1.4 SYSCFG 中断线 0 状态寄存器 (SYSCFG_ITLINE0)

SYSCFG interrupt line 0 status register

器件上实现了一组专用的寄存器，用于将与每条中断线相关联的所有挂起中断源收集到一个寄存器中。这样，当有多个中断源与中断线相关联时，用户只需通过单次读操作即可确定哪个外设需要服务。
这些寄存器中的所有位都是只读的，在有相应的中断请求挂起时由硬件置 1，通过复位外设寄存器中的中断源标志来清零。

偏移地址：0x80

系统复位值：0x0000 0000

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

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

位 31:1 保留，必须保持复位值。

位 0 **WWDG**：窗口看门狗中断挂起标志 (Window watchdog interrupt pending flag)

7.1.5 SYSCFG 中断线 2 状态寄存器 (SYSCFG_ITLINE2)

SYSCFG interrupt line 2 status register

偏移地址：0x88

系统复位值：0x0000 0000

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

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

位 31:2 保留，必须保持复位值。

位 1 **RTC**：RTC 中断请求挂起 (RTC interrupt request pending)（EXTI 线 19）

位 0 保留，必须保持复位值。

7.1.6 SYSCFG 中断线 3 状态寄存器 (SYSCFG_ITLINE3)

SYSCFG interrupt line 3 status register

偏移地址：0x8C

系统复位值：0x0000 0000

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

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

位 31:2 保留，必须保持复位值。

位 1 **FLASH_ITF**：Flash 接口中断请求挂起 (Flash interface interrupt request pending)

位 0 保留，必须保持复位值。

7.1.7 SYSCFG 中断线 4 状态寄存器 (SYSCFG_ITLINE4)

SYSCFG interrupt line 4 status register

偏移地址：0x90

系统复位值：0x0000 0000

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

位 31:1 保留，必须保持复位值。

位 0 **RCC**：复位和时钟控制中断请求挂起 (Reset and clock control interrupt request pending)

7.1.8 SYSCFG 中断线 5 状态寄存器 (SYSCFG_ITLINE5)

SYSCFG interrupt line 5 status register

偏移地址：0x94

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI1	EXTI0
														r	r

位 31:2 保留，必须保持复位值。

位 1 **EXTI1**：EXTI 线 1 中断请求挂起 (EXTI line 1 interrupt request pending)

位 0 **EXTI0**：EXTI 线 0 中断请求挂起 (EXTI line 0 interrupt request pending)

7.1.9 SYSCFG 中断线 6 状态寄存器 (SYSCFG_ITLINE6)

SYSCFG interrupt line 6 status register

偏移地址：0x98

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI3	EXTI2
														r	r

位 31:2 保留，必须保持复位值。

位 1 **EXTI3**：EXTI 线 3 中断请求挂起 (EXTI line 3 interrupt request pending)

位 0 **EXTI2**：EXTI 线 2 中断请求挂起 (EXTI line 2 interrupt request pending)

7.1.10 SYSCFG 中断线 7 状态寄存器 (SYSCFG_ITLINE7)

SYSCFG interrupt line 7 status register

偏移地址：0x9C

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	EXTI15	EXTI14	EXTI13	EXTI12	EXTI11	EXTI10	EXTI9	EXTI8	EXTI7	EXTI6	EXTI5	EXTI4
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留，必须保持复位值。

位 11 **EXTI15**：EXTI 线 15 中断请求挂起 (EXTI line 15 interrupt request pending)

位 10 **EXTI14**：EXTI 线 14 中断请求挂起 (EXTI line 14 interrupt request pending)

位 9 **EXTI13**：EXTI 线 13 中断请求挂起 (EXTI line 13 interrupt request pending)

位 8 **EXTI12**：EXTI 线 12 中断请求挂起 (EXTI line 12 interrupt request pending)

位 7 **EXTI11**：EXTI 线 11 中断请求挂起 (EXTI line 11 interrupt request pending)

位 6 **EXTI10**：EXTI 线 10 中断请求挂起 (EXTI line 10 interrupt request pending)

位 5 **EXTI9**：EXTI 线 9 中断请求挂起 (EXTI line 9 interrupt request pending)

位 4 **EXTI8**：EXTI 线 8 中断请求挂起 (EXTI line 8 interrupt request pending)

位 3 **EXTI7**：EXTI 线 7 中断请求挂起 (EXTI line 7 interrupt request pending)

位 2 **EXTI6**：EXTI 线 6 中断请求挂起 (EXTI line 6 interrupt request pending)

位 1 **EXTI5**：EXTI 线 5 中断请求挂起 (EXTI line 5 interrupt request pending)

位 0 **EXTI4**：EXTI 线 4 中断请求挂起 (EXTI line 4 interrupt request pending)

7.1.11 SYSCFG 中断线 9 状态寄存器 (SYSCFG_ITLINE9)

SYSCFG interrupt line 9 status register

偏移地址：0xA4

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA1_CH1
															r

位 31:1 保留，必须保持复位值。

位 0 **DMA1_CH1**：DMA1 通道 1 中断请求挂起 (DMA1 channel 1interrupt request pending)

7.1.12 SYSCFG 中断线 10 状态寄存器 (SYSCFG_ITLINE10)

SYSCFG interrupt line 10 status register

偏移地址：0xA8

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA1_CH3	DMA1_CH2
														r	r

位 31:2 保留，必须保持复位值。

位 1 **DMA1_CH3**：DMA1 通道 3 中断请求挂起 (DMA1 channel 3 request pending)

位 0 **DMA1_CH2**：DMA1 通道 2 中断请求挂起 (DMA1 channel 2 request pending)

7.1.13 SYSCFG 中断线 11 状态寄存器 (SYSCFG_ITLINE11)

SYSCFG interrupt line 11 status register

偏移地址：0xAC

系统复位值：0x0000 0000

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

位 31:1 保留，必须保持复位值。

位 0 **DMAMUX**：DMAMUX 中断请求挂起 (DMAMUX interrupt request pending)

7.1.14 SYSCFG 中断线 12 状态寄存器 (SYSCFG_ITLINE12)

SYSCFG interrupt line 12 status register

偏移地址：0xB0

系统复位值：0x0000 0000

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

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

位 31:1 保留，必须保持复位值。

位 0 **ADC**：ADC 中断请求挂起 (ADC interrupt request pending)

7.1.15 SYSCFG 中断线 13 状态寄存器 (SYSCFG_ITLINE13)

SYSCFG interrupt line 13 status register

偏移地址：0xB4

系统复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM1_BRK	TIM1_UPD	TIM1_TRG	TIM1_CCU
												r	r	r	r

位 31:4 保留，必须保持复位值。

位 3 **TIM1_BRK**: 定时器 1 刹车中断请求挂起 (Timer 1 break interrupt request pending)

位 2 **TIM1_UPD**：定时器 1 更新中断请求挂起 (Timer 1 update interrupt request pending)

位 1 **TIM1_TRG**：定时器 1 触发中断请求挂起 (Timer 1 trigger interrupt request pending)

位 0 **TIM1_CCU**：定时器 1 换向中断请求挂起 (Timer 1 commutation interrupt request pending)

7.1.16 SYSCFG 中断线 14 状态寄存器 (SYSCFG_ITLINE14)

SYSCFG interrupt line 14 status register

偏移地址：0xB8

系统复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM1_CC
															r

位 31:1 保留，必须保持复位值。

位 0 **TIM1_CC**：定时器 1 捕获/比较中断请求挂起 (Timer 1 capture compare interrupt request pending)

7.1.17 SYSCFG 中断线 16 状态寄存器 (SYSCFG_ITLINE16)

SYSCFG interrupt line 16 status register

偏移地址：0xC0

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM3
															r

位 31:1 保留，必须保持复位值。

位 0 **TIM3**：定时器 3 中断请求挂起 (Timer 3 interrupt request pending)

7.1.18 SYSCFG 中断线 19 状态寄存器 (SYSCFG_ITLINE19)

SYSCFG interrupt line 19 status register

偏移地址：0xCC

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM14
															r

位 31:1 保留，必须保持复位值。

位 0 **TIM14**：定时器 14 中断请求挂起 (Timer 14 interrupt request pending)

7.1.19 SYSCFG 中断线 21 状态寄存器 (SYSCFG_ITLINE21)

SYSCFG interrupt line 21 status register

偏移地址：0xD4

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM16
															r

位 31:1 保留，必须保持复位值。

位 0 **TIM16**：定时器 16 中断请求挂起 (Timer 16 interrupt request pending)

7.1.20 SYSCFG 中断线 22 状态寄存器 (SYSCFG_ITLINE22)

SYSCFG interrupt line 22 status register

偏移地址：0xD8

系统复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM17
															r

位 31:1 保留，必须保持复位值。

位 0 **TIM17**：定时器 17 中断请求挂起 (Timer 17 interrupt request pending)

7.1.21 SYSCFG 中断线 23 状态寄存器 (SYSCFG_ITLINE23)

SYSCFG interrupt line 23 status register

偏移地址：0xDC

系统复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C1
															r

位 31:1 保留，必须保持复位值。

位 0 **I2C1**：I2C1 中断请求挂起 (I2C1 interrupt request pending)（与 EXTI 线 23 结合使用）

7.1.22 SYSCFG 中断线 25 状态寄存器 (SYSCFG_ITLINE25)

SYSCFG interrupt line 25 status register

偏移地址：0xE4

系统复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI1
															r

位 31:1 保留，必须保持复位值。

位 0 **SPI1**：SPI1 中断请求挂起 (SPI1 interrupt request pending)

7.1.23 SYSCFG 中断线 27 状态寄存器 (SYSCFG_ITLINE27)

SYSCFG interrupt line 27 status register

偏移地址：0xEC

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USART1
															r

位 31:1 保留，必须保持复位值。

位 0 **USART1**：USART1 中断请求挂起 (USART1 interrupt request pending)（与 EXTI 线 25 结合使用）

7.1.24 SYSCFG 中断线 28 状态寄存器 (SYSCFG_ITLINE28)

SYSCFG interrupt line 28 status register

偏移地址：0xF0

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USART2
															r

位 31:1 保留，必须保持复位值。

位 0 **USART2**：USART2 中断请求挂起 (USART2 interrupt request pending)（EXTI 线 26）

7.1.25 SYSCFG 寄存器映射

下表列出了 SYSCFG 寄存器映射和复位值。

表 26. SYSCFG 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**SYSCFG_CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C_PC14_FMP	I2C_PA10_FMP	I2C_PA9_FMP	Res.	I2C1_FMP	I2C_PB9_FMP	I2C_PB8_FMP	I2C_PB7_FMP	I2C_PB6_FMP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOOSTEN	IR_MOD		IR_POL	PA12_RMP	PA11_RMP	Res.	MEM_MODE[1:0]	
	复位值								0	0	0		0	0	0	0	0								0	0	0	0	0	0		X	X
0x04 to 0x17	**保留**	保留																															
0x18	**SYSCFG_CFGR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCUP_LOCK
	复位值																																0
0x1C to 0x3B	**保留**	保留																															
0x3C	**SYSCFG_CFGR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PINMUX5		PINMUX4		PINMUX3		PINMUX2		PINMUX1		PINMUX0	
	复位值																					0	0	0	0	0	0	0	0	0	0	0	0
0x40 to 0x7F	**保留**	保留																															
0x80	**SYSCFG_ITLINE0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WWDG
	复位值																																
0x84	**保留**	保留																															
0x88	**SYSCFG_ITLINE2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTC	
	复位值																															0	
0x8C	**SYSCFG_ITLINE3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLASH_ITF	
	复位值																															0	
0x90	**SYSCFG_ITLINE4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCC
	复位值																																0
0x94	**SYSCFG_ITLINE5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI1	EXTI0
	复位值																															0	0
0x98	**SYSCFG_ITLINE6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI3	EXTI2
	复位值																															0	0
0x9C	**SYSCFG_ITLINE7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI15	EXTI14	EXTI13	EXTI12	EXTI11	EXTI10	EXTI9	EXTI8	EXTI7	EXTI6	EXTI5	EXTI4
	复位值																					0	0	0	0	0	0	0	0	0	0	0	0
0xA0	**保留**	保留																															
0xA4	**SYSCFG_ITLINE9**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA1_CH1
	复位值																																0

表 26. SYSCFG 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xA8	**SYSCFG_ITLINE10**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA1_CH3	DMA1_CH2
	复位值																															0	0
0xAC	**SYSCFG_ITLINE11**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAMUX
	复位值																																0
0xB0	**SYSCFG_ITLINE12**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC
	复位值																																0
0xB4	**SYSCFG_ITLINE13**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM1_BRK	TIM1_UPD	TIM1_TRG	TIM1_CCU
	复位值																													0	0	0	0
0xB8	**SYSCFG_ITLINE14**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM1_CC
	复位值																																0
0xBC	**保留**	保留																															
0xC0	**SYSCFG_ITLINE16**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM3
	复位值																																0
0xC4-0xC8	**保留**	保留																															
0xCC	**SYSCFG_ITLINE19**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM14
	复位值																																0
0xD0	**保留**	保留																															
0xD4	**SYSCFG_ITLINE21**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM16
	复位值																																0
0xD8	**SYSCFG_ITLINE22**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM17
	复位值																																0
0xDC	**SYSCFG_ITLINE23**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C1
	复位值																																0
0xE0	**保留**	保留																															
0xE4	**SYSCFG_ITLINE25**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI1
	复位值																																0
0xE8	**保留**	保留																															
	复位值																																
0xEC	**SYSCFG_ITLINE27**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USART1
	复位值																																0
0xF0	**SYSCFG_ITLINE28**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USART2
	复位值																																0

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

8 互连矩阵

8.1 简介

多个外设间有直接连接。

这可以在外设间实现自动通信和/或同步，节省了 CPU 资源，进而降低功耗。

此外，这些硬件连接消除了软件延迟，允许设计可预测系统。

这些互连可以在运行、睡眠和停止模式下工作，具体取决于外设。

关于不同 STM32C0x1 产品的外设可用性，请参见*第 1.4 节：外设可用性*。

8.2 连接汇总

表 27. 互连矩阵(1)(2)

源	目标							
	TIM1	TIM3	TIM14	TIM16	TIM17	ADC1	DMAMUX	IRTIM
TIM1	-	8.3.1	-	-	-	8.3.2	-	-
TIM3	8.3.1	-	-	-	-	8.3.2	-	-
TIM14	-	8.3.1	-	-	-	-	8.3.8	-
TIM16	-	-	-	-	-	-	-	8.3.7
TIM17	8.3.1	-	-	-	-	-	-	8.3.7
USART1	-	-	-	-	-	-	-	8.3.7
USART2	-	-	-	-	-	-	-	8.3.7
ADC	8.3.3	-	-	-	-	-	-	-
温度传感器	-	-	-	-	-	8.3.5	-	-
VREFINT	-	-		-	-	8.3.5	-	-
HSE	-	-	8.3.4	-	8.3.4	-	-	-
LSE	-	-	-	8.3.4	-	-	-	-
LSI	-	-	-	8.3.4	-	-	-	-
MCO	-	-	8.3.4	-	8.3.4	-	-	-
MCO2	-	-	8.3.4	8.3.4	8.3.4	-	-	-
EXTI	-	-	-	-	-	8.3.2	8.3.2	-
RTC	-	-	8.3.4	-	-	-	-	-
SYST ERR	8.3.6	-	-	8.3.6	8.3.6	-	-	-

1. 表中的编号是*第 8.3 节：互连详细信息*中相应小节的链接。
2. 灰色单元格中的“-”符号表示“无互连”。

8.3 互连详细信息

8.3.1 从 TIM1、TIM3、TIM14 和 TIM17 到 TIM1 和 TIM3

目的

某些 TIMx 定时器从内部连接在一起，以实现定时器同步或链接。

当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。

关于功能描述，请参见：*第 16.3.19 节：定时器同步*。

以下章节对同步模式进行了详细说明：

- *第 15.3.26 节：定时器同步*（面向高级控制定时器 TIM1）
- *第 16.3.18 节：定时器与外部触发同步*（面向通用定时器 TIM3）

触发信号

在可配置定时器事件发生后，输出（来自主器件）出现在 TIMx_TRGO 信号（和 TIMx_TRGOx）上。

对于没有触发输出的 TIM14 和 TIM17 定时器，改用输出比较 1 代替。

输入（到从器件）信号在 TIMx_ITR0/ITR1/ITR2/ITR3 上。

TIM1 的输入和输出信号如*图 55：高级控制定时器框图*所示。

表 65：TIM1 内部触发连接 给出了可能的主/从连接。

相关功耗模式

这些互连在运行模式和睡眠模式下工作。

8.3.2 从 TIM1、TIM3 和 EXTI 到 ADC

目的

通用定时器 TIM3、高级控制定时器 TIM1 和 EXTI 可用于生成 ADC 触发事件。

关于 TIMx 同步的说明，请参见：*第 15.3.27 节：ADC 同步*。

关于 ADC 同步的说明，请参见：*第 14.5 节：外部触发转换和触发极性（EXTSEL 和 EXTEN）*。

触发信号

输出（来自定时器）位于 TIMx_TRGO、TIMx_TRGO2 信号或 TIMx_CCx 事件上。

输入（到 ADC）位于 TRG[7:0] 信号上。

表 53：外部触发器 给出了定时器和 ADC 之间的连接。

相关功耗模式

这些互连在运行模式和睡眠模式下工作。

8.3.3 从 ADC 到 TIM1

目的

ADC 可通过看门狗信号向高级控制定时器 TIM1 提供触发事件。

关于 ADC 模拟看门狗设置的说明，请参见：*第 14.8 节：模拟窗口看门狗*。

关于定时器的触发设置，请参见：*第 15.3.4 节：外部触发输入*。

触发信号

输出（来自 ADC）位于信号 ADC_AWDx_OUT 上，其中 x = 1、2、3（每个 ADC 具有三个看门狗）；输入（到定时器）位于信号 TIMx_ETR（外部触发信号）上。

相关功耗模式

该互连在运行模式和睡眠模式下工作。

8.3.4 从 HSE、LSE、LSI、MCO、MCO2 和 RTC 到 TIM14、TIM16 和 TIM17

目的

可以选择将外部时钟（HSE 和 LSE）、内部时钟 (LSI)、微控制器输出时钟（MCO 和 MCO2）、RTC 时钟以及 GPIO 作为一些 TIM14/16/TIM17 定时器的捕获通道 1 的输入。

定时器允许校准或精确测量内部时钟（如 HSI48 或 LSI）以及将精确的时钟（如 LSE 或 HSE/32）用于提供参考时序。有关详细信息，请参见*第 5.2.13 节：基于 TIM14/TIM16/TIM17 的内部/外部时钟测量*。

使用低速外部 (LSE) 振荡器时，无需额外的硬件连接。

相关功耗模式

这些互连在运行模式和睡眠模式下工作。

8.3.5 从内部模拟源到 ADC

目的

内部温度传感器输出电压 V_{TS} 和内部参考电压 V_{REFINT} 通道连接到 ADC 输入通道。

更多信息，请参见：

- *第 14.2 节：ADC 主要特性*
- *第 14.4.8 节：通道选择（CHSEL、SCANDIR、CHSELRMOD）*
- *第 14.10 节：温度传感器和内部参考电压*

相关功耗模式

这些互连在运行模式和睡眠模式下工作。

8.3.6 从系统错误到 TIM1、TIM16 和 TIM17

目的

CSS、CPU HardFault 和 RAM 奇偶校验错误能够以面向 TIM1、TIM16 和 TIM17 的定时器刹车形式生成系统错误。

刹车功能的目的是保护由这些定时器生成的 PWM 信号所驱动的功率器件。

相关信息请参见：

- *第 15.3.16 节：使用刹车功能* (TIM1)
- *第 18.3.11 节：使用刹车功能* (TIM16/TIM17)
- *图 180：TIM16/TIM17 框图*

相关功耗模式

这些互连在运行模式和睡眠模式下工作。

8.3.7 从 TIM16、TIM17、USART1 和 USART2 到 IRTIM

目的

与 USART1 或 USART2 传输信号相关的 TIM17 定时器的 TIMx_OC1 输出通道可生成红外输出波形。

第 19 节：红外接口 (IRTIM) 对此功能进行了介绍。

相关功耗模式

这些互连在运行模式和睡眠模式下工作。

8.3.8 从 TIM14 和 EXTI 到 DMAMUX

目的

TIM14 通用定时器和 EXTI 可用作 DMAMUX 的触发事件。

相关功耗模式

这些互连在运行模式和睡眠模式下工作。

9 直接存储器访问控制器 (DMA)

9.1 简介

直接存储器访问 (DMA) 控制器是一个主控总线系统外设。

当通过控制为 CPU 减轻负担时，DMA 用于在存储器映射的外设和/或存储器之间进行可编程的数据传输。

DMA 控制器采用单 AHB 主控总线架构。

有一个3通道的DMA控制器。

每个通道都专门管理来自一个或更多外设的存储器访问请求。该 DMA 有一个仲裁器，用于处理 DMA 请求间的优先级。

9.2 DMA 主要特性

- 单 AHB 主控总线
- 外设到存储器、存储器到外设、存储器到存储器以及外设到外设的数据传输
- 可将映射到存储器空间上的片上设备作为源或目标进行访问，例如 Flash、SRAM、AHB 和 APB 外设
- 所有 DMA 通道均可单独配置：
  - 每个通道均与来自外设的 DMA 请求信号或存储器到存储器传输中的软件触发信号相关联。此配置由软件执行。
  - 各请求之间的优先级可用软件编程（每通道 4 个级别：非常高、高、中、低），在软件优先级相同的情况下可以通过硬件决定优先级（例如，通道 1 请求的优先级高于通道 2 请求的优先级）。
  - 源和目标的传输大小（字节、半字、字）彼此独立，模拟打包和拆包。源和目标的地址必须根据传输数据的大小进行对齐。
  - 支持外设与存储器之间的双向传输，并支持循环缓冲区管理
  - 可编程的待传输数据数目：0 到 2^{16} - 1
- 每个通道生成一个中断请求。每个中断请求均因以下三个 DMA 事件中的任意一个而引起：传输完成、半传输或传输错误。

9.3 DMA 实现

9.3.1 DMA1

DMA1 使用下表所示的硬件配置参数实现。

表 28.DMA 实现

特性	DMA
通道数	3

9.3.2 DMA 请求映射

DMA 控制器通过 DMAMUX 外设连接至来自 AHB/APB 外设的 DMA 请求。

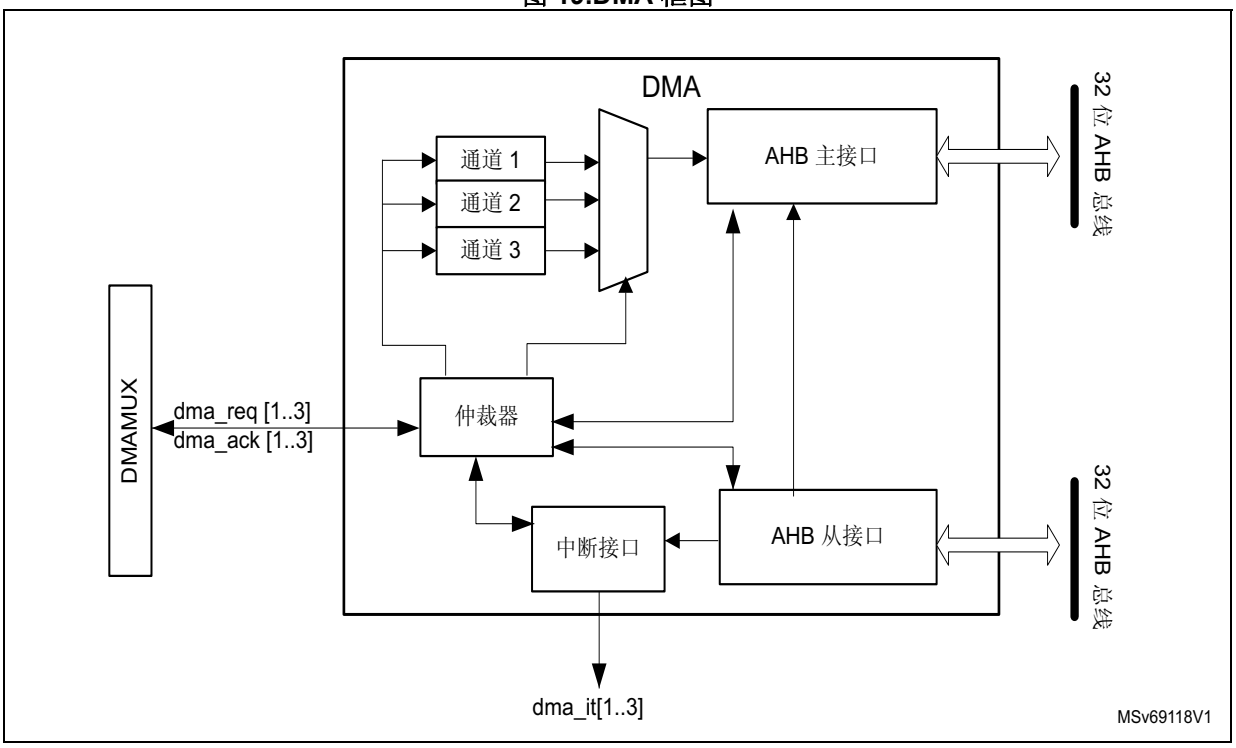
有关不同请求的映射，请参见 DMAMUX 部分。

9.4 DMA 功能描述

9.4.1 DMA 框图

DMA 框图如下图所示。

图 19.DMA 框图



DMA 控制器通过与其他系统主设备共用 AHB 系统总线来执行直接存储器传输。总线矩阵执行循环调度。当 CPU 与 DMA 访问目标（存储器或外设）相同时，DMA 请求会将 CPU 对系统总线的访问停止数个总线周期。

根据通过 AHB 从接口实现的相应配置，DMA 控制器会在 DMA 通道及其接收的相关请求之间进行仲裁。DMA 控制器还会通过单一 AHB 端口主器件调度 DMA 数据传输。

DMA 控制器按通道向中断控制器生成中断。

9.4.2 DMA 引脚和内部信号

表 29.DMA 内部输入/输出信号

信号名称	信号类型	说明
dma_req[x]	输入	DMA 通道 x 请求

表 29.DMA 内部输入/输出信号（续）

信号名称	信号类型	说明
dma_ack[x]	输出	DMA 通道 x 应答
dma_it[x]	输出	DMA 通道 x 中断

9.4.3 DMA 传输

软件在通道级配置 DMA 控制器，以便执行块传输，此类传输由一系列 AHB 总线传输组成。

DMA 块传输可从外设请求，或在进行存储器到存储器传输时由软件触发。

触发该事件后，会按照以下步骤进行单次 DMA 传输：

1. 外设向 DMA 控制器发送单个 DMA 请求信号。
2. DMA 控制器按照与此外设请求相关的通道的优先级来处理该请求。
3. 只要 DMA 控制器授权给外设，DMA 控制器就会向外设发送确认信号。
4. 外设获得 DMA 控制器的确认信号后，便会立即释放其请求。
5. 一旦外设使请求失效，DMA 控制器就会释放确认信号。

外设可继续发送请求，再次启动 DMA 传输。

不论外设是传输源还是传输目标，都使用请求/应答协议。例如，对于存储器到外设传输，外设通过对 DMA 控制器驱动其单个请求信号来启动传输。DMA 控制器随后读取存储器中的单个数据，并将该数据写入外设。

对于给定通道 x，DMA 块传输由以下内容的重复序列组成：

- 单次 DMA 传输，其中封装单个数据的两次 AHB 传输（通过 DMA AHB 总线主控实现）：
  - 通过内部当前外设/存储器地址寄存器进行寻址，从外设数据寄存器或存储器单元中读取单个数据（字节、半字或字）。
    用于首次传输的起始地址是在 DMA_CPARx 或 DMA_CMARx 寄存器中编程的外设或存储器基址。
  - 通过内部当前外设/存储器地址寄存器进行寻址，向外设数据寄存器或存储器单元中写入单个数据（字节、半字或字）。
    用于首次传输的起始地址是在 DMA_CPARx 或 DMA_CMARx 寄存器中编程的外设或存储器基址。
- 编程的 DMA_CNDTRx 寄存器在传输后递减
  该寄存器包含待传输数据项的剩余数目（AHB“先读后写”传输次数）。

重复执行该序列，直至 DMA_CNDTRx 为空。

注意： *AHB 主控总线源地址/目标地址必须根据传输至源/目标的单个数据的设定大小进行对齐。*

9.4.4 DMA 仲裁

DMA 仲裁器管理不同通道间的优先级。

当仲裁器为某个有效通道 x 授予优先权后（硬件请求或软件触发），会发起单次 DMA 传输（例如，单个数据的一次 AHB“先读后写”传输）。随后仲裁器会再次对有效通道组进行仲裁，并选择优先级最高的通道。

优先级管理分为两个阶段：

- 软件：每个通道的优先级均在 DMA_CCRx 寄存器中配置为以下四个不同级别之一：
  - 非常高
  - 高
  - 中
  - 低
- 硬件：如果两个请求的软件优先级相同，则索引编号最低的通道优先。例如，通道 1 的优先级高于通道 2。

当通道 x 被编程为在存储器到存储器模式下进行块传输时，在此通道 x 的各个单次 DMA 传输之间将进行重新仲裁。每当存在其他同时带有有效请求的通道时，DMA 仲裁器便会自动授权切换到其他有有效请求的通道中具有最高优先级的通道，此通道的优先级可能低于存储器到存储器通道。

9.4.5 DMA 通道

各个通道均能处理外设寄存器（位于固定地址中）与存储器单元之间的 DMA 传输。待传输数据项的数目可编程。每个传输完成后，包含待传输的数据项数目的寄存器都会递减。

DMA 通道以块传输的方式进行编程设置。

数据大小可编程

要传输到外设和存储器的单个数据的大小（字节、半字或字）可分别通过 DMA_CCRx 寄存器的 PSIZE[1:0] 和 MSIZE[1:0] 位域进行编程。

指针递增

根据 DMA_CCRx 寄存器的 PINC 位和 MINC 位的状态，外设和存储器指针在每次传输后可自动递增。

如果使能了**递增模式**（PINC 或 MINC 置 1），则下次传输的地址是前一次传输的地址加上 1、2 或 4，具体取决于 PSIZE[1:0] 或 MSIZE[1:0] 中定义的数据大小。首次传输的地址可在 DMA_CPARx 或 DMA_CMARx 寄存器中进行编程。在传输过程中，这些寄存器将保持初始编程的值。软件无法获得当前传输的地址（这个地址位于控制器内部的地址寄存器中，用于存储正在传输中的外设或存储器地址）。

如果将通道 x 配置为**非循环模式**，则在最后一次数据传输完成后（即待传输的单个数据数目达到零后），将不处理任何 DMA 请求。必须禁止 DMA 通道，这样才能将新的数据项数目重新加载到 DMA_CNDTRx 寄存器。

注意： *如果禁止通道 x，DMA 寄存器不会被复位。在通道配置阶段，DMA 通道寄存器（DMA_CCRx、DMA_CPARx 和 DMA_CMARx）仍保留初始编程值。*

在**循环模式**下，最后一次数据传输完成后，DMA_CNDTRx 寄存器将自动重新加载初始编程值。当前的内部地址寄存器重新加载 DMA_CPARx 和 DMA_CMARx 寄存器中的基址值。

通道配置流程

配置 DMA 通道 x 时需按照以下步骤操作：

1. 在 DMA_CPARx 寄存器中设置外设寄存器地址。
   在外设事件发生后，或在存储器到存储器模式下使能通道后，会将数据从该地址移至存储器或从存储器移至该地址。

2. 设置 DMA_CMARx 寄存器中的存储器地址。
   在外设事件发生后，或在存储器到存储器模式下使能通道后，会将数据写入存储器或从存储器读取数据。
3. 在 DMA_CNDTRx 寄存器中配置待传输的总数据数。
   每次数据传输后，该值都会递减。
4. 在 DMA_CCRx 寄存器中配置下列参数：
   – 通道优先级
   – 数据传输方向
   – 循环模式
   – 外设和存储器递增模式
   – 外设和存储器数据大小
   – 传输完成一半和/或全部完成以及/或者出现传输错误时的中断使能
5. 将 DMA_CCRx 寄存器中的 EN 位置 1 以激活通道。

通道在使能后可处理来自此通道所连接外设的任意 DMA 请求，或者启动存储器到存储器块传输。

注意： *通道配置流程的最后两步可合并为对 DMA_CCRx 寄存器进行单次访问来配置和使能通道。*

通道状态和禁止通道

处于活动状态的通道 x 为已使能通道（读取 DMA_CCRx.EN = 1）。活动通道 x 为必须由软件使能（DMA_CCRx.EN 设置为 1），且之后未发生传输错误 (DMA_ISR.TEIFx = 0) 的通道。如果存在传输错误，通道将由硬件自动禁止 (DMA_CCRx.EN = 0)。

可能会出现以下三种用例：

- 挂起和恢复通道

  这对应于以下两个操作：
  – 活动通道由软件禁止（写入 DMA_CCRx.EN = 0，而 DMA_CCRx.EN = 1）。
  – 软件再次使能通道（DMA_CCRx.EN 设置为 1），而无需重新配置其他通道寄存器（例如 DMA_CNDTRx、DMA_CPARx 和 DMA_CMARx）。

  这种情况不受 DMA 硬件支持，无法保证正确执行剩余数据传输。
- 停止和中止通道

  如果应用程序不再需要此活动通道，则可通过软件将其禁止。通道会停止并中止，但 DMA_CNDTRx 寄存器内容可能无法正确反映相对终止的源和目标缓冲区/寄存器，剩余数据传输的个数。
- 中止并重新启动通道

  这对应于软件序列：禁止活动通道，然后重新配置通道并再次将其使能。

  如果满足以下条件，则硬件支持此用例：
  – 应用程序可保证：在软件禁止通道时，相应的主设备端口的 DMA 传输已经完成。例如，应用程序可先在 DMA 模式下禁止外设，以确保没有来自该外设的待处理硬件 DMA 请求。
  – 软件必须对 DMA_CCRx 寄存器进行分步的写访问。首先禁止通道。然后，在需要进行配置更改的情况下，重新配置通道以进行下一次块传输，其中包括 DMA_CCRx。DMA_CCRx.EN=1 时，DMA_CCRx 寄存器存在只读的字段。最后，再次使能通道。

发生通道传输错误时，DMA_CCRx 寄存器的 EN 位由硬件清零。在 DMA_ISR 寄存器的 TEIFx 位置 1 之前，不能通过软件将该 EN 位置 1 来重新激活通道 x。

循环模式（在存储器到外设/外设到存储器传输中）

循环模式可用于处理循环缓冲区和连续数据流（例如 ADC 扫描模式）。可使用 DMA_CCRx 寄存器中的 CIRC 位使能此功能。

注意： *在存储器到存储器模式下，不能使用循环模式。在循环模式 (CIRC = 1) 下使能通道前，软件必须将 DMA_CCRx 寄存器的 MEM2MEM 位清零。如果循环模式已激活，则待传输数据的数目将自动重新装载为在通道配置阶段设置的初始值，并继续响应 DMA 请求。*

为停止循环传输，软件需要在禁止 DMA 通道前使外设停止生成 DMA 请求（例如退出 ADC 扫描模式）。
软件必须在启动/使能传输前，以及在停止循环传输后，明确设定 DMA_CNDTRx 值。

存储器到存储器模式

DMA 通道在没有外设请求触发的情况下同样可以工作。该模式被称为存储器到存储器模式，由软件启动。

如果 DMA_CCRx 寄存器的 MEM2MEM 位置 1，则通道（如果使能）会启动传输。DMA_CNDTRx 寄存器达到零后，传输停止。

注意： *在循环模式下，不得使用存储器到存储器模式。在存储器到存储器模式 (MEM2MEM = 1) 下使能通道前，软件必须将 DMA_CCRx 寄存器的 CIRC 位清零。*

外设到外设模式

在以下情况下，任意 DMA 通道均可工作在外设到外设模式下：

- 选择来自外设的硬件请求触发 DMA 通道时
  此外设为 DMA 发起方，并且控制自身与属于其他存储器映射外设（此外设未配置为 DMA 模式）的寄存器之间的数据传输。
- 未选择任何外设请求和未将任何外设请求连接至 DMA 通道时
  软件通过将 DMA_CCRx 寄存器的 MEM2MEM 位置 1 来配置寄存器到寄存器的传输。

设定传输方向，分配源/目标

DMA_CCRx 寄存器的 DIR 位值用于设置传输方向，因此会标识源和目标，这与源/目标类型（外设或存储器）无关：

- **DIR = 1** 通常用于定义存储器到外设的传输。一般而言，如果 DIR = 1：
  - **源**属性将由 DMA_MARx 寄存器以及 DMA_CCRx 寄存器的 MSIZE[1:0] 位域和 MINC 位定义。
    无论其常用的命名规则如何，上述“存储器”寄存器、位域和位都用于在外设到外设模式下定义源外设。
  - **目标**属性将由 DMA_PARx 寄存器以及 DMA_CCRx 寄存器的 PSIZE[1:0] 位域和 PINC 位定义。
    无论其常用的命名规则如何，上述“外设”寄存器、位域和位都用于在存储器到存储器模式下定义目标存储器。
- **DIR = 0** 通常用于定义外设到存储器的传输。一般而言，如果 DIR = 0：
  - **源**属性将由 DMA_PARx 寄存器以及 DMA_CCRx 寄存器的 PSIZE[1:0] 位域和 PINC 位定义。
    无论其常用的命名规则如何，上述“外设”寄存器、位域和位都用于在存储器到存储器模式下定义源存储器。
  - **目标**属性将由 DMA_MARx 寄存器以及 DMA_CCRx 寄存器的 MSIZE[1:0] 位域和 MINC 位定义。

无论其常用的命名规则如何，上述“存储器”寄存器、位域和位都用于在外设到外设模式下定义目标外设。

9.4.6 DMA 数据宽度、对齐和字节序

如果 PSIZE[1:0] 和 MSIZE[1:0] 不相等，则 DMA 控制器将按照下表所述方式进行数据对齐。

表 30. 可编程的数据宽度和字节序（PINC = MINC = 1 时）

源端口宽度（DIR = 1 时为 MSIZE，否则为 PSIZE）	目标端口宽度（DIR = 1 时为 PSIZE，否则为 MSIZE）	要传输的数据项的数目 (NDT)	源内容：地址或/数据（DIR = 1 时为 DMA_CMARx，否则为 DMA_CPARx）	DMA 传输	目标内容：地址/数据（DIR = 1 时为 DMA_CPARx，否则为 DMA_CMARx）
8	8	4	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1：读取 B0[7:0] @0x0，然后写入 B0[7:0] @0x0 2：读取 B1[7:0] @0x1，然后写入 B1[7:0] @0x1 3：读取 B2[7:0] @0x2，然后写入 B2[7:0] @0x2 4：读取 B3[7:0] @0x3，然后写入 B3[7:0] @0x3	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3
8	16	4	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1：读取 B0[7:0] @0x0，然后写入 00B0[15:0] @0x0 2：读取 B1[7:0] @0x1，然后写入 00B1[15:0] @0x2 3：读取 B2[7:0] @0x2，然后写入 00B2[15:0] @0x4 4：读取 B3[7:0] @0x3，然后写入 00B3[15:0] @0x6	@0x0/00B0 @0x2/00B1 @0x4/00B2 @0x6/00B3
8	32	4	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1：读取 B0[7:0] @0x0，然后写入 000000B0[31:0] @0x0 2：读取 B1[7:0] @0x1，然后写入 000000B1[31:0] @0x4 3：读取 B2[7:0] @0x2，然后写入 000000B2[31:0] @0x8 4：读取 B3[7:0] @0x3，然后写入 000000B3[31:0] @0xC	@0x0/000000B0 @0x4/000000B1 @0x8/000000B2 @0xC/000000B3
16	8	4	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6	1：读取 B1B0[15:0] @0x0，然后写入 B0[7:0] @0x0 2：读取 B3B2[15:0] @0x2，然后写入 B2[7:0] @0x1 3：读取 B5B4[15:0] @0x4，然后写入 B4[7:0] @0x2 4：读取 B7B6[15:0] @0x6，然后写入 B6[7:0] @0x3	@0x0/B0 @0x1/B2 @0x2/B4 @0x3/B6
16	16	4	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6	1：读取 B1B0[15:0] @0x0，然后写入 B1B0[15:0] @0x0 2：读取 B3B2[15:0] @0x2，然后写入 B3B2[15:0] @0x2 3：读取 B5B4[15:0] @0x4，然后写入 B5B4[15:0] @0x4 4：读取 B7B6[15:0] @0x6，然后写入 B7B6[15:0] @0x6	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6
16	32	4	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6	1：读取 B1B0[15:0] @0x0，然后写入 0000B1B0[31:0] @0x0 2：读取 B3B2[15:0] @0x2，然后写入 0000B3B2[31:0] @0x4 3：读取 B5B4[15:0] @0x4，然后写入 0000B5B4[31:0] @0x8 4：读取 B7B6[15:0] @0x6，然后写入 0000B7B6[31:0] @0xC	@0x0/0000B1B0 @0x4/0000B3B2 @0x8/0000B5B4 @0xC/0000B7B6
32	8	4	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC	1：读取 B3B2B1B0[31:0] @0x0，然后写入 B0[7:0] @0x0 2：读取 B7B6B5B4[31:0] @0x4，然后写入 B4[7:0] @0x1 3：读取 BBBAB9B8[31:0] @0x8，然后写入 B8[7:0] @0x2 4：读取 BFBEBDBC[31:0] @0xC，然后写入 BC[7:0] @0x3	@0x0/B0 @0x1/B4 @0x2/B8 @0x3/BC
32	16	4	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC	1：读取 B3B2B1B0[31:0] @0x0，然后写入 B1B0[15:0] @0x0 2：读取 B7B6B5B4[31:0] @0x4，然后写入 B5B4[15:0] @0x2 3：读取 BBBAB9B8[31:0] @0x8，然后写入 B9B8[15:0] @0x4 4：读取 BFBEBDBC[31:0] @0xC，然后写入 BDBC[15:0] @0x6	@0x0/B1B0 @0x2/B5B4 @0x4/B9B8 @0x6/BDBC
32	32	4	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC	1：读取 B3B2B1B0[31:0] @0x0，然后写入 B3B2B1B0[31:0] @0x0 2：读取 B7B6B5B4[31:0] @0x4，然后写入 B7B6B5B4[31:0] @0x4 3：读取 BBBAB9B8[31:0] @0x8，然后写入 BBBAB9B8[31:0] @0x8 4：读取 BFBEBDBC[31:0] @0xC，然后写入 BFBEBDBC[31:0] @0xC	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC

解决 AHB 外设不支持字节/半字写传输的问题

如果 DMA 控制器启动 AHB 字节或半字写传输，则 32 位 AHB 主控数据总线 (HWDATA[31:0]) 的未使用数据线上将会重复所传输的数据。

如果 AHB 从外设不支持字节或半字写传输且不会产生任何错误，则 DMA 控制器会对 HWDATA 的 32 个位执行写操作，如下列两个示例所示：

- 要写入半字 0xABCD，DMA 控制器会将 HWDATA 总线设为 0xABCDABCD，并采用半字数据大小（在 AHB 主控总线中，将 HSIZE 设为 HalfWord）。
- 要写入字节 0xAB，DMA 控制器会将 HWDATA 总线设为 0xABABABAB，并采用字节数据大小（在 AHB 主控总线中，将 HSIZE 设为 Byte）。

假设 AHB/APB 桥为 AHB 32 位从外设，且该外设不考虑 HSIZE 数据，则任何 AHB 字节或半字传输都将转换为 32 位 APB 传输，如下所述：

- 将 AHB 字节写传输转换为 APB 字写传输，如向 0x0、0x1、0x2 或 0x3 地址之一写入 0xB0 转换为向 0x0 地址写入 0xB0B0B0B0。
- 将 AHB 半字写传输转换为 APB 字写传输，如向 0x0 或 0x2 地址写入 0xB1B0 转换为向 0x0 地址写入 0xB1B0B1B0。

9.4.7 DMA 错误管理

当对保留的地址空间执行读写操作时，将生成 DMA 传输错误。如果在 DMA 读或写访问过程中生成了 DMA 传输错误，则硬件会将相应 DMA_CCRx 寄存器的 EN 位清零，从而自动禁止出错的通道 x。

DMA_ISR 寄存器的 TEIFx 位置 1。如果 DMA_CCRx 寄存器的 TEIE 位置 1，还将产生中断。

在 DMA_ISR 寄存器的 TEIFx 位清零（通过将 DMA_IFCR 寄存器的 CTEIFx 位置 1）前，DMA_CCRx 寄存器的 EN 位不能再次由软件置 1（通道 x 重新激活）。

软件在收到与外设相关的通道出现传输错误的通知后，首先要停止这个在 DMA 模式下的外设，以禁止任何挂起或后续的 DMA 请求。随后软件可以正常地将 DMA 和外设重新配置为 DMA 模式，以便进行新的传输。

9.5 DMA 中断

对于每个 DMA 通道 x，在发生“半传输”、“传输完成”或“传输错误”时都会生成中断。可以使用单独的中断使能位以提高灵活性。

表 31. DMA 中断请求

中断请求	中断事件	事件标志	中断使能位
通道 x 中断	通道 x 上发生半传输	HTIFx	HTIEx
	通道 x 上传输完成	TCIFx	TCIEx
	通道 x 上发生传输错误	TEIFx	TEIEx
	通道 x 上发生半传输、传输完成或传输错误	GIFx	-

9.6 DMA 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.2 节*。

DMA 寄存器必须按字（32 位）进行访问。

9.6.1 DMA 中断状态寄存器 (DMA_ISR)

DMA interrupt status register

偏移地址：0x00

复位值：0x0000 0000

每个状态位在软件将 DMA_IFCR 寄存器中相应的清零位或相应的全局清零位 CGIFx 置 1 后由硬件清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留，必须保持复位值。

位 11 **TEIF3**：通道 3 的传输错误 (TE) 标志 (Transfer error (TE) flag for channel 3)
0：无 TE 事件
1：发生 TE 事件

位 10 **HTIF3**：通道 3 的半传输 (HT) 标志 (Half transfer (HT) flag for channel 3)
0：无 HT 事件
1：发生 HT 事件

位 9 **TCIF3**：通道 3 的传输完成 (TC) 标志 (Transfer complete (TC) flag for channel 3)
0：无 TC 事件
1：发生 TC 事件

位 8 **GIF3**：通道 3 的全局中断标志 (Global interrupt flag for channel 3)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

位 7 **TEIF2**：通道 2 的传输错误 (TE) 标志 (Transfer error (TE) flag for channel 2)
0：无 TE 事件
1：发生 TE 事件

位 6 **HTIF2**：通道 2 的半传输 (HT) 标志 (Half transfer (HT) flag for channel 2)
0：无 HT 事件
1：发生 HT 事件

位 5 **TCIF2**：通道 2 的传输完成 (TC) 标志 (Transfer complete (TC) flag for channel 2)
0：无 TC 事件
1：发生 TC 事件

位 4 **GIF2**：通道 2 的全局中断标志 (Global interrupt flag for channel 3)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

位 3 **TEIF1**：通道 1 的传输错误 (TE) 标志 (Transfer error (TE) flag for channel 1)
0：无 TE 事件
1：发生 TE 事件

位 2 **HTIF1**：通道 1 的半传输 (HT) 标志 (Half transfer (HT) flag for channel 1)
0：无 HT 事件
1：发生 HT 事件

位 1 **TCIF1**：通道 1 的传输完成 (TC) 标志 (Transfer complete (TC) flag for channel 1)
0：无 TC 事件
1：发生 TC 事件

位 0 **GIF1**：通道 1 的全局中断标志 (Global interrupt flag for channel 1)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

9.6.2 DMA 中断标志清零寄存器 (DMA_IFCR)

DMA interrupt flag clear register

偏移地址：0x04

复位值：0x0000 0000

将该 DMA_IFCR 寄存器中通道 x 的全局清零位 CGIFx 置 1 后，DMA 硬件会清零 DMA_ISR 寄存器中的相应 GIFx 位以及各个 TEIFx、HTIFx 和 TCIFx 标志。

将该 DMA_IFCR 寄存器中的各个 CTEIFx、CHTIFx 和 CTCIFx 清零位均置 1 后，DMA 硬件会清零 DMA_ISR 寄存器中相应的单独标志和全局标志 GIFx，前提是其他两个单独的标志均未置 1。

对任何标志清零位写入 0 均不起作用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
				w	w	w	w	w	w	w	w	w	w	w	w

位 31:12 保留，必须保持复位值。

位 11 **CTEIF3**：通道 3 的传输错误标志清零 (Transfer error flag clear for channel 3)

位 10 **CHTIF3**：通道 3 的半传输标志清零 (Half transfer flag clear for channel 3)

位 9 **CTCIF3**：通道 3 的传输完成标志清零 (Transfer complete flag clear for channel 3)

位 8 **CGIF3**：通道 3 的全局中断标志清零 (Global interrupt flag clear for channel 3)

位 7 **CTEIF2**：通道 2 的传输错误标志清零 (Transfer error flag clear for channel 2)

位 6 **CHTIF2**：通道 2 的半传输标志清零 (Half transfer flag clear for channel 2)

位 5 **CTCIF2**：通道 2 的传输完成标志清零 (Transfer complete flag clear for channel 2)

位 4 **CGIF2**：通道 2 的全局中断标志清零 (Global interrupt flag clear for channel 2)

位 3 **CTEIF1**：通道 1 的传输错误标志清零 (Transfer error flag clear for channel 1)

位 2 **CHTIF1**：通道 1 的半传输标志清零 (Half transfer flag clear for channel 1)

位 1 **CTCIF1**：通道 1 的传输完成标志清零 (Transfer complete flag clear for channel 1)

位 0 **CGIF1**：通道 1 的全局中断标志清零 (Global interrupt flag clear for channel 1)

9.6.3 DMA 通道 x 配置寄存器 (DMA_CCRx)

DMA channel x configuration register

偏移地址：0x08 + 0x14 * (x - 1)（x = 1 到 3）

复位值：0x0000 0000

EN = 1 时，寄存器位域/位 MEM2MEM、PL[1:0]、MSIZE[1:0]、PSIZE[1:0]、MINC、PINC 和 DIR 为只读。

MEM2MEM 和 CIRC 位的状态不能同时处于高电平。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:15 保留，必须保持复位值。

位 14 **MEM2MEM**：存储器到存储器模式

0：禁止

1：使能

注意：该位由软件置 1 和清零。使能通道 (EN = 1) 后禁止写入。使能通道 (EN = 1) 后为只读。

位 13:12 **PL[1:0]**：优先级 (Priority level)

00：低

01：中

10：高

11：非常高

注意：该位域由软件置 1 和清零。使能通道 (EN = 1) 后禁止写入。使能通道 (EN = 1) 后为只读。

位 11:10 **MSIZE[1:0]**：存储器容量

定义到标识存储器的每次 DMA 传输的数据大小。

在存储器到存储器模式下，如果 DIR = 1，则该位域标识存储器源；如果 DIR = 0，则该位域标识存储器目标。

在外设到外设模式下，如果 DIR = 1，则该位域标识外设源；如果 DIR = 0，则该位域标识外设目标。

00：8 位

01：16 位

10：32 位

11：保留

注意：该位域由软件置 1 和清零。使能通道 (EN = 1) 后禁止写入。使能通道 (EN = 1) 后为只读。

位 9:8 **PSIZE[1:0]**：外设容量

定义到标识外设的每次 DMA 传输的数据大小。

在存储器到存储器模式下，如果 DIR = 1，则该位域标识存储器目标；如果 DIR = 0，则该位域标识存储器源。

在外设到外设模式下，如果 DIR = 1，则该位域标识外设目标；如果 DIR = 0 则该位域标识外设源。

00：8 位

01：16 位

10：32 位

11：保留

注意：该位域由软件置 1 和清零。使能通道 (EN = 1) 后禁止写入。使能通道 (EN = 1) 后为只读。

位 7 **MINC**：存储器递增模式 (Memory increment mode)

定义到标识存储器的每次 DMA 传输的递增模式。

在存储器到存储器模式下，如果 DIR = 1，则该位标识存储器源；如果 DIR = 0，则该位标识存储器目标。

在外设到外设模式下，如果 DIR = 1，则该位标识外设源；如果 DIR = 0，则该位标识外设目标。

0：禁止

1：使能

注意：该位由软件置 1 和清零。使能通道 (EN = 1) 后禁止写入。使能通道 (EN = 1) 后为只读。

位 6 **PINC**：外设递增模式 (Peripheral increment mode)

定义到标识外设的每次 DMA 传输的递增模式。

在存储器到存储器模式下，如果 DIR = 1，则该位标识存储器目标；如果 DIR = 0，则该位标识存储器源。

在外设到外设模式下，如果 DIR = 1，则该位标识外设目标；如果 DIR = 0 则该位标识外设源。

0：禁止

1：使能

注意：该位由软件置 1 和清零。使能通道 (EN = 1) 后禁止写入。使能通道 (EN = 1) 后为只读。

位 5 **CIRC**：循环模式 (Circular mode)

0：禁止

1：使能

注意：该位由软件置 1 和清零。使能通道 (EN = 1) 后禁止写入。使能通道 (EN = 1) 后并非只读。

位 4 **DIR**：数据传输方向 (Data transfer direction)

该位仅在存储器到外设和外设到存储器模式下才能置 1。

0：从外设读取

- 源属性由 PSIZE 和 PINC 以及 DMA_CPARx 寄存器定义。这同样适用于存储器到存储器模式。
- 目标属性由 MSIZE 和 MINC 以及 DMA_CMARx 寄存器定义。这同样适用于外设到外设模式。

1：从存储器读取

- 目标属性由 PSIZE 和 PINC 以及 DMA_CPARx 寄存器定义。这同样适用于存储器到存储器模式。
- 源属性由 MSIZE 和 MINC 以及 DMA_CMARx 寄存器定义。这同样适用于外设到外设模式。

注意：该位由软件置 1 和清零。使能通道 (EN = 1) 后禁止写入。使能通道 (EN = 1) 后为只读。

位 3 **TEIE**：传输错误中断使能 (Transfer error interrupt enable)

0：禁止

1：使能

注意：该位由软件置 1 和清零。使能通道 (EN = 1) 后禁止写入。使能通道 (EN = 1) 后并非只读。

位 2 **HTIE**：半传输中断使能 (Half transfer interrupt enable)

0：禁止

1：使能

注意：该位由软件置 1 和清零。使能通道 (EN = 1) 后禁止写入。使能通道 (EN = 1) 后并非只读。

位 1 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)

0：禁止

1：使能

注意：该位由软件置 1 和清零。使能通道 (EN = 1) 后禁止写入。使能通道 (EN = 1) 后并非只读。

位 0 **EN**：通道使能 (Channel enable)

发生通道传输错误后，该位由硬件清零。在 DMA_ISR 寄存器的 TEIFx 位清零（通过将 DMA_IFCR 寄存器的 CTEIFx 位置 1）前，该位不能再次由软件置 1（通道 x 重新激活）。

0：禁止

1：使能

注意：该位由软件置 1 和清零。

9.6.4 DMA 通道 x 待传输数据数量寄存器 (DMA_CNDTRx)

DMA channel x number of data to transfer register

偏移地址：0x0C + 0x14 * (x - 1)（x = 1 到 3）

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:0 **NDT[15:0]**：待传输的数据数目 (Number of data to transfer)（0 到 2^{16} - 1）

通道使能后，该位域由硬件更新：

- 在每次 DMA“先读后写”传输后，该位域都会递减，指示剩余的待传输数据项数目。
- 如果通道未处于循环模式（DMA_CCRx 寄存器中的 CIRC = 0），则在达到设定的待传输数据数目时，该位域会保持为零。
- 如果通道处于循环模式 (CIRC = 1)，则在传输完成后该位域会自动重新装载之前设定的值。

如果该位域为零，则无论通道状态如何（使能或未使能），都不会处理任何传输。

注意：该位域由软件置 1 和清零。使能通道 (EN = 1) 后禁止写入。使能通道 (EN = 1) 后为只读。

9.6.5 DMA 通道 x 外设地址寄存器 (DMA_CPARx)

DMA channel x peripheral address register

偏移地址：0x10 + 0x14 * (x - 1)（x = 1 到 3）

复位值：0x0000 0000

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

位 31:0 **PA[31:0]**：外设地址 (Peripheral address)

其中包含读/写数据的外设数据寄存器的基址。

PSIZE[1:0] = 01（16 位）时，忽略 PA[31:0] 的位 0。访问将自动对齐到半字地址。

PSIZE[1:0] = 10（32 位）时，忽略 PA[31:0] 的位 1 和位 0。访问将自动对齐到字地址。

在存储器到存储器模式下，如果 DIR = 1，则该位域标识存储器目标地址；如果 DIR = 0，则该位域标识存储器源地址。

在外设到外设模式下，如果 DIR = 1，则该位域标识外设目标地址；如果 DIR = 0，则该位域标识外设源地址。

注意： 该位域由软件置 1 和清零。使能通道 (EN = 1) 后禁止写入。使能通道 (EN = 1) 后并非只读。

9.6.6 DMA 通道 x 存储器地址寄存器 (DMA_CMARx)

DMA channel x memory address register

偏移地址：0x14 + 0x14 * (x - 1)（x = 1 到 3）

复位值：0x0000 0000

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

位 31:0 **MA[31:0]**：外设地址 (Peripheral address)

其中包含读/写数据的存储器的基址。

MSIZE[1:0] = 01（16 位）时，忽略 MA[31:0] 的位 0。访问将自动对齐到半字地址。

MSIZE[1:0] = 10（32 位）时，忽略 MA[31:0] 的位 1 和位 0。访问将自动对齐到字地址。

在存储器到存储器模式下，如果 DIR = 1，则该位域标识存储器源地址；如果 DIR = 0，则该位域标识存储器目标地址。

在外设到外设模式下，如果 DIR = 1，则该位域标识外设源地址；如果 DIR = 0，则该位域标识外设目标地址。

注意： 该位域由软件置 1 和清零。使能通道 (EN = 1) 后禁止写入。使能通道 (EN = 1) 后并非只读。

9.6.7 DMA 寄存器映射

表 32.DMA 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	DMA_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
	复位值																					0	0	0	0	0	0	0	0	0	0	0	0
0x004	DMA_IFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
	复位值																					0	0	0	0	0	0	0	0	0	0	0	0
0x008	DMA_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	复位值																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	DMA_CNDTR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	DMA_CPAR1	PA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	DMA_CMAR1	MA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	保留	保留。																															
0x01C	DMA_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	复位值																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	DMA_CNDTR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	DMA_CPAR2	PA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x028	DMA_CMAR2	MA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x02C	保留	保留。																															
0x030	DMA_CCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	复位值																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034	DMA_CNDTR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	复位值																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x038	DMA_CPAR3	PA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03C	DMA_CMAR3	MA[31:0]																															
	复位值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 2.2 节*。

10 DMA 请求复用器 (DMAMUX)

10.1 简介

外设通过设置其 DMA 请求信号来指示存在 DMA 传输请求。DMA 请求信号在 DMA 控制器处理 DMA 请求并生成 DMA 确认信号之前一直处于挂起状态，之后相应的 DMA 请求信号就将变为无效。

在本文档中，DMA 请求/确认协议所需的控制信号组并未明确列出和说明，而是称作 DMA 请求线。

DMAMUX 请求复用器可在产品的外设和 DMA 控制器之间重新配置（路由）DMA 请求线。该路由功能通过可编程的多通道 DMA 请求线复用器来确保实现。每条通道可不受限制地选择一个 DMA 请求线，或者按照与来自其 DMAMUX 同步输入的事件相同步的方式选择一个 DMA 请求线。此外，DMAMUX 还可用作其输入触发信号的可编程事件的 DMA 请求发生器。

*第 10.3.1 节*规定了 DMAMUX 实例的数量及其主要特性。

以下的信号分配取决于产品实现：DMAMUX 请求复用器输入到外设 DMA 请求线和DMAMUX 请求发生器输出的分配，DMAMUX 请求复用器输出到 DMA 控制器通道的分配以及 DMAMUX同步和触发输入到内部和外部信号的分配，详细描述见*第 10.3.2 节*。

10.2 DMAMUX 主要特性

- 3 通道可编程 DMA 请求线复用器输出
- 4 通道 DMA 请求发生器
- 23 个 DMA 请求发生器的触发输入
- 23 个同步输入
- 每个 DMA 请求发生器通道均具有：
  - DMA 请求触发输入选择器
  - DMA 请求计数器
  - 所选 DMA 请求触发输入的事件溢出标志
- 每个 DMA 请求线复用器通道输出均具有：
  - 多达 49 个来自外设的输入 DMA 请求线
  - 一个 DMA 请求线输出
  - 同步输入选择器
  - DMA 请求计数器
  - 所选同步输入的事件溢出标志
  - 一个事件输出，用于 DMA 请求链接

10.3 DMAMUX 实现

10.3.1 DMAMUX 实例化

DMAMUX 通过下表中列出的硬件配置参数进行实例化。

表 33. DMAMUX 实例化

特性	DMAMUX
DMAMUX 输出请求通道数	3
DMAMUX 请求发生器通道数	4
DMAMUX 请求触发输入数	23
DMAMUX 同步输入数	23
DMAMUX 外设请求输入数	49

10.3.2 DMAMUX 映射

资源到 DMAMUX 的映射通过硬接线实现。

表 34. DMAMUX：复用器输入到资源的分配

DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源
1	dmamux1_req_gen0	19	保留	37	TIM3_UP
2	dmamux1_req_gen1	20	TIM1_CH1	38	保留
3	dmamux1_req_gen2	21	TIM1_CH2	39	保留
4	dmamux1_req_gen3	22	TIM1_CH3	40	保留
5	adc_dma	23	TIM1_CH4	41	保留
6	保留	24	TIM1_TRIG_COM	42	保留
7	保留	25	TIM1_UP	43	保留
8	保留	26	保留	44	TIM16_CH1
9	保留	27	保留	45	TIM16_TRIG_COM
10	i2c1_rx_dma	28	保留	46	TIM16_UP
11	i2c1_tx_dma	29	保留	47	TIM17_CH1
12	保留	30	保留	48	TIM17_TRIG_COM
13	保留	31	保留	49	TIM17_UP
14	保留	32	TIM3_CH1	50	usart1_rx_dma
15	保留	33	TIM3_CH2	51	usart1_tx_dma
16	spi1_rx_dma	34	TIM3_CH3	52	usart2_rx_dma
17	spi1_tx_dma	35	TIM3_CH4	53	usart2_tx_dma
18	保留	36	TIM3_TRIG	54	保留

表 35. DMAMUX：触发输入到资源的分配

触发输入	资源	触发输入	资源
0	EXTI 线 0	12	EXTI 线 12
1	EXTI 线 1	13	EXTI 线 13
2	EXTI 线 2	14	EXTI 线 14
3	EXTI 线 3	15	EXTI 线 15
4	EXTI 线 4	16	dmamux_evt0
5	EXTI 线 5	17	dmamux_evt1
6	EXTI 线 6	18	dmamux_evt2
7	EXTI 线 7	19	保留
8	EXTI 线 8	20	保留
9	EXTI 线 9	21	TIM14
10	EXTI 线 10	22	保留
11	EXTI 线 11	23	保留

表 36. DMAMUX：同步输入到资源的分配

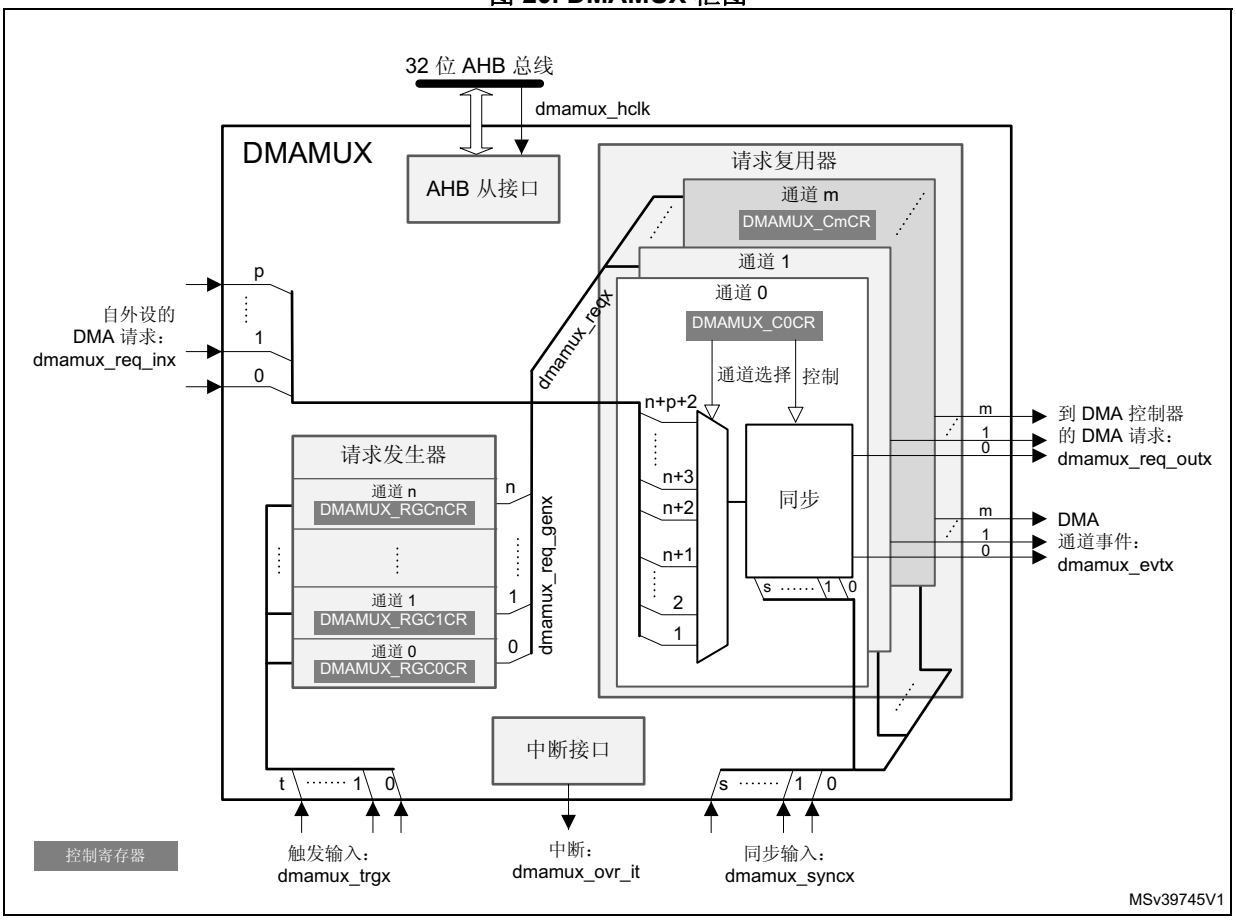
触发输入	资源	触发输入	资源
0	EXTI 线 0	12	EXTI 线 12
1	EXTI 线 1	13	EXTI 线 13
2	EXTI 线 2	14	EXTI 线 14
3	EXTI 线 3	15	EXTI 线 15
4	EXTI 线 4	16	dmamux_evt0
5	EXTI 线 5	17	dmamux_evt1
6	EXTI 线 6	18	dmamux_evt2
7	EXTI 线 7	19	保留
8	EXTI 线 8	20	保留
9	EXTI 线 9	21	TIM14
10	EXTI 线 10	22	保留
11	EXTI 线 11	23	保留

10.4 DMAMUX 功能描述

10.4.1 DMAMUX 框图

图 20 显示了 DMAMUX 框图。

图 20. DMAMUX 框图



DMAMUX 具有两个主要子模块：请求线复用器和请求线发生器。

该实现可：

- 分配 DMAMUX 请求复用器子模块输入 (dmamux_reqx)，这些输入来自外设 (dmamux_req_inx) 和 DMAMUX 请求发生器子模块的通道 (dmamux_req_genx)
- 将 DMAMUX 请求输出分配到 DMA 控制器的通道 (dmamux_req_outx)
- 将内部或外部信号分配到 DMA 请求触发输入 (dmamux_trgx)
- 将内部或外部信号分配到同步输入 (dmamux_syncx)

10.4.2 DMAMUX 信号

表 37 列出了 DMAMUX 信号。

表 37. DMAMUX 信号

信号名称	说明
dmamux_hclk	DMAMUX AHB 时钟
dmamux_req_inx	DMAMUX DMA 请求线输入（来自外设）
dmamux_trgx	DMAMUX DMA 请求触发输入（到请求发生器子模块）
dmamux_req_genx	DMAMUX 请求发生器子模块通道输出
dmamux_reqx	DMAMUX 请求复用器子模块输入（来自外设请求和请求发生器通道）
dmamux_syncx	DMAMUX 同步输入（到请求复用器子模块）
dmamux_req_outx	DMAMUX 请求输出（到 DMA 控制器）
dmamux_evtx	DMAMUX 事件输出

10.4.3 DMAMUX 通道

DMAMUX 通道是一个可能包括额外的 DMAMUX 请求发生器通道的请求复用器通道，具体取决于所选的请求复用器输入。

一个DMAMUX 请求复用器通道连接到 DMA 控制器的一个通道且专用于这个通道。

通道配置流程

请遵循以下顺序来配置 DMAMUX x 通道和相关的 DMA 通道 y：

1. 对 DMA 通道 y 进行完整的设置和配置，但不要使能通道 y。
2. 对相关 DMAMUX y 通道进行完整的设置和配置。
3. 最后，将 DMA y 通道寄存器中的 EN 位置 1 以激活 DMA 通道。

10.4.4 DMAMUX 请求线复用器

DMAMUX 请求复用器及其多条通道可确保 DMA 请求/确认控制信号（称为 DMA 请求线）的实际路由。

每个 DMA 请求线并行连接到 DMAMUX 请求线复用器的所有通道。

DMA 请求来自外设或 DMAMUX 请求发生器。

DMAMUX 请求线复用器通道 x 按照 DMAMUX_CxCR 寄存器中的 DMAREQ_ID 位域的配置来选择 DMA 请求线编号。

注意： *位域 DMAREQ_ID 为空值表示未选择任何 DMA 请求线。*

小心： 除非应用程序可保证所连接的两个 DMA 通道不会同时激活，否则不得为不同的 x 和 y DMAMUX 请求复用器通道编程相同的非空 DMAREQ_ID（通过 DMAMUX_CxCR 和 DMAMUX CyCR）。

除了 DMA 请求选择外，还可根据需要配置和使能同步模式和/或事件生成。

同步模式和通道事件生成

每个 DMAMUX 请求线复用器通道 x 可单独同步，方法是将 DMAMUX_CxCR 寄存器中的同步使能 (SE) 位置 1。

DMAMUX 具有多个同步输入。同步输入并行连接到请求复用器的所有通道。

同步输入通过给定通道 x 的 DMAMUX_CxCR 寄存器中的 SYNC_ID 位域选择。

当某个通道处于此同步模式时，如果通过 DMAMUX_CxCR 寄存器的 SPOL[1:0] 位域在所选输入同步信号上检测到可配置的上升沿/下降沿，则所选输入 DMA 请求线信号将被传送到复用器通道输出。

此外，DMAMUX 请求复用器内部有一个可编程 DMA 请求计数器，可用于实现通道请求输出生成功能以及事件生成功能。通道 x 输出上的事件生成功能通过 DMAMUX_CxCR 寄存器的 EGE 位（事件生成使能）使能。

如*图 22* 所示，检测到同步输入的边沿后，挂起的所选输入 DMA 请求线将连接到 DMAMUX 复用器通道 x 输出。

注意： *如果在不存在挂起的所选输入 DMA 请求线的情况下发生同步事件，则会将其丢弃。再次发生同步事件之前，后续有效的输入请求线不会连接至 DMAMUX 复用器通道输出。*

此后，DMA 控制器每次处理连接的 DMAMUX 请求时（已处理请求被禁止），DMAMUX 请求计数器都会递减。下溢时，DMA 请求计数器将自动加载 DMAMUX_CxCR 寄存器的 NBREQ 位域中的值，输入 DMA 请求线将与复用器通道 x 输出断开。

因此，检测到同步事件后传送到复用器通道 x 输出的 DMA 请求数等于 NBREQ 位域中的值加 1。

注意： *只能在相应复用器通道 x 的同步使能位 (SE) 和事件生成使能位 (EGE) 均禁止时通过软件写入 NBREQ 位域的值。*

图 21. DMAMUX 请求线复用器通道的同步模式

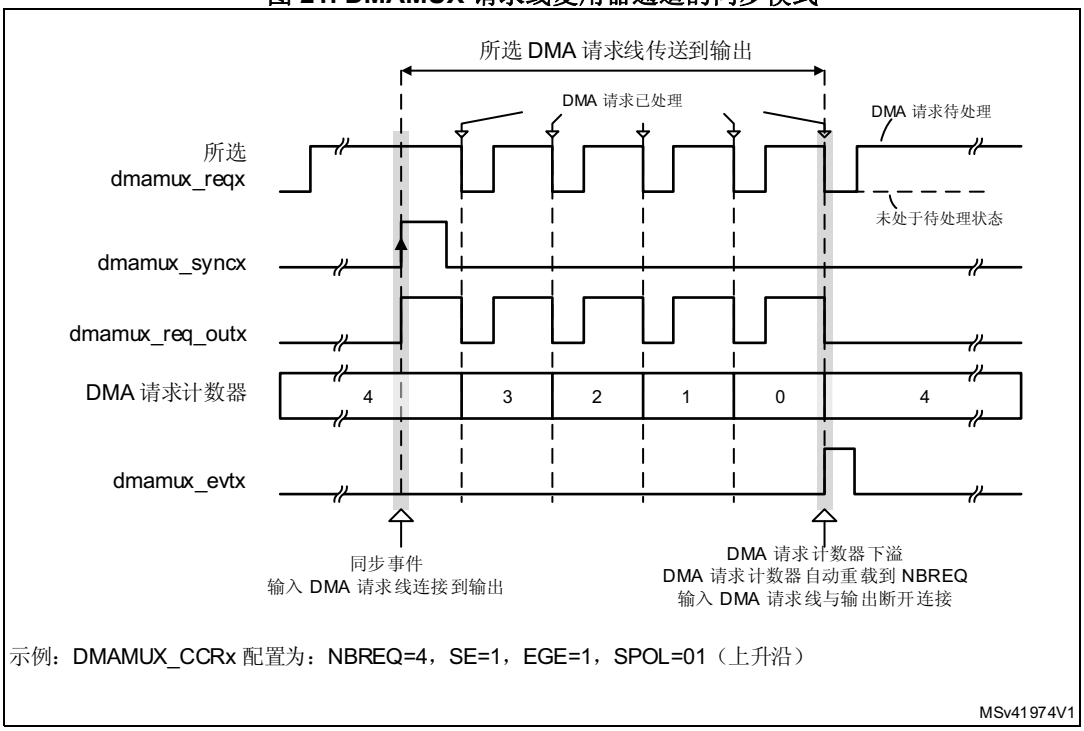
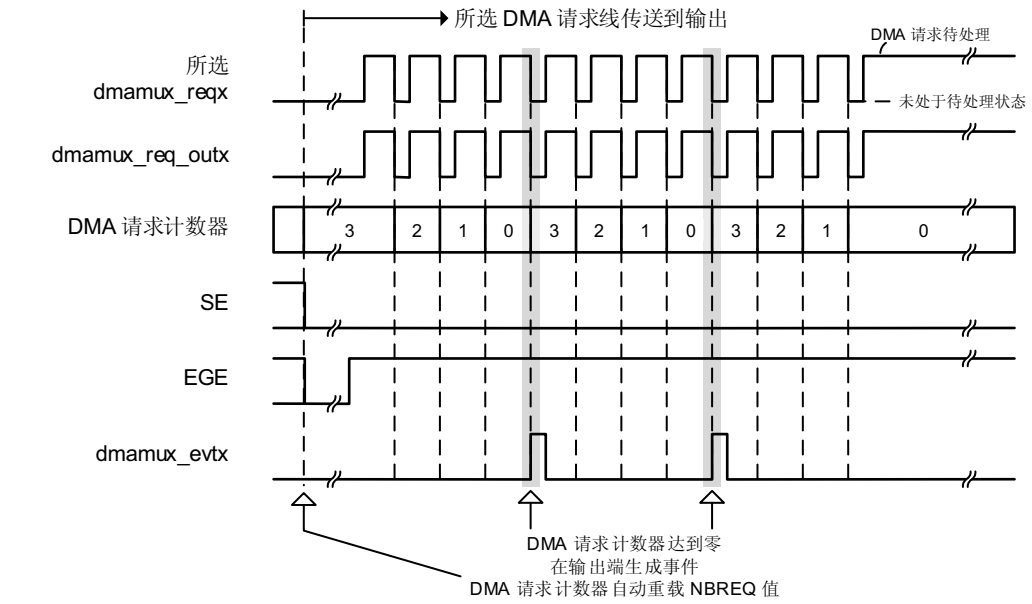


图 22. DMA 请求线复用器通道的事件生成



如果 EGE 使能，当复用器通道的 DMA 请求计数器自动重新装入已编程 NBREQ 位域的值时，复用器通道会生成一个通道事件（即，一个 AHB 时钟周期的脉冲），如*图 21* 和*图 22* 所示。

注意： *如果 EGE 使能且 NBREQ = 0，则在每个处理的 DMA 请求后都会生成一个事件。*

注意： *如果边沿后的状态保持稳定的时间持续 2 个 AHB 时钟周期以上，则会检测到同步事件（边沿）。*

写入 DMAMUX_CxCR 寄存器后，同步事件将被屏蔽 3 个 AHB 时钟周期。

同步溢出和中断

如果在请求计数器下溢（通过 DMAMUX_CxCR 寄存器的 NBREQ 位域编程的内部请求计数器）之前发生新的同步事件，则 DMAMUX_CSR 寄存器中的同步溢出标志位 SOFx 会置 1。

注意： *DMA 控制器的相关通道使用结束后，必须禁止请求复用器通道 x 同步 (DMAMUX_CxCR.SE = 0)。否则，在新检测到同步事件后，由于未从 DMA 控制器收到 DMA 确认（即无已处理请求），将会发生同步溢出。*

溢出标志 SOFx 的复位方式是将 DMAMUX_CFR 寄存器中的相关清除同步溢出标志位 CSOFx 置 1。

如果 DMAMUX_CxCR 寄存器中的同步溢出中断使能位 SOIE 置 1，则将同步溢出标志置 1 会产生中断。

10.4.5 DMAMUX 请求发生器

DMAMUX 请求发生器会在其 DMA 请求触发输入上出现触发事件后产生 DMA 请求。

DMAMUX 请求发生器有多个通道。DMA 请求触发输入并行连接到所有通道。

DMAMUX 请求发生器通道的输出是 DMAMUX 请求线复用器的输入。

每个 DMAMUX 请求发生器通道 x 在相应的 DMAMUX_RGxCR 寄存器中都有一个使能位 GE（发生器使能）。

DMAMUX 请求发生器通道 x 的 DMA 请求触发输入通过相应 DMAMUX_RGxCR 寄存器中的 SIG_ID（触发信号 ID）位域选择。

DMA 请求触发输入上的触发事件可以是上升沿、下降沿或任一边沿。有效边沿通过相应 DMAMUX_RGxCR 寄存器中的 GPOL（发生器极性）位域选择。

触发事件后，相应发生器通道开始在其输出上生成 DMA 请求。连接的 DMA 控制器每次处理 DMAMUX 生成的请求时（已处理请求被禁止），内置（DMAMUX 请求发生器内）DMA 请求计数器都会递减。出现下溢时，请求生成器通道会停止生成 DMA 请求，DMA 请求计数器将在收到下一个触发事件时自动重新加载其设定值。

因此，触发事件后生成的 DMA 请求的数量为 GNBREQ + 1。

注意： *只能在相应发生器通道 x 的使能位 GE 被禁止时通过软件写入 GNBREQ 位域的值。*

没有硬件写保护。

如果边沿后的状态保持稳定的时间持续 2 个 AHB 时钟周期以上，则会检测到触发事件（边沿）。

写入 DMAMUX_RGxCR 寄存器后，触发事件将被屏蔽 3 个 AHB 时钟周期。

触发溢出和中断

如果在 DMAMUX 请求发生器计数器下溢（通过 DMAMUX_RGxCR 寄存器的 GNBREQ 位域编程的内部计数器）之前发生新的 DMA 请求触发事件，并且请求发生器通道 x 通过 GE 使能，则 DMAMUX_RGSR 寄存器中的请求触发事件溢出标志位 OFx 将通过硬件置为有效。

注意： *DMA 控制器的相关通道使用结束后，必须禁止请求发生器通道 x (DMAMUX_RGxCR.GE = 0)。否则，在新检测到触发事件后，由于未从 DMA 收到确认（即无已处理请求），将会发生触发溢出。*

溢出标志 OFx 的复位方式是将 DMAMUX_RGCFR 寄存器中的相关清除溢出标志位 COFx 置 1。

如果 DMAMUX_RGxCR 寄存器中的 DMA 请求触发事件溢出中断使能位 OIE 置 1，则将 DMAMUX 请求触发溢出标志置 1 会产生中断。

10.5 DMAMUX 中断

发生以下事件时会产生中断：

- 每个 DMA 请求线复用器通道发生同步事件溢出
- 每个 DMA 请求发生器通道发生触发事件溢出

对于任一事件，每个通道的中断使能、状态和清除标志寄存器位均可单独使用。

表 38. DMAMUX 中断

中断信号	中断事件	事件标志	清除位	使能位
dmamuxovr_it	DMAMUX 请求线复用器的通道 x 上发生同步事件溢出	SOFx	CSOFx	SOIE
	DMAMUX 请求发生器的通道 x 上发生触发事件溢出	OFx	COFx	OIE

10.6 DMAMUX 寄存器

有关 DMAMUX 基址的信息，请参见包括寄存器边界地址的表格。

DMAMUX 寄存器可按字节（8 位）、半字（16 位）或字（32 位）进行访问。地址必须与数据大小对齐。

10.6.1 DMAMUX 请求线复用器通道 x 配置寄存器 (DMAMUX_CxCR)

DMAMUX request line multiplexer channel x configuration register

偏移地址：0x000 + 0x04 * x（x = 0 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
						rw	rw			rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **SYNC_ID[4:0]**：同步标识 (Synchronization identification)
选择同步输入（请参见*表 36：DMAMUX：同步输入到资源的分配*）。

位 23:19 **NBREQ[4:0]**：要转发的 DMA 请求数减 1 (Number of DMA requests minus 1 to forward)
定义在同步事件后要转发到 DMA 控制器的 DMA 请求的数量，和/或生成输出事件前的 DMA 请求的数量。
只有当 SE 和 EGE 位均为低电平时，才能写入该位域。

位 18:17 **SPOL[1:0]**：同步极性 (Synchronization polarity)
定义所选同步输入的边沿极性：
00：无事件（无同步也无检测）。
01：上升沿
10：下降沿
11：上升沿和下降沿

位 16 **SE**：同步使能 (Synchronization enable)
0：禁止同步
1：使能同步

位 15:10 保留，必须保持复位值。

位 9 **EGE**：事件生成使能 (Event generation enable)
0：禁止事件生成
1：使能事件生成

位 8 **SOIE**：同步上溢中断使能 (Synchronization overrun interrupt enable)
0：禁止中断
1：使能中断

位 7:6 保留，必须保持复位值。

位 5:0 **DMAREQ_ID[5:0]**：DMA 请求标识 (DMA request identification)
选择输入 DMA 请求。有关复用器输入到资源的分配的信息，请参见 DMAMUX 表。

10.6.2 DMAMUX 请求线复用器中断通道状态寄存器 (DMAMUX_CSR)

DMAMUX request line multiplexer interrupt channel status register

偏移地址：0x080

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOF2	SOF1	SOF0
													r	r	r

位 31:3 保留，必须保持复位值。

位 2:0 **SOF[2:0]**：同步溢出事件标志 (Synchronization overrun event flag)
当 DMA 请求线复用器通道 x 上发生同步事件且 DMA 请求计数器的值小于 NBREQ 时，该标志将置 1。
该标志的清零方式是向 DMAMUX_CFR 寄存器中的相应 CSOFx 位写入 1。

10.6.3 DMAMUX 请求线复用器中断清除标志寄存器 (DMAMUX_CFR)

DMAMUX request line multiplexer interrupt clear flag register

偏移地址：0x084

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSOF2	CSOF1	CSOF0
													w	w	w

位 31:3 保留，必须保持复位值。

位 2:0 **CSOF[2:0]**：清零同步溢出事件标志 (Clear synchronization overrun event flag)
将 1 写入每个位时，DMAMUX_CSR 寄存器中相应的溢出标志 SOFx 将清零。

10.6.4 DMAMUX 请求发生器通道 x 配置寄存器 (DMAMUX_RGxCR)

DMAMUX request generator channel x configuration register

偏移地址：0x100 + 0x04 * x（x = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL[1:0]		GE
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
							rw				rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:19 **GNBREQ[4:0]**：要生成的 DMA 请求数（减 1）(Number of DMA requests to be generated (minus 1))

定义触发事件后生成的 DMA 请求的数量。生成的 DMA 请求的实际数量为 GNBREQ+1。

Note: 只有在禁止 GE 后才能写入该位域。

位 18:17 **GPOL[1:0]**：DMA 请求发生器触发极性 (DMA request generator trigger polarity)

定义所选触发输入的边沿极性

00：无事件，即无触发检测和生成。

01：上升沿

10：下降沿

11：上升沿和下降沿

位 16 **GE**：DMA 请求发生器通道 x 使能 (DMA request generator channel x enable)

0：禁止 DMA 请求发生器通道 x

1：使能 DMA 请求发生器通道 x

位 15:9 保留，必须保持复位值。

位 8 **OIE**：触发上溢中断使能 (Trigger overrun interrupt enable)

0：禁止在出现触发溢出事件时产生中断。

1：使能在出现触发溢出事件时产生中断。

位 7:5 保留，必须保持复位值。

位 4:0 **SIG_ID[4:0]**：信号标识 (Signal identification)

选择用于 DMA 请求发生器的通道 x 的 DMA 请求触发输入

10.6.5 DMAMUX 请求发生器中断状态寄存器 (DMAMUX_RGSR)

DMAMUX request generator interrupt status register

偏移地址：0x140

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OF3	OF2	OF1	OF0
												r	r	r	r

位 31:4 保留，必须保持复位值。

位 3:0 **OF[3:0]**：触发溢出事件标志 (Trigger overrun event flag)

在请求计数器下溢（通过 DMAMUX_RGxCR 寄存器的 GNBREQ 位域编程的内部请求计数器）之前，在 DMA 请求发生器通道 x 上发生新触发事件时，该标志将置 1。

该标志的清零方式是向 DMAMUX_RGCFR 寄存器中的相应 COFx 位写入 1。

10.6.6 DMAMUX 请求发生器中断清除标志寄存器 (DMAMUX_RGCFR)

DMAMUX request generator interrupt clear flag register

偏移地址：0x144

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COF3	COF2	COF1	COF0
												w	w	w	w

位 31:4 保留，必须保持复位值。

位 3:0 **COF[3:0]**：清零触发溢出事件标志 (Clear trigger overrun event flag)

将 1 写入每个位时，DMAMUX_RGSR 寄存器中相应的溢出标志 OFx 将清零。

10.6.7 DMAMUX 寄存器映射

下表汇总了 DMAMUX 寄存器和复位值。有关 DMAMUX 寄存器的基址，请参见寄存器边界地址表。

表 39. DMAMUX 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	DMAMUX_C0CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	复位值				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x004	DMAMUX_C1CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	复位值				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x008	DMAMUX_C2CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	复位值				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x0 - 0x07C	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x080	DMAMUX_CSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOF2	SOF1	SOF0
	复位值																														0	0	0
0x084	DMAMUX_CFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSOF2	CSOF1	CSOF0
	复位值																														0	0	0
0x088 - 0x0FC	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x100	DMAMUX_RG0CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL[1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	复位值									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x104	DMAMUX_RG1CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL[1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	复位值									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x108	DMAMUX_RG2CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL[1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	复位值									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x10C	DMAMUX_RG3CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL[1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	复位值									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x110 - 0x13C	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x140	DMAMUX_RGSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OF3	OF2	OF1	OF0
	复位值																													0	0	0	0
0x144	DMAMUX_RGCFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COF3	COF2	COF1	COF0
	复位值																													0	0	0	0
0x148 - 0x3FC	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

有关寄存器边界地址的信息，请参见 *Section 2.2 on page 37*。

11 嵌套向量中断控制器 (NVIC)

11.1 主要特性

- 32 个可屏蔽中断通道（不包括 16 根 Cortex®-M0+ 中断线）
- 4 个可编程优先级（使用了 2 位中断优先级）
- 低延迟异常和中断处理
- 电源管理控制
- 系统控制寄存器的实现

NVIC 和处理器内核接口紧密配合，可以实现低延迟的中断处理和晚到中断的高效处理。

包括内核异常在内的所有中断均通过 NVIC 进行管理。关于异常和 NVIC 编程的更多信息，请参见编程手册 PM0223。

11.2 SysTick 校准值寄存器

SysTick 校准值设为 1000。SysTick 重载值寄存器可以根据实际的 HCLK 频率和所需的时间周期进行调整，更多详细信息请参见 PM0223。

11.3 中断和异常向量

表 40 为向量表。外设相关信息仅适用于包含相应外设的器件。

表 40. 向量表(1)

位置	优先级	优先级类型	缩略语	说明	地址
-	-	-	-	保留	0x0000_0000
-	-3	固定	复位	复位	0x0000_0004
-	-2	固定	NMI_Handler	不可屏蔽中断。SRAM 奇偶校验错误、HSE CSS 和 LSE CSS 连接到 NMI 向量。	0x0000_0008
-	-1	固定	HardFault_Handler	所有类型的错误	0x0000_000C
-	-	-	-	保留	0x0000_0010 0x0000_0014 0x0000_0018 0x0000_001C 0x0000_0020 0x0000_0024 0x0000_0028
-	3	可设置	SVC_Handler	通过 SWI 指令调用的系统服务	0x0000_002C
-	-	-	-	保留	0x0000_0030 0x0000_0034
-	5	可设置	PendSV_Handler	可挂起的系统服务请求	0x0000_0038
-	6	可设置	SysTick_Handler	系统节拍定时器	0x0000_003C

表 40. 向量表[1]（续）

位置	优先级	优先级类型	缩略语	说明	地址
0	7	可设置	WWDG	窗口看门狗中断	0x0000_0040
1	-	-	-	保留	0x0000_0044
2	9	可设置	RTC	RTC 中断（EXTI 线 19）	0x0000_0048
3	10	可设置	FLASH	Flash 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 0 和 1 中断	0x0000_0054
6	13	可设置	EXTI2_3	EXTI 线 2 和 3 中断	0x0000_0058
7	14	可设置	EXTI4_15	EXTI 线 4 到 15 中断	0x0000_005C
8	-	-	-	保留	0x0000_0060
9	16	可设置	DMA1_Channel1	DMA1 通道 1 中断	0x0000_0064
10	17	可设置	DMA1_Channel2_3	DMA1 通道 2 和 3 中断	0x0000_0068
11	18	可设置	DMAMUX	DMAMUX 中断	0x0000_006C
12	19	可设置	ADC	ADC 中断	0x0000_0070
13	20	可设置	TIM1_BRK_UP_TRG_COM	TIM1 刹车、更新、触发和换向中断	0x0000_0074
14	21	可设置	TIM1_CC	TIM1 捕获比较中断	0x0000_0078
15	-	-	-	保留	0x0000_007C
16	23	可设置	TIM3	TIM3 全局中断	0x0000_0080
17	-	-	-	保留	0x0000_0084
18	-	-	-	保留	0x0000_0088
19	26	可设置	TIM14	TIM14 全局中断	0x0000_008C
20	-	-	-	保留	0x0000_0090
21	28	可设置	TIM16	TIM16 全局中断	0x0000_0094
22	29	可设置	TIM17	TIM17 全局中断	0x0000_0098
23	30	可设置	I2C1	I2C1 全局中断（与 EXTI 23 结合使用）	0x0000_009C
24	-	-	-	保留	0x0000_00A0
25	32	可设置	SPI1	SPI1/I2S1 全局中断	0x0000_00A4
26	-	-	-	保留	0x0000_00A8
27	34	可设置	USART1	USART1 全局中断（与 EXTI 25 结合使用）	0x0000_00AC
28	35	可设置	USART2	USART2 全局中断	0x0000_00B0
29	-	-	-	保留	0x0000_00B4
30	-	-	-	保留	0x0000_00B8
31	-	-	-	保留	0x0000_00BC

1. 灰色单元格对应于 Cortex®-M0+ 中断。

12 扩展中断和事件控制器 (EXTI)

扩展中断和事件控制器 (EXTI) 通过可配置的事件输入（线）和直接事件输入（线）来管理 CPU 和系统唤醒。它可以为电源控制提供唤醒请求，给 CPU NVIC 产生中断请求以及给 CPU 的事件输入产生事件。对于 CPU，需通过额外的事件生成模块 (EVG) 来生成 CPU 事件信号。

EXTI 唤醒请求可以唤醒处于停止模式的系统。

此外，还可以在运行模式下生成中断请求和事件请求。

EXTI 还包含 EXTI I/O 端口复用器。

12.1 EXTI 主要特性

EXTI 的主要特性如下：

- 任一输入端发生事件后都会唤醒系统
- 唤醒标志和 CPU 中断生成，用于在源外设中没有唤醒标志的事件
- 事件可以配置（来自 I/O、没有相关中断挂起状态位的外设或产生脉冲的外设）
  - 可选择的有效触发边沿
  - 上升沿和下降沿中断挂起状态位互不关联
  - 独立的中断和事件生成掩码，用于调节 CPU 唤醒、中断和事件生成
  - 支持软件触发
- 直接事件（来自具有相关标志和中断挂起状态位的外设）
  - 固定上升沿有效触发
  - EXTI 中没有中断挂起状态位
  - 独立的中断和事件生成掩码，用于调节 CPU 唤醒和事件生成
  - 不支持软件触发
- I/O 端口选择器

12.2 EXTI 框图

EXTI 包含一个通过 AHB 接口访问的寄存器模块、一个事件输入触发模块一个屏蔽模块和一个 EXTI 复用器，如*图 23* 所示。

寄存器模块包含所有 EXTI 寄存器。

事件输入触发模块提供事件输入边沿触发逻辑。

屏蔽模块为不同的唤醒、中断和事件输出及其屏蔽功能提供事件输入分配。

针对 EXTI 事件信号，EXTI 复用器可以选择 I/O 端口。

图 23. EXTI 框图

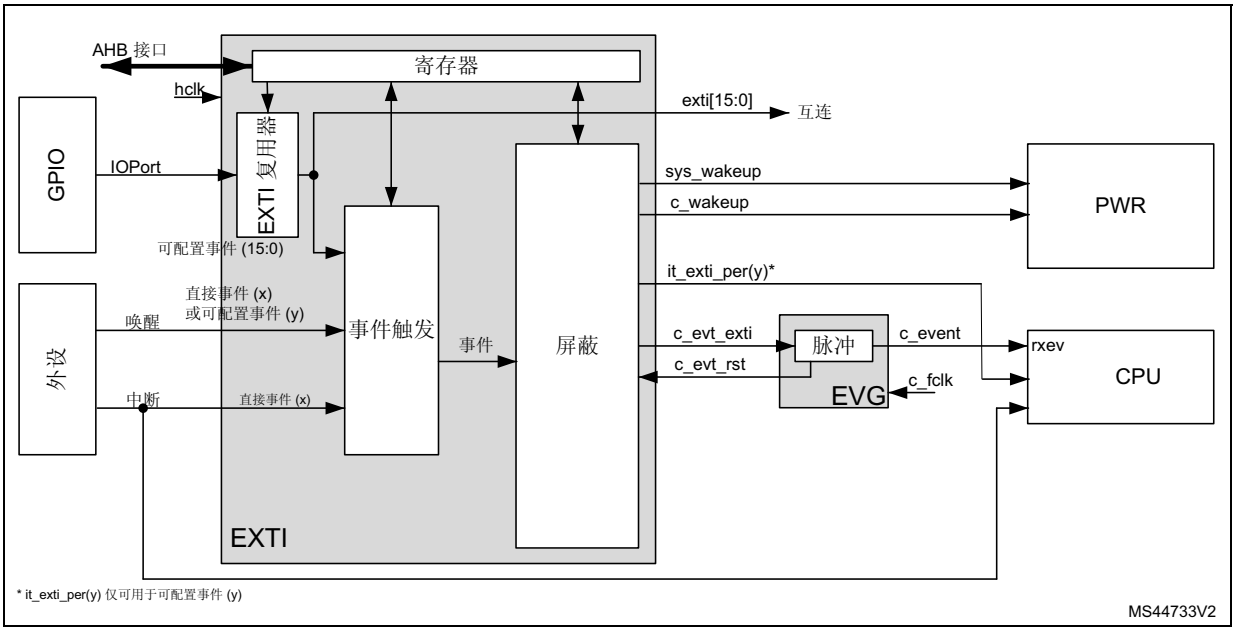


表 41. EXTI 信号概述

信号名称	I/O	说明
AHB 接口	I/O	EXTI 寄存器总线接口。将一个事件配置为允许安全性时，AHB 接口支持安全访问
hclk	I	AHB 总线时钟和 EXTI 系统时钟
可配置事件 (y)	I	来自没有相关中断和标志的外设的异步唤醒事件
直接事件 (x)	I	来自具有相关中断和标志的外设的同步与异步唤醒事件
IOPort(n)	I	GPIO 端口 [15:0]
exti[15:0]	O	用于触发其他 IP 的 EXTI 输出端口
it_exti_per (y)	O	与可配置事件 (y) 关联的 CPU 的中断
c_evt_exti	O	CPU 的高电平触发事件输出，与 hclk 同步
c_evt_rst	I	用于清除 c_evt_exti 的异步复位输入
sys_wakeup	O	面向 ck_sys 和 hclk 的 PWR 的异步系统唤醒请求
c_wakeup	O	面向 CPU 的 PWR 的唤醒请求，与 hclk 同步

表 42. EVG 引脚概述

引脚名称	I/O	说明
c_fclk	I	CPU 自由运行时钟
c_evt_in	I	来自 EXTI 的高电平触发事件输入，与 CPU 时钟异步
c_event	O	事件脉冲，与 CPU 时钟同步
c_evt_rst	O	事件复位信号，与 CPU 时钟同步

12.2.1 外设和 CPU 之间的 EXTI 连接

对于能够在系统处于停止模式时生成唤醒或中断事件的外设，将连接至 EXTI。

- 对于生成脉冲或在外设中没有中断状态位的外设唤醒信号，将连接至 EXTI 可配置线。对于这类事件，EXTI 提供的状态挂起位需清零。与状态位相关的 EXTI 中断会中断 CPU。
- 对于在外设中有状态位（需要在外设中清零）的外设中断和唤醒信号，将连接至 EXTI 直接线。EXTI 中没有状态挂起位。中断或唤醒由外设中的 CPU 清除。外设中断会直接中断 CPU。
- EXTI 复用器的所有 GPIO 端口输入，可以选择端口以便通过可配置的事件来唤醒系统。

EXTI 可配置事件中断连接到 CPU 的 NVIC(a)。

专用 EXTI/EVG CPU 事件连接至 CPU rxev 输入。

EXTI CPU 唤醒信号连接至 PWR 模块，用于唤醒系统和 CPU 子系统总线时钟。

12.3 EXTI 功能描述

根据 EXTI 线类型和唤醒目标，将使用不同的逻辑实现。适用的功能以及相关的控制或状态寄存器如下：

- 上升沿和下降沿事件使能，具体包括
  - *EXTI 上升沿触发选择寄存器 (EXTI_RTSR1)*
  - *EXTI 下降沿触发选择寄存器 1 (EXTI_FTSR1)*
- 软件触发，具体包括 *EXTI 软件中断事件寄存器 1 (EXTI_SWIER1)*
- 挂起中断标志，具体包括
  - *EXTI 上升沿挂起寄存器 1 (EXTI_RPR1)*
  - *EXTI 下降沿挂起寄存器 1 (EXTI_FPR1)*
  - *EXTI 外部中断选择寄存器 (EXTI_EXTICRx)*
- CPU 唤醒和中断使能，具体包括
  - *EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR1)*
- CPU 唤醒和事件使能，具体包括
  - *EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR1)*

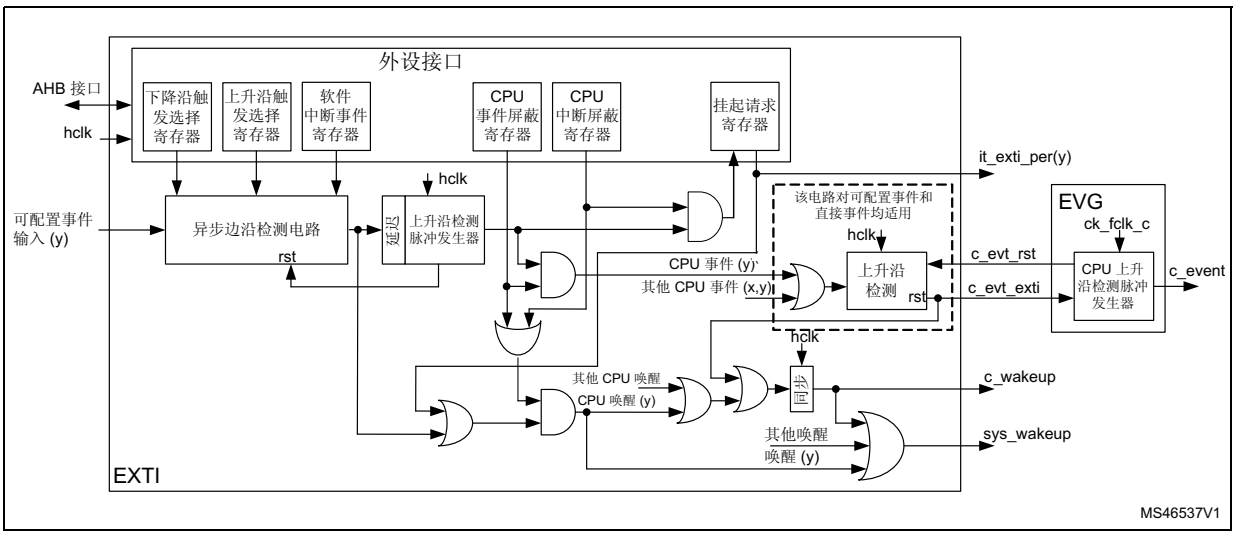
表 43. EXTI 事件输入配置和寄存器控制

事件输入类型	逻辑实现	EXTI_RTSR1	EXTI_FTSR1	EXTI_SWIER1	EXTI_R/FPR1	EXTI_IMR1	EXTI_EMR1
可配置	可配置事件输入唤醒逻辑	x	x	x	x	x	x
直接	直接事件输入唤醒逻辑	-	-	-	-	x	x

12.3.1 EXTI 可配置事件输入唤醒

图 24 所示为具有如下用途的可配置事件输入的相关逻辑详图：唤醒 CPU 子系统总线时钟并生成 CPU 的 EXTI 挂起标志与中断和 / 或 CPU 唤醒事件。

图 24. 可配置事件触发逻辑 CPU 唤醒



软件中断事件寄存器允许通过软件写入相应的寄存器位来触发可配置事件，与边沿选择设置无关。

上升沿和下降沿选择寄存器允许使能和选择可配置事件有效触发边沿或两种边沿触发都选。

CPU 具有专用的中断屏蔽寄存器和专用的事件屏蔽寄存器。使能的事件允许在 CPU 上生成一个事件。CPU 的所有事件会一起进行逻辑或运算，生成单个 CPU 事件信号。对于未屏蔽的 CPU 事件，事件挂起寄存器（EXTI_RPR1 和 EXTI_FPR1）不会置 1。

可配置事件具有唯一的中断挂起请求寄存器（由 CPU 使用）。挂起寄存器只会针对未屏蔽的中断而置位。每个可配置事件均提供对 CPU 的中断。可配置事件中断需要由软件通过 EXTI_RPR1 和 / 或 EXTI_FPR1 寄存器确认。

使能 CPU 中断或 CPU 事件时，异步边沿检测电路由时钟延迟和上升沿检测脉冲发生器复位。这可以确保 EXTI hclk 时钟在异步边沿检测电路复位之前唤醒。

注意： *检测到的可配置事件中断挂起请求可由 CPU 清除。只要中断挂起请求处于活动状态，系统就无法进入低功耗模式。*

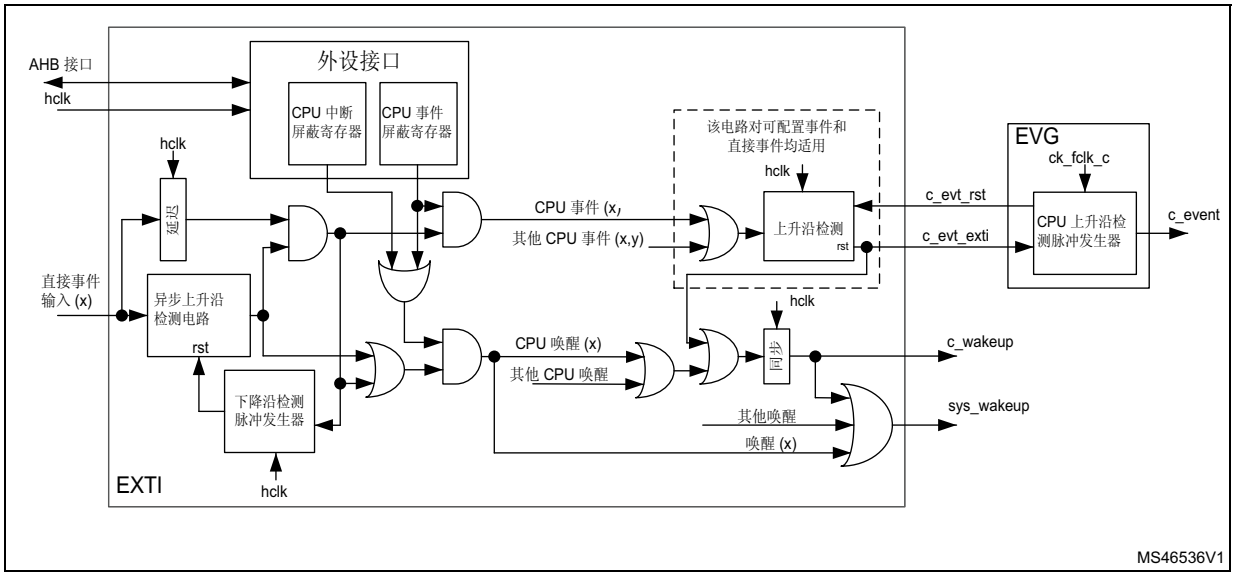
12.3.2 EXTI 直接事件输入唤醒

图 25 所示为唤醒系统的直接事件输入的相关逻辑详图。

直接事件没有关联的 EXTI 中断。EXTI 仅唤醒系统和 CPU 子系统时钟，并可能生成 CPU 唤醒事件。与直接唤醒事件相关的外设同步中断会唤醒 CPU。

EXTI 直接事件能够生成 CPU 事件。此 CPU 事件会唤醒 CPU。该 CPU 事件可以在相关外设的中断标志置 1 之前发生。

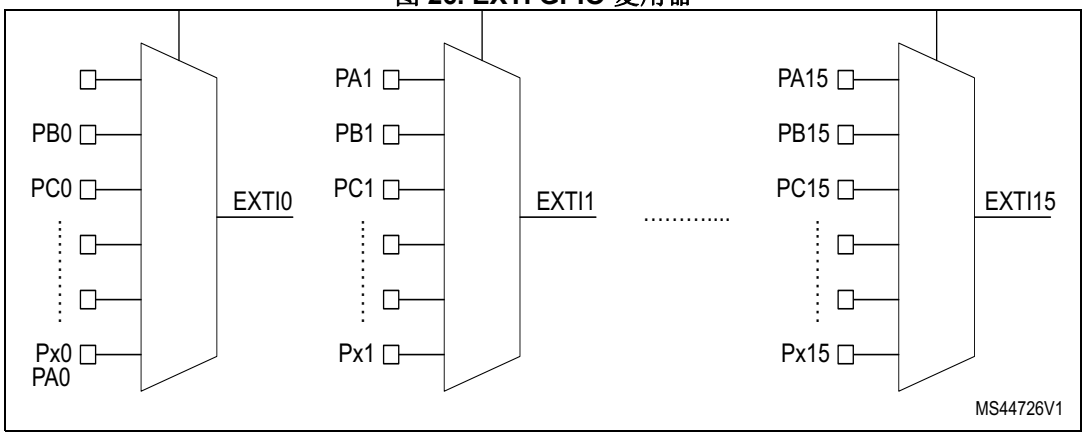
图 25. 直接事件触发逻辑 CPU 唤醒



12.3.3 EXTI 复用器

EXTI 复用器可以选择将 GPIO 作为中断和唤醒。各 GPIO 通过 16 根 EXTI 复用器线连接到前 16 个 EXTI 事件（均为可配置事件）。通过 *EXTI 外部中断选择寄存器 (EXTI_EXTICRx)* 寄存器来选择作为 EXTI 复用器输出的具体 GPIO 端口。

图 26. EXTI GPIO 复用器



EXTI 复用器输出可作为 EXTI 的输出信号，用于触发其他功能模块。EXTI 复用器输出通过 EXTI_IMR 和 EXTI_EMR 寄存器提供，与屏蔽设置无关。

EXTI 线（事件输入）的连接如下表所示。

表 44. EXTI 线连接

EXTI 线	线源	线类型
0-15	GPIO	可配置
16	保留	-
17	保留	-
18	保留	-
19	RTC	直接
20	保留	-
21	保留	-
22	保留	-
23	I2C1 唤醒	直接
24	保留	-
25	USART1 唤醒	直接
26	保留	-
27	保留	-
28	保留	-
29	保留	-
30	保留	-
31	LSE_CSS	直接
32	保留	-
33	保留	-

12.4 EXTI 功能行为

在生成唤醒事件的相应外设中使能直接事件输入。通过使能至少一个触发沿来使能可配置事件。

使能事件输入后，是否生成 CPU 唤醒取决于 CPU 中断屏蔽和 CPU 事件屏蔽。

表 45. 屏蔽功能

CPU 中断使能 EXTI_IMR.IMn	CPU 事件使能 EXTI_EMR.EMn	可配置事件输入 EXTI_RPR.RPIFn EXTI_FPR.FPIFn	exti(n) 中断(1)	CPU 事件	CPU 唤醒
0	0	无	已屏蔽	已屏蔽	已屏蔽
	1	无	已屏蔽	有	有
1	0	状态已锁存	有	已屏蔽	有(2)
	1	状态已锁存	有	有	有

1. 单个 exti(n) 中断将面向 CPU。如果 CPU 不需要中断，则必须在 CPU NVIC 中屏蔽 exti(n) 中断。
2. 仅限 EXTI_IMR.IMn 中已使能 CPU 中断的情况。

对于可配置事件输入，当事件输入上出现使能的边沿（上升沿和/或下降沿）后，会生成一个事件请求。当相关 CPU 中断未被屏蔽时，EXTI_RPR 或 / 和 EXTI_FPR 寄存器中相应的 RPIFn 和 / 或 FPIFn 位将置 1，从而唤醒 CPU 子系统并激活 CPU 中断信号。RPIFn 和 / 或 FPIFn 挂起位通过写 1 来清零，与此同时也会清除 CPU 中断请求。

对于直接事件输入，当在相关外设中使能时，只会在上升沿生成事件请求。EXTI 中没有相应的 CPU 挂起位。当相关 CPU 中断未被屏蔽时，会唤醒相应的 CPU 子系统。CPU 由外设同步中断唤醒（中断）。

要生成事件，CPU 事件必须处于未屏蔽状态。当事件输入上出现使能的边沿后，会生成 CPU 事件脉冲。这时没有事件挂起位。

对于可配置事件输入，可通过软件将软件中断/事件寄存器 EXTI_SWIER1 的相应位置 1（效果与事件输入上出现上升沿相同）来生成事件请求。此时，EXTI_RPR1 寄存器中的挂起上升沿事件标志将置 1，与 EXTI_RTSR1 寄存器设置无关。

12.5 EXTI 寄存器

EXTI 寄存器映射分为以下几个部分：

表 46. EXTI 寄存器映射的各个部分

地址	说明
0x000 - 0x01C	通用可配置事件 [31:0] 配置
0x060 - 0x06C	EXTI I/O 端口复用器
0x080 - 0x0BC	CPU 输入事件配置

所有寄存器均可支持按字（32 位）、半字（16 位）或字节（8 位）访问。

12.5.1 EXTI 上升沿触发选择寄存器 (EXTI_RTSR1)

EXTI rising trigger selection register

偏移地址：0x000

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **RTx**：可配置线 x 的上升沿触发事件配置位 (Rising trigger event configuration bit of configurable line x)（x = 15 到 0）(1)

每个位会针对相应线上的事件和中断使能 / 禁止上升沿触发。

0：禁止

1：使能

1. 可配置线为边沿触发，在这些输入上不能出现毛刺信号。
如果在对寄存器执行写操作时可配置线上出现上升沿，则相关挂起位不会被置 1。
对于已使能下降沿触发的线，依然可以设置上升沿触发。在这种情况下，两种边沿均会生成触发信号。

12.5.2 EXTI 下降沿触发选择寄存器 1 (EXTI_FTSR1)

EXTI falling trigger selection register 1

偏移地址：0x004

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **FTx**：可配置线 x 的下降沿触发事件配置位 (Falling trigger event configuration bit of configurable line x)（x = 15 到 0）(1)

每个位会针对相应线上的事件和中断使能 / 禁止下降沿触发。

0：禁止

1：使能

1. 可配置线为边沿触发，在这些输入上不能出现毛刺信号。
如果在对寄存器执行写操作时可配置线上出现下降沿，则相关挂起位不会被置 1。
对于已使能上升沿触发的线，依然可以设置下降沿触发。在这种情况下，两种边沿均会生成触发信号。

12.5.3 EXTI 软件中断事件寄存器 1 (EXTI_SWIER1)

EXTI software interrupt event register 1

偏移地址：0x008

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWI 15	SWI 14	SWI 13	SWI 12	SWI 11	SWI 10	SWI9	SWI8	SWI7	SWI6	SWI5	SWI4	SWI3	SWI2	SWI1	SWI0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **SWIx**：线 x 上的软件上升沿事件触发 (Software rising edge event trigger on line x)（x = 15 到 0）

通过软件将任何位置 1 都会在相应的线 x 上触发上升沿事件，与 EXTI_RTSR1 和 EXTI_FTSR1 的设置无关。这些位由硬件自动清零。读取任何位都将始终返回 0。

0：无影响

1：在相应线上生成上升沿事件，然后生成中断

12.5.4 EXTI 上升沿挂起寄存器 1 (EXTI_RPR1)

EXTI rising edge pending register 1

偏移地址：0x00C

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPIF15	RPIF14	RPIF13	RPIF12	RPIF11	RPIF10	RPIF9	RPIF8	RPIF7	RPIF6	RPIF5	RPIF4	RPIF3	RPIF2	RPIF1	RPIF0
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

位 31:16 保留，必须保持复位值。

位 15:0 **RPIFx**：可配置线 x 的上升沿事件挂起 (Rising edge event pending for configurable line x)（x = 15 到 0）

每个位都会在相应线上出现由硬件或软件（通过 EXTI_SWIER1 寄存器）生成的上升沿事件后置 1。每个位都通过写 1 来清零。

0：未发生上升沿触发请求

1：发生了上升沿触发请求

12.5.5 EXTI 下降沿挂起寄存器 1 (EXTI_FPR1)

EXTI falling edge pending register 1

偏移地址：0x010

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FPIF15	FPIF14	FPIF13	FPIF12	FPIF11	FPIF10	FPIF9	FPIF8	FPIF7	FPIF6	FPIF5	FPIF4	FPIF3	FPIF2	FPIF1	FPIF0
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

位 31:16 保留，必须保持复位值。

位 15:0 **FPIFx**：可配置线 x 的下降沿事件挂起 (Falling edge event pending for configurable line x)（x = 15 到 0）

每个位都会在相应线上出现由硬件或软件（通过 EXTI_SWIER1 寄存器）生成的下降沿事件后置 1。每个位都通过写 1 来清零。

0：未发生下降沿触发请求

1：发生了下降沿触发请求

12.5.6 EXTI 外部中断选择寄存器 (EXTI_EXTICRx)

EXTI external interrupt selection register

偏移地址：0x060 + 0x4 * (x - 1)（x = 1 到 4）

复位值：0x0000 0000

EXTIm 位域包含的位数与 nb_ioport 配置的位数完全一致。

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

位 31:24 **EXTIm+3[7:0]**：EXTIm+3 GPIO 端口选择 (EXTIm+3 GPIO port selection) (m = 4 * (x - 1))

这些位通过软件写入，用于选择 EXTIm+3 外部中断的源输入。

0x00：PA[m+3] 引脚

0x01：PB[m+3] 引脚

0x02：PC[m+3] 引脚

0x03：PD[m+3] 引脚

0x04：保留

0x05：PF[m+3] 引脚

其他值：保留

位 23:16 **EXTIm+2[7:0]**：EXTIm+2 GPIO 端口选择 (EXTIm+2 GPIO port selection) (m = 4 * (x - 1))

这些位通过软件写入，用于选择 EXTIm+2 外部中断的源输入。

0x00：PA[m+2] 引脚

0x01：PB[m+2] 引脚

0x02：PC[m+2] 引脚

0x03：PD[m+2] 引脚

0x04：保留

0x05：PF[m+2] 引脚

其他值：保留

位 15:8 **EXTIm+1[7:0]**：EXTIm+1 GPIO 端口选择 (EXTIm+1 GPIO port selection) (m = 4 * (x - 1))

这些位通过软件写入，用于选择 EXTIm+1 外部中断的源输入。

0x00：PA[m+1] 引脚

0x01：PB[m+1] 引脚

0x02：PC[m+1] 引脚

0x03：PD[m+1] 引脚

0x04：保留

0x05：PF[m+1] 引脚

其他值：保留

位 7:0 **EXTIm[7:0]**：EXTIm GPIO 端口选择 (EXTIm GPIO port selection) (m = 4 * (x - 1))
这些位通过软件写入，用于选择 EXTIm 外部中断的源输入。
0x00：PA[m] 引脚
0x01：PB[m] 引脚
0x02：PC[m] 引脚
0x03：PD[m] 引脚
0x04：保留
0x05：PF[m] 引脚
其他值：保留

12.5.7 EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR1)

EXTI CPU wakeup with interrupt mask register

偏移地址：0x080 (EXTI_IMR1)

复位值：0xFFF8 0000

包含可配置事件和直接事件的寄存器位。

复位值的设置如下：默认使能直接线的中断并禁止可配置线的中断。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IM31	Res.	Res.	Res.	Res.	Res.	IM25	Res.	IM23	Res.	Res.	Res.	IM19	Res.	Res.	Res.
rw						rw		rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **IM31**：线 31 上的 CPU 唤醒中断屏蔽 (CPU wakeup with interrupt mask on line 31)
将该位置 1/ 清零会取消屏蔽 / 屏蔽相应线上的 CPU 唤醒中断（通过事件实现）。
0：屏蔽唤醒中断
1：取消屏蔽唤醒中断

位 30:26 保留，必须保持复位值。

位 25 **IM25**：线 25 上的 CPU 唤醒中断屏蔽 (CPU wakeup with interrupt mask on line 25)
将每个位置 1/ 清零会取消屏蔽 / 屏蔽相应线上的 CPU 唤醒中断（通过事件实现）。
0：屏蔽唤醒中断
1：取消屏蔽唤醒中断

位 24 保留，必须保持复位值。

位 23 **IM23**：线 23 上的 CPU 唤醒中断屏蔽 (CPU wakeup with interrupt mask on line 23)
将每个位置 1/ 清零会取消屏蔽 / 屏蔽相应线上的 CPU 唤醒中断（通过事件实现）。
0：屏蔽唤醒中断
1：取消屏蔽唤醒中断

位 22:20 保留，必须保持复位值。

位 19 **IM19**：线 19 上的 CPU 唤醒中断屏蔽 (CPU wakeup with interrupt mask on line 19)
将该位置 1/ 清零会取消屏蔽 / 屏蔽相应线上的 CPU 唤醒中断（通过事件实现）。
0：屏蔽唤醒中断
1：取消屏蔽唤醒中断

位 18:16 保留，必须保持复位值。

位 15:0 **IMx**：线 x 上的 CPU 唤醒中断屏蔽 (CPU wakeup with interrupt mask on line x)（x = 15 到 0）
将每个位置 1/ 清零会取消屏蔽 / 屏蔽相应线上的 CPU 唤醒中断（通过事件实现）。
0：屏蔽唤醒中断
1：取消屏蔽唤醒中断

12.5.8 EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR1)

EXTI CPU wakeup with event mask register

偏移地址：0x084 (EXTI_EMR1)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EM31	Res.	Res.	Res.	Res.	Res.	EM25	Res.	EM23	Res.	Res.	Res.	EM19	Res.	Res.	Res.
rw						rw		rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EM15	EM14	EM13	EM12	EM11	EM10	EM9	EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **EM31**：线 31 上的 CPU 唤醒事件生成屏蔽 (CPU wakeup with event generation mask on line 31)
将该位置 1/ 清零会取消屏蔽 / 屏蔽相应线上的 CPU 唤醒事件生成。
0：屏蔽唤醒事件生成
1：取消屏蔽唤醒事件生成

位 30:26 保留，必须保持复位值。

位 25 **EM25**：线 25 上的 CPU 唤醒事件生成屏蔽 (CPU wakeup with event generation mask on line 25)
将该位置 1/ 清零会取消屏蔽 / 屏蔽相应线上的 CPU 唤醒事件生成。
0：屏蔽唤醒事件生成
1：取消屏蔽唤醒事件生成

位 24 保留，必须保持复位值。

位 23 **EM23**：线 23 上的 CPU 唤醒事件生成屏蔽 (CPU wakeup with event generation mask on line 23)
将该位置 1/ 清零会取消屏蔽 / 屏蔽相应线上的 CPU 唤醒事件生成。
0：屏蔽唤醒事件生成
1：取消屏蔽唤醒事件生成

位 22:20 保留，必须保持复位值。

位 19 **EM19**：线 19 上的 CPU 唤醒事件生成屏蔽 (CPU wakeup with event generation mask on line 19)
将该位置 1/ 清零会取消屏蔽 / 屏蔽相应线上的 CPU 唤醒事件生成。
0：屏蔽唤醒事件生成
1：取消屏蔽唤醒事件生成

位 18:16 保留，必须保持复位值。

位 15:0 **EMx**：线 x 上的 CPU 唤醒事件生成屏蔽 (CPU wakeup with event generation mask on line x)（x = 15 到 0）
将每个位置 1/ 清零会取消屏蔽 / 屏蔽相应线上的 CPU 唤醒事件生成。
0：屏蔽唤醒事件生成
1：取消屏蔽唤醒事件生成

12.5.9 EXTI 寄存器映射

下表列出了 EXTI 寄存器映射和复位值。

表 47. EXTI 控制器寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**EXTI_RTSR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	**EXTI_FTSR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x008	**EXTI_SWIER1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWI[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	**EXTI_RPR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RPIF[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	**EXTI_FPR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FPIF[16:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014-0x05C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x060	**EXTI_EXTICR1**	EXTI3[7:0]								EXTI2[7:0]								EXTI1[7:0]								EXTI0[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x064	**EXTI_EXTICR2**	EXTI7[7:0]								EXTI6[7:0]								EXTI5[7:0]								EXTI4[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x068	**EXTI_EXTICR3**	EXTI11[7:0]								EXTI10[7:0]								EXTI9[7:0]								EXTI8[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x06C	**EXTI_EXTICR4**	EXTI15[7:0]								EXTI14[7:0]								EXTI13[7:0]								EXTI12[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x080	**EXTI_IMR1**	IM31	Res.	Res.	Res.	Res.	Res.	IM25	Res.	IM23	Res.	Res.	Res.	IM19	Res.	Res.	Res.	IM[15:0]															
	Reset value	1						1		1				1				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	**EXTI_EMR1**	EM31	Res.	Res.	Res.	Res.	Res.	EM25	Res.	EM23	Res.	Res.	Res.	EM19	Res.	Res.	Res.	EM[15:0]															
	Reset value	0						0		0				0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x088-0x08C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

13 循环冗余校验计算单元 (CRC)

13.1 简介

CRC（循环冗余校验）计算单元使用多项式发生器从一个 8 位 /16 位 /32 位的数据字中产生 CRC 码。

在众多的应用中，基于 CRC 的技术还常用来验证数据传输或存储的完整性。根据功能安全标准的规定，这些技术提供了验证 Flash 完整性的方法。CRC 计算单元有助于在运行期间计算软件的签名，并将该签名与链接时生成并存储在指定存储单元的参考签名加以比较。

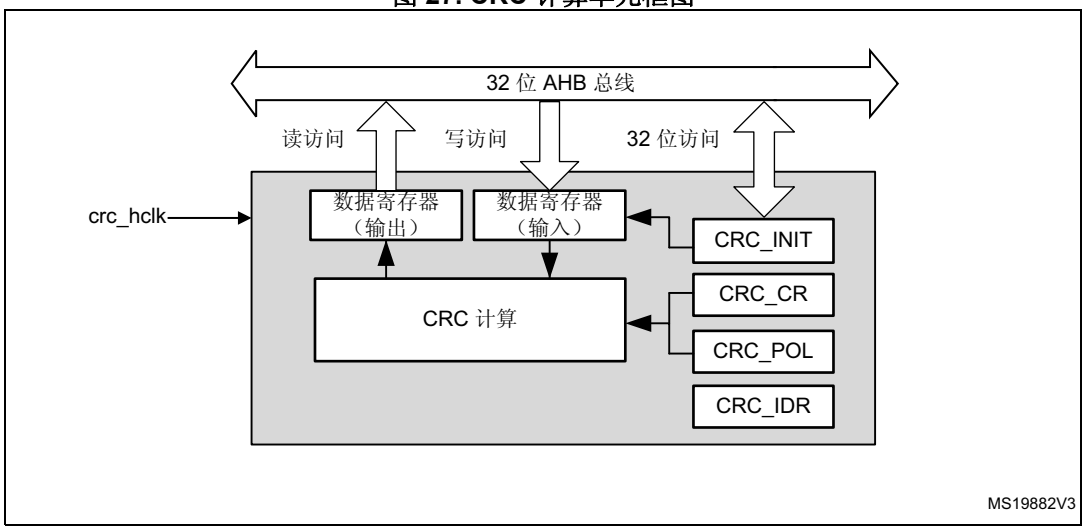
13.2 CRC 主要特性

- 使用 CRC-32（以太网）多项式：0x4C11DB7
  $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 或者，使用位数可编程的（7 位、8 位、16 位和 32 位）的完全可编程多项式
- 处理 8 位、16 位、32 位数据大小
- 可编程 CRC 初始值
- 单输入 / 输出 32 位数据寄存器
- 输入缓冲器可避免计算期间发生总线阻塞
- 对于 32 位数据大小，CRC 计算在 4 个 AHB 时钟周期 (HCLK) 内完成
- 8 位通用寄存器（可用于临时存储）
- I/O 数据有可逆性选项
- 仅通过 AHB 从外设按 32 位字进行访问（例外情况：CRC_DR 寄存器可按字、右对齐半字和右对齐字节进行访问）

13.3 CRC 功能描述

13.3.1 CRC 框图

图 27. CRC 计算单元框图



13.3.2 CRC 内部信号

表 48. CRC 内部输入 / 输出信号

信号名称	信号类型	说明
crc_hclk	数字输入	AHB 时钟

13.3.3 CRC 运算

CRC 计算单元具有单个 32 位读 / 写数据寄存器 (CRC_DR)。它用于输入新数据（写访问）和保存之前 CRC 计算的结果（读访问）。

对数据寄存器的每个写操作都会对之前的 CRC 值（存储在 CRC_DR 中）和新值再做一次 CRC 计算。CRC 计算针对整个 32 位数据字或逐个字节完成，具体取决于数据的写入格式。

CRC_DR 寄存器可按字、右对齐半字和右对齐字节进行访问。对于其他寄存器，只允许按 32 位进行访问。

计算时间取决于数据宽度：

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

输入缓冲器中可立即写入第二个数据，无需因之前的 CRC 计算而等待任何等待周期。

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

为管理各种数据存放方式（大端/小端）输入数据的顺序可以反转。可对 8 位、16 位和 32 位数据执行反转操作，具体取决于 CRC_CR 寄存器中的 REV_IN[1:0] 位。

例如，0x1A2B3C4D 输入数据在 CRC 计算中用作：

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

通过将 CRC_CR 寄存器中 REV_OUT 位置 1 也可以将输出数据反转。

该操作按位进行。例如，输出数据 0x11223344 将转换为 0x22CC4488。

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

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

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

多项式可编程

多项式系数可完全通过 CRC_POL 寄存器进行编程，通过编程 CRC_CR 寄存器中的 POLYSIZE[1:0] 位可将多项式大小配置为 7 位、8 位、16 位或 32 位。不支持偶数多项式。

注意： *偶数多项式的形式为 $X+X^2+..+X^n$，而奇数多项式的形式为 $1+X+X^2+..+X^n$。*

如果 CRC 数据小于 32 位，可从 CRC_DR 寄存器的最低有效位读取 CRC 的值。

为实现可靠的 CRC 计算，CRC 计算期间不能实时更改多项式的值或大小。因此，如果正在执行 CRC 计算，则在更改多项式前，应用程序必须先复位，或者先执行 CRC_DR 读操作。

默认的多项式值为 CRC-32（以太网）多项式：0x4C11DB7。

13.4 CRC 寄存器

CRC_DR 寄存器可按字、右对齐半字和右对齐字节进行访问。对于其他寄存器，只允许按 32 位访问。

13.4.1 CRC 数据寄存器 (CRC_DR)

CRC data register

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[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DR[31:0]**：数据寄存器位

该寄存器用于向 CRC 计算器写入新数据。

读取寄存器时可读出之前的 CRC 计算结果。

如果数据大小小于 32 位，则最低有效位可用于写入 / 读取正确值。

13.4.2 CRC 独立数据寄存器 (CRC_IDR)

CRC independent data register

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
IDR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IDR[31:0]**：通用的 32 位数据寄存器位

这些位可用作四个字节的临时存储单元。

该寄存器不受 CRC_CR 寄存器中 RESET 位产生的 CRC 复位影响

13.4.3 CRC 控制寄存器 (CRC_CR)

CRC control register

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

位 31:8 保留，必须保持复位值。

位 7 **REV_OUT**：反转输出数据 (Reverse output data)

该位用于控制输出数据位顺序的反转。

0：不影响位顺序

1：位反转输出格式

位 6:5 **REV_IN[1:0]**：反转输入数据 (Reverse input data)

该位域用于控制输入数据位顺序的反转

00：不影响位顺序

01：按字节执行位反转

10：按半字执行位反转

11：按字执行位反转

位 4:3 **POLYSIZE[1:0]**：多项式大小 (Polynomial size)

这些位用于控制多项式的大小。

00：32 位多项式

01：16 位多项式

10：8 位多项式

11：7 位多项式

位 2:1 保留，必须保持复位值。

位 0 **RESET**：RESET 位

该位由软件置 1，用于复位 CRC 计算单元并将数据寄存器设置为存储在 CRC_INIT 寄存器中的值。该位只能置 1，将由硬件自动进行清零

13.4.4 CRC 初始值 (CRC_INIT)

CRC initial value

偏移地址：0x10

复位值：0xFFFF FFFF

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

位 31:0 **CRC_INIT[31:0]**：可编程 CRC 初始值 (Programmable initial CRC value)
该寄存器用于写入 CRC 初始值。

13.4.5 CRC 多项式 (CRC_POL)

CRC polynomial

偏移地址：0x14

复位值：0x04C1 1DB7

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

位 31:0 **POL[31:0]**：可编程多项式 (Programmable polynomial)
该寄存器用于写入要用于 CRC 计算的多项式系数。
如果多项式大小小于 32 位，则必须使用最低有效位编程正确值。

13.4.6 CRC 寄存器映射

表 49. CRC 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CRC_DR	DR[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x04	CRC_IDR	IDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	CRC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV_OUT	REV_IN[1:0]		POLYSIZE[1:0]		Res.	Res.	RESET
	Reset value																									0	0	0	0	0			0
0x10	CRC_INIT	CRC_INIT[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x14	CRC_POL	POL[31:0]																															
	Reset value	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	1	1	1	0	1	1	0	1	1	0	1	1	1

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

14 模数转换器 (ADC)

14.1 简介

12 位 ADC 是逐次趋近型模数转换器。它具有多达 23 个复用通道，可测量来自 19 个外部源和 4 个内部源的信号。各种不同通道的 A/D 转换可在单次、连续、扫描或不连续采样模式下进行。ADC 的结果存储在一个左对齐或右对齐的 16 位数据寄存器中。

ADC 具有模拟看门狗特性，允许应用检测输入电压是否超过了用户自定义的阈值上限或下限。

采用了高效的低功耗模式，在低频下可实现极低的功耗。

内置硬件过采样器，可提高模拟性能，同时还能减轻 CPU 进行相关计算的负担。

14.2 ADC 主要特性

- 高性能
  - 可配置 12 位、10 位、8 位或 6 位分辨率
  - ADC 转换时间：12 位分辨率对应的转换时间为 0.4 µs (2.5 Msps)，若降低分辨率，可进一步缩短转换时间。
  - 自校准
  - 可编程采样时间
  - 内置数据一致性，可确保数据对齐
  - 支持 DMA
- 低功耗
  - 应用可降低 PCLK 频率从而以低功耗运行，同时仍可保持最优的 ADC 性能。例如，无论 PCLK 的频率如何，都会保持 0.4 µs 的转换时间
  - 等待模式：防止 ADC 在低频 PCLK 应用中溢出
  - 自动关闭模式：ADC 会自动断电，但正在进行转换时除外。这可大幅降低 ADC 的功耗。
- 模拟输入通道
  - 最多 19 个外部模拟输入
  - 1 个用于内部温度传感器 (V_{SENSE}) 的通道
  - 1 个用于内部参考电压 (V_{REFINT}) 的通道
  - 2 个用于监视内部电源 (V_{DDA}/V_{SSA}) 的通道
- 可通过以下方式启动转换过程：
  - 通过软件
  - 通过极性可配置的硬件触发器（定时器事件或 GPIO 输入事件）
- 转换模式
  - 可转换单条通道，也可扫描一系列通道
  - 单次模式会在每次触发时对选定的输入执行一次转换
  - 连续模式可连续转换选定的输入
  - 不连续模式
- 在采样结束、转换结束、序列转换结束以及发生模拟看门狗或溢出事件时产生中断
- 模拟看门狗
- 过采样器
  - 16 位数据寄存器
  - 过采样率可在 2x 到 256x 之间进行调整
  - 可编程数据最多可移位 8 位
- ADC 输入范围：$V_{SSA} \leq V_{IN} \leq V_{DDA}$

14.3 ADC 实现

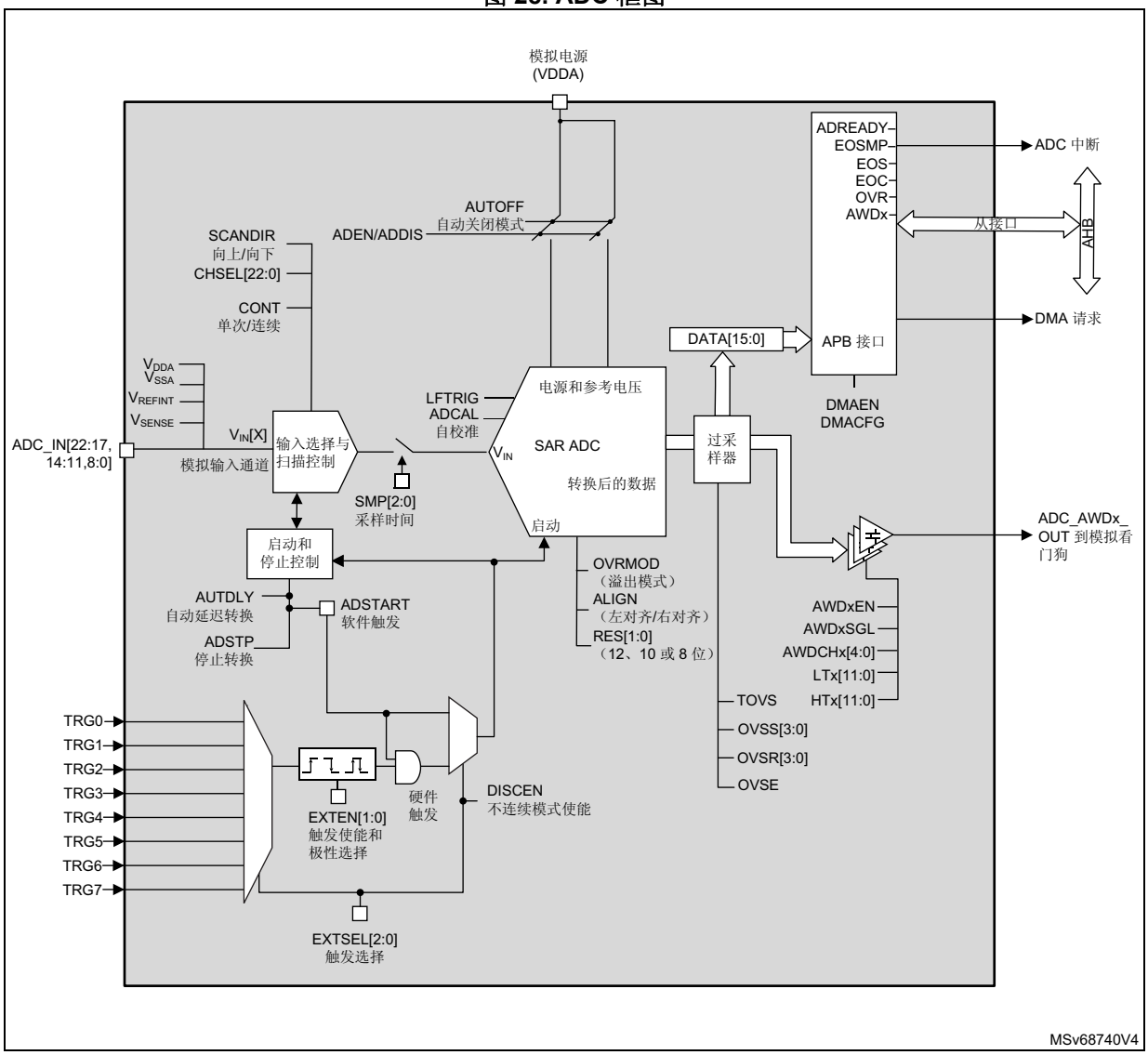
表 50. ADC 主要特性

ADC 模式 / 特性	STM32C01xx	STM32C03xx
分辨率	12 位	
最大采样速度	2.5 Msps	
硬件偏移校准	X	
单端输入	X	
差分输入	-	
过采样模式	X	
数据寄存器	16 位	
支持 DMA	X	
模拟看门狗数	3	
外部通道数	13	19
内部通道数	4	

14.4 ADC 功能描述

图 28 给出了 ADC 框图，*表 51* 列出了 ADC 的引脚说明。

图 28. ADC 框图



1. TRGi 映射到产品级。请参见*第 14.4.1 节：ADC 引脚和内部信号*中的*外部触发器表*。

14.4.1 ADC 引脚和内部信号

表 51. ADC 输入 / 输出引脚

名称	信号类型	备注
VDDA	模拟电源输入	ADC 的模拟电源和正参考电压
VSSA	模拟电源地输入	模拟电源地
ADC_INx	模拟输入信号	最多 19 个外部模拟输入通道

表 52. ADC 内部输入 / 输出信号

内部信号名称	信号类型	说明
$V_{IN}[x]$	模拟输入通道	连接到内部通道或 ADC_IN*i* 外部通道
TRGx	输入	ADC 转换触发信号
V_{SENSE}	输入	内部温度传感器输出电压
V_{REFINT}	输入	内部参考电压输出电压
ADC_AWDx_OUT	输出	内部模拟看门狗输出信号，连接到片上定时器（x = 模拟看门狗编号 = 1、2、3）。

表 53. 外部触发器

名称	源	EXTSEL[2:0]
TRG0	TIM1_TRGO2	000
TRG1	TIM1_CC4	001
TRG2	保留	010
TRG3	TIM3_TRGO	011
TRG4	保留	100
TRG5	保留	101
TRG6	保留	110
TRG7	EXTI11	111

14.4.2 ADC 调压器 (ADVREGEN)

ADC 配有特定的内部调压器，使用 ADC 之前，调压器必须使能并处于稳定状态。

可通过将 ADC_CR 寄存器中的 ADVREGEN 位置 1 来使能 ADC 内部调压器。软件必须等到 ADC 调压器的启动时间 ($t_{ADCVREG\_STUP}$) 过后，才能开始校准或使能 ADC。该延迟必须通过软件管理（有关 $t_{ADCVREG\_STUP}$ 的详细信息，请参见器件数据手册）。

ADC 操作完成后，禁止 ADC (ADEN = 0)。为了保持低功耗，务必在进入低功耗模式（LPRun、LPSleep 或停止模式）前禁止 ADC 调压器。（请参见 *ADC 调压器禁止序列*一节）。

注意： *在内部调压器禁止后，ADC 的模拟校准值依旧保持。*

电源控制单元提供的模拟参考

内部 ADC 调压器在内部使用的模拟参考是由电源控制单元通过缓冲器提供。当电源控制单元的主调压器处于正常运行模式时，该缓冲器始终使能（请参见复位和时钟控制部分以及电源控制部分）。

如果主调压器进入低功耗模式（例如低功耗运行模式），则会禁止该缓冲器并且无法使用 ADC。

ADC 调压器使能序列

要使能 ADC 调压器，需将 ADC_CR 寄存器中的 ADVREGEN 位置 1。

ADC 调压器禁止序列

要禁止 ADC 调压器，需执行以下序列：

1. 确保 ADC 已禁止 (ADEN = 0)。
2. 将 ADC_CR 寄存器中的 ADVREGEN 位清零。

14.4.3 校准 (ADCAL)

ADC 具有校准功能。校准过程中，ADC 会计算校准系数，ADC 下一次掉电之前，会在内部对 ADC 应用该校准系数。校准过程中，应用不得使用 ADC，必须等待校准完成。

校准应在启动 A/D 转换之前进行。校准可消除偏移误差，由于制造工艺的不同，各芯片的偏移误差也有所不同。

校准通过软件将 ADCAL 位置 1 发起。只有在满足以下所有条件时，才能发起校准：

- ADC 调压器已使能（ADVREGEN = 1 且 LDORDY = 1），
- ADC 已禁止 (ADEN = 0)，且
- 自动模式已禁止 (AUTOFF = 0)。

ADCAL 位在所有校准序列过程中保持为 1。校准完成后，该位会立即由硬件清零。随后，可从 ADC_DR 寄存器中读取校准系数（位 6 到位 0）。

如果禁止 ADC (ADEN = 0)，则会保留内部模拟校准。如果 ADC 工作条件发生变化（V_{DDA} 变化是造成 ADC 偏移变化的主要原因，温度变化影响较小），建议重新运行校准周期。

在下列情况下，校准系数会丢失：

- ADC 掉电（例如，产品进入待机 模式时）
- ADC 外设复位。

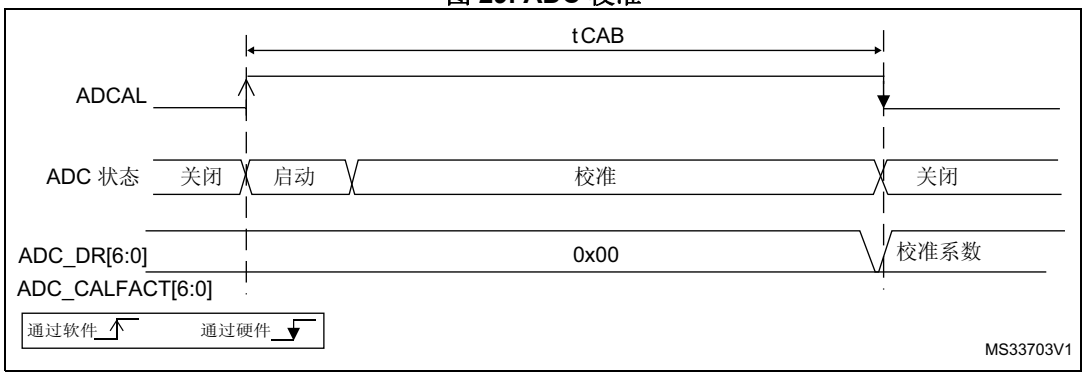
每次 ADC 掉电校准系数都会丢失（例如，产品进入待机 模式时）。仍可通过软件保存并恢复校准系数，以节省 ADC 重启时间（前提是 ADC 掉电期间的温度和电压保持稳定）。

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

软件校准流程

1. 确保 ADEN = 0、AUTOFF = 0、ADVREGEN = 1 且 DMAEN = 0。
2. 将 ADCAL 置 1。
3. 等待，直至 ADCAL = 0（或直至 EOCAL = 1）。如果已通过将 ADC_IER 寄存器中的 EOCALIE 位置 1 来使能中断，可通过中断进行处理。
4. 校准系数可从 ADC_DR 或 ADC_CALFACT 寄存器的位 6:0 读取。
5. 要降低校准系数提取的噪声效应，可通过软件求八个 CALFACT[6:0] 值（可选）的平均值。

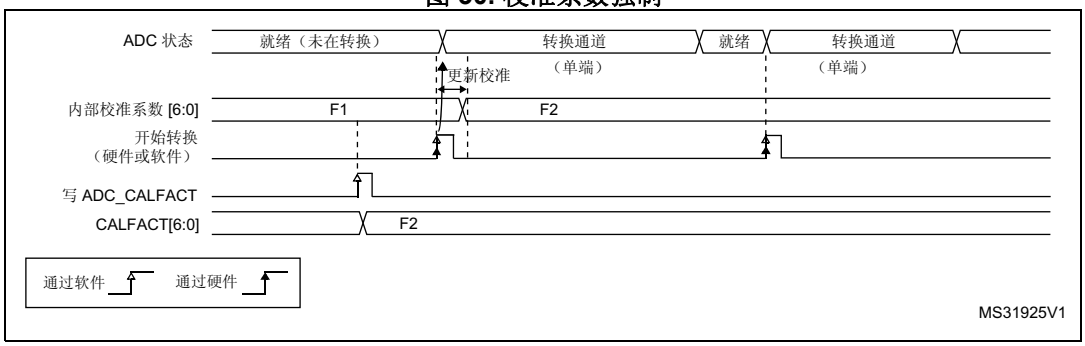
图 29. ADC 校准



校准系数强制软件程序

1. 确保 ADEN = 1 且 ADSTART = 0（ADC 启动时没有进行任何转换）
2. 将保存的校准系数写入 ADC_CALFACT
3. 启动新的转换后，将立即使用校准系数。

图 30. 校准系数强制



14.4.4 ADC 开关控制（ADEN、ADDIS、ADRDY）

上电时，会禁止 ADC 并进入掉电模式 (ADEN = 0)。

如*图 31* 所示，ADC 在开始精确转换之前需要一段稳定时间 t_{STAB}。

以下两个控制位可用于使能或禁止 ADC：

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

随后可通过将 ADSTART 置 1（请参见*第 232 页的第 14.5 节：外部触发转换和触发极性（EXTSEL 和 EXTEN）*）开始进行转换，如果使能了外部触发，则在外部事件发生时开始进行转换。

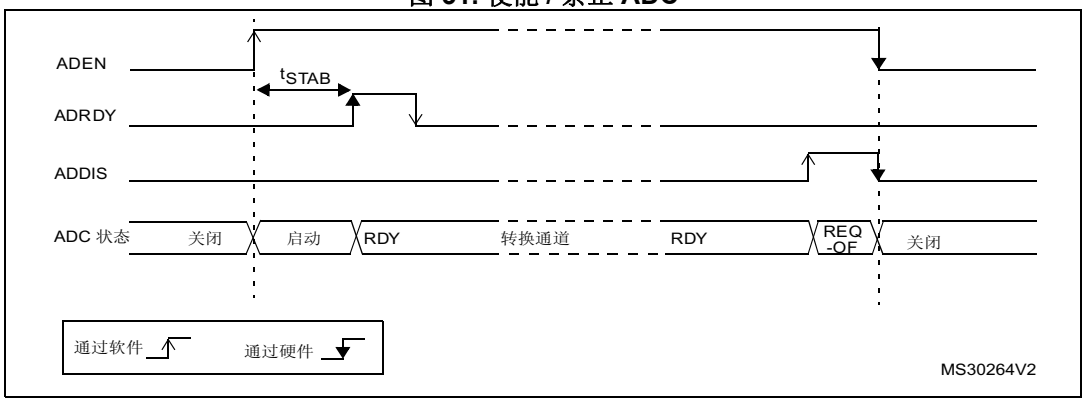
请按照以下流程使能 ADC：

1. 将 ADC_ISR 寄存器中的 ADRDY 位编程为 1，将该位清零。
2. 将 ADC_CR 寄存器中的 ADEN 位置 1。
3. 等待，直至 ADC_ISR 寄存器中的 ADRDY = 1（ADRDY 会在 ADC 启动时间后置 1）。如果已通过将 ADC_IER 寄存器中的 ADRDYIE 位置 1 来使能中断，可通过中断进行处理。

请按照以下流程禁止 ADC：

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

图 31. 使能 / 禁止 ADC



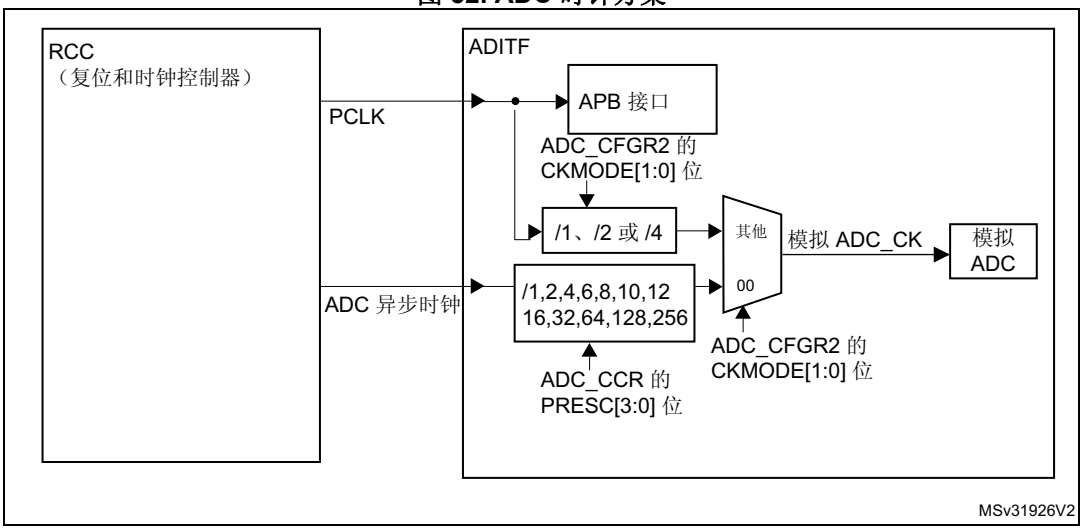
注意： *在自动关闭模式下 (AUTOFF = 1)，上电 / 掉电阶段是由硬件自动执行的，不会将 ADRDY 标志置 1。*

当总线时钟比模拟 ADC_CK 时钟快很多时，ADEN 和 ADDIS 位设置之间必须遵循十个模拟 ADC_CK 周期的最短延迟。

14.4.5 ADC 时钟（CKMODE、PRESC[3:0]）

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

图 32. ADC 时钟方案



1. 有关如何使能 PCLK 时钟和 ADC 异步时钟的信息，请参见*复位和时钟控制 (RCC) 部分*。

模拟 ADC 的输入时钟可在两个不同的时钟源之间进行选择（请参见*图 32：ADC 时钟方案*，了解 PCLK 时钟和 ADC 异步时钟的使能方式）：

a) ADC 时钟可以是名为“ADC 异步时钟”的特定时钟源，该时钟源独立于 APB 时钟，并与 APB 时钟异步。

有关生成该时钟源的更多信息，请参见 RCC 部分。

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

b) ADC 时钟可由 ADC 总线接口的 APB 时钟除以一个可编程系数（1、2 或 4，具体根据 CKMODE[1:0] 位而定）来提供。

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

在选项 a) 中，对 ADC_CCR 寄存器中的 PRESC[3:0] 位进行编程时，生成的 ADC 时钟最后会除以预分频系数（1、2、4、6、8、10、12、16、32、64、128、256）。

选项 a) 的优点在于无论选择哪种 APB 时钟图，都可以达到最大 ADC 时钟频率。

选项 b) 的优势在于绕过了时钟域重新同步。如果 ADC 由定时器触发，并且应用要求 ADC 精确触发（不存在任何不确定性），可使用该选项（否则，重新同步两个时钟域会为触发时刻带来不确定性）。

表 54. 触发与转换开始之间的延迟(1)

ADC 时钟源	CKMODE[1:0]	触发事件与 转换开始之间的延迟
SYSCLK 或 HSIKER 时钟(2)	00	延迟是不确定的（存在抖动）
PCLK 2 分频	01	延迟是确定的（无抖动），等于 3.25 个 ADC 时钟周期
PCLK 4 分频	10	延迟是确定的（无抖动），等于 3.125 个 ADC 时钟周期
PCLK 1 分频	11	延迟是确定的（无抖动），等于 3 个 ADC 时钟周期

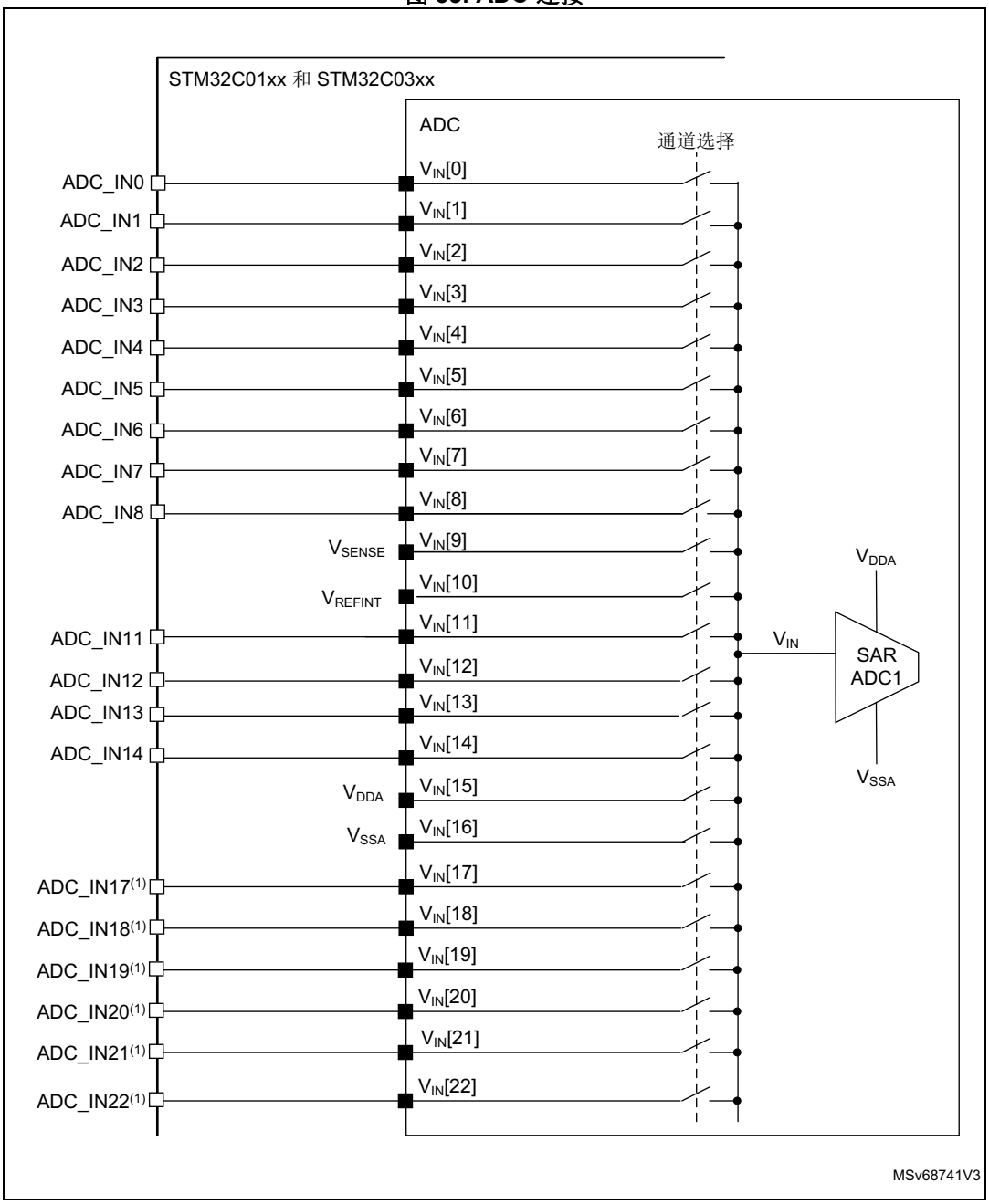
1. 有关最大 ADC_CLK 频率，请参见器件数据手册。
2. 通过 RCC_CCIPR 寄存器的 ADCSEL 位域选择。

小心： 选择 CKMODE[1:0] = 11（PCLK 1 分频）时，用户必须确保 PCLK 的占空比为 50%。实现方式是选择占空比为 50% 的系统时钟，并在 RCC 中将 APB 预分频器配置为旁路模式（请参见复位和时钟控制器部分）。如果选择内部源时钟，则 AHB 和 APB 预分频器不会对时钟进行分频。

14.4.6 ADC 连接

ADC 输入连接到外部通道和内部源，如*图 33* 所示。

图 33. ADC 连接



1. ADC_IN17 到 ADC_IN22 仅在 STM32C03xx 器件上可用。

14.4.7 配置 ADC

仅当 ADC 已禁止时（ADEN 必须清零），软件才能写入 ADC_CR 寄存器中的 ADCAL 和 ADEN 位并配置 ADC_CFGR1 和 ADC_CFGR2 寄存器。

只有在 ADC 已使能、并且没有待处理的禁止 ADC 的请求（ADEN = 1 且 ADDIS = 0）时，软件才能对 ADC_CR 寄存器中的 ADSTART 位和 ADDIS 位执行写操作。

对于 ADC_IER、ADC_SMPR、ADC_CHSELR 和 ADC_CCR 寄存器中的所有其他控制位，请参见*第 14.12 节：ADC 寄存器*中的相应控制位的说明。

在转换进行过程中，可以修改 ADC_AWDTRx 寄存器。

只有在 ADC 已使能、并且没有待处理的禁止 ADC 的请求（ADSTART = 1 且 ADDIS = 0）时，软件才能对 ADC_CR 寄存器中的 ADSTP 位执行写操作。

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

14.4.8 通道选择（CHSEL、SCANDIR、CHSELRMOD）

复用通道多达 23 个：

- 多达 19 路来自 GPIO 引脚 (ADC_INx) 的模拟输入
- 4 路内部模拟输入（温度传感器、内部参考电压、模拟电源和模拟地）

可转换单个通道，也可以转换一系列通道。

待转换通道的顺序可在 ADC_CHSELR 通道选择寄存器中进行编程：每个模拟输入通道都有专用的选择位 (CHSELx)。

ADC 扫描序列发生器可以在两种不同模式下使用：

- 序列发生器不可完全配置：

  通过通道编号定义通道的扫描顺序（必须将 ADC_CFGR1 寄存器中的 CHSELRMOD 位清零）：
  - 通过 ADC_CHSELR 寄存器中的 CHSELx 位配置序列长度
  - 序列方向：可按正向（从最低通道编号到最高通道编号）或反向（从最高通道编号到最低通道编号）扫描通道，具体取决于 SCANDIR 位的值（SCANDIR = 0：正向扫描； SCANDIR = 1：反向扫描）
  - 任何通道均可属于这些序列
- 序列发生器可完全配置

  ADC_CFGR1 寄存器中的 CHSELRMOD 位置 1。
  - 序列发生器长达 8 个通道
  - 通道的扫描顺序与通道编号无关。可通过 ADC_CHSELR 寄存器中的 SQ1[3:0] 到 SQ8[3:0] 位配置任何顺序。
  - 只能在该序列中选择通道 0 到通道 14
  - 如果序列发生器检测到 SQx[3:0] = 0b1111，将忽略之后的 SQx[3:0] 寄存器。
  - 如果未在 SQx[3:0] 中编程 0b1111，序列发生器将扫描全部八个通道。

编程 ADC CHSELR、SCANDIR 和 CHSELRMOD 位后，必须等待 CCRDY 标志出现后才能开始转换。该标志指示已应用新的通道设置。如果需要新的配置，必须将 CCRDY 标志清零才能开始转换。

仅当 ADSTART 位清零时（这可确保当前未进行任何转换），才允许通过软件对 CHSEL、SCANDIR 和 CHSELRMOD 位进行编程。

温度传感器、V_{REFINT} 以及 VDDA 和 VSSA 内部通道

温度传感器连接到通道 ADC V_{IN}[9]。

内部参考电压 V_{REFINT} 连接到通道 ADC V_{IN}[10]。

V_{DDA} 通道连接到 ADC V_{IN}[15]。

V_{SSA} 通道连接到 ADC V_{IN}[16]。

14.4.9 可编程采样时间 (SMPx[2:0])

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

使用可编程采样时间后，可根据输入电压源的输入电阻微调转换速度。

ADC 会在数个 ADC 时钟周期内对输入电压进行采样，时钟周期数可使用 ADC_SMPR 寄存器中的 SMP1[2:0] 和 SMP2[2:0] 位进行修改。

每个通道可通过 ADC_SMPR 寄存器中的 SMPSELx 位，从 SMP1[2:0] 和 SMP2[2:0] 位域配置的两个采样时间中选择一个。

总转换时间的计算公式如下：

t_{CONV} = 采样时间 + 12.5 x ADC 时钟周期

示例：

如果 ADC_CLK = 16 MHz，采样时间为 1.5 个 ADC 时钟周期：

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

ADC 通过将 EOSMP 标志置 1 来指示采样阶段结束。

14.4.10 单次转换模式 (CONT = 0)

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

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

在序列中，每次转换完成后：

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

转换序列完成后：

- EOS（序列结束）标志置 1
- EOSIE 位置 1 时将产生中断

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

注意： *要转换单个通道，可将序列长度编程为 1。*

14.4.11 连续转换模式 (CONT = 1)

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

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

在序列中，每次转换完成后：

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

转换序列完成后：

- EOS（序列结束）标志置 1
- EOSIE 位置 1 时将产生中断

随后，会立即重启新序列，ADC 会继续重复执行转换序列。

注意： *要转换单个通道，可将序列长度编程为 1。*

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

14.4.12 开始转换 (ADSTART)

软件通过将 ADSTART 置 1 的方式开始进行 ADC 转换。

ADSTART 置 1 后：

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

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

ADSTART 位由硬件清零：

- 在使用软件触发的单次模式下（CONT = 0，EXTEN = 00）
  - 只要转换序列结束 (EOS = 1) 就清零
- 在使用软件触发的不连续模式下（CONT = 0，DISCEN = 1，EXTEN = 00）
  - 转换结束时 (EOC = 1) 清零
- 在所有情况下（CONT = x，EXTEN = XX）
  - 执行由软件调用的 ADSTP 程序之后（参见*第 232 页的第 14.4.14 节：停止正在进行的转换 (ADSTP)*）清零

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

如果在单次模式下选择了硬件触发（CONT = 0 且 EXTEN = 01），则 EOS 标志置 1 时，ADSTART 不会由硬件清零（DMAEN = 1 且 DMACFG = 0 时除外，这种情况下 ADSTART 会在 DMA 传输结束时清零）。这样，便无需通过软件将 ADSTART 再次置 1，并可确保不会错过下一触发事件。

更改通道选择配置（通过编程 ADC_CHSELR 寄存器或更改 CHSELRMOD 或 SCANDIR）后，必须等待至 CCRDY 标志置为有效之后才能将 ADSTART 置为有效，否则会忽略写入 ADSTART 的值。

14.4.13 时序

从转换开始到转换结束所经过的时间是配置的采样时间与逐次逼近时间（具体取决于数据分辨率）的总和：

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

$t_{CONV} = t_{SMPL} + t_{SAR} = 42.9\ ns\ _{|min} + 357.1\ ns\ _{|12bit} = 0.400\ \mu s\ _{|min}$（对于 $f_{ADC\_CLK} = 35\ MHz$）

图 34. 模数转换时间

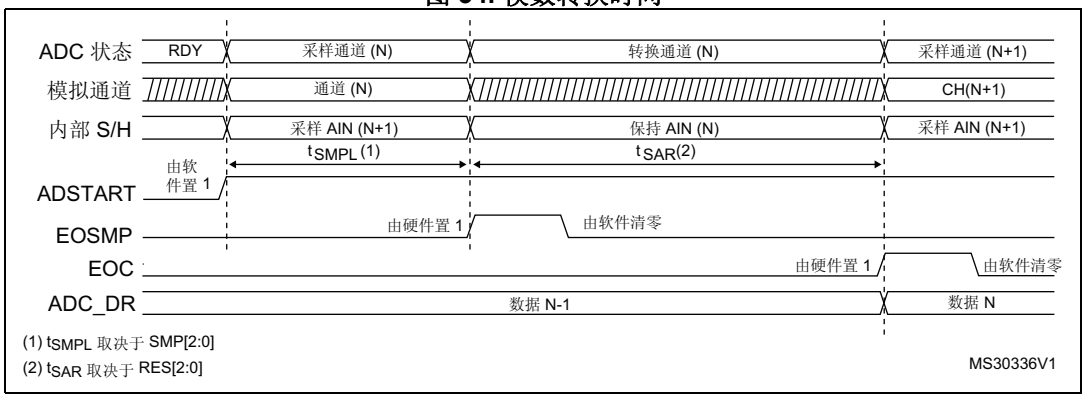
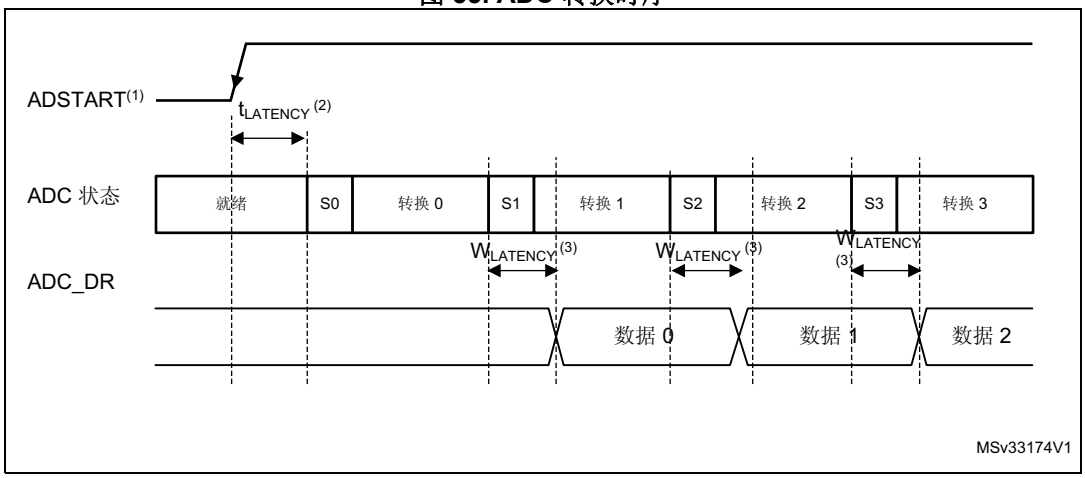


图 35. ADC 转换时序



1. EXTEN = 00 或 EXTEN ≠ 00
2. 触发延迟（更多详细信息，请参见数据手册）
3. ADC_DR 寄存器写入延迟（更多详细信息，请参见数据手册）

14.4.14 停止正在进行的转换 (ADSTP)

可通过软件将 ADC_CR 寄存器中的 ADSTP 置 1，停止任何正在进行的转换。

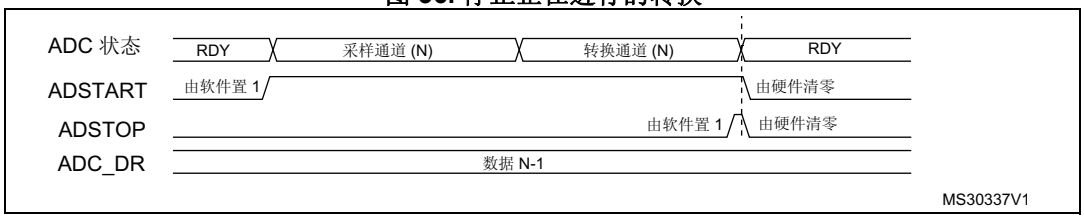
该操作会复位 ADC 操作，ADC 将处于空闲状态，准备好进行新操作。

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

扫描序列也会中止并会复位（这意味着重启 ADC 将重新开始新的序列）。

该程序完成后，ADSTP 位和 ADSTART 位均会由硬件清零，软件必须等待 ADSTART=0，然后才能开始进行新的转换。

图 36. 停止正在进行的转换



14.5 外部触发转换和触发极性（EXTSEL 和 EXTEN）

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

在转换进行时发生的硬件触发会被忽略。

如果 ADSTART 位 = 0，则会忽略发生的任何硬件触发。

表 55 提供了 EXTEN[1:0] 值与触发极性之间的对应关系。

表 55. 配置触发极性

源	EXTEN[1:0]
禁止触发检测	00
在上升沿检测	01
在下降沿检测	10
在上升沿和下降沿均检测	11

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

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

有关所有可用于进行常规转换的外部触发，请参见*第 14.4.1 节：ADC 引脚和内部信号*中的*表 53：外部触发器*。

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

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

14.5.1 不连续模式 (DISCEN)

可将 ADC_CFGR1 寄存器中的 DISCEN 位置 1 来使能该模式。

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

示例：

- DISCEN = 1，要转换的通道 = 0、3、7、10
  - 第一次触发：转换通道 0 并生成 EOC 事件
  - 第二次触发：转换通道 3 并生成 EOC 事件
  - 第三次触发：转换通道 7 并生成 EOC 事件
  - 第四次触发：转换通道 10 并同时生成 EOC 和 EOS 事件
  - 第五次触发：转换通道 0 并生成 EOC 事件
  - 第六次触发：转换通道 3 并生成 EOC 事件
  - ...
- DISCEN = 0，要转换的通道 = 0、3、7、10
  - 第一次触发：转换整个序列：通道 0、然后是通道 3、7 和 10。每次转换都会生成 EOC 事件，最后一次转换还会生成 EOS 事件。
  - 任何后续触发事件都将重启整个序列。

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

14.5.2 可编程分辨率 (RES)—— 快速转换模式

可通过降低 ADC 分辨率缩短转换时间 (t_{SAR})。

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

注意： *RES[1:0] 位必须在 ADEN 位复位后才能进行更改。*

转换结果的宽度始终为 12 位，任何未使用的 LSB 位都会读为零。

降低分辨率可缩短逐次趋近步骤所需的转换时间，如*表 56* 所示。

表 56. t_{SAR} 与分辨率的对应关系

RES[1:0] 位	t_{SAR} （ADC 时钟周期）	f_{ADC} = 35 MHz 时的 t_{SAR} (ns)	$t_{SMPL\ (min)}$ （ADC 时钟周期）	t_{CONV} （ADC 时钟周期）（具有最小 t_{SMPL}）	f_{ADC} = 35 MHz 时的 t_{CONV} (ns)
12	12.5	357	1.5	14	400
10	10.5	300	1.5	12	343
8	8.5	243	1.5	10	286
6	6.5	186	1.5	8	229

14.5.3 转换结束、采样阶段结束（EOC、EOSMP 标志）

ADC 指示每个转换结束 (EOC) 事件。

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

ADC 还通过将 ADC_ISR 寄存器中的 EOSMP 标志置 1 来指示采样阶段结束。EOSMP 标志可通过由软件向其写入 1 来清零。如果 ADC_IER 寄存器中的 EOSMPIE 位置 1，可产生中断。

该中断用于使处理与转换进行同步。通常情况下，可在转换阶段的隐藏时间中访问模拟复用器，这样在下次采样开始时便可放置好复用器。

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

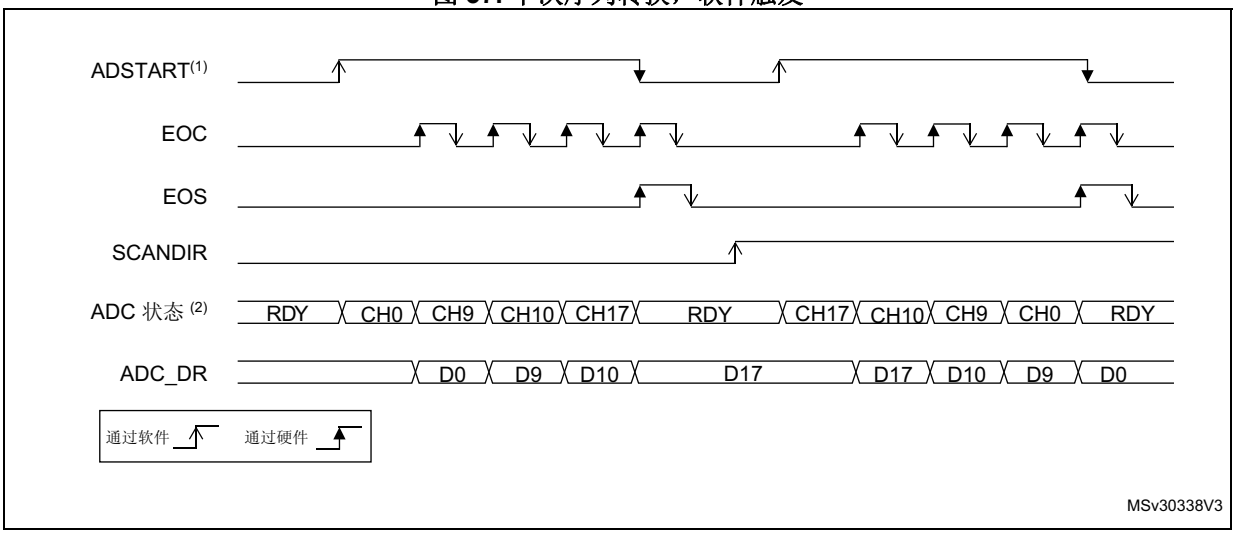
14.5.4 转换序列结束（EOS 标志）

每次序列 (EOS) 事件结束时，ADC 都会通知应用。

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

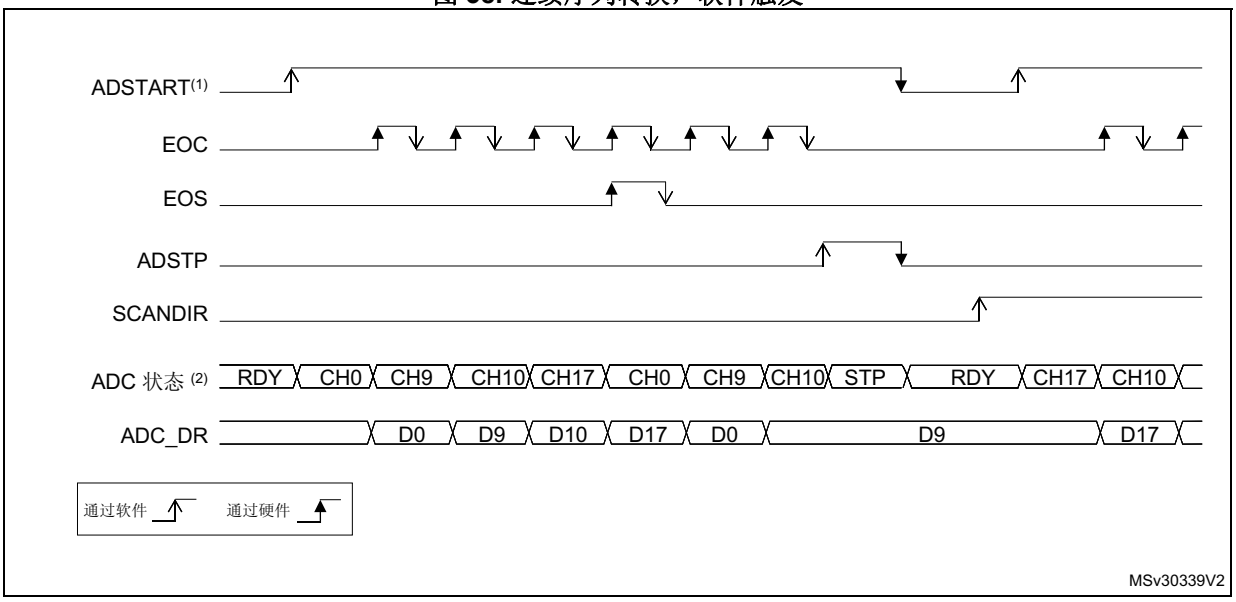
14.5.5 时序图示例（单次 / 连续模式硬件 / 软件触发）

图 37. 单次序列转换，软件触发



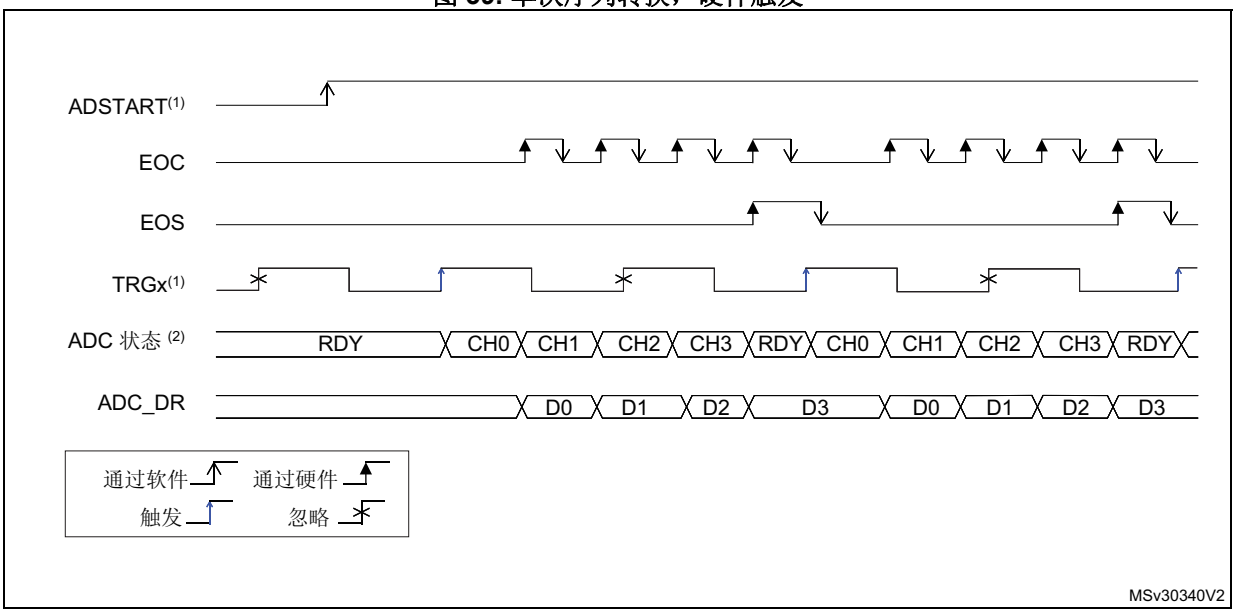
1. EXTEN = 00，CONT = 0
2. CHSEL = 0x20601，WAIT = 0，AUTOFF = 0

图 38. 连续序列转换，软件触发



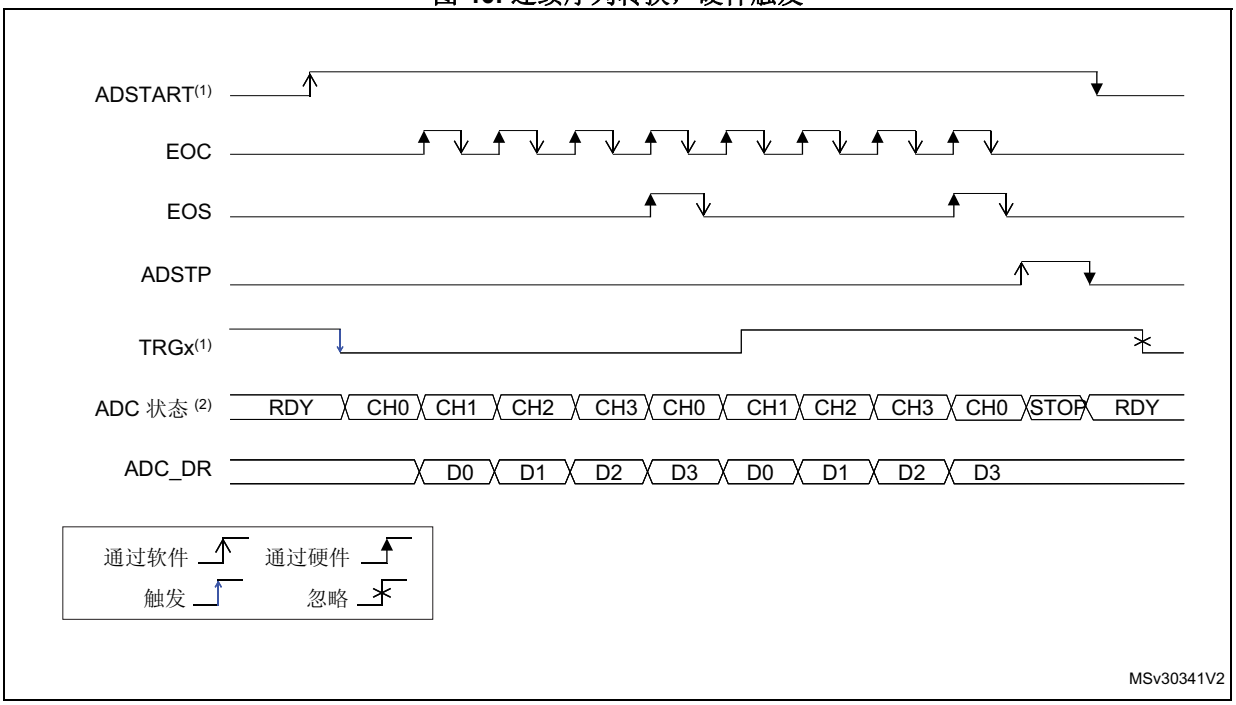
1. EXTEN = 00，CONT = 1,
2. CHSEL = 0x20601，WAIT = 0，AUTOFF = 0

图 39. 单次序列转换，硬件触发



1. EXTSEL = TRGx（过频），EXTEN = 01（上升沿），CONT = 0
2. CHSEL = 0xF，SCANDIR = 0，WAIT = 0，AUTOFF = 0

图 40. 连续序列转换，硬件触发



1. EXTSEL = TRGx，EXTEN = 10（下降沿），CONT = 1
2. CHSEL = 0xF，SCANDIR = 0，WAIT = 0，AUTOFF = 0

14.5.6 低频触发模式

在 ADC 已使能或最后一次 ADC 转换完成后，ADC 会立即准备开始新的转换。ADC 需要在预定义的时间 (t_{idle}) 开始，否则 ADC 转换数据可能会因晶体管泄漏而受到损坏（有关 t_{idle} 的最大值，请参见器件数据手册）。

如果应用必须支持长于最大 t_{idle} 值的时间（即，单次转换模式下相邻两次触发之间的时间，或者 ADC 使能与第一次 ADC 转换之间的时间），则需要重新调整 ADC 内部状态。将 ADC_CFGR2 寄存器中的 LFTRIG 位置 1，可使能该机制。通过将该位置 1，任何触发（软件或硬件）均可向 ADC 发送重新调整命令。相对于 LFTRIG 为零的情况，ADC 延迟 1 个 ADC 时钟周期后开始转换。

AUTOFF 位置 1 时，不必使用该模式。对于等待模式，只有第一次触发会生成内部重新调整命令。

14.6 数据管理

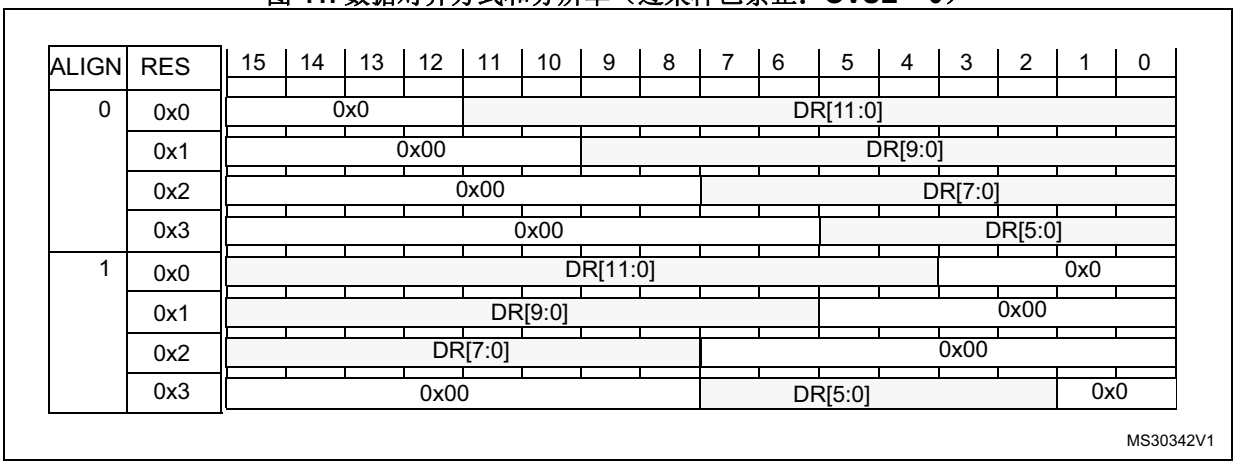
14.6.1 数据寄存器和数据对齐（ADC_DR、ALIGN）

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

ADC_DR 的格式取决于配置的数据对齐方式和分辨率。

ADC_CFGR1 寄存器中的 ALIGN 位用于选择转换后存储的数据的对齐方式。数据可右对齐 (ALIGN = 0) 或左对齐 (ALIGN = 1)，如*图 41* 所示。

图 41. 数据对齐方式和分辨率（过采样已禁止：OVSE = 0）



14.6.2 ADC 溢出（OVR、OVRMOD）

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

如果新转换完成时 EOC 标志仍为“1”，则 ADC_ISR 寄存器中的 OVR 标志会置 1。如果 ADC_IER 寄存器中的 OVRIE 位置 1，可产生中断。

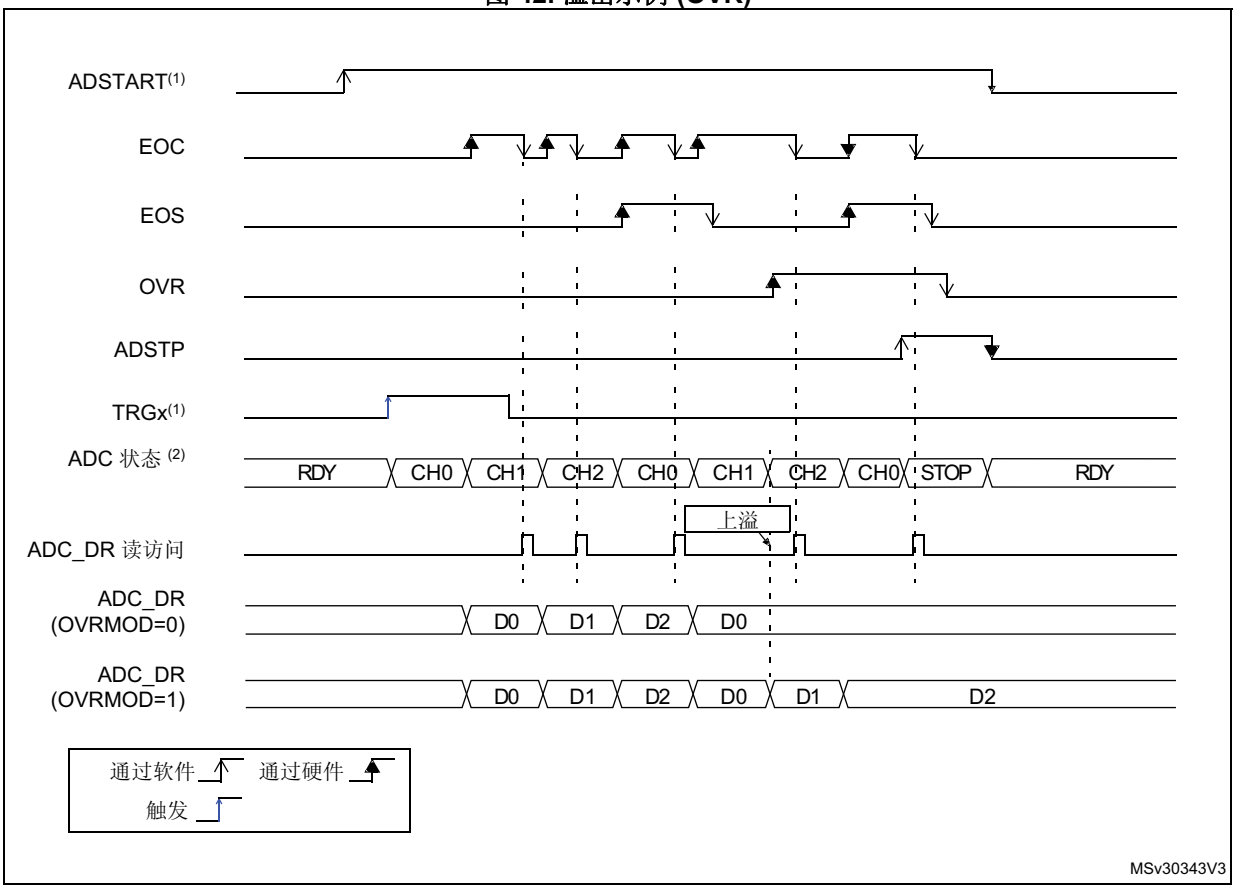
如果发生溢出情况，ADC 会保持工作状态并可继续进行转换，除非通过软件将 ADC_CR 寄存器中的 ADSTP 位置 1，从而停止并复位序列。

OVR 标志可通过由软件向其写入 1 的方式来清零。

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

- OVRMOD = 0
  - 溢出事件会保留数据寄存器的数据，防止其被覆盖：会保留原数据，并会丢弃新的转换结果。如果 OVR 保持为 1，可继续进行转换，但会丢弃所得的数据。
- OVRMOD = 1
  - 数据寄存器会由上一次转换结果覆盖，之前未读取的数据会丢失。如果 OVR 保持为 1，可继续进行转换，ADC_DR 寄存器始终包含最新转换得出的数据。

图 42. 溢出示例 (OVR)



14.6.3 在不使用 DMA 的情况下管理转换的数据序列

如果转换过程足够慢，则可使用软件来处理转换序列。在这种情况下，软件必须使用 EOC 标志及其相关中断来处理各个数据结果。每次转换完成时，都会将 ADC_ISR 寄存器中的 EOC 位置 1，并可读取 ADC_DR 寄存器。ADC_CFGR1 寄存器中的 OVRMOD 位应配置为 0，以便将溢出事件作为错误进行管理。

14.6.4 在不使用 DMA 且不发生溢出的情况下管理转换的数据

使 ADC 在转换一条或多条通道时无需在每次转换后都读取数据可能会很有用。在这种情况下，OVRMOD 位必须配置为 1，OVR 标志应被软件忽略。如果 OVRMOD = 1，溢出事件不会阻止 ADC 继续进行转换，ADC_DR 寄存器始终包含最新的转换数据。

14.6.5 使用 DMA 管理转换的数据

由于转换得出的所有通道值都存储在单个数据寄存器中，因此，在转换多条通道时使用 DMA 可提高效率。这样可以避免存储在 ADC_DR 寄存器中的转换数据结果丢失。

若 DMA 模式已使能（ADC_CFGR1 寄存器中的 DMAEN 位置 1），则会在每个通道转换后生成 DMA 请求。这样便可将转换的数据从 ADC_DR 寄存器传输到用软件选择的目标位置。

注意： *ADC 校准阶段完成后，必须将 ADC_CFGR1 寄存器中的 DMAEN 位置 1。*

尽管如此，如果因 DMA 无法及时处理 DMA 传输请求而发生溢出 (OVR = 1)，ADC 会停止生成 DMA 请求，新转换对应的数据不会通过 DMA 进行传输。这意味着可将传输到 RAM 的所有数据都视为有效数据。

根据 OVRMOD 位的配置，可保留或覆盖数据（请参见*第 237 页的第 14.6.2 节：ADC 溢出（OVR、OVRMOD）*）。

DMA 传输请求会禁止，直至软件将 OVR 位清零。

根据应用用途的不同，推荐使用两种不同的 DMA 模式，并使用 ADC_CFGR1 寄存器中的 DMACFG 位配置相应的模式：

- DMA 单次模式 (DMACFG = 0)。
  如果计划将 DMA 设置为传输固定数目的数据字，应选择该模式。
- DMA 循环模式 (DMACFG = 1)
  如果计划将 DMA 设置为循环模式或双缓冲区模式，应选择该模式。

DMA 单次模式 (DMACFG = 0)

在该模式下，每次出现新的转换数据字时，ADC 都会生成 DMA 传输请求，DMA 到达最后一个 DMA 传输操作时（发生传输完成中断，请参见*第 169 页的第 9 节：直接存储器访问控制器 (DMA)*），即使转换已再次开始，ADC 也会停止生成 DMA 请求。

DMA 传输完成后（在 DMA 控制器中配置的所有传输操作均已完成后）：

- ADC 数据寄存器的内容会冻结。
- 任何正在进行的转换都会中止，其部分结果会被丢弃
- 不会将任何新的 DMA 请求发送到 DMA 控制器。如果仍存在已开始的转换，这样可避免生成溢出错误。
- 扫描序列会停止并复位
- DMA 会停止

DMA 循环模式 (DMACFG = 1)

在该模式下，每次数据寄存器中出现新的转换数据字时，ADC 都会生成 DMA 传输请求，即使 DMA 已到达最后一次 DMA 传输操作也不例外。这样可将 DMA 配置为循环模式，从而处理连续的模拟输入数据流。

14.7 低功耗特性

14.7.1 等待模式转换

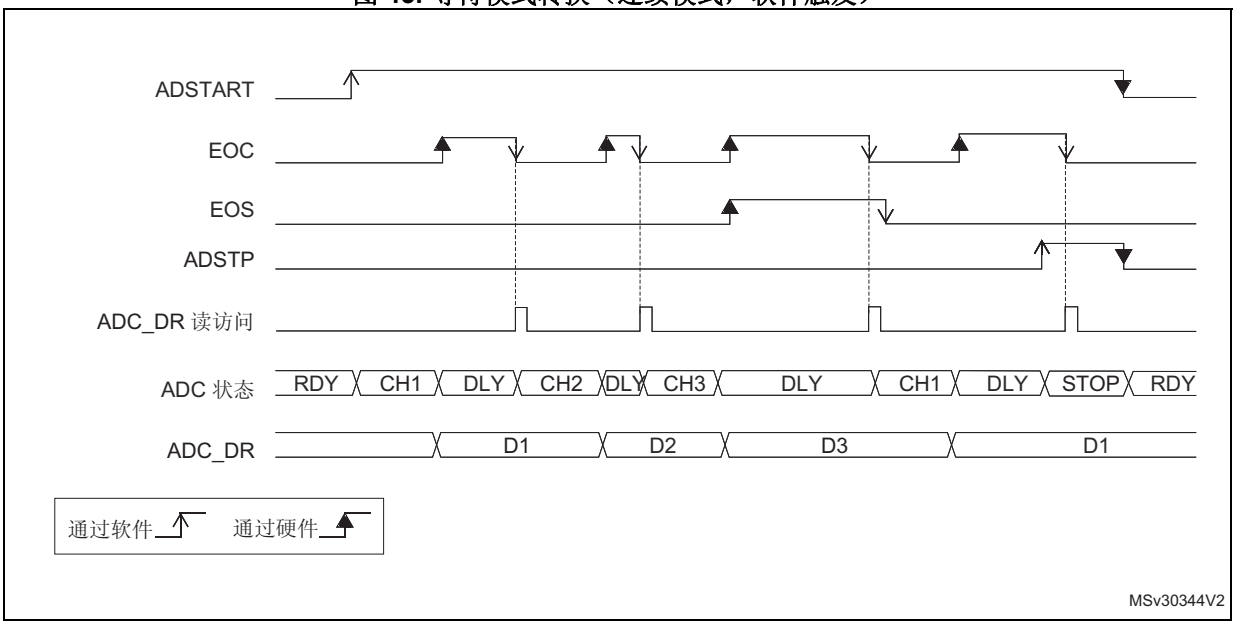
等待模式转换可用于简化软件，并可优化采用低频时钟的应用（此类应用可能存在 ADC 溢出的风险）的性能。

如果 ADC_CFGR1 寄存器中的 WAIT 位置 1，则仅当之前的数据已进行处理、ADC_DR 寄存器已读取或者 EOC 位已清零后，才会开始新的转换。

通过这种方式，可自动调整 ADC 的速度，使其适应系统读取数据的速度。

注意： *转换进行时发生的或在读访问之前的等待时间内发生的硬件触发会被忽略。*

图 43. 等待模式转换（连续模式，软件触发）



1. EXTEN = 00，CONT = 1
2. CHSEL = 0x3，SCANDIR = 0，WAIT = 1，AUTOFF = 0

14.7.2 自动关闭模式 (AUTOFF)

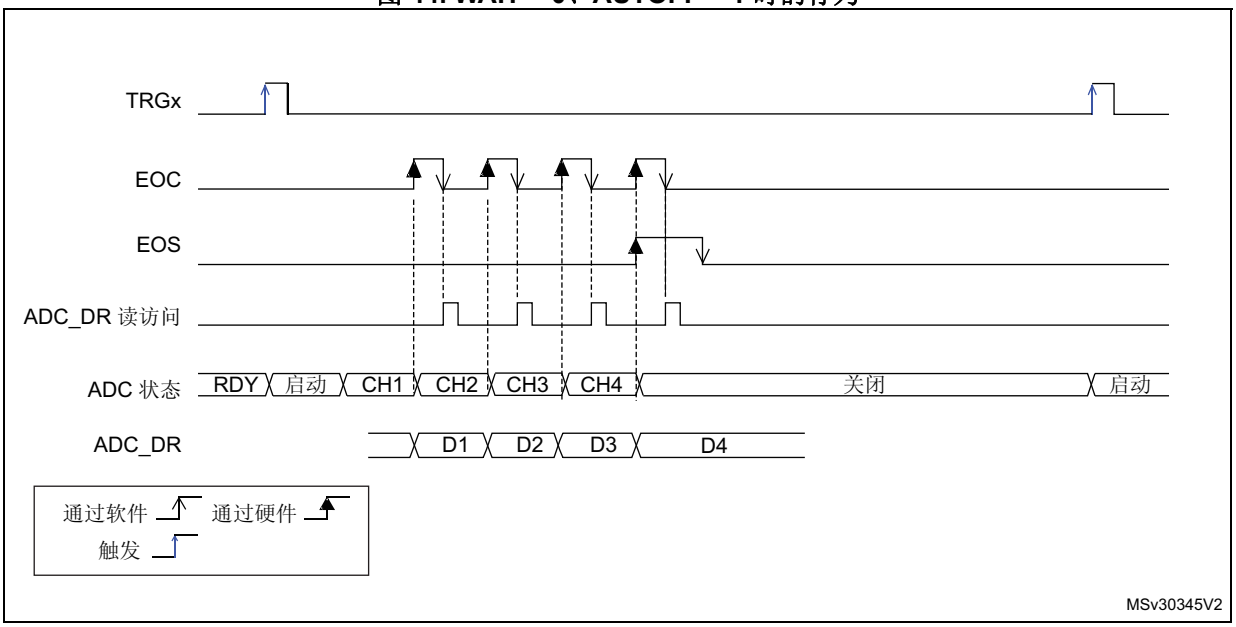
ADC 具有自动电源管理功能，也称为自动关闭模式，将 ADC_CFGR1 寄存器中的 AUTOFF 位置 1 可使能该模式。

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

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

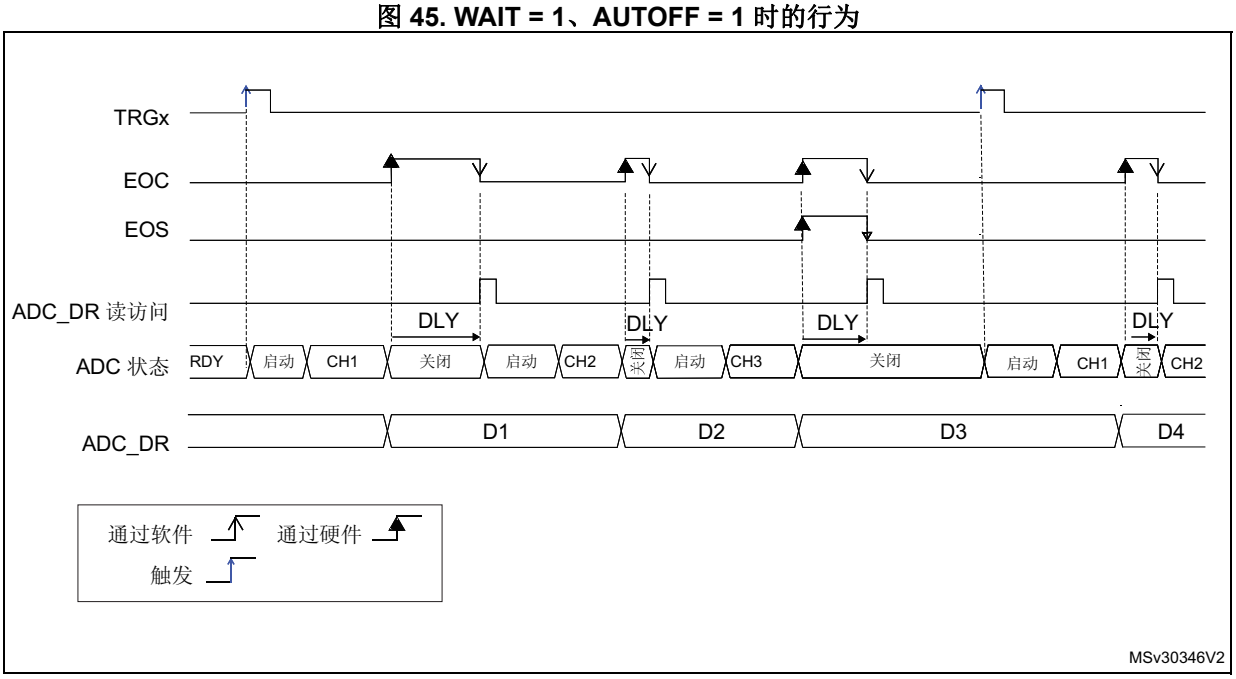
对于采用低频时钟的应用，可将自动关闭模式与等待模式转换 (WAIT = 1) 结合使用，如果 ADC 在等待过程中自动掉电、并会在应用读取 ADC_DR 寄存器后立即重启，这种组合可显著降低功耗（请参见*图 45：WAIT = 1、AUTOFF = 1 时的行为*）。

图 44. WAIT = 0、AUTOFF = 1 时的行为



1. EXTSEL = TRGx，EXTEN = 01（上升沿），CONT = x，ADSTART = 1，CHSEL = 0xF，SCANDIR = 0，WAIT = 1，AUTOFF = 1

图 45. WAIT = 1、AUTOFF = 1 时的行为



1. EXTSEL = TRGx，EXTEN = 01（上升沿），CONT = x，ADSTART = 1，CHSEL = 0xF，SCANDIR = 0，WAIT = 1，AUTOFF = 1

14.8 模拟窗口看门狗

三个 AWD 模拟看门狗会监测一些通道是否保持在配置的电压范围（窗口）内。

14.8.1 模拟看门狗 1 说明

通过将 ADC_CFGR1 寄存器中的 AWD1EN 位置 1 来使能 AWD1 模拟看门狗。该功能用于监测一条选定的通道或所有已使能的通道（请参见*表 58：模拟看门狗 1 通道选择*）是否仍处于所配置的电压范围（窗口）内，如*图 46* 所示。

如果 ADC 转换的模拟电压低于阈值下限或高于阈值上限，则 AWD1 模拟看门狗状态位会置 1。这些阈值在 ADC_AWD1TR 寄存器的 HT1[11:0] 和 LT1[11:0] 位中编程设定。可以通过将 ADC_IER 寄存器中的 AWD1IE 位置 1 来使能中断。

AWD1 标志可通过由软件编程为 1 的方式来清零。

如果转换的数据的分辨率小于 12 位（取决于 DRES[1:0] 位），由于始终会在内部对完整的 12 位原始转换数据进行比较（左对齐），因此编程阈值的 LSB 必须保持清零状态。

表 57 介绍了如何对所有可能的分辨率进行比较。

表 57. 模拟看门狗比较

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

1. 进行任何对齐计算之前，会对原始转换数据进行看门狗比较。

表 58 介绍了如何配置 ADC_CFGR1 寄存器中的 AWD1SGL 和 AWD1EN 位，以使能一条或多条通道上的模拟看门狗。

图 46. 模拟看门狗的保护区域

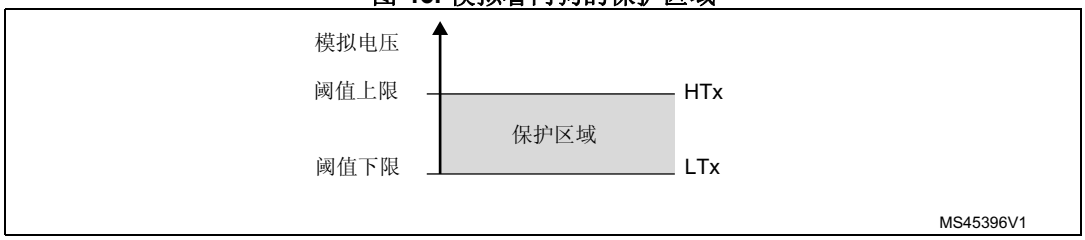


表 58. 模拟看门狗 1 通道选择

模拟看门狗保护的通道	AWD1SGL 位	AWD1EN 位
无	x	0
所有通道	0	1
单(1)通道	1	1

1. 通过 AWD1CH[4:0] 位选择

14.8.2 模拟看门狗 2 和 3 说明

第二个和第三个模拟看门狗更加灵活，可通过编程 ADC_AWDxCR 中的 AWDxCHy（x = 2 和 3）来保护多条已选通道。

ADC_AWDxCR 寄存器中的任何 AWDxCHy 位（x = 2、3）置 1 时，都会使能相应的看门狗。

如果转换的数据的分辨率小于 12 位（通过 DRES[1:0] 位配置），由于始终会在内部对完整的 12 位原始转换数据进行比较（左对齐），因此编程阈值的 LSB 必须保持清零状态。

表 57 介绍了如何对所有可能的分辨率进行比较。

如果 ADC 转换的模拟电压低于阈值下限或高于阈值上限，则 AWD2/3 模拟看门狗状态位会置 1。这些阈值在 ADC_AWDxTR 寄存器（x = 2 或 3）的 HTx[11:0] 和 LTx[11:0] 中编程设定。可以通过将 ADC_IER 寄存器中的 AWDxIE 位置 1 来使能中断。

AWD2 和 ADW3 标志可通过由软件编程为 1 的方式来清零。

14.8.3 ADC_AWDx_OUT 输出信号生成

每个模拟看门狗都关联到一个内部硬件信号 ADC_AWDx_OUT（x 为看门狗编号），该信号直接连接到一些片上定时器的 ETR 输入（外部触发）（有关如何选择 ADC_AWDx_OUT 信号作为 ETR 的详细信息，请参见定时器部分）。

当关联的模拟看门狗使能时，ADC_AWDx_OUT 会激活：

- 当受保护的转换超出编程阈值时，ADC_AWDx_OUT 会置 1。
- 在编程阈值范围内的下一受保护转换结束后，ADC_AWDx_OUT 会复位。如果下一受保护转换仍超出编程阈值范围，该位仍保持置 1。
- 禁止 ADC 时（将 ADDIS 置 1 时），ADC_AWDx_OUT 也保持复位状态。请注意，停止转换（ADSTP 置 1）可能会清除 ADC_AWDx_OUT 状态。
- ADC 转换不受保护的通道时，ADC_AWDx_OUT 状态不会更改（请参见*图 49*）

AWDx 标志由硬件置 1，由软件复位。AWDx 标志对 ADC_AWDx_OUT 的生成没有影响（例如：如果软件未将 AWDx 标志清零，即 AWDx 标志会保持为 1，此时 ADC_AWDx_OUT 仍可翻转）。

ADC_AWDx_OUT 信号通过 ADC_CLK 域生成。即使 APB 时钟停止，也可生成该信号。

AWD 比较在每次 ADC 转换结束时执行。比较操作后的两个 ADC_CLK 时钟周期后会出现 ADC_AWDx_OUT 上升沿和下降沿。

由于 ADC_AWDx_OUT 通过 ADC_CLK 域生成，而 AWD 标志通过 APB 时钟域生成，因此这两个信号的上升沿不同步。

图 47. ADC_AWDx_OUT 信号生成

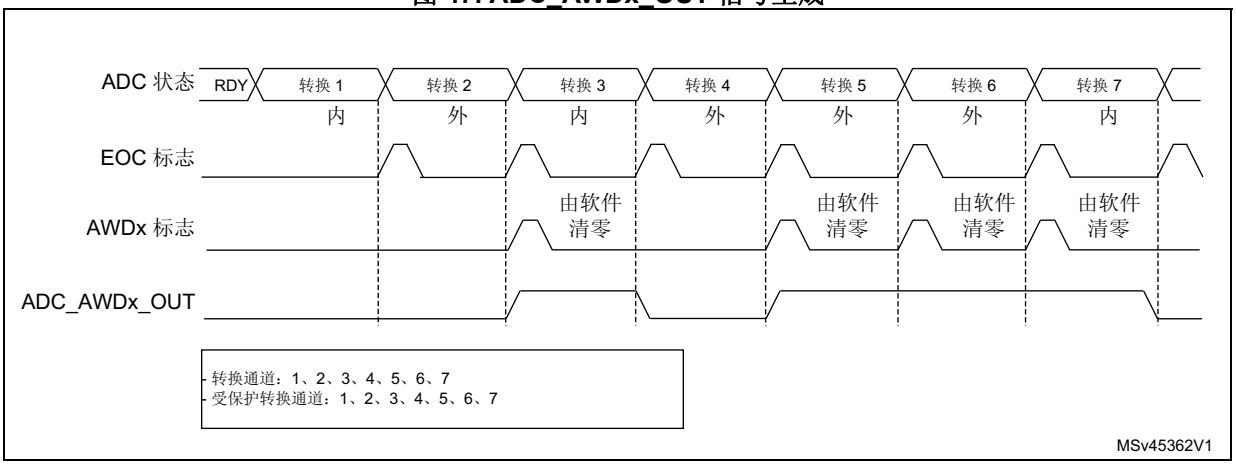


图 48. ADC_AWDx_OUT 信号生成（AWDx 标志未通过软件清零）

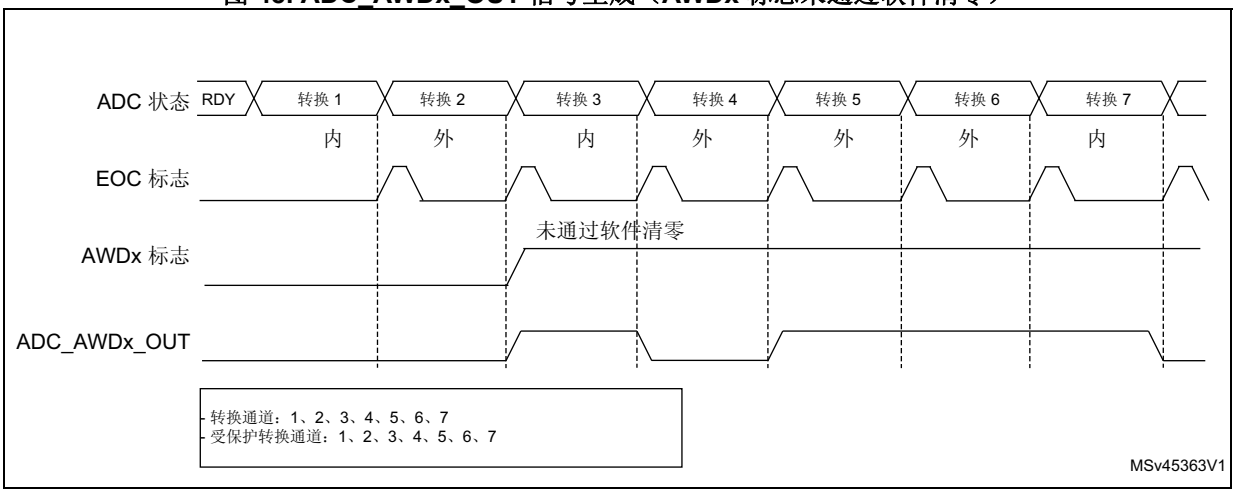
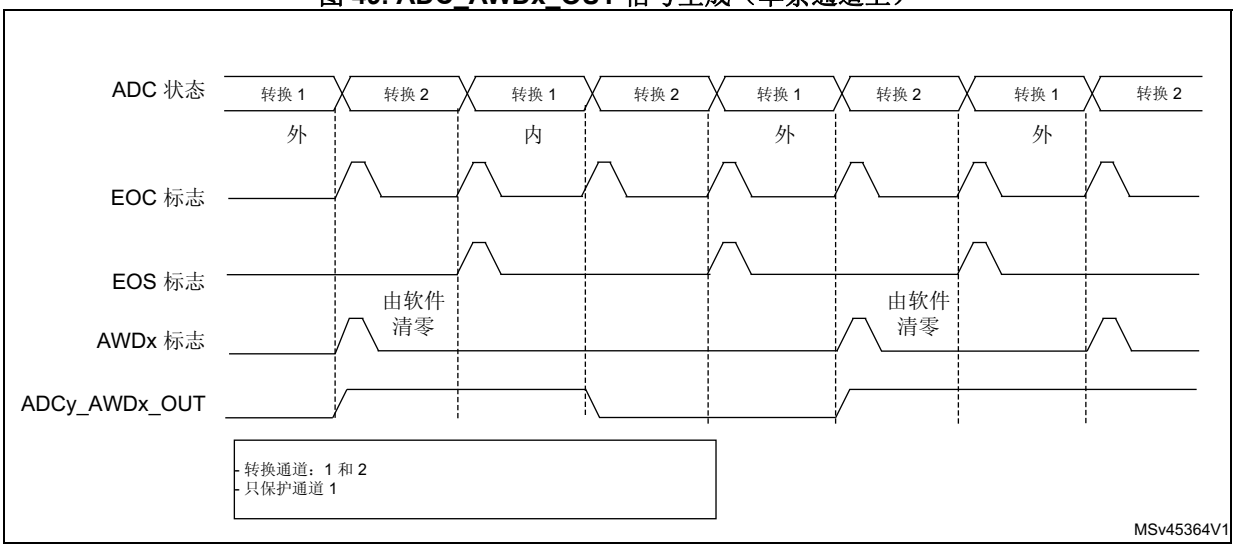


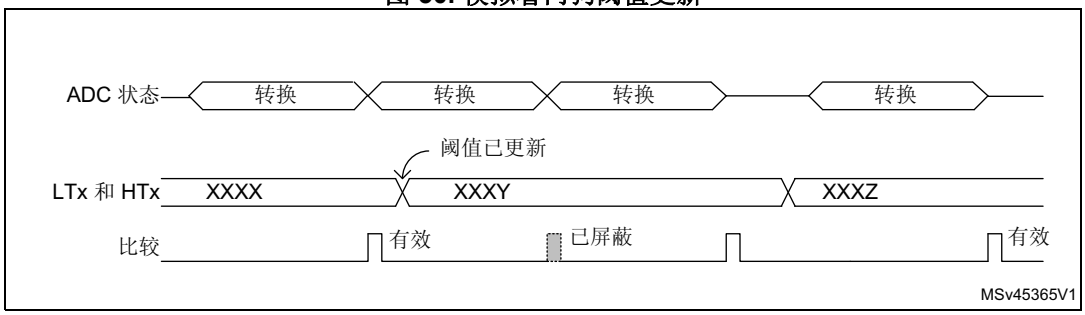
图 49. ADC_AWDx_OUT 信号生成（单条通道上）



14.8.4 模拟看门狗阈值控制

LTx[11:0] 和 HTx[11:0] 可在模数转换期间（即，从 ADC 内部状态转换开始到转换结束）更改。如果在 ADC 保护通道转换期间编程设定 HTx 和 LTx 位，该转换将屏蔽看门狗功能。开始新的转换时会取消该屏蔽，并应用更改后的新 AWD 阈值，然后开始计算下一个 ADC 转换结果。AWD 比较在每次转换结束时执行。如果当前 ADC 数据超出新的阈值范围，则不会生成任何中断或 ADC_AWDx_OUT 信号。只有在阈值更新后开始的 ADC 转换结束时，才会生成中断和 ADC_AWDx_OUT 信号。如果 ADC_AWDx_OUT 已置为有效，则编程设定新的阈值不会将 ADC_AWDx_OUT 信号置为无效。

图 50. 模拟看门狗阈值更新



14.9 过采样器

过采样单元会进行数据预处理，以减轻 CPU 的负担。过采样单元可处理多个转换，并计算多个转换结果的平均值，得到数据宽度增加（高达 16 位）的单个数据。

它提供的结果采用以下形式，其中的 N 和 M 可以进行调整：

$$\text{结果} = \frac{1}{M} \times \sum_{n=0}^{n=N-1} \text{转换}(t_n)$$

允许通过硬件执行以下功能：计算平均值、降低数据速率、改进 SNR 以及基本滤波。

过采样率 N 通过 ADC_CFGR2 寄存器中的 OVFS[2:0] 位进行定义，它的范围是 2x 到 256x。分频系数 M 由向右移位构成，最多可移 8 位。它可通过 ADC_CFGR2 寄存器中的 OVSS[3:0] 位进行配置。

求和单元可得出多达 20 位（256 x 12 位）的结果，结果会先右移。随后，会将结果的高位截断，只留下 16 个最低有效位，这些位使用移位后剩下的最低有效位四舍五入为最接近的数值，然后将最终得到的结果传输到 ADC_DR 数据寄存器中。

注意： *如果移位后得到的中间结果超过 16 位，则会截断结果的高位。*

图 51. 20 位到 16 位结果的截断过程

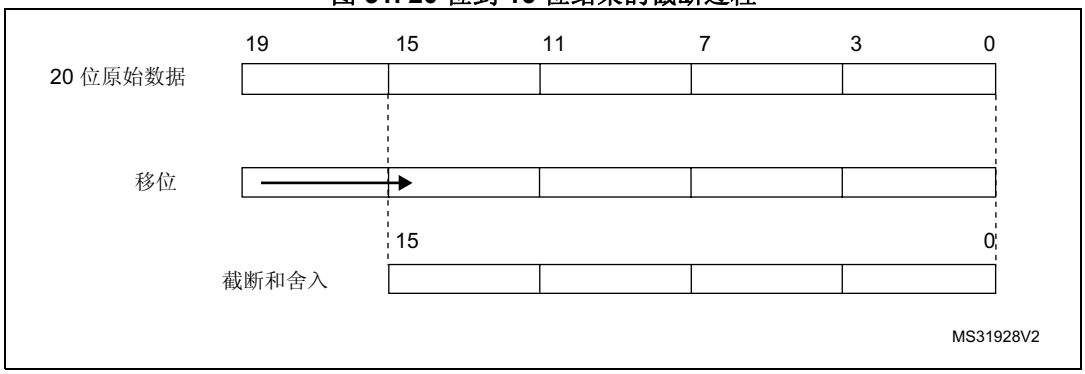
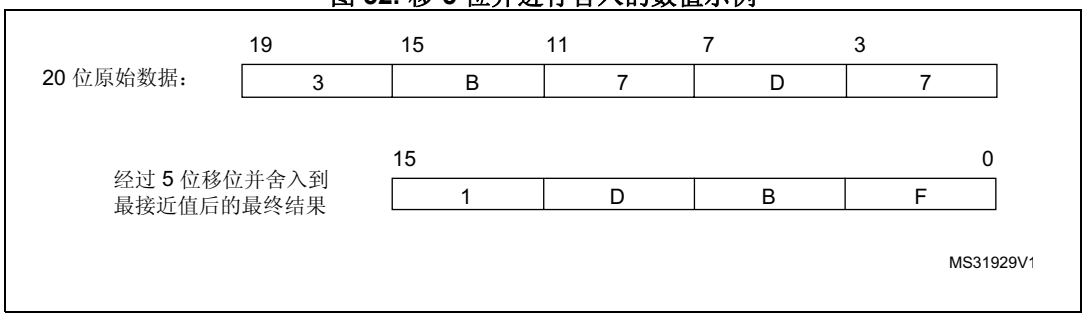


图 52 介绍了从原始的 20 位累加数据到最终 16 位结果的数值处理过程。

图 52. 移 5 位并进行舍入的数值示例



以下*表 59* 列出了原始转换数据等于 0xFFF 时对应的各种 N 和 M 组合的数据格式。

表 59. 最大输出结果与 N 和 M 的对应关系。呈灰色显示的数值表示截断的部分

过采样率	最大原始数据	不移位 OVSS = 0000	移 1 位 OVSS = 0001	移 2 位 OVSS = 0010	移 3 位 OVSS = 0011	移 4 位 OVSS = 0100	移 5 位 OVSS = 0101	移 6 位 OVSS = 0110	移 7 位 OVSS = 0111	移 8 位 OVSS = 1000
2x	0x1FFE	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040	0x0020
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

与标准转换模式相比，过采样模式下的转换时序不会发生变化：整个过采样序列中，采样时间保持相等。每完成 N 次转换都会提供新数据，等效延迟等于 N x t_{CONV} = N x (t_{SMPL} + t_{SAR})。各标志的情况如下：

- 每个采样阶段后都会将采样阶段结束标志 (EOSMP) 置 1
- 转换结束事件 (EOC) 在每 N 次转换后产生，过采样转换结果可用。
- 过采样数据序列完成后（即 N x 序列长度次转换之后），会发生序列结束事件 (EOCSEQ)

14.9.1 过采样时支持的 ADC 工作模式

在过采样模式下，大部分 ADC 工作模式都可用：

- 单次或连续模式转换、向前或向后扫描序列
- 可由软件或触发器启动 ADC 转换
- ADC 在转换过程中停止（中止）
- 通过 CPU 或 DMA 在支持溢出检测的情况下读取数据
- 低功耗模式（WAIT、AUTOFF）
- 可编程分辨率：在这种情况下，会按照与 12 位转换相同的方式对分辨率降低的转换值（根据 ADC_CFGR1 寄存器中的 RES[1:0] 位）进行累加、截断、四舍五入和移位

注意： *处理过采样数据时，不可使用对齐模式。ADC_CFGR1 中的 ALIGN 位会被忽略，且数据始终采用右对齐格式。*

14.9.2 模拟看门狗

可以使用模拟看门狗功能，但存在以下区别：

- 会忽略 RES[1:0] 位，始终会使用完整的 12 位值 HTx[11:0] 和 LTx[11:0] 进行比较。
- 会比较 16 位过采样结果 ADC_DR[15:4] 的 12 个最高有效位

注意： *移位位数较大时必须多加留意，因为这样会缩小比较范围。例如，如果过采样结果移了 4 位，得到 12 位右对齐数据，那么只能对 8 个数据位执行有效的模拟看门狗比较。比较操作会在 ADC_DR[11:4] 与 HTx[7:0]/LTx[[7:0] 之间进行，并且 HTx[11:8]/LTx[11:8] 必须保持复位状态。*

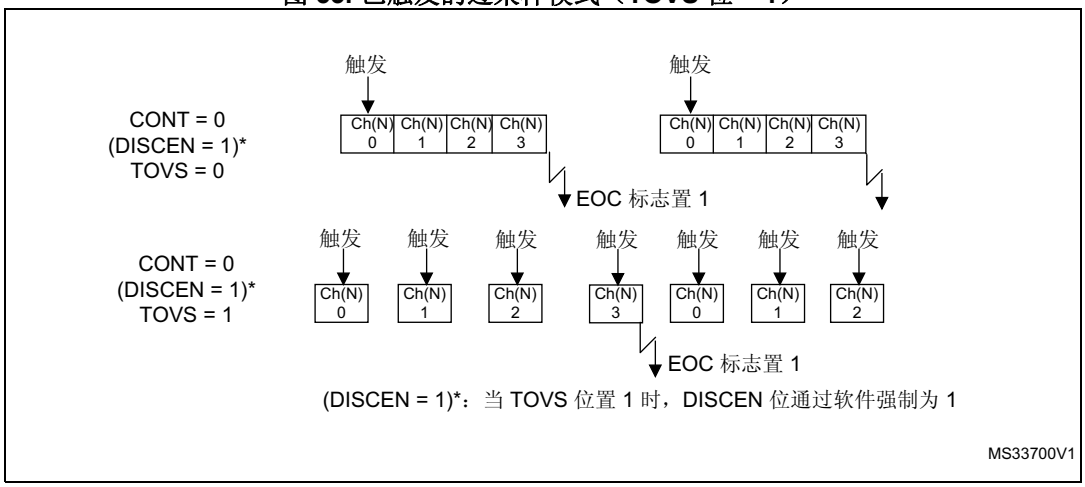
14.9.3 触发模式

平均值计算单元还可用于基本滤波，虽然它不是效率非常高的滤波器（衰减缓慢、停止频段衰减受限），但可用作陷波滤波器，用于抑制恒定的寄生频率（通常来自电源或切换模式电源）。为此，可使用 ADC_CFGR2 中的 TOVS 位使能特定的不连续模式，以获得由用户定义、且与转换时间本身无关的过采样频率。

图 53 显示了在不连续模式下如何响应触发从而开始转换。

如果 TOVS 位置 1，则会忽略 DISCEN 位的内容，并将该位视为 1。

图 53. 已触发的过采样模式（TOVS 位 = 1）



14.10 温度传感器和内部参考电压

温度传感器可用于测量器件的结温 (T_J)。温度传感器在内部连接到 ADC V_{IN}[9] 输入通道，该通道用于将传感器输出电压转换为数字值。温度传感器模拟引脚的采样时间必须大于数据手册中指定的最小 T_{S_temp} 值。不使用时可将传感器置于掉电模式。

内部参考电压 (VREFINT) 为 ADC 提供了一个稳定的（带隙）电压输出。VREFINT 内部连接到 ADC V_{IN}[10] 输入通道。VREFINT 的精确电压由 ST 在生产测试期间对每部分单独测量，储存于系统存储区。

图 54 显示的是温度传感器、内部参考电压与 ADC 之间连接的框图。

必须将 TSEN 位置 1 才能使能 ADC V_{IN}[9]（温度传感器）的转换，必须将 VREFEN 位置 1 才能使能 ADC V_{IN}[10] (V_{REFINT}) 的转换。

温度传感器的输出电压随温度线性变化。由于工艺不同，该线的偏移量取决于各个芯片（芯片之间的温度变化可达 45 °C）。

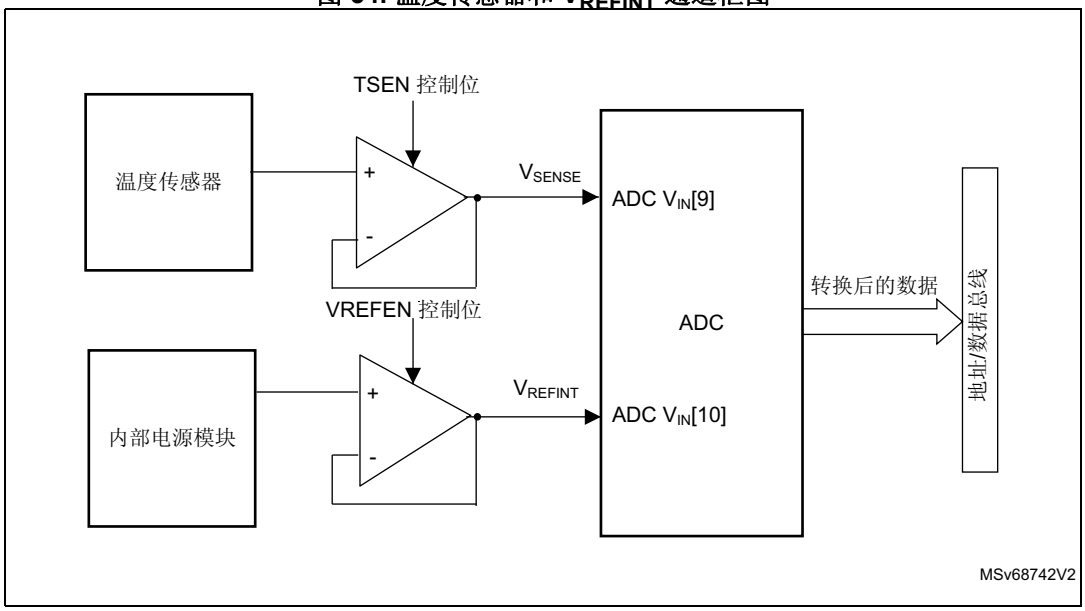
未校准的内部温度传感器更适用于对温度变量而非绝对温度进行测量的应用。为提高温度传感器测量的准确性，ST 在生产过程中将校准值存储在每个器件的系统存储器中。

在制造过程中，会将温度传感器的校准数据和内部参考电压存储在系统存储区。随后，用户应用可读取这些数据，并使用这些数据提高温度传感器或内部参考的准确性。其他相关信息，请参见数据手册。

主要特性

- 线性度：最高 ±2 °C，精度取决于校准情况

图 54. 温度传感器和 V_{REFINT} 通道框图



读取温度

1. 选择 ADC V_{IN}[9] 输入通道。
2. 选择器件数据手册中规定的合适的采样时间 (T_{S_temp})。
3. 将 ADC_CCR 寄存器中的 TSEN 位置 1，将温度传感器从掉电模式中唤醒，并等待其稳定时间 (t_{START})。
4. 将 ADC_CR 寄存器中的 ADSTART 位置 1（或通过外部触发），开始 ADC 转换。
5. 读取 ADC_DR 寄存器中生成的 V_{SENSE} 数据。
6. 使用以下公式计算温度

$$\text{温度 (°C)} = \frac{\text{Sense\_DATA} - \text{TS\_CAL1}}{\text{Avg\_Slope\_Code}} + \text{TS\_CAL1\_TEMP}$$

$$\text{Avg\_Slope\_Code} = \text{Avg\_Slope} \times 4096 / 3000$$

$$\text{Sense\_DATA} = \text{TS\_DATA} \times V_{DDA} / 3.0$$

其中：

- TS_CAL1 是在 TS_CAL1_TEMP 下获得的温度传感器校准值（有关 TS_CAL1 值的信息，请参见数据手册）
- TS_DATA 是由 ADC 转换得到的实际温度传感器输出值
  有关 TS_CAL1 校准点的更多信息，请参见特定器件数据手册。
- Avg_Slope 是用 mV/°C 表示的温度传感器输出电压系数（有关 Avg_Slope 值的信息，请参见数据手册）。

注意： *传感器从掉电模式中唤醒后需要一个启动时间，启动时间过后其才能正确输出 V_{SENSE}。ADC 在上电后同样需要一个启动时间，因此，为尽可能缩短延迟时间，应同时将 ADEN 和 TSEN 位置 1。*

使用内部参考电压计算实际的 V_{DDA} 电压

施加给器件的 V_{DDA} 电源电压可能会有变化，或无法获得准确值。在制造过程中由 ADC 在 V_{DDA_Charac} 下获得的内置内部参考电压 (V_{REFINT}) 及其校准数据可用于评估实际的 V_{DDA} 电压。

以下公式可求得为器件供电的实际 V_{DDA} 电压：

$$V_{DDA} = V_{DDA\_Charac} \times VREFINT\_CAL / VREFINT\_DATA$$

其中：

- V_{DDA_Charac} 是制造过程中 V_{DDA} 电压在 V_{REFINT} 下的特性值。该值在器件数据手册中指定。
- VREFINT_CAL 是 VREFINT 校准值
- VREFINT_DATA 是由 ADC 转换得到的实际 VREFINT 输出值

将电源相关的 ADC 测量值转换为绝对电压值

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

$$V_{CHANNELx} = \frac{V_{DDA}}{NUM\_CODES} \times ADC\_DATA_x$$

对于 V_{DDA} 值未知的应用，必须使用内部参考电压，V_{DDA} 可替换为*使用内部参考电压计算实际的 V_{DDA} 电压*一节提供的表达式，从而得出以下公式：

$$V_{CHANNELx} = \frac{V_{DDA\_Charac} \times VREFINT\_CAL \times ADC\_DATA_x}{VREFINT\_DATA \times NUM\_CODES}$$

其中：

- V_{DDA_Charac} 是制造过程中 V_{DDA} 电压在 V_{REFINT} 下的特性值。该值在器件数据手册中指定。
- VREFINT_CAL 是 VREFINT 校准值
- ADC_DATA_x 是由 ADC 在通道 x 上测得的值（右对齐）
- VREFINT_DATA 是由 ADC 转换得到的实际 VREFINT 输出值
- NUM_CODES 是 ADC 输出代码数。例如，如果分辨率为 12 位，该值为 $2^{12} = 4096$；如果分辨率为 8 位，该值为 $2^8 = 256$。

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

14.11 ADC 中断

发生下列任一事件均可生成中断：

- 校准结束（EOCAL 标志）
- ADC 就绪后，ADC 上电（ADRDY 标志）
- 任何转换结束（EOC 标志）
- 转换序列结束（EOS 标志）
- 发生模拟看门狗检测时（AWD1、AWD2 和 AWD3 标志）
- 通道配置就绪时（CCRDY 标志）
- 采样阶段结束时（EOSMP 标志）
- 发生数据溢出时（OVR 标志）

可以使用单独的中断使能位以提高灵活性。

表 60. ADC 中断

中断事件	事件标志	使能控制位
校准结束	EOCAL	EOCALIE
ADC 就绪	ADRDY	ADRDYIE
转换结束	EOC	EOCIE
转换序列结束	EOS	EOSIE
模拟看门狗 1 状态位置 1	AWD1	AWD1IE
模拟看门狗 2 状态位置 1	AWD2	AWD2IE
模拟看门狗 3 状态位置 1	AWD3	AWD3IE
通道配置就绪	CCRDY	CCRDYIE
采样阶段结束	EOSMP	EOSMPIE
上溢	OVR	OVRIE

14.12 ADC 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.2 节*。

14.12.1 ADC 中断和状态寄存器 (ADC_ISR)

ADC interrupt and status register

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CCRDY	Res.	EOCAL	Res.	AWD3	AWD2	AWD1	Res.	Res.	OVR	EOS	EOC	EOSMP	ADRDY
		rc_w1		rc_w1		rc_w1	rc_w1	rc_w1			rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:14 保留，必须保持复位值。

位 13 **CCRDY**：通道配置就绪标志 (Channel Configuration Ready flag)

当在对 ADC_CHSELR 寄存器进行编程或者更改 CHSELRMOD 或 SCANDIR 后应用通道配置时，该标志位由硬件置 1。通过软件编程可将该位清零。

0：未应用通道配置更新。

1：应用了通道配置更新。

注意：当通过软件配置通道（通过编程 ADC_CHSELR 或者更改 CHSELRMOD 或 SCANDIR）时，必须等待至 CCRDY 标志置 1 之后才能再次进行配置或开始转换，否则会忽略新的配置（或 START 位）。该标志置为有效后，如果需要通过软件再次配置通道，必须将 CCRDY 标志清零才能继续新的配置。

位 12 保留，必须保持复位值。

位 11 **EOCAL**：校准结束标志 (End Of Calibration flag)

校准完成时，该位由硬件置 1。通过软件写入 1 可将该位清零。

0：校准未完成

1：校准已完成

位 10 保留，必须保持复位值。

位 9 **AWD3**：模拟看门狗 3 标志 (Analog watchdog 3 flag)

当转换后的电压超过在 ADC_AWD3TR 和 ADC_AWD3TR 寄存器中编程设定的值时，会通过硬件将该位置 1。通过软件编程为 1 可将该位清零。

0：未发生模拟看门狗事件（或标志事件已通过软件确认并清零）。

1：发生模拟看门狗事件

位 8 **AWD2**：模拟看门狗 2 标志 (Analog watchdog 2 flag)

当转换后的电压超过在 ADC_AWD2TR 和 ADC_AWD2TR 寄存器中编程设定的值时，会通过硬件将该位置 1。通过软件编程可将该位清零。

0：未发生模拟看门狗事件（或标志事件已通过软件确认并清零）。

1：发生模拟看门狗事件

位 7 **AWD1**：模拟看门狗 1 标志 (Analog watchdog 1 flag)

当转换后的电压超过在 ADC_TR1 和 ADC_HR1 寄存器中编程设定的值时，会通过硬件将该位置 1。通过软件编程为 1 可将该位清零。

0：未发生模拟看门狗事件（或标志事件已通过软件确认并清零）。

1：发生模拟看门狗事件

位 6:5 保留，必须保持复位值。

位 4 **OVR**：ADC 溢出 (ADC overrun)

该位在发生溢出事件时由硬件置 1，这意味着在 EOC 标志已置 1 时，新转换已完成。通过软件写入 1 可将该位清零。

0：未发生溢出（或标志事件已通过软件确认并清零）

1：发生溢出

位 3 **EOS**：序列结束标志 (End of sequence flag)

在由 CHSEL 位选择的一系列通道转换结束时，会通过硬件将该位置 1。通过软件写入 1 可将该位清零。

0：转换序列未完成（或标志事件已通过软件确认并清零）。

1：转换序列已完成

位 2 **EOC**：转换结束标志 (End of conversion flag)

当通道的每次转换结束，新数据结果出现在 ADC_DR 寄存器时，会通过硬件将该位置 1。通过软件向该位写入 1，或读取 ADC_DR 寄存器都可将该位清零。

0：通道转换未完成（或标志事件已通过软件确认并清零）。

1：通道转换已完成

位 1 **EOSMP**：采样结束标志 (End of sampling flag)

在转换过程中，当采样阶段结束时该位由硬件置 1。通过软件将该位编程为“1”，可将该位清零。

0：采样阶段未结束（或标志事件已通过软件确认并清零）

1：采样阶段已结束

位 0 **ADRDY**：ADC 就绪 (ADC ready)

DC 使能后 (ADEN = 1) 以及 ADC 达到准备好接收转换请求的状态时，会通过硬件将该位置 1。

通过软件写入 1 可将该位清零。

0：ADC 未准备好开始转换（或标志事件已通过软件确认并清零）

1：ADC 已准备好开始转换

14.12.2 ADC 中断使能寄存器 (ADC_IER)

ADC interrupt enable register

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CCRDYIE	Res.	EOCALIE	Res.	AWD3IE	AWD2IE	AWD1IE	Res.	Res.	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE
		rw		rw		rw	rw	rw			rw	rw	rw	rw	rw

位 31:14 保留，必须保持复位值。

位 13 **CCRDYIE**：通道配置就绪中断使能 (Channel Configuration Ready Interrupt enable)
该位由软件置 1 和清零，用于使能 / 禁止通道配置就绪中断。
0：禁止通道配置就绪中断
1：使能通道配置就绪中断
注意：仅当 ADSTART 位清零时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 12 保留，必须保持复位值。

位 11 **EOCALIE**：校准结束中断使能 (End of calibration interrupt enable)
该位由软件置 1 和清零，用于使能 / 禁止校准结束中断。
0：禁止校准结束中断
1：使能校准结束中断
注意：仅当 ADSTART 位清零时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 10 保留，必须保持复位值。

位 9 **AWD3IE**：模拟看门狗 3 中断使能 (Analog watchdog 3 interrupt enable)
通过软件将该位置 1 和清零可使能 / 禁止模拟看门狗中断。
0：禁止模拟看门狗中断
1：使能模拟看门狗中断
注意：仅当 ADSTART 位清零时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 8 **AWD2IE**：模拟看门狗 2 中断使能 (Analog watchdog 2 interrupt enable)
通过软件将该位置 1 和清零可使能 / 禁止模拟看门狗中断。
0：禁止模拟看门狗中断
1：使能模拟看门狗中断
注意：仅当 ADSTART 位清零时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 7 **AWD1IE**：模拟看门狗 1 中断使能 (Analog watchdog 1 interrupt enable)
通过软件将该位置 1 和清零可使能 / 禁止模拟看门狗中断。
0：禁止模拟看门狗中断
1：使能模拟看门狗中断
注意：仅当 ADSTART 位清零时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 6:5 保留，必须保持复位值。

位 4 **OVRIE**：溢出中断使能 (Overrun interrupt enable)
该位由软件置 1 和清零，用于使能 / 禁止溢出中断。
0：禁止溢出中断
1：使能溢出中断。OVR 位置 1 时产生中断。
注意：仅当 ADSTART 位清零时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 3 **EOSIE**：转换序列结束中断使能 (End of conversion sequence interrupt enable)
该位由软件置 1 和清零，用于使能 / 禁止转换序列结束中断。
0：禁止 EOS 中断
1：使能 EOS 中断 EOS 位置 1 时产生中断。
注意：仅当 ADSTART 位清零时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 2 **EOCIE**：转换结束中断使能 (End of conversion interrupt enable)
通过软件将该位置 1 和清零可使能 / 禁止转换结束中断。
0：禁止 EOC 中断
1：使能 EOC 中断 EOC 位置 1 时产生中断。
注意：仅当 ADSTART 位清零时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 1 **EOSMPIE**：采样结束标志中断使能 (End of sampling flag interrupt enable)
该位由软件置 1 和清零，用于使能 / 禁止采样阶段结束中断。
0：禁止 EOSMP 中断。
1：使能 EOSMP 中断。EOSMP 位置 1 时产生中断。
注意：仅当 ADSTART 位清零时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 0 **ADRDYIE**：ADC 就绪中断使能 (ADC ready interrupt enable)
该位由软件置 1 和清零，用于使能 / 禁止 ADC 就绪中断。
0：禁止 ADRDY 中断。
1：使能 ADRDY 中断。ADRDY 位置 1 时产生中断。
注意：仅当 ADSTART 位清零时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

14.12.3 ADC 控制寄存器 (ADC_CR)

ADC control register

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAL	Res.	Res.	ADVREGEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs			rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADSTP	Res.	ADSTART	ADDIS	ADEN
											rs		rs	rs	rs

位 31 **ADCAL**：ADC 校准 (ADC calibration)
该位由软件置 1，用于开始 ADC 校准。
校准完成后，该位由硬件清零。
0：校准已完成
1：写入 1 可校准 ADC。读取值为 1 表示正在进行校准。
注意：仅当 ADC 已禁止时（ADCAL = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0、AUTOFF = 0 且 ADEN = 0），才允许通过软件将 ADCAL 位置 1。
仅当 ADEN = 1 且 ADSTART = 0 时（ADC 已使能，当前未进行任何转换），才允许通过软件对 ADC_CALFACT 执行写操作来更新校准系数。

位 30:29 保留，必须保持复位值。

位 28 **ADVREGEN**：ADC 调压器使能 (ADC Voltage Regulator Enable)
该位由软件置 1，用于使能 ADC 内部调压器。调压器输出在 $t_{ADCVREG\_STUP}$ 后可用。
该位由软件清零，用于禁止调压器。仅当 ADEN 设置为 0 时，该位才能清零。
0：禁止 ADC 调压器
1：使能 ADC 调压器
注意：仅当 ADC 已禁止时（ADCAL = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0 且 ADEN = 0），才允许通过软件对该位域进行编程。

位 27:5 保留，必须保持复位值。

位 4 **ADSTP**：ADC 停止转换命令 (ADC stop conversion command)

该位由软件置 1，用于停止和丢弃正在进行的转换（ADSTP 命令）。

当转换已有效丢弃、并且 ADC 已准备好接收新的开始转换命令时，会通过硬件将该位清零。

0：当前未执行 ADC 停止转换命令

1：写入 1 可停止 ADC。读取值为 1 表示正在执行 ADSTP 命令。

注意：仅当 ADSTART = 1 且 ADDIS = 0 时（ADC 已使能、可能正在进行转换、并且没有任何待处理的禁止 ADC 的请求），将 ADSTP 置 1 才会生效。

位 3 保留，必须保持复位值。

位 2 **ADSTART**：ADC 开始转换命令 (ADC start conversion command)

该位由软件置 1，用于开始 ADC 转换。根据 EXTEN [1:0] 配置位的值，可以立即开始转换（软件触发配置），也可以在发生硬件触发事件后开始转换（硬件触发配置）。

该位通过硬件清零：

– 在单次转换模式下（CONT = 0、DISCEN = 0），如果选择了软件触发 (EXTEN = 00)：出现转换序列结束 (EOS) 标志时清零。

– 在不连续转换模式下（CONT = 0、DISCEN = 1），如果选择了软件触发 (EXTEN = 00)：出现转换结束 (EOC) 标志时清零。

– 在所有其他情况下：执行完 ADSTP 命令后，由硬件将 ADSTP 位清零的同时清零。

0：当前未进行 ADC 转换。

1：写入 1 可开始 ADC。读取值为 1 表示 ADC 正在工作，可能正在进行转换。

注意：仅当 ADEN = 1 且 ADDIS = 0 时（ADC 已使能，并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 ADSTART 置 1。

在写入 ADC_CHSELR 寄存器或者更改 CHSELRMOD 或 SCANDIRW 后，必须等待至 CCRDY 标志置为有效之后才能将 ADSTART 置 1，否则会忽略写入 ADSTART 的值。

位 1 **ADDIS**：ADC 禁止命令 (ADC disable command)

该位由软件置 1，用于禁止 ADC（ADDIS 命令）并使其进入掉电状态（OFF 状态）。

ADC 已有效禁止后，会立即通过硬件将该位清零（此时也会通过硬件将 ADEN 清零）。

0：当前未执行 ADDIS 命令

1：写入 1 可禁止 ADC。读取值为 1 表示正在执行 ADDIS 命令。

注意：仅当 ADEN = 1 且 ADSTART = 0 时（这可确保当前未进行任何转换），将 ADDIS 置 1 才能生效。

位 0 **ADEN**：ADC 使能命令 (ADC enable command)

该位由软件置 1，用于使能 ADC。ADRDY 标志置 1 后，ADC 将立即准备好运行。

如果 ADC 已禁止，则执行 ADDIS 命令后，将通过硬件对该位清零。

0：禁止 ADC（OFF 状态）

1：写入 1 可使能 ADC。

注意：仅当 ADC_CR 寄存器的所有位均为 0 时（ADCAL = 0、ADSTP = 0、ADSTART = 0、ADDIS = 0 且 ADEN = 0），才允许通过软件将 ADEN 位置 1。

14.12.4 ADC 配置寄存器 1 (ADC_CFGR1)

ADC configuration register 1

偏移地址：0x0C

复位值：0x0000 0000

仅当 ADC_CR 中的 ADEN 清零时，才允许通过软件编程 ADC_CFGR1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AWD1CH[4:0]					Res.	Res.	AWD1EN	AWD1SGL	CHSELRMOD	Res.	Res.	Res.	Res.	DISCEN
	rw	rw	rw	rw	rw			rw	rw	rw					rw

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

位 31 保留，必须保持复位值。

位 30:26 **AWD1CH[4:0]**：模拟看门狗通道选择 (Analog watchdog channel selection)

这些位由软件置 1 和清零，用于选择由模拟看门狗监控的输入通道。

00000：通过 AWD 监控 ADC 模拟输入通道 0

00001：通过 AWD 监控 ADC 模拟输入通道 1

.....

10110：通过 AWD 监控 ADC 模拟输入通道 22

其他值：保留

注意：由 AWDCH[4:0] 位选择的通道也必须设置到 CHSELR 寄存器中。

位 25:24 保留，必须保持复位值。

位 23 **AWD1EN**：模拟看门狗使能 (Analog watchdog enable)

该位由软件置 1 和清零。

0：禁止模拟看门狗 **1**

1：使能模拟看门狗 **1**

注意：

位 22 **AWD1SGL**：在单一通道或所有通道上使能看门狗 (Enable the watchdog on a single channel or on all channels)

该位由软件置 1 和清零，用于使能由 AWDCH[4:0] 位确定的通道或所有通道上的模拟看门狗。

0：在所有通道上使能模拟看门狗 **1**

1：在单一通道上使能模拟看门狗 **1**

注意：

位 21 **CHSELRMOD**：ADC_CHSELR 寄存器的模式选择 (Mode selection of the ADC_CHSELR register)

该位由软件置 1 和清零，用于控制 ADC_CHSELR 功能：

0：ADC_CHSELR 寄存器的每个位用于使能一个输入

1：ADC_CHSELR 寄存器能够对多达 8 个通道进行排序

注意：如果在通道配置后 CCRDY 尚未置为有效（写入 ADC_CHSELR 寄存器或者更改 CHSELRMOD 或 SCANDIR），则写入该位的值会被忽略。

位 20:17 保留，必须保持复位值。

位 16 **DISCEN**：不连续模式 (Discontinuous mode)

该位由软件置 1 和清零，用于使能 / 禁止不连续模式。

0：禁止不连续模式

1：使能不连续模式

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

位 15 **AUTOFF**：自动关闭模式 (Auto-off mode)

该位由软件置 1 和清零，用于使能 / 禁止自动关闭模式。

0：禁止自动关闭模式

1：使能自动关闭模式

注意：

位 14 **WAIT**：等待转换模式 (Wait conversion mode)

该位由软件置 1 和清零，用于使能 / 禁止等待转换模式。

0：等待转换模式关闭

1：等待转换模式开启

注意：

位 13 **CONT**：单次 / 连续转换模式 (Single/continuous conversion mode)

该位由软件置 1 和清零。该位置 1 时，转换将持续进行，直到该位清零。

0：单次转换模式

1：连续转换模式

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

位 12 **OVRMOD**：溢出管理模式 (Overrun management mode)

该位由软件置 1 和清零，用于配置数据溢出的管理方式。

0：如果检测到溢出，ADC_DR 寄存器会保留原有数据。

1：如果检测到溢出，ADC_DR 寄存器会被上一转换结果覆盖。

注意：

位 11:10 **EXTEN[1:0]**：外部触发使能和极性选择 (External trigger enable and polarity selection)

这些位由软件置 1 和清零，用于选择外部触发极性并使能触发。

00：禁止硬件触发检测（可通过软件开始转换）

01：在上升沿执行硬件触发检测

10：在下降沿执行硬件触发检测

11：在上升沿和下降沿都执行硬件触发检测

注意：

位 9 保留，必须保持复位值。

位 8:6 **EXTSEL[2:0]**：外部触发选择 (External trigger selection)

这些位可选择用于触发转换开始的外部事件（有关详细信息，请参见*表 53：外部触发器*）：

000：TRG0

001：TRG1

010：TRG2

011：TRG3

100：TRG4

101：TRG5

110：TRG6

111：TRG7

注意：

位 5 **ALIGN**：数据对齐 (Data alignment)

该位由软件置 1 和清零，用于选择右对齐或左对齐。请参见*第 237 页的图 41：数据对齐方式和分辨率（过采样已禁止：OVSE = 0）*。

0：右对齐

1：左对齐

注意：

位 4:3 **RES[1:0]**：数据分辨率 (Data resolution)

通过软件写入这些位可选择转换的分辨率。

00：12 位

01：10 位

10：8 位

11：6 位

位 2 **SCANDIR**：扫描序列方向 (Scan sequence direction)

该位由软件置 1 和清零，用于选择扫描序列中各通道的方向。仅当 CHSELMOD 位清零时生效。

0：向前扫描（从 CHSEL0 到 CHSEL22）

1：向后扫描（从 CHSEL22 到 CHSEL0）

注意：

如果在通道配置后 CCRDY 尚未置为有效（写入 ADC_CHSELR 寄存器或者更改 CHSELRMOD 或 SCANDIR），则写入该位的值会被忽略。

位 1 **DMACFG**：直接存储器访问配置 (Direct memory access configuration)

该位由软件置 1 和清零，用于在 DMA 两种工作模式之间进行选择，仅当 DMAEN = 1 时，该位才有效。

0：选择 DMA 单次模式

1：选择 DMA 循环模式

更多详细信息，请参见*第 238 页的第 14.6.5 节：使用 DMA 管理转换的数据*

注意：

位 0 **DMAEN**：直接存储器访问使能 (Direct memory access enable)

该位由软件置 1 和清零，用于使能 DMA 请求的生成。这样便可使用 DMA 控制器自动管理转换的数据。更多详细信息，请参见*第 238 页的第 14.6.5 节：使用 DMA 管理转换的数据*。

0：禁止 DMA

1：使能 DMA

注意：

14.12.5 ADC 配置寄存器 2 (ADC_CFGR2)

ADC configuration register 2

偏移地址：0x10

复位值：0x0000 0000

仅当 ADC_CR 中的 ADEN 清零时，才允许通过软件编程 ADC_CFGR2。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CKMODE[1:0]		LFTRIG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TOVS	OVSS[3:0]				OVSR[2:0]			Res.	OVSE
						rw	rw	rw	rw	rw	rw	rw	rw		rw

位 31:30 **CKMODE[1:0]**：ADC 时钟模式 (ADC clock mode)

该位由软件置 1 和清零，用于定义为模拟 ADC 提供时钟的方式：

00：ADCCLK（异步时钟模式），在产品级生成（请参见 RCC 部分）

01：PCLK/2（同步时钟模式）

10：PCLK/4（同步时钟模式）

11：PCLK（同步时钟模式）。仅当 PCLK 的时钟占空比为 50% 时，才必须使能该配置（必须旁路 RCC 中配置的 APB 预分频器，并且系统时钟占空比必须为 50% ）。

在所有同步时钟模式下，从定时器触发到转换开始的延迟过程中，不存在抖动。

注意：仅当 ADC 已禁止时（ADCAL = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0 且 ADEN = 0），才允许通过软件对这些位执行写操作。

位 29 **LFTRIG**：低频触发模式使能 (Low frequency trigger mode enable)

该位由软件置 1 和清零。

0：禁止低频触发模式

1：使能低频触发模式

注意：仅当 ADEN 位清零时，才允许通过软件对该位执行写操作。

位 28:10 保留，必须保持复位值。

位 9 **TOVS**：已触发的过采样 (Triggered Oversampling)

该位由软件置 1 和清零。

0：会在触发后连续完成某一通道的所有过采样转换

1：某一通道的每个过采样转换都需要触发

注意：仅当 ADEN 位清零时，才允许通过软件对该位执行写操作。

位 8:5 **OVSS[3:0]**：过采样移位 (Oversampling shift)

该位由软件置 1 和清零。

0000：不移位

0001：移 1 位

0010：移 2 位

0011：移 3 位

0100：移 4 位

0101：移 5 位

0110：移 6 位

0111：移 7 位

1000：移 8 位

其他值：保留

注意：仅当 ADEN 位清零时，才允许通过软件对该位执行写操作。

位 4:2 **OVSR[2:0]**：过采样率 (Oversampling ratio)

该位域定义过采样率的数值。

000：2x

001：4x

010：8x

011：16x

100：32x

101：64x

110：128x

111：256x

注意：仅当 ADEN 位清零时，才允许通过软件对该位执行写操作。

位 1 保留，必须保持复位值。

位 0 **OVSE**：过采样器使能 (Oversampler Enable)

该位由软件置 1 和清零。

0：禁止过采样器

1：使能过采样器

注意：仅当 ADEN 位清零时，才允许通过软件对该位执行写操作。

14.12.6 ADC 采样时间寄存器 (ADC_SMPR)

ADC sampling time register

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.	SMPSEL22	SMPSEL21	SMPSEL20	SMPSEL19	SMPSEL18	SMPSEL17	SMPSEL16	SMPSEL15	SMPSEL14	SMPSEL13	SMPSEL12	SMPSEL11	SMPSEL10	SMPSEL9	SMPSEL8
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMPSEL7	SMPSEL6	SMPSEL5	SMPSEL4	SMPSEL3	SMPSEL2	SMPSEL1	SMPSEL0	Res.	SMP2[2:0]			Res.	SMP1[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw		rw	rw	rw

位 31 保留，必须保持复位值。

位 30:8 **SMPSELx**：通道 x 采样时间选择 (Channel-x sampling time selection)（x = 22 到 0）
通过软件写入这些位可定义使用的采样时间。
0：CHANNELx 的采样时间使用 SMP1[2:0] 寄存器的设置。
1：CHANNELx 的采样时间使用 SMP2[2:0] 寄存器的设置。
注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。
有关最大通道数，请参见第 14.3 节：ADC 实现。

位 7 保留，必须保持复位值。

位 6:4 **SMP2[2:0]**：采样时间选择 2 (Sampling time selection 2)
这些位由软件写入，用于选择应用于所有通道的采样时间。
000：1.5 个 ADC 时钟周期
001：3.5 个 ADC 时钟周期
010：7.5 个 ADC 时钟周期
011：12.5 个 ADC 时钟周期
100：19.5 个 ADC 时钟周期
101：39.5 个 ADC 时钟周期
110：79.5 个 ADC 时钟周期
111：160.5 个 ADC 时钟周期
注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 3 保留，必须保持复位值。

位 2:0 **SMP1[2:0]**：*采样时间选择 1 (Sampling time selection 1)*
这些位由软件写入，用于选择应用于所有通道的采样时间。
000：1.5 个 ADC 时钟周期
001：3.5 个 ADC 时钟周期
010：7.5 个 ADC 时钟周期
011：12.5 个 ADC 时钟周期
100：19.5 个 ADC 时钟周期
101：39.5 个 ADC 时钟周期
110：79.5 个 ADC 时钟周期
111：160.5 个 ADC 时钟周期
注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

14.12.7 ADC 看门狗阈值寄存器 (ADC_AWD1TR)

ADC watchdog threshold register

偏移地址：0x20

复位值：0x0FFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	HT1[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LT1[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **HT1[11:0]**：模拟看门狗 1 高阈值 （Analog watchdog 1 higher threshold）
通过软件写入这些位可为模拟看门狗定义高阈值。
请参见*第 242 页的第 14.8 节：模拟窗口看门狗*。

位 15:12 保留，必须保持复位值。

位 11:0 **LT1[11:0]**：模拟看门狗 *1* 低阈值 （Analog watchdog *1* lower threshold）
通过软件写入这些位可为模拟看门狗定义低阈值。
请参见*第 242 页的第 14.8 节：模拟窗口看门狗*。

14.12.8 ADC 看门狗阈值寄存器 (ADC_AWD2TR)

ADC watchdog threshold register

偏移地址：0x24

复位值：0x0FFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	HT2[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LT2[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **HT2[11:0]**：模拟看门狗 2 高阈值 （Analog watchdog 2 higher threshold）
通过软件写入这些位可为模拟看门狗定义高阈值。
请参见*第 242 页的第 14.8 节：模拟窗口看门狗*。

位 15:12 保留，必须保持复位值。

位 11:0 **LT2[11:0]**：模拟看门狗 2 低阈值 （Analog watchdog 2 lower threshold）
通过软件写入这些位可为模拟看门狗定义低阈值。
请参见*第 242 页的第 14.8 节：模拟窗口看门狗*。

14.12.9 ADC 通道选择寄存器 (ADC_CHSELR)

ADC channel selection register

偏移地址：0x28

复位值：0x0000 0000

同一寄存器可以在两种不同模式下使用：

– 每个 ADC_CHSELR 位使能一个输入（ADC_CFGR1 中的 CHSELRMOD = 0）。请参见本节。

– ADC_CHSELR 能够对多达 8 个通道进行排序（ADC_CFGR1 中的 CHSELRMOD = 1）。请参见下一节。

ADC_CFGR1 中的 CHSELRMOD = 0

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

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

位 31:23 保留，必须保持复位值。

位 22:0 **CHSEL[22:0]**：通道 x 选择 (Channel-x selection)

这些位由软件写入，用于定义将哪些通道纳入要转换的通道序列。有关连接到外部通道和内部源的 ADC 输入，请参见*图 33：ADC 连接*。

0：未选择输入通道 x 进行转换

1：已选择输入通道 x 进行转换

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。如果在通道配置后 CCRDY 尚未置为有效（写入 ADC_CHSELR 寄存器或者更改 CHSELRMOD 或 SCANDIR），则写入该位的值会被忽略。

14.12.10 ADC 通道选择寄存器 [复用] (ADC_CHSELR)

ADC channel selection register [alternate]

偏移地址：0x28

复位值：0x0000 0000

同一寄存器可以在两种不同模式下使用：

– 每个 ADC_CHSELR 位使能一个输入（ADC_CFGR1 中的 CHSELRMOD = 0）。请参见上一节。

– ADC_CHSELR 能够对多达 8 个通道进行排序（ADC_CFGR1 中的 CHSELRMOD = 1）。请参见本节。

ADC_CFGR1 中的 CHSELRMOD = 1：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SQ8[3:0]				SQ7[3:0]				SQ6[3:0]				SQ5[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ4[3:0]				SQ3[3:0]				SQ2[3:0]				SQ1[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 **SQ8[3:0]**：序列的第八次转换 (8th conversion of the sequence)

通过软件编程这些位，并将通道编号 (0..14) 分配给序列中的第八次转换。0b1111 表示序列结束。

当 0b1111（序列结束）编程到较低序列通道时，这些位会被忽略。

0000：CH0

0001：CH1

...

1100：CH12

1101：CH13

1110：CH14

1111：未选择通道（序列结束）

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 27:24 **SQ7[3:0]**：序列的第七次转换 (7th conversion of the sequence)

通过软件编程这些位，并将通道编号 (0..14) 分配给序列中的第七次转换。0b1111 表示序列结束。

当 0b1111（序列结束）编程到较低序列通道时，这些位会被忽略。

有关通道选择的定义，请参见 SQ8[3:0]。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 23:20 **SQ6[3:0]**：序列的第六次转换 (6th conversion of the sequence)

通过软件编程这些位，并将通道编号 (0..14) 分配给序列中的第六次转换。0b1111 表示序列结束。

当 0b1111（序列结束）编程到较低序列通道时，这些位会被忽略。

有关通道选择的定义，请参见 SQ8[3:0]。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 19:16 **SQ5[3:0]**：序列的第五次转换 (5th conversion of the sequence)

通过软件编程这些位，并将通道编号 (0..14) 分配给序列中的第五次转换。0b1111 表示序列结束。

当 0b1111（序列结束）编程到较低序列通道时，这些位会被忽略。

有关通道选择的定义，请参见 SQ8[3:0]。

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 15:12 **SQ4[3:0]**：序列的第四次转换 (4th conversion of the sequence)
通过软件编程这些位，并将通道编号 (0..14) 分配给序列中的第四次转换。0b1111 表示序列结束。
当 0b1111（序列结束）编程到较低序列通道时，这些位会被忽略。
有关通道选择的定义，请参见 SQ8[3:0]。
注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 11:8 **SQ3[3:0]**：序列的第三次转换 (3rd conversion of the sequence)
通过软件编程这些位，并将通道编号 (0..14) 分配给序列中的第三次转换。0b1111 表示序列结束。
当 0b1111（序列结束）编程到较低序列通道时，这些位会被忽略。
有关通道选择的定义，请参见 SQ8[3:0]。
注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 7:4 **SQ2[3:0]**：序列的第二次转换 (2nd conversion of the sequence)
通过软件编程这些位，并将通道编号 (0..14) 分配给序列中的第二次转换。0b1111 表示序列结束。
当 0b1111（序列结束）编程到较低序列通道时，这些位会被忽略。
有关通道选择的定义，请参见 SQ8[3:0]。
注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 3:0 **SQ1[3:0]**：序列的第一次转换 (1st conversion of the sequence)
通过软件编程这些位，并将通道编号 (0..14) 分配给序列中的第一次转换。0b1111 表示序列结束。
当 0b1111（序列结束）编程到较低序列通道时，这些位会被忽略。
有关通道选择的定义，请参见 SQ8[3:0]。
注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

14.12.11 ADC 看门狗阈值寄存器 (ADC_AWD3TR)

ADC watchdog threshold register

偏移地址：0x2C

复位值：0x0FFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	HT3[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LT3[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **HT3[11:0]**：模拟看门狗 3 高阈值 (Analog watchdog 3 higher threshold)
通过软件写入这些位可为模拟看门狗定义高阈值。
请参见*第 242 页的第 14.8 节：模拟窗口看门狗*。

位 15:12 保留，必须保持复位值。

位 11:0 **LT3[11:0]**：模拟看门狗 *3* 低阈值 (Analog watchdog *3*lower threshold)
通过软件写入这些位可为模拟看门狗定义低阈值。
请参见*第 242 页的第 14.8 节：模拟窗口看门狗*。

14.12.12 ADC 数据寄存器 (ADC_DR)

ADC data register

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

位 31:16 保留，必须保持复位值。

位 15:0 **DATA[15:0]**：转换后的数据 (Converted data)

这些位为只读。其中包含上一转换通道的转换结果。数据可以采用左对齐和右对齐，如*第 237 页的图 41：数据对齐方式和分辨率（过采样已禁止：OVSE = 0）*所示。

校准完成后，DATA[6:0] 会立即包含校准系数。

14.12.13 ADC 模拟看门狗 2 配置寄存器 (ADC_AWD2CR)

ADC analog watchdog 2 configuration register

偏移地址：0xA0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD2 CH22	AWD2 CH21	AWD2 CH20	AWD2 CH19	AWD2 CH18	AWD2 CH17	AWD2 CH16
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD2 CH15	AWD2 CH14	AWD2 CH13	AWD2 CH12	AWD2 CH11	AWD2 CH10	AWD2 CH9	AWD2 CH8	AWD2 CH7	AWD2 CH6	AWD2 CH5	AWD2 CH4	AWD2 CH3	AWD2 CH2	AWD2 CH1	AWD2 CH0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。

位 22:0 **AWD2CH[22:0]**：模拟看门狗通道选择 (Analog watchdog channel selection)

这些位由软件置 1 和清零，用于使能和选择由模拟看门狗 2 (AWD2) 监控的输入通道。

0：不通过 AWD2 监控 ADC 模拟通道 x

1：通过 AWD2 监控 ADC 模拟通道 x

注意：通过 ADC_AWD2CR 选择的通道必须也在 ADC_CHSELR 寄存器中配置。仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

14.12.14 ADC 模拟看门狗 3 配置寄存器 (ADC_AWD3CR)

ADC Analog Watchdog 3 Configuration register

偏移地址：0xA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD3 CH22	AWD3 CH21	AWD3 CH20	AWD3 CH19	AWD3 CH18	AWD3 CH17	AWD3 CH16
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD3 CH15	AWD3 CH14	AWD3 CH13	AWD3 CH12	AWD3 CH11	AWD3 CH10	AWD3 CH9	AWD3 CH8	AWD3 CH7	AWD3 CH6	AWD3 CH5	AWD3 CH4	AWD3 CH3	AWD3 CH2	AWD3 CH1	AWD3 CH0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。

位 22:0 **AWD3CH[22:0]**：模拟看门狗通道选择 (Analog watchdog channel selection)

这些位由软件置 1 和清零，用于使能和选择由模拟看门狗 3 (AWD3) 监控的输入通道。

0：不通过 AWD3 监控 ADC 模拟通道 x

1：通过 AWD3 监控 ADC 模拟通道 x

注意：通过 ADC_AWD3CR 选择的通道必须也在 ADC_CHSELR 寄存器中配置。仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

14.12.15 ADC 校准系数 (ADC_CALFACT)

ADC calibration factor

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

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

位 31:7 保留，必须保持复位值。

位 6:0 **CALFACT[6:0]**：校准系数

这些位可由硬件或软件写入。

– 校准完成后，会立即由硬件更新为校准系数。

– 软件可向这些位写入新的校准系数。如果新校准系数不同于当前存储于模拟 ADC 中的校准系数，新的转换启动后，会立即应用新校准系数。

– 校准完成后，DATA[6:0] 会立即包含校准系数。

注意：仅当 ADEN=1 时（ADC 已使能、当前未执行任何校准和转换），才能通过软件对这些位执行写操作。

14.12.16 ADC 通用配置寄存器 (ADC_CCR)

ADC common configuration register

偏移地址：0x308

复位值：0x0000 0000

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

位 31:25 保留，必须保持复位值。

位 24 保留，必须保持复位值。

位 23 **TSEN**：温度传感器使能 (Temperature sensor enable)

该位由软件置 1 和清零，用于使能 / 禁止温度传感器。

0：禁止温度传感器

1：使能温度传感器

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 22 **VREFEN**：V_{REFINT} 使能 (V_{REFINT} enable)

该位由软件置 1 和清零，用于使能 / 禁止 V_{REFINT}。

0：禁止 V_{REFINT}

1：使能 V_{REFINT}

注意：仅当 ADSTART = 0 时（这可确保当前未进行任何转换），才允许通过软件对该位执行写操作。

位 21:18 **PRESC[3:0]**：ADC 预分频器 （ADC prescaler）

由软件置 1 和清零，用于选择 ADC 的时钟频率。

0000：输入 ADC 时钟未分频

0001：输入 ADC 时钟 2 分频

0010：输入 ADC 时钟 4 分频

0011：输入 ADC 时钟 6 分频

0100：输入 ADC 时钟 8 分频

0101：输入 ADC 时钟 10 分频

0110：输入 ADC 时钟 12 分频

0111：输入 ADC 时钟 16 分频

1000：输入 ADC 时钟 32 分频

1001：输入 ADC 时钟 64 分频

1010：输入 ADC 时钟 128 分频

1011：输入 ADC 时钟 256 分频

其他值：保留

注意：仅当 ADC 已禁止时（ADCAL = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0 且 ADEN = 0），才允许通过软件对这些位执行写操作。

位 17:0 保留，必须保持复位值。

14.13 ADC 寄存器映射

下表对 ADC 寄存器进行了汇总。

表 61. ADC 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**ADC_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCRDY.	Res.	EOCAL	Res.	AWD3	AWD2	AWD1	Res.	Res.	OVR	EOS	EOC	EOSMP	ADRDY
	Reset value																			0		0		0	0	0			0	0	0	0	0
0x04	**ADC_IER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCRDYIE.	Res.	EOCALIE	Res.	AWD3IE	AWD2IE	AWD1IE	Res.	Res.	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE
	Reset value																			0		0		0	0	0			0	0	0	0	0
0x08	**ADC_CR**	ADCAL	Res.	Res.	ADVREGEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADSTP	Res.	ADSTART	ADDIS	ADEN
	Reset value	0			0																								0		0	0	0
0x0C	**ADC_CFGR1**	Res.	AWDCH[4:0]					Res.	Res.	AWD1EN	AWD1SGL	CHSELRMOD	Res.	Res.	Res.	Res.	DISCEN	AUTOFF	WAIT	CONT	OVRMOD	EXTEN[1:0]		Res.	EXTSEL [2:0]			ALIGN	RES [1:0]		SCANDIR	DMACFG	DMAEN
	Reset value		0	0	0	0	0			0	0	0					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0
0x10	**ADC_CFGR2**	CKMODE[1:0]		LFTRIG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TOVS	OVSS[3:0]				OVSR[2:0]			Res.	OVSE
	Reset value	0	0	0																				0	0	0	0	0	0	0	0	0	0
0x14	**ADC_SMPR**	Res.	SMPSEL22	SMPSEL21	SMPSEL20	SMPSEL19	SMPSEL18	SMPSEL17	SMPSEL16	SMPSEL15	SMPSEL14	SMPSEL13	SMPSEL12	SMPSEL11	SMPSEL10	SMPSEL9	SMPSEL8	SMPSEL7	SMPSEL6	SMPSEL5	SMPSEL4	SMPSEL3	SMPSEL2	SMPSEL1	SMPSEL0	Res.	SMP2 [2:0]			Res.	SMP1 [2:0]		
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0		0	0	0
0x18	Reserved	Reserved																															
0x1C	Reserved	Reserved																															
0x20	**ADC_AWD1TR**	Res.	Res.	Res.	Res.	HT1[11:0]												Res.	Res.	Res.	Res.	LT1[11:0]											
	Reset value					1	1	1	1	1	1	1	1	1	1	1	1					0	0	0	0	0	0	0	0	0	0	0	0
0x24	**ADC_AWD2TR**	Res.	Res.	Res.	Res.	HT2[11:0]												Res.	Res.	Res.	Res.	LT2[11:0]											
	Reset value					1	1	1	1	1	1	1	1	1	1	1	1					0	0	0	0	0	0	0	0	0	0	0	0
0x28	**ADC_CHSELR** (CHSELRMOD= 0)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHSEL22	CHSEL21	CHSEL20	CHSEL19	CHSEL18	CHSEL17	CHSEL16	CHSEL15	CHSEL14	CHSEL13	CHSEL12	CHSEL11	CHSEL10	CHSEL9	CHSEL8	CHSEL7	CHSEL6	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0
	Reset value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**ADC_CHSELR** (CHSELRMOD= 1)	SQ8[3:0]				SQ7[3:0]				SQ6[3:0]				SQ5[3:0]				SQ4[3:0]				SQ3[3:0]				SQ2[3:0]				SQ1[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**ADC_AWD3TR**	Res.	Res.	Res.	Res.	HT3[11:0]												Res.	Res.	Res.	Res.	LT3[11:0]											
	Reset value					1	1	1	1	1	1	1	1	1	1	1	1					0	0	0	0	0	0	0	0	0	0	0	0
0x30 0x34 0x38 0x3C	Reserved	Reserved																															

表 61. ADC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x40	**ADC_DR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	Reserved	Reserved																															
0xA0	**ADC_AWD2CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD2CH22	AWD2CHL21	AWD2CH20	AWD2CH19	AWD2CH18	AWD2CH17	AWD2CH16	AWD2CH15	AWD2CH14	AWD2CH13	AWD2CH12	AWD2CH11	AWD2CH10	AWD2CH9	AWD2CH8	AWD2CH7	AWD2CH6	AWD2CH5	AWD2CH4	AWD2CH3	AWD2CH2	AWD2CH1	AWD2CH0
	Reset value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA4	**ADC_AWD3CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD3CH22	AWD3CHL21	AWD3CH20	AWD3CH19	AWD3CH18	AWD3CH17	AWD3CH16	AWD3CH15	AWD3CH14	AWD3CH13	AWD3CH12	AWD3CH11	AWD3CH10	AWD3CH9	AWD3CH8	AWD3CH7	AWD3CH6	AWD3CH5	AWD3CH4	AWD3CH3	AWD3CH2	AWD3CH1	AWD3CH0
	Reset value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	Reserved	Reserved																															
0xB4	**ADC_CALFACT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT[6:0]						
	Reset value																										0	0	0	0	0	0	0
...	Reserved	Reserved																															
0x308	**ADC_CCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSEN	VREFEN	PRESC3	PRESC2	PRESC1	PRESC0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value									0	0	0	0	0	0																		

有关寄存器边界地址的信息，请参见*第 2.2 节*。

15 高级控制定时器 (TIM1)

在本章中，应将“TIMx”理解为“TIM1”，因为本参考手册所适用的产品中只有一个此类定时器的实例。

15.1 TIM1 简介

高级控制定时器 (TIM1) 包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

此类定时器可用于多种用途，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输出比较、PWM 和带死区插入的互补 PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

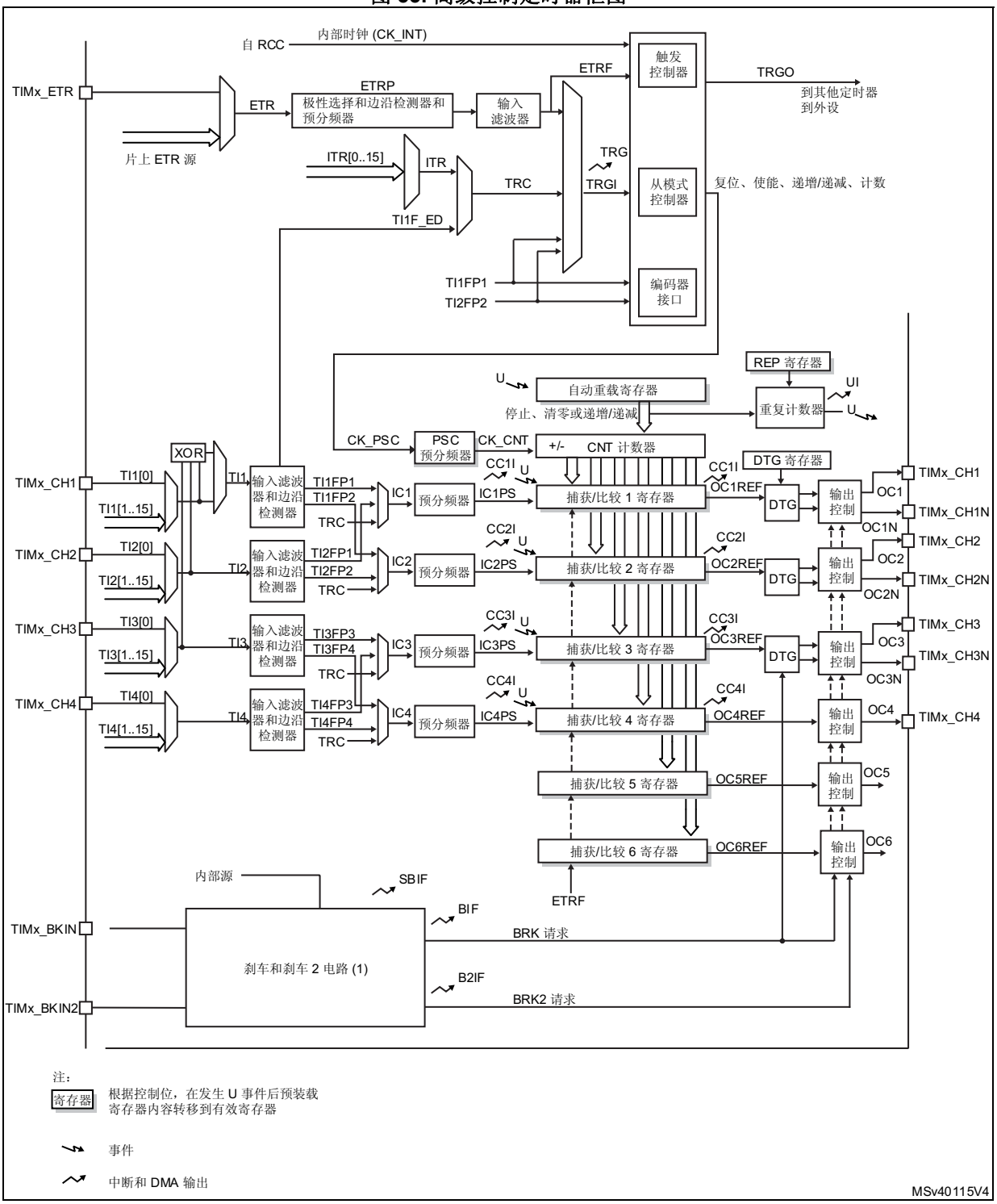
高级控制定时器 (TIM1) 和通用定时器 (TIMy) 彼此完全独立，不共享任何资源。
如*第 15.3.26 节：定时器同步*中所述，它们可以一起同步操作。

15.2 TIM1 主要特性

TIM1 定时器主要特性：

- 16 位递增、递减、递增/递减自动重载计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 到 65536 之间。
- 多达 6 个独立通道，可用于：
  - 输入捕获（但通道 5 和通道 6 除外）
  - 输出比较
  - PWM 生成（边沿和中心对齐模式）
  - 单脉冲模式输出
- 带可编程死区的互补输出
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路。
- 重复计数器，用于仅在给定数目的计数周期后更新定时器寄存器。
- 2 个刹车输入，用于将定时器的输出信号置于用户选择的安全配置中。
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较
- 支持定位用增量（正交）编码器和霍尔传感器电路
- 触发输入可用作外部时钟或用于逐周期电流管理

图 55. 高级控制定时器框图



1. 内部刹车事件源可以是：
   - CSS 生成的时钟故障事件。有关 CSS 的详细信息，请参见*第 5.2.6 节：时钟安全系统 (CSS)*
   - SRAM 奇偶校验错误信号
   - Cortex®-M0+LOCKUP (Hardfault) 输出。

15.3 TIM1 功能描述

15.3.1 时基单元

可编程高级控制定时器的主要模块是一个 16 位计数器及其相关的自动重载寄存器。计数器可递增计数、递减计数或交替进行递增和递减计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)
- 重复计数器寄存器 (TIMx_RCR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以实时传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将产生更新事件。该更新事件也可由软件产生。下文将详细介绍每个配置中更新事件的产生方式。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意，计数器将在 TIMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器说明

预分频器可对计数器时钟进行分频，分频系数介于 1 和 65536 之间。该分频器基于 16 位寄存器 TIMx_PSC 所控制的 16 位计数器而工作。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。新的预分频比将在下一更新事件发生时被采用。

图 56 和*图 57* 以一些示例说明在预分频比实时变化时计数器的行为：

图 56. 预分频器分频由 1 变为 2 时的计数器时序图

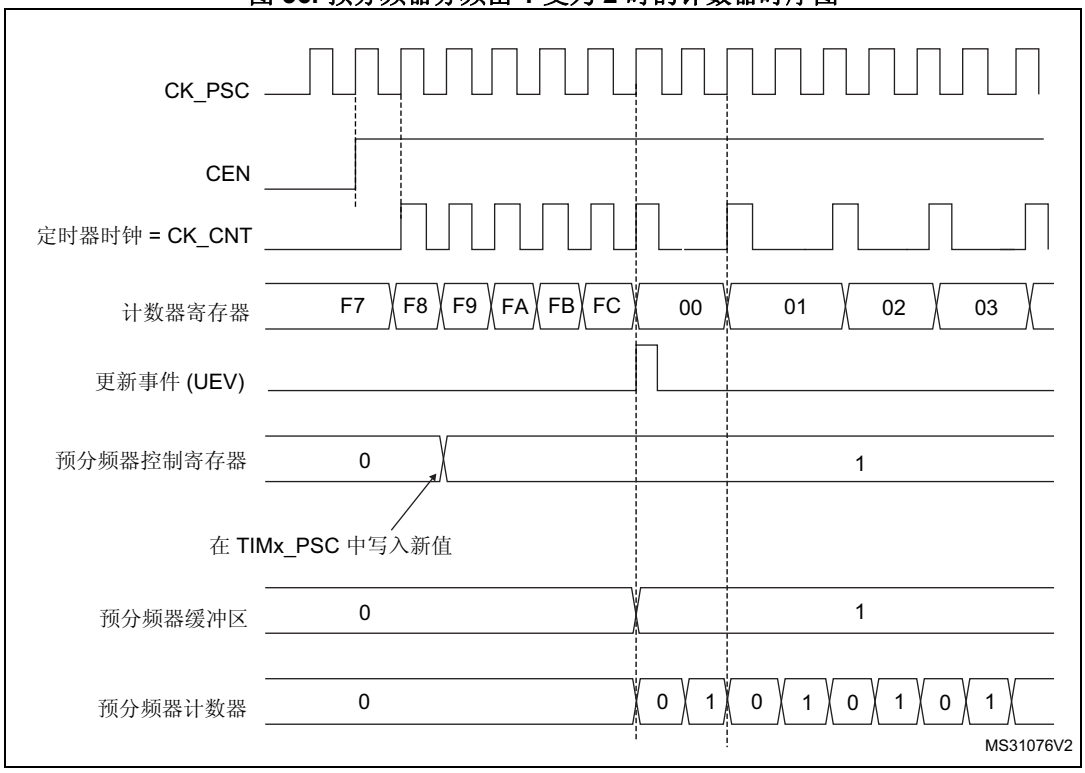
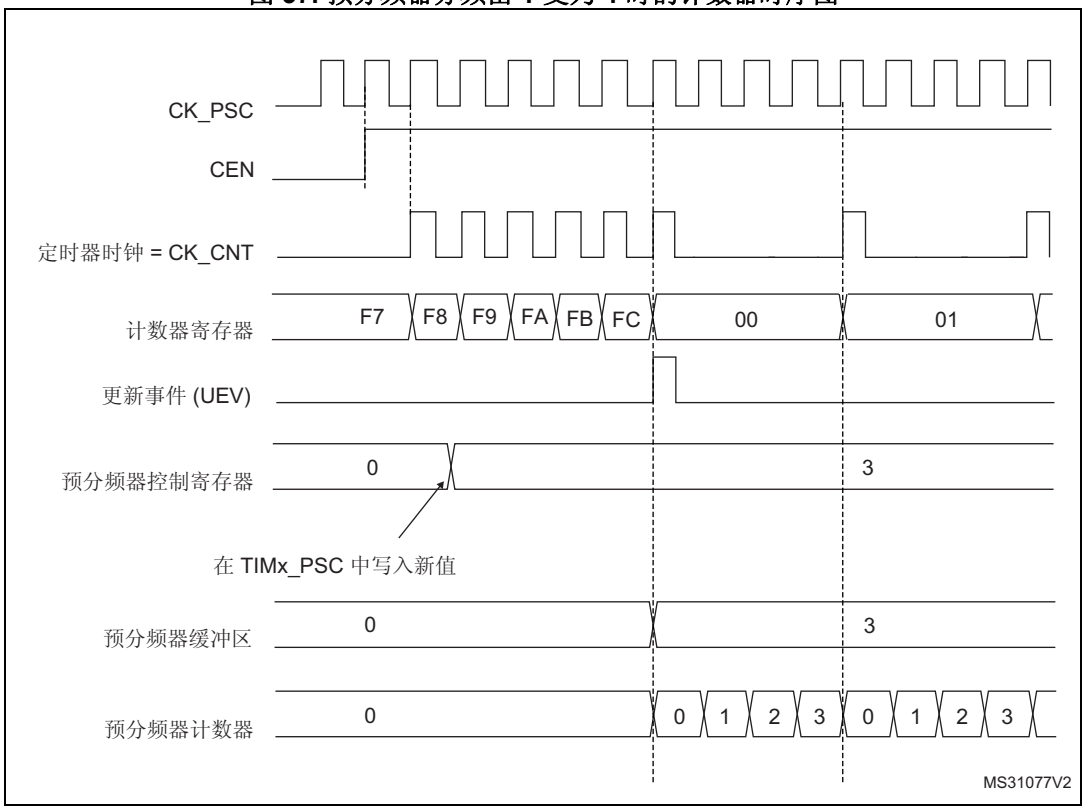


图 57. 预分频器分频由 1 变为 4 时的计数器时序图



15.3.2 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

对于带重复计数器的计数单元，当递增计数的重复次数达到重复计数器寄存器中设定的次数加一次（(TIMx_RCR) + 1）时，将生成更新事件 (UEV)。否则，将在每次计数器上溢时产生更新事件。注：TIMx_RCR寄存器的复位默认值为0。

将 TIMx_EGR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。利用这一特性，在定时器的输入捕获操作时可以通过对 UG 位置 1 让计数器清零而不至于触发更新中断，也就避免了在捕获中断里触发更新中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容，
- 自动重载影子寄存器将以预装载值 (TIMx_ARR) 进行更新，
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 58. 计数器时序图，1 分频内部时钟

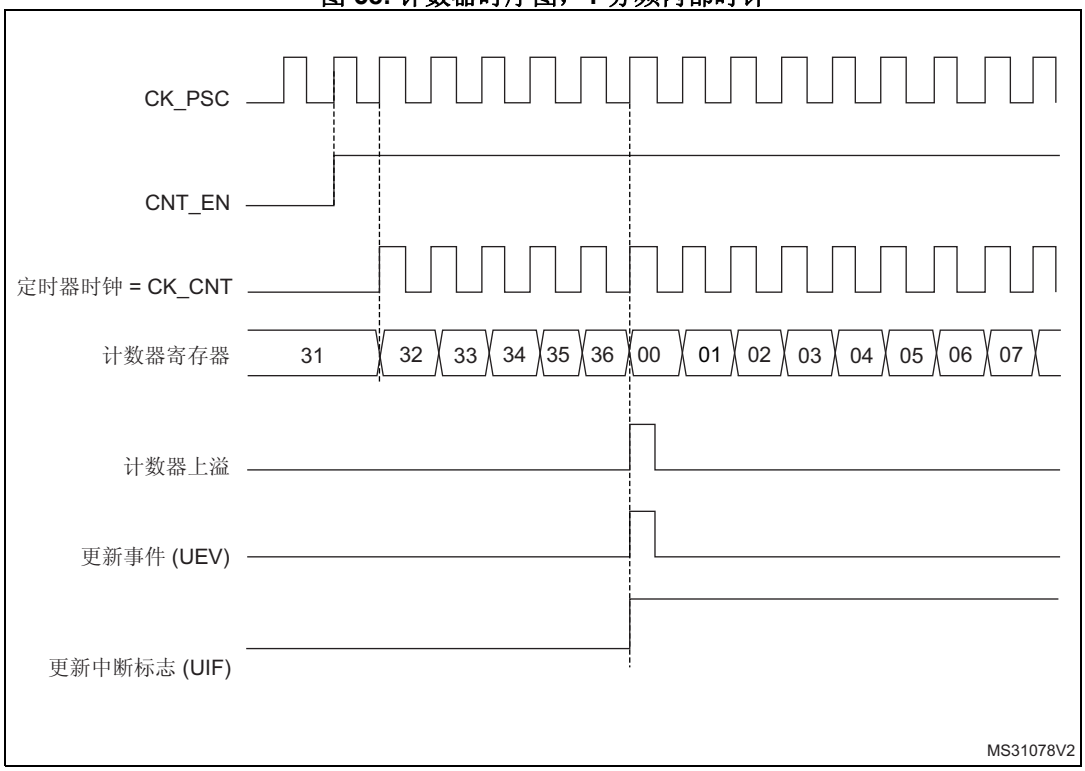


图 59. 计数器时序图，2 分频内部时钟

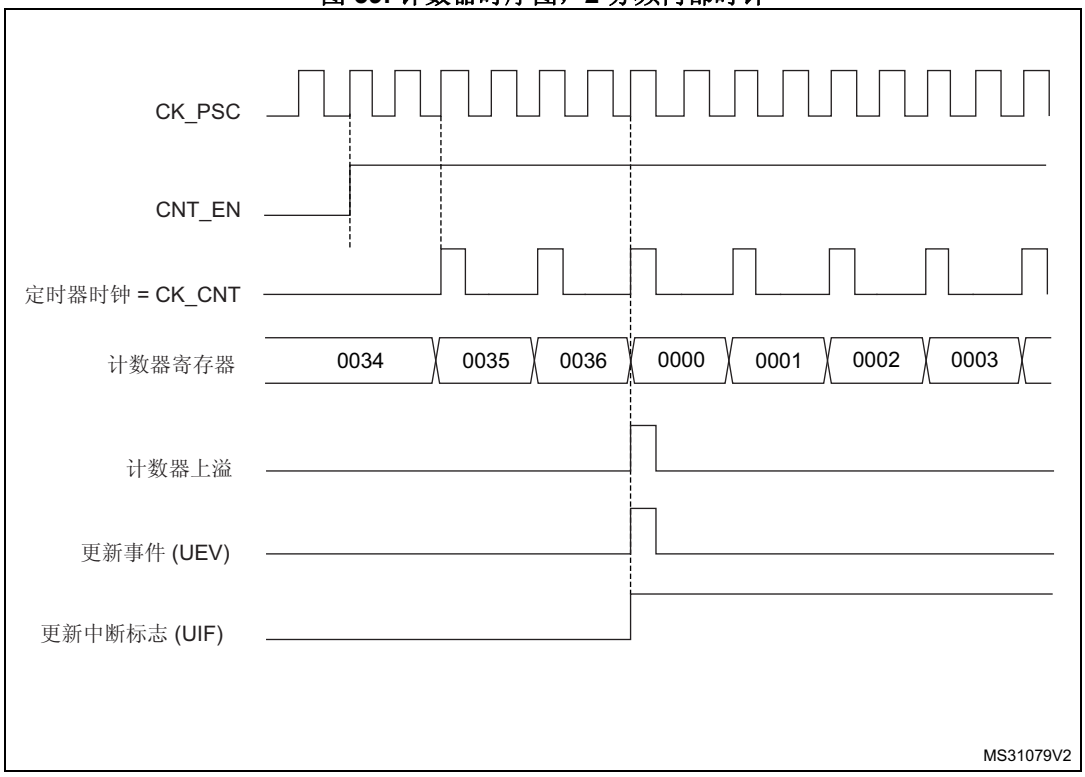


图 60. 计数器时序图，4 分频内部时钟

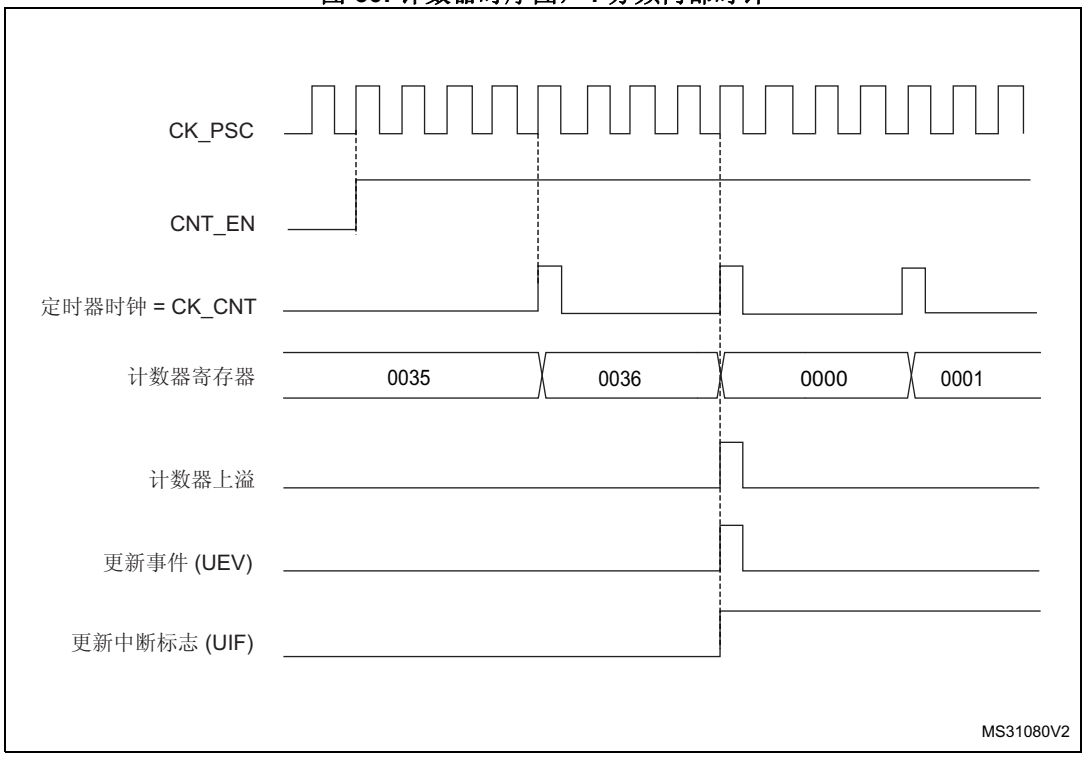


图 61. 计数器时序图，N 分频内部时钟

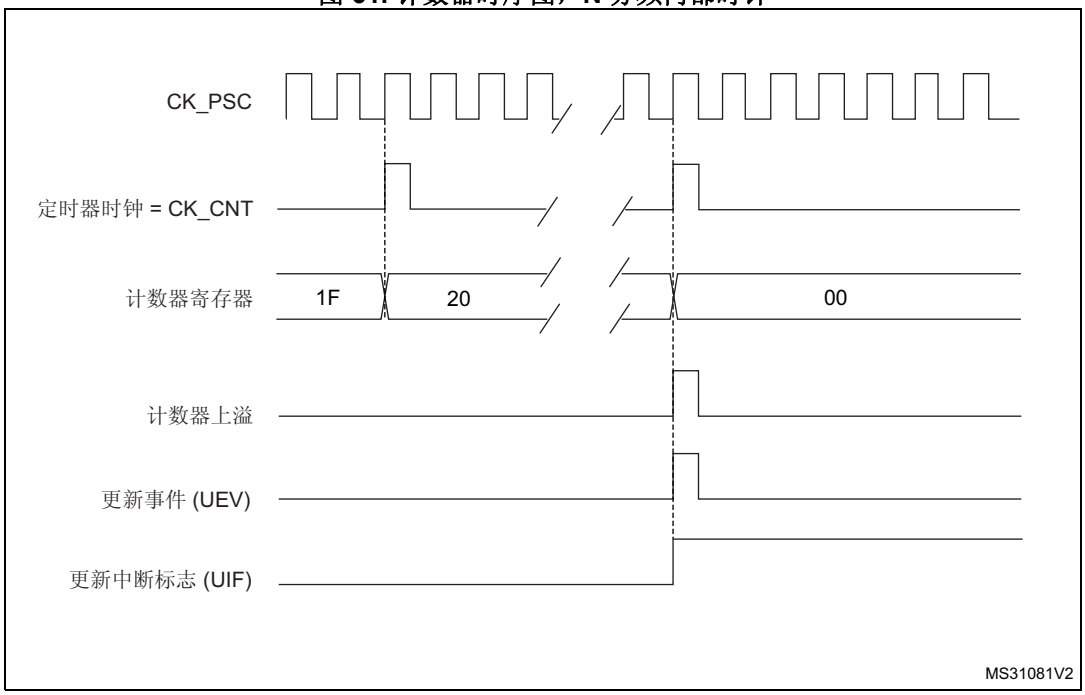


图 62. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

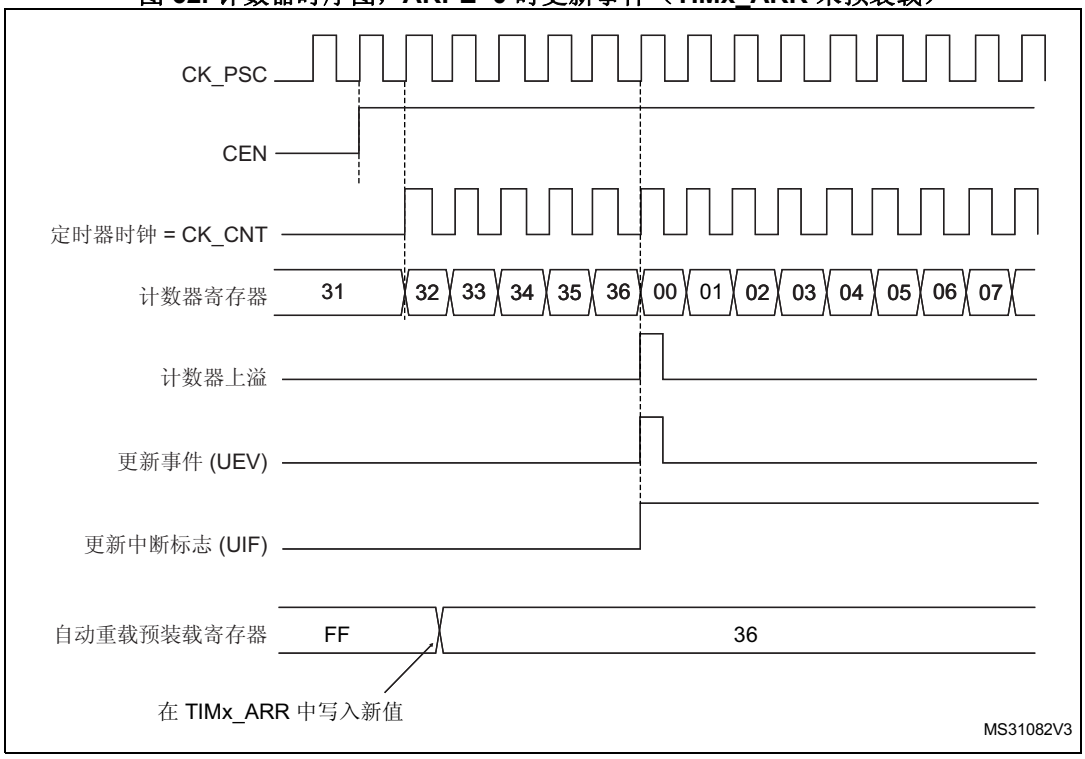
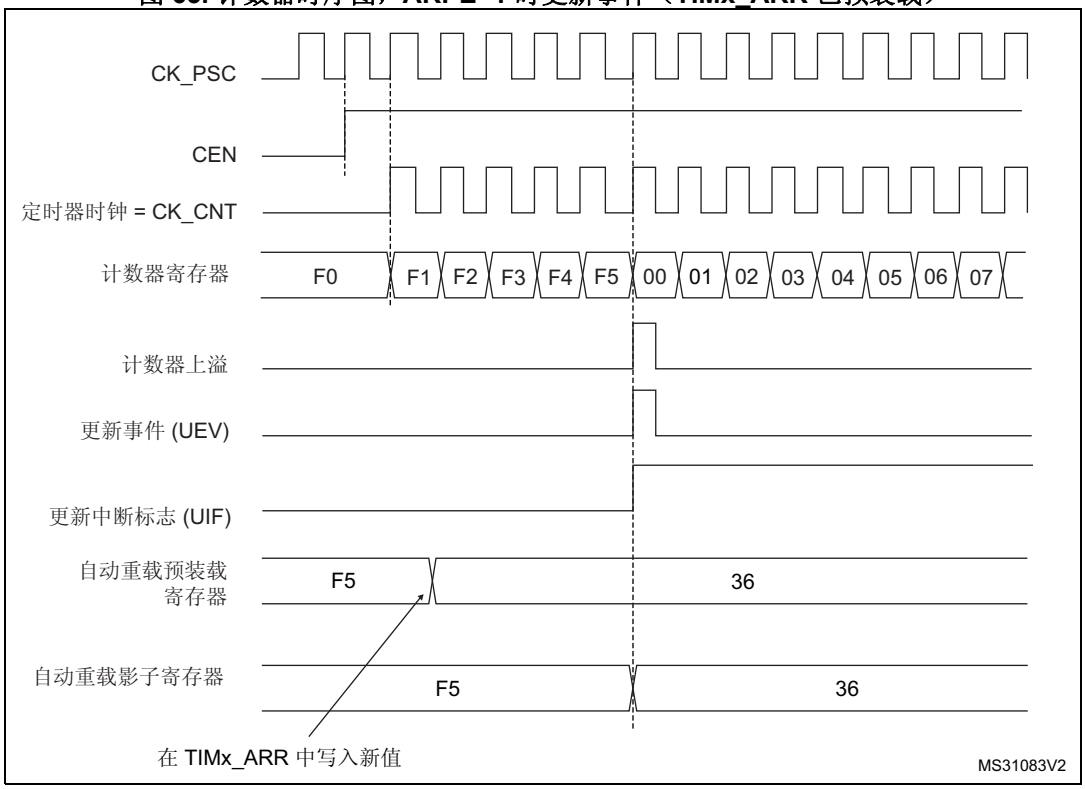


图 63. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载）



递减计数模式

在递减计数模式下，计数器从自动重载值（TIMx_ARR 寄存器的内容）开始递减计数到 0，然后重新从自动重载值开始计数并生成计数器下溢事件。

对于带重复计数器的计数单元，当递减计数的重复次数达到重复计数器寄存器中编程的次数加一次（(TIMx_RCR) + 1）时，将产生更新事件 (UEV)。否则，将在每次计数器下溢时产生更新事件。注：TIMx_RCR寄存器的复位默认值为0。

将 TIMx_EGR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会从当前自动重载值开始重新计数，而预分频器计数器则从 0 开始重新计数（但预分频比保持不变）。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。利用这一特性，在定时器的输入捕获操作时可以通过对UG位置1让计数器清零而不至于触发更新中断，也就避免了在捕获中断里触发更新中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容。
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载影子寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，ARR 寄存器更新在计数器重载之前被更新，因此下一个周期就是预期的值。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 64. 计数器时序图，1 分频内部时钟

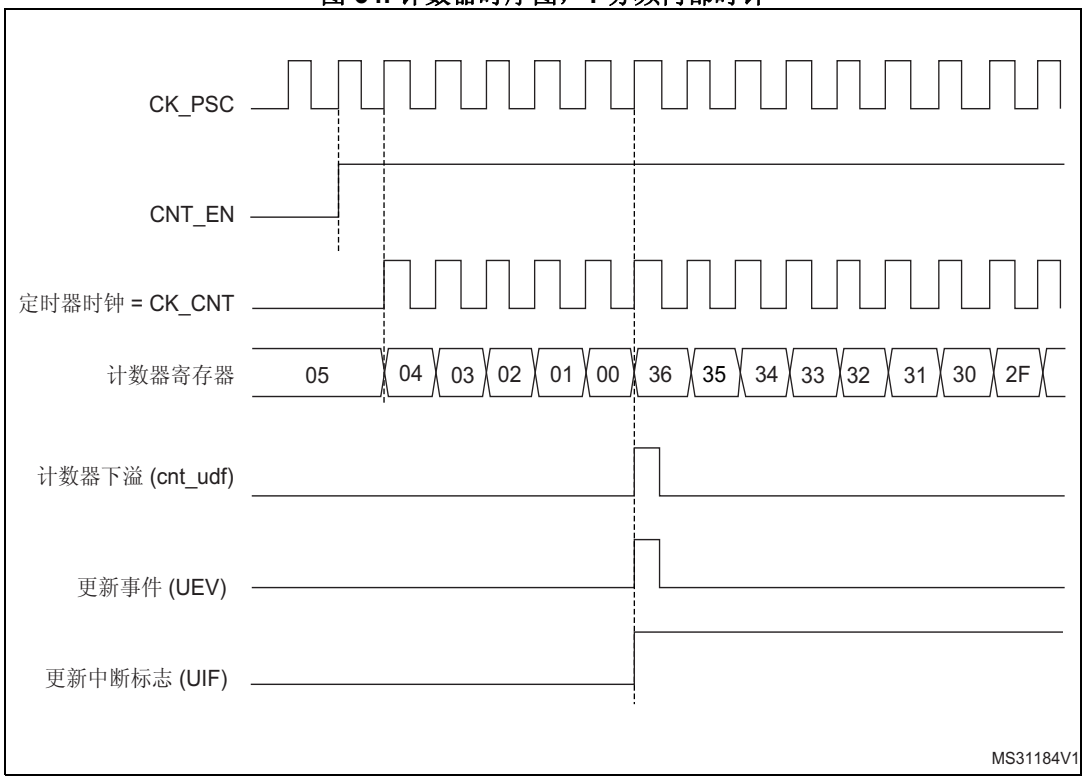


图 65. 计数器时序图，2 分频内部时钟

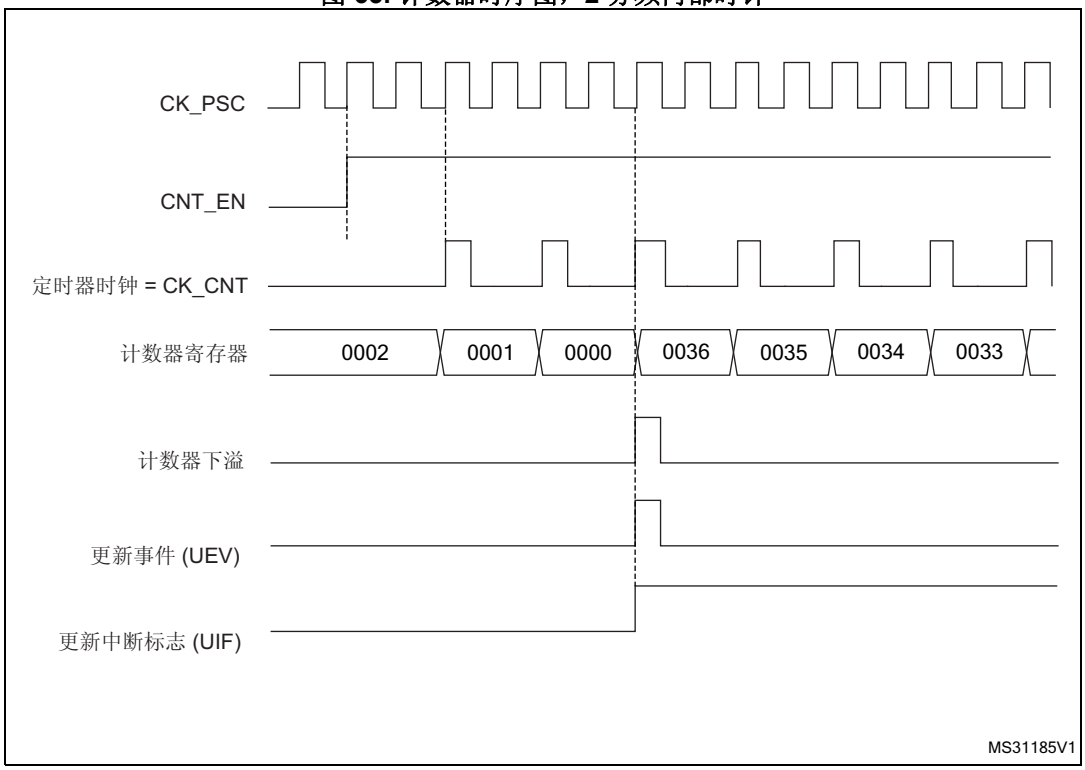


图 66. 计数器时序图，4 分频内部时钟

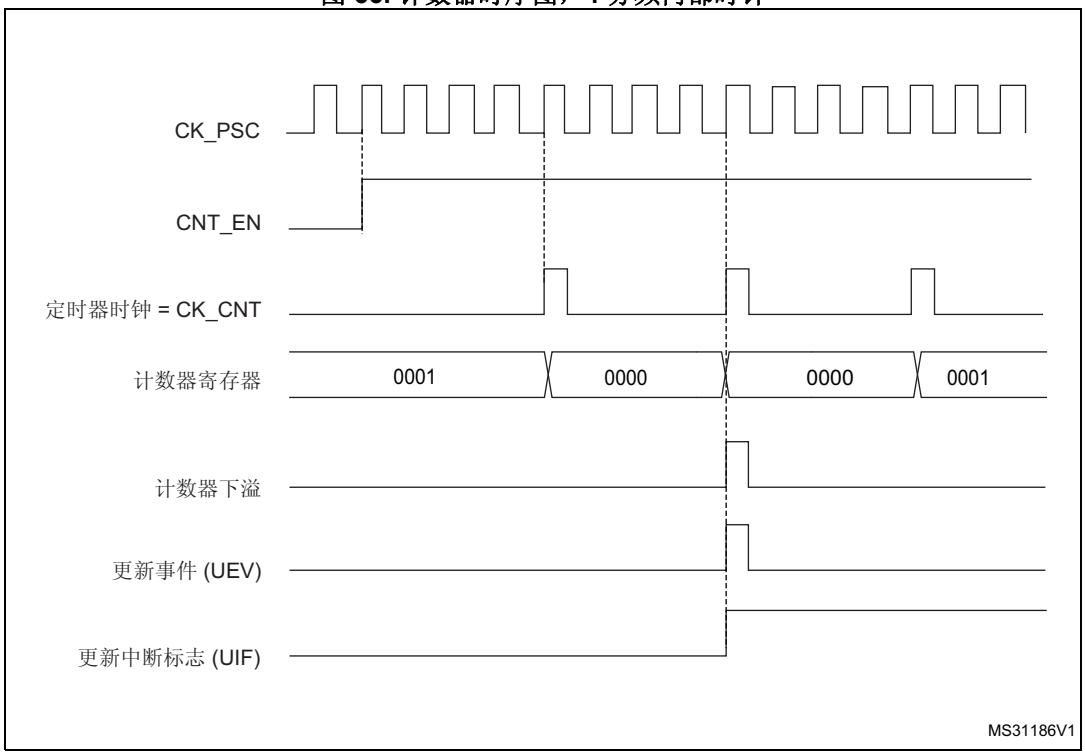


图 67. 计数器时序图，N 分频内部时钟

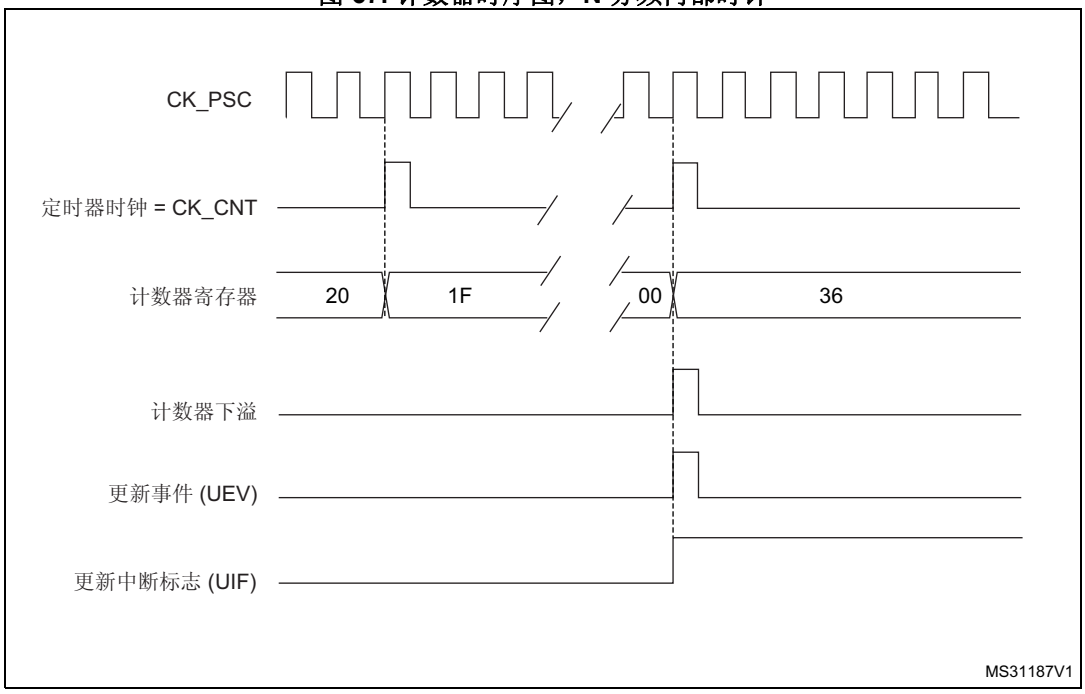
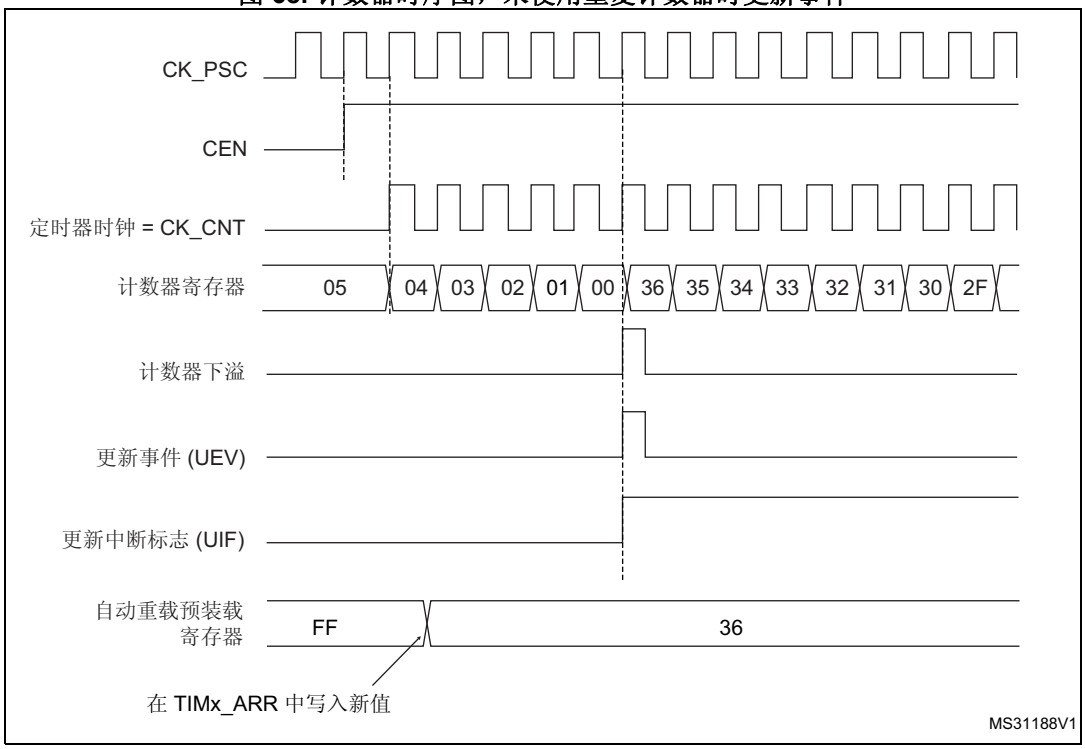


图 68. 计数器时序图，未使用重复计数器时更新事件



中心对齐模式（递增/递减计数）

在中心对齐模式下，计数器从 0 开始计数到自动重载值（TIMx_ARR 寄存器的内容）– 1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中心对齐模式 1，CMS = “01”）、计数器递增计数（中心对齐模式 2，CMS = “10”）以及计数器递增/递减计数（中心对齐模式 3，CMS = “11”）。

在此模式下，TIMx_CR1 寄存器的 DIR 方向位不可写入值，而是由硬件更新并指示当前计数器方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。

UEV 更新事件可通过软件禁止，只需将 TIMx_CR1 寄存器中的 UDIS 位置 1。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

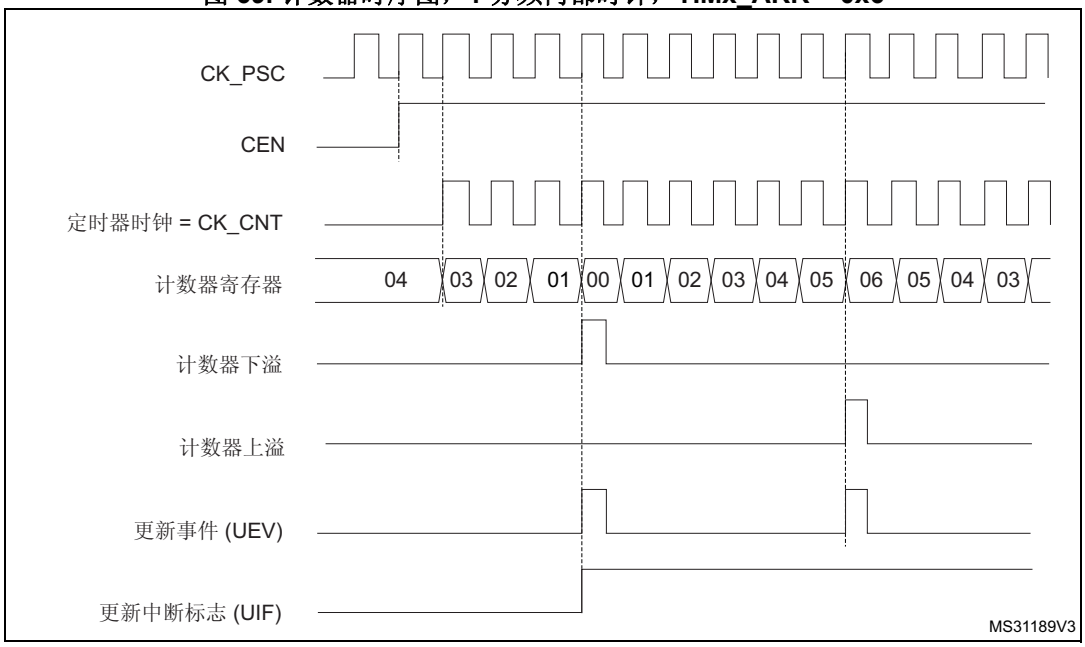
此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成 UEV 更新事件，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。利用这一特性，在定时器的输入捕获操作时可以通过对UG位置1让计数器清零而不至于触发更新中断，也就避免了在捕获中断里触发更新中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）
- 自动重载影子寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，如果更新操作是由计数器上溢触发的，则 ARR 寄存器在计数器重载之前更新，因此，下一个计数周期就是我们所希望的新的周期长度（计数器被重载新的值）。

以下各图以一些示例说明不同时钟频率下计数器的行为。

图 69. 计数器时序图，1 分频内部时钟，TIMx_ARR = 0x6



1. 此处使用中心对齐模式 1（有关详细信息，请参见*第 15.4 节：TIM1 寄存器*）。

图 70. 计数器时序图，2 分频内部时钟

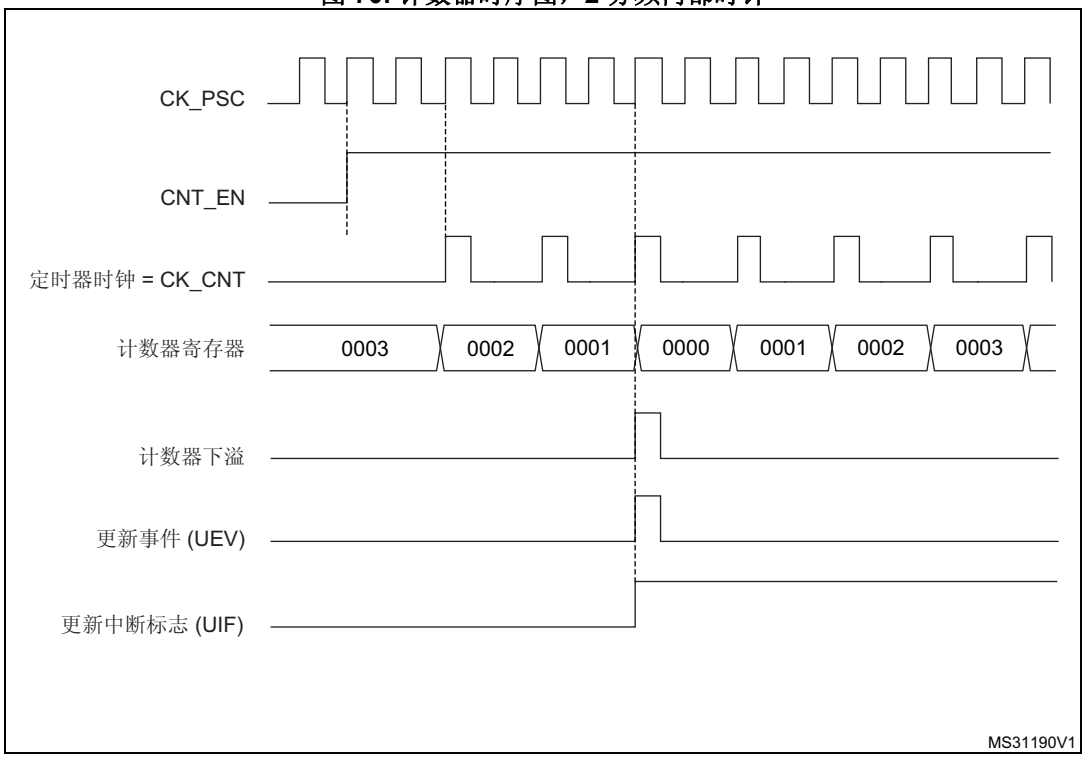


图 71. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36

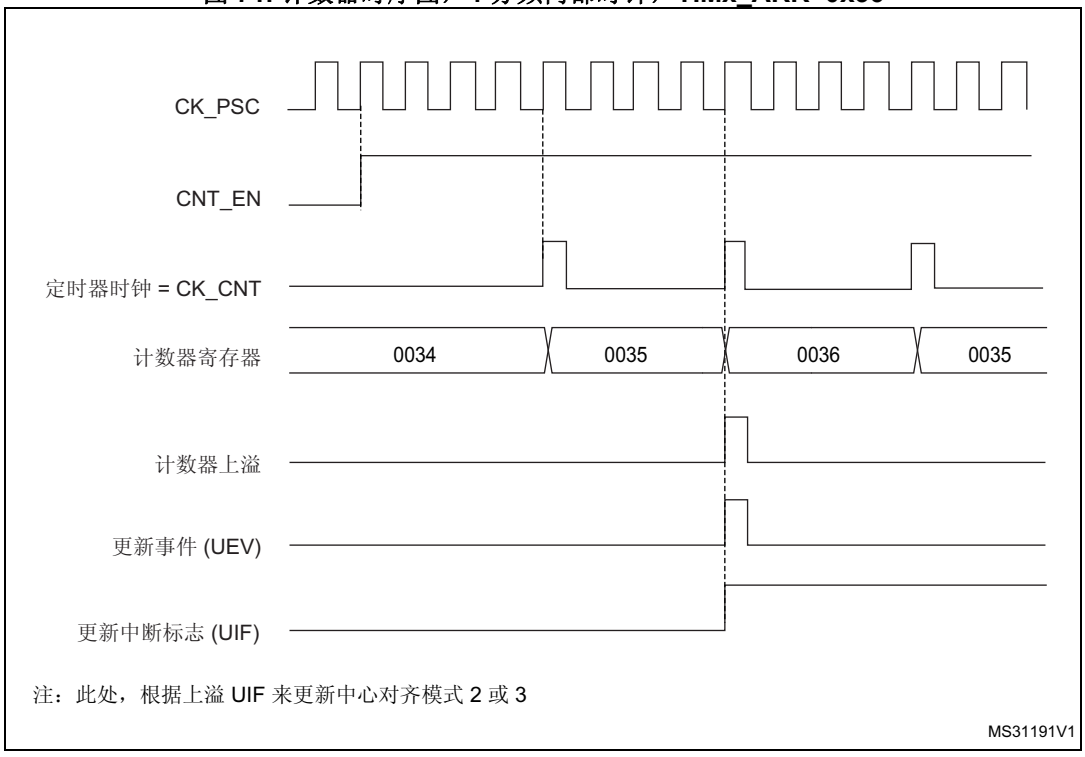


图 72. 计数器时序图，N 分频内部时钟

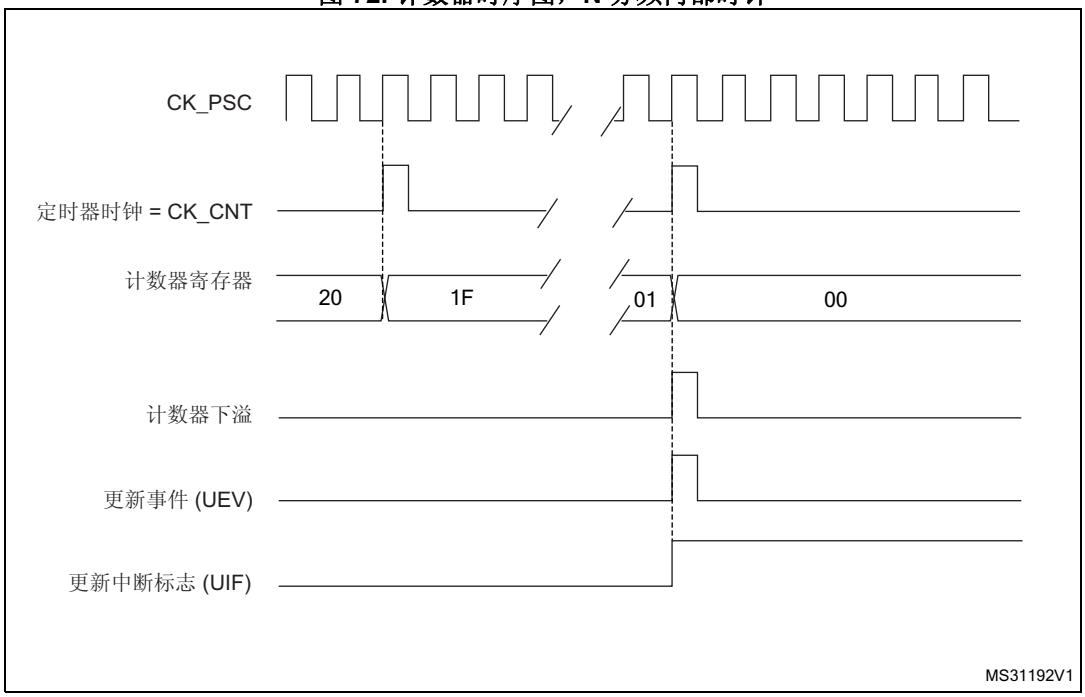


图 73. 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

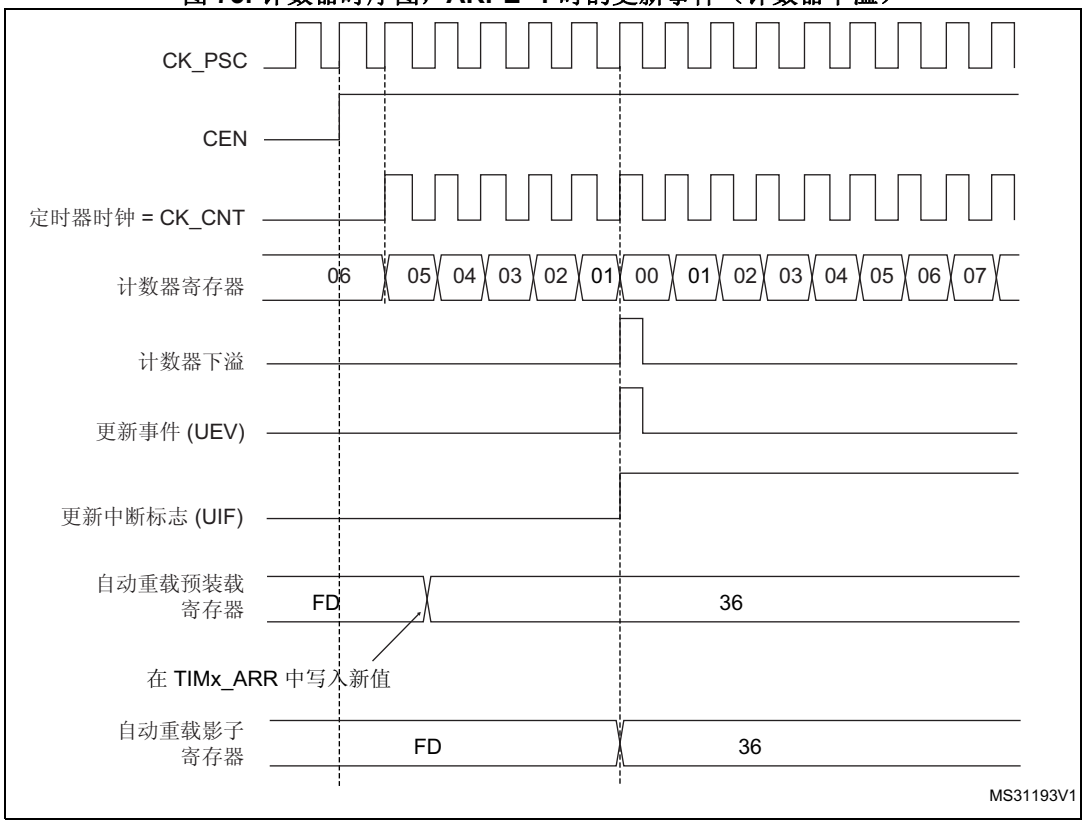
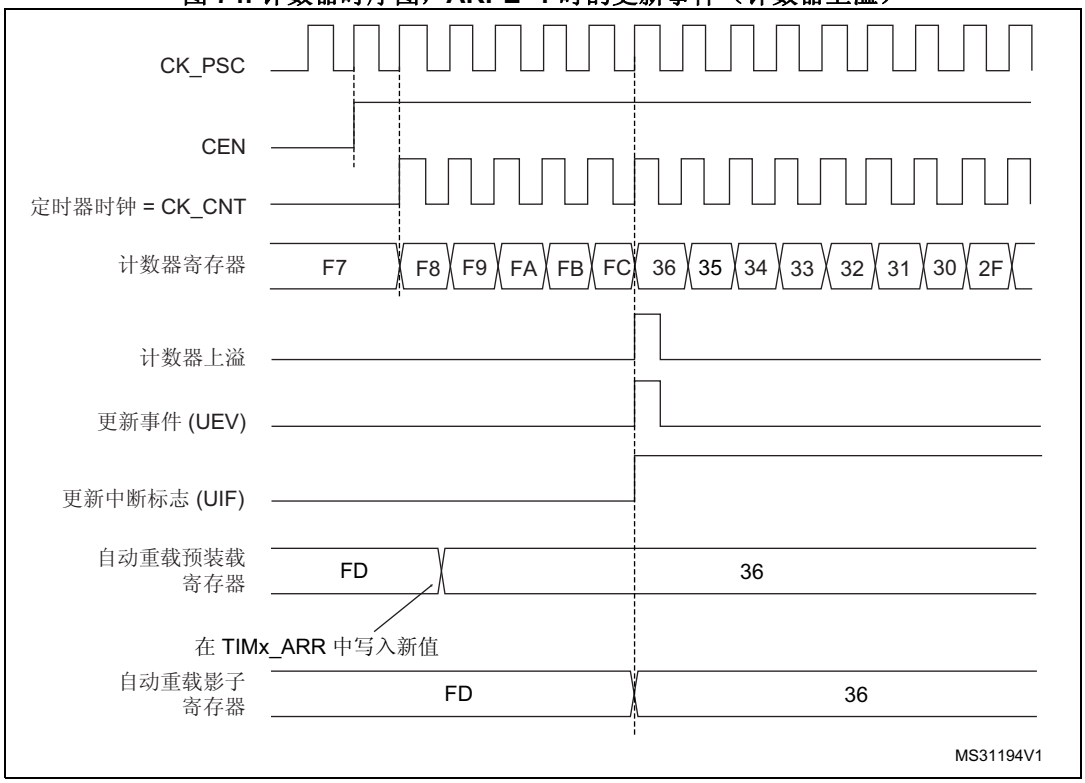


图 74. 计数器时序图，ARPE=1 时的更新事件（计数器上溢）



15.3.3 重复计数器

*第 15.3.1 节：时基单元*介绍如何因计数器上溢/下溢而生成更新事件 (UEV)。实际上，只有当重复计数器达到零时，才会生成更新事件。这在生成 PWM 信号时很有用。

这意味着，每当发生 N+1 个计数器上溢或下溢（其中，N 是 TIMx_RCR 重复计数器寄存器中的值），数据就将从预装载寄存器转移到影子寄存器（TIMx_ARR 自动重载寄存器、TIMx_PSC 预分频器寄存器以及比较模式下的 TIMx_CCRx 捕获/比较寄存器）。

重复计数器在下列情况下递减：

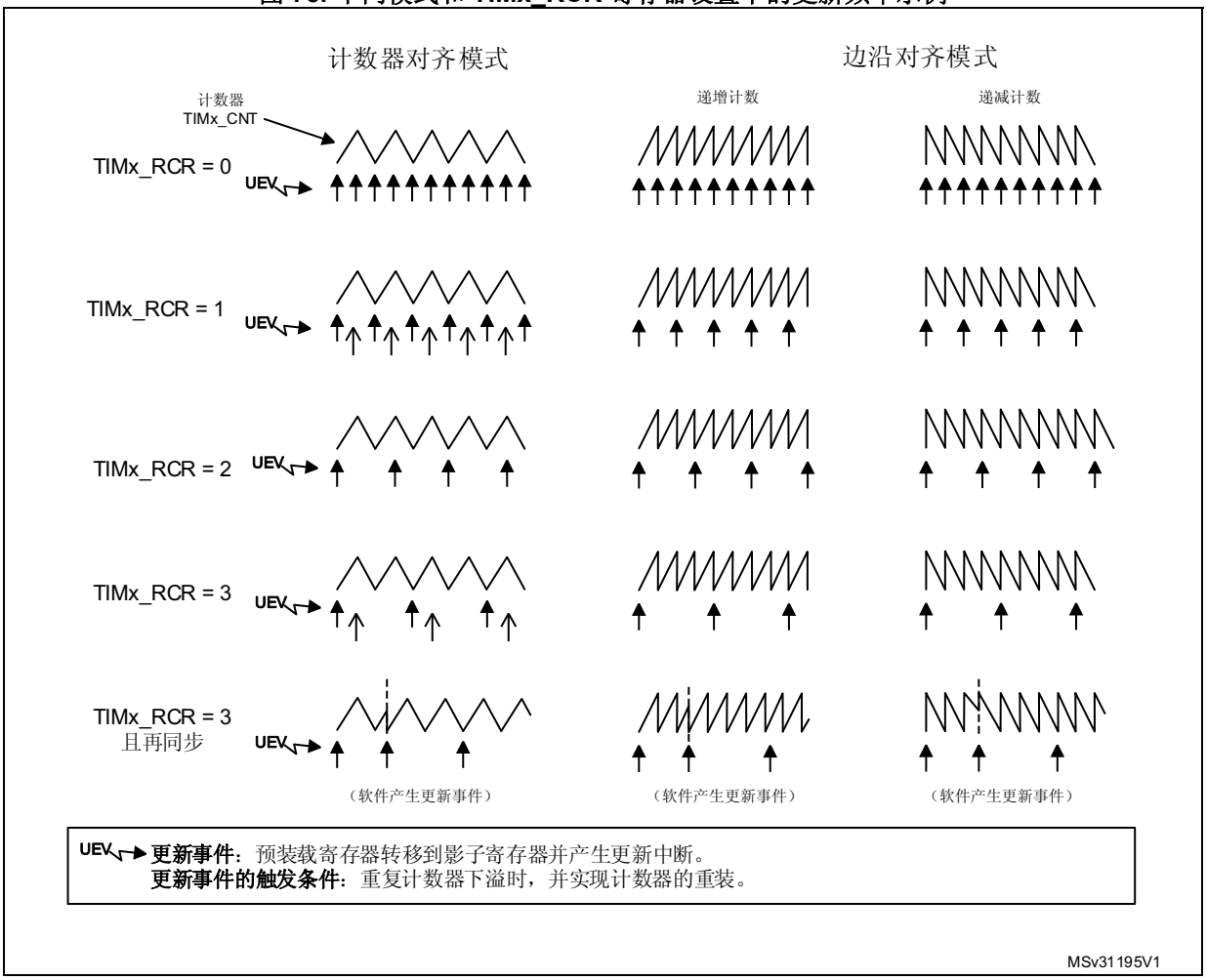
- 递增计数模式下的每个计数器上溢，
- 递减计数模式下的每个计数器下溢，
- 中心对齐模式下计数器的每个上溢或下溢动作。尽管这使得最大重复次数不超过 32768 个 PWM 周期，但在每个 PWM 周期内可更新占空比两次。当在中心对齐模式下，每个 PWM 周期仅刷新一次比较寄存器时，由于模式的对称性，最大分辨率为 $2xT_{ck}$。

重复计数器是自动重载类型；其重复率为 TIMx_RCR 寄存器所定义的值（请参见*图 75*）。当更新事件由软件（通过将 TIMx_EGR 寄存器的 UG 位置 1）或硬件（通过从模式控制器）生成时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 TIMx_RCR 寄存器的内容。

在中心对齐模式下，如果 RCR 值为奇数，更新事件将在上溢或下溢时发生，这取决于何时写入 RCR 寄存器以及何时启动计数器：如果在启动计数器前写入 RCR，则 UEV 在下溢时发生。如果在启动计数器后写入 RCR，则 UEV 在上溢时发生。

例如，如果 RCR = 3，UEV 将在每个周期的第四个上溢或下溢事件时产生（取决于何时写入 RCR）。

图 75. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例



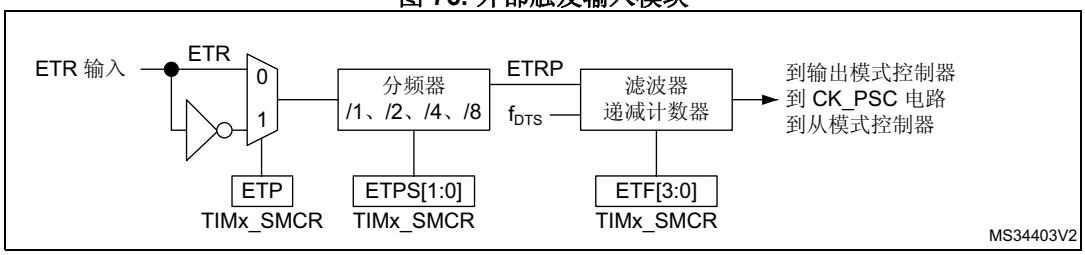
15.3.4 外部触发输入

定时器具有一个外部触发输入 ETR。它的用途是：

- 外部时钟（外部时钟模式 2，请参见*第 15.3.5 节*）
- 用于从模式的触发信号（请参见*第 15.3.26 节*）
- 用于逐周期电流调节的 PWM 复位输入（请参见*第 15.3.7 节*）

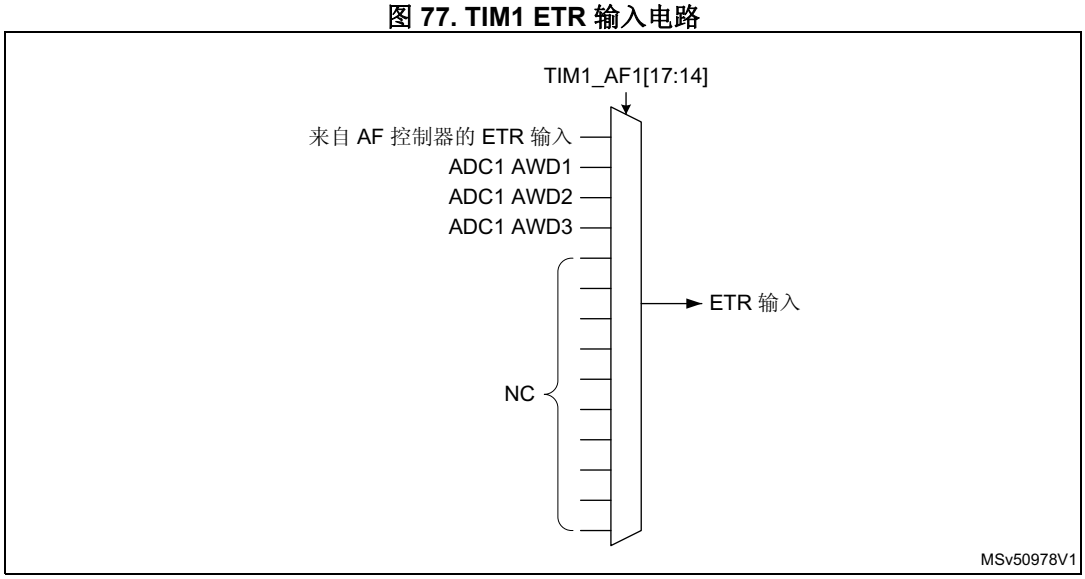
下面的*图 76* 介绍了 ETR 输入的调节过程。输入极性通过 TIMx_SMCR 寄存器中的 ETP 位定义。触发信号可通过 ETPS[1:0] 位域编程的分频比进行预分频，然后通过 ETF[3:0] 位域设置的滤波方式进行数字滤波。

图 76. 外部触发输入模块



ETR 输入来自多个源：输入引脚（默认配置）和模拟看门狗。使用 ETRSEL[3:0] 位域进行选择。

图 77. TIM1 ETR 输入电路



15.3.5 时钟选择

计数器时钟可由下列时钟源提供：

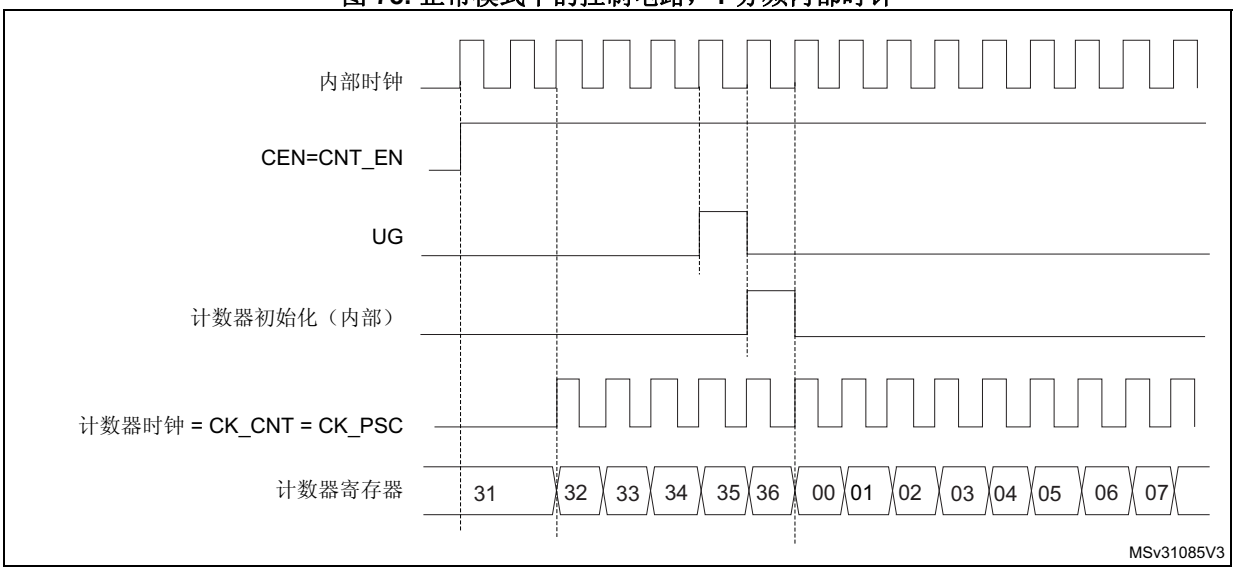
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 编码器模式

内部时钟源 (CK_INT)

如果禁止从模式控制器 (SMS=000)，则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍保持自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 78 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

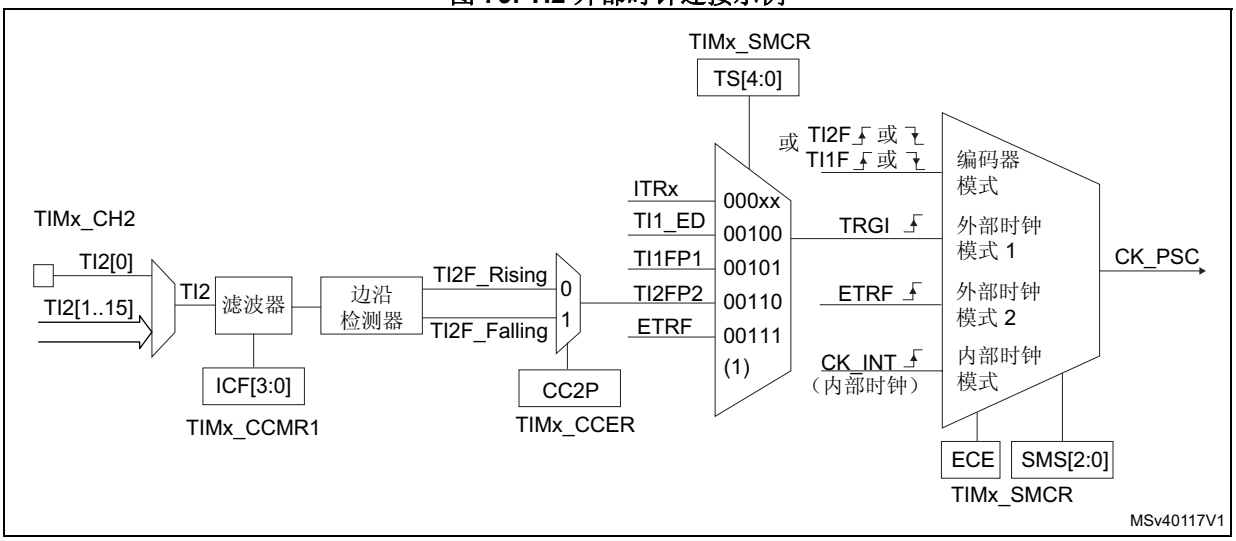
图 78. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 79. TI2 外部时钟连接示例



1. 保留从 01000 到 11111 的编码

例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

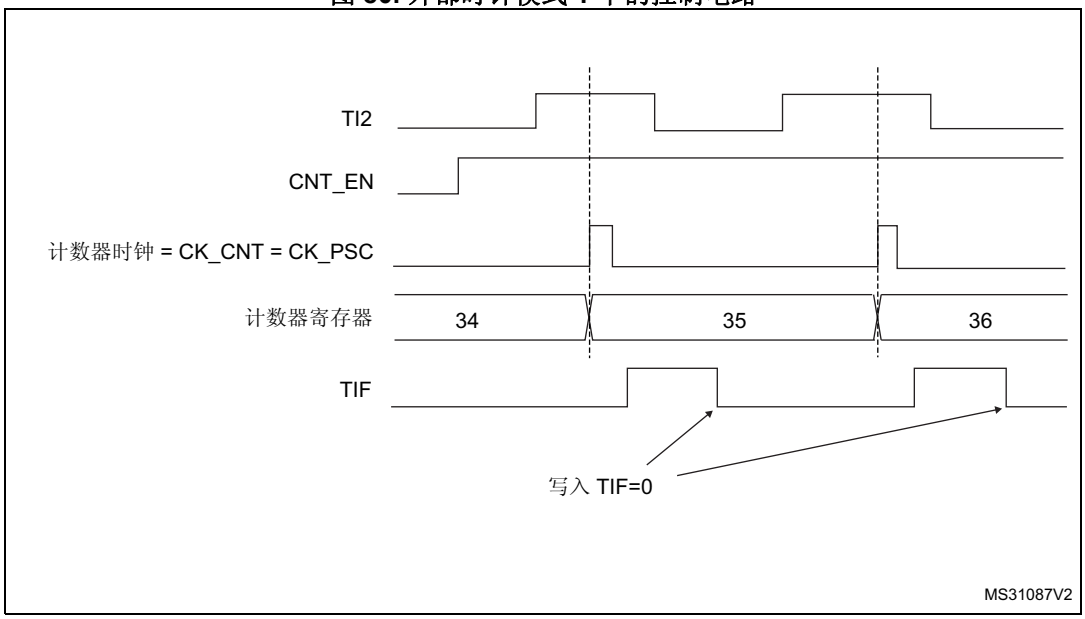
1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2x 源（内部或外部）。
2. 通过在 TIMx_CCMR1 寄存器中写入 CC2S = “01”来配置通道 2，使其能够检测 TI2 输入的上升沿。
3. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。
4. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
5. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
6. 通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为触发输入源。
7. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

注： *由于捕获预分频器不用于触发操作，因此用户无需对其进行配置。*

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 80. 外部时钟模式 1 下的控制电路



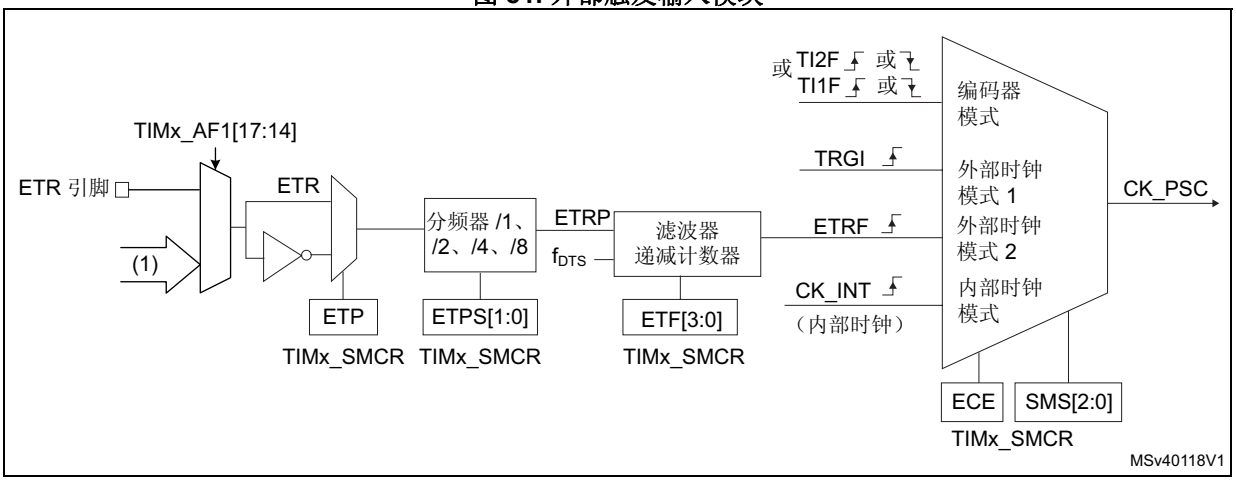
外部时钟源模式 2

通过在 TIMx_SMCR 寄存器中写入 ECE=1 可选择此模式。

计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

图 81 简要介绍了外部触发输入模块。

图 81. 外部触发输入模块



1. 请参见*图 77：TIM1 ETR 输入电路*。

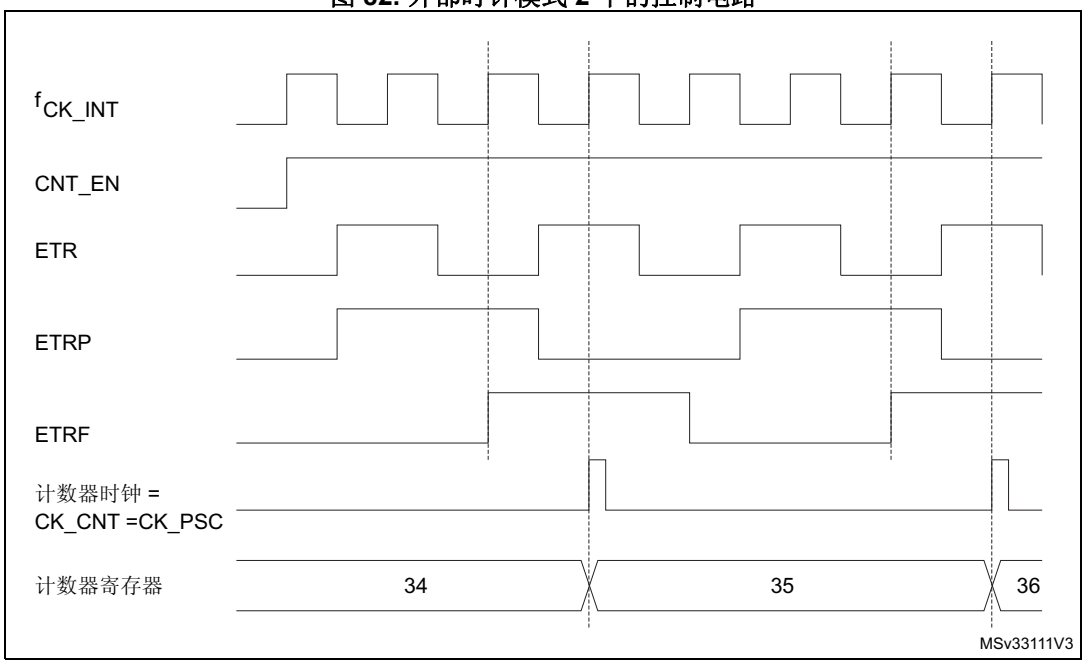
例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 由于此例中不需滤波器，因此在 TIMx_SMCR 寄存器中写入 ETF[3:0]=0000。
2. 通过在 TIMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
3. 通过在 TIMx_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
4. 通过在 TIMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
5. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

ETR 的上升沿与实际计数器时钟之间的延迟是由于 ETRP 信号的重新同步电路引起的。因此，计数器能够正确捕获的最大频率不超过 TIMxCLK 频率的 ¼。当 ETRP 信号较快时，用户应通过适当的 ETPS 预分频器设置对外部信号进行分频。

图 82. 外部时钟模式 2 下的控制电路



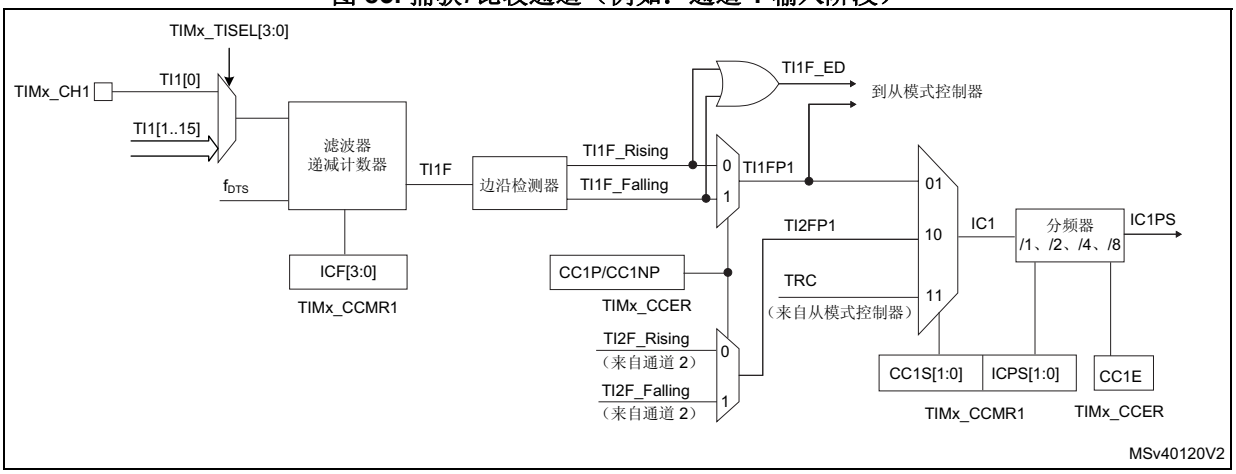
15.3.6 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入单元（数字滤波、多路复用和预分频器，通道 5 和通道 6 除外）和一个输出单元（比较器和输出控制）构建而成。

图 83 到*图 86* 简要介绍了一个捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 83. 捕获/比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端输出电平由极性选择位控制。

图 84. 捕获/比较通道 1 主电路

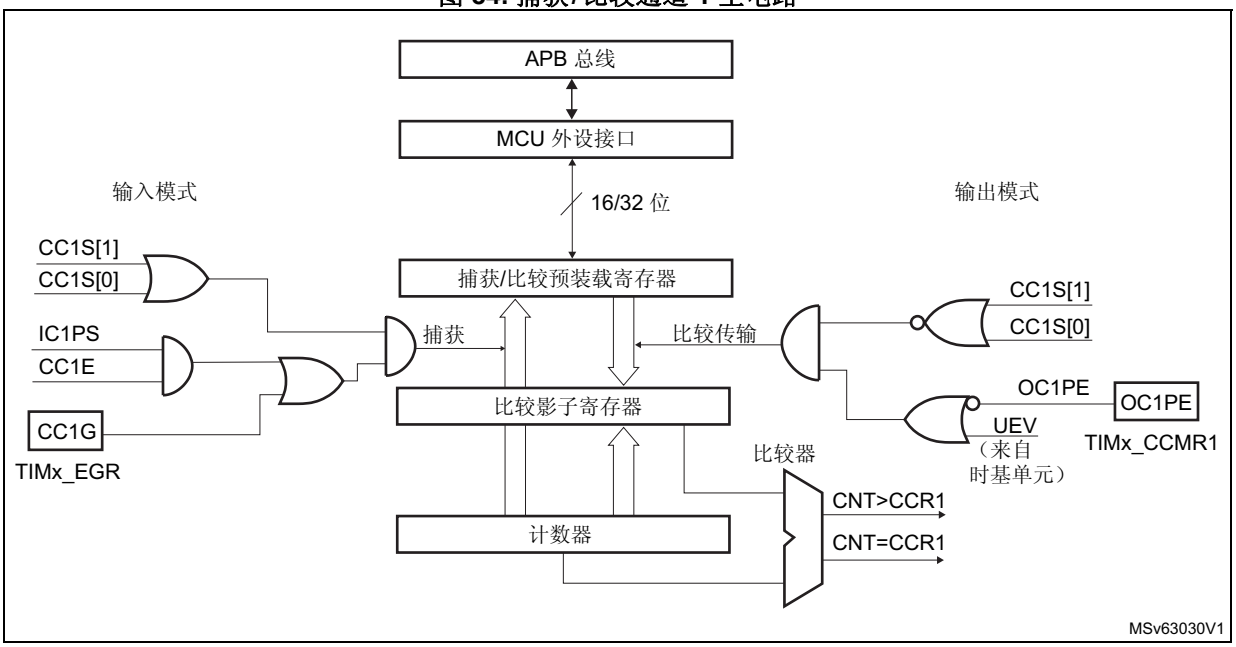
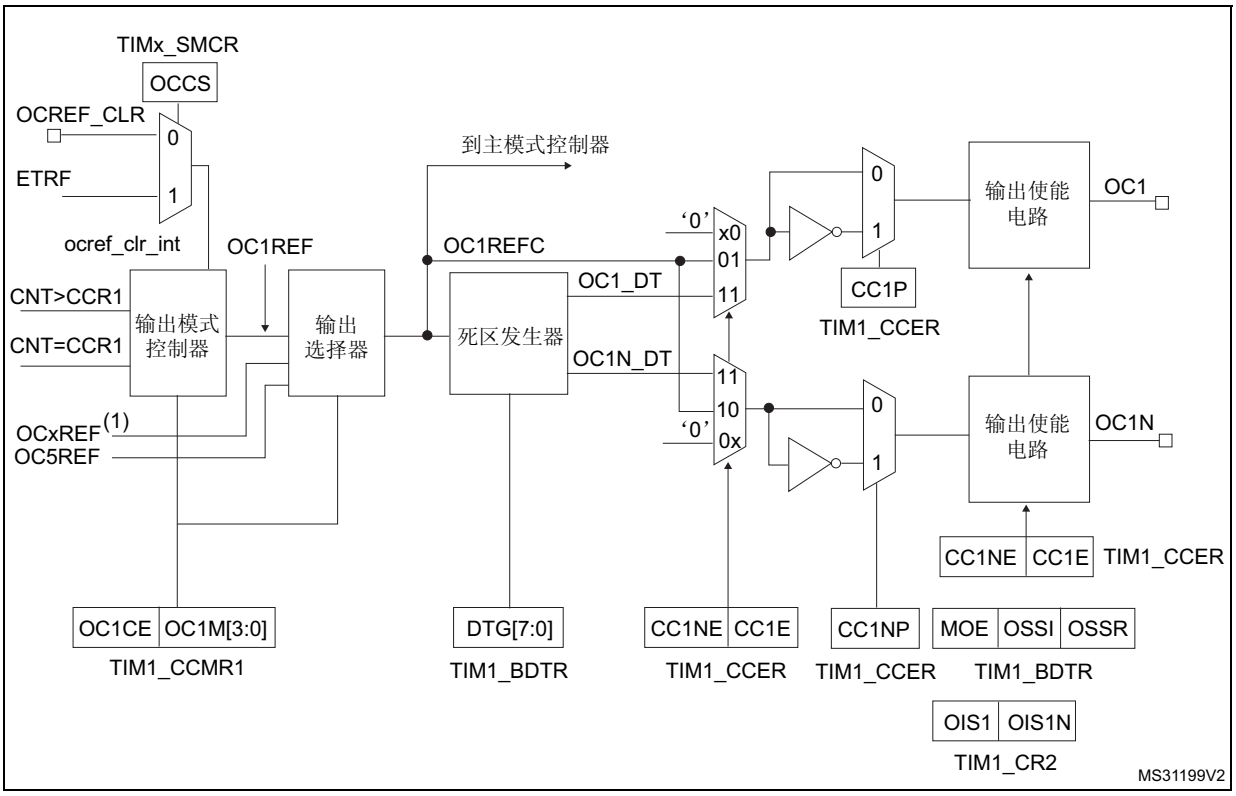


图 85. 捕获/比较通道的输出阶段（通道 1、通道 2 和通道 3）



1. OCxREF，其中 x 为互补通道的序号

图 86. 捕获/比较通道的输出阶段（通道 4）

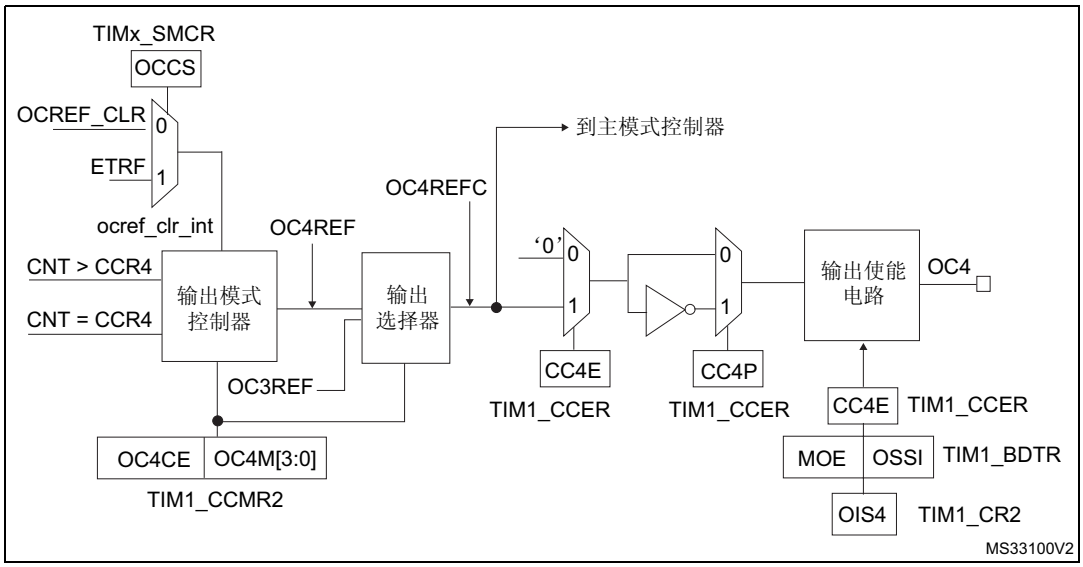
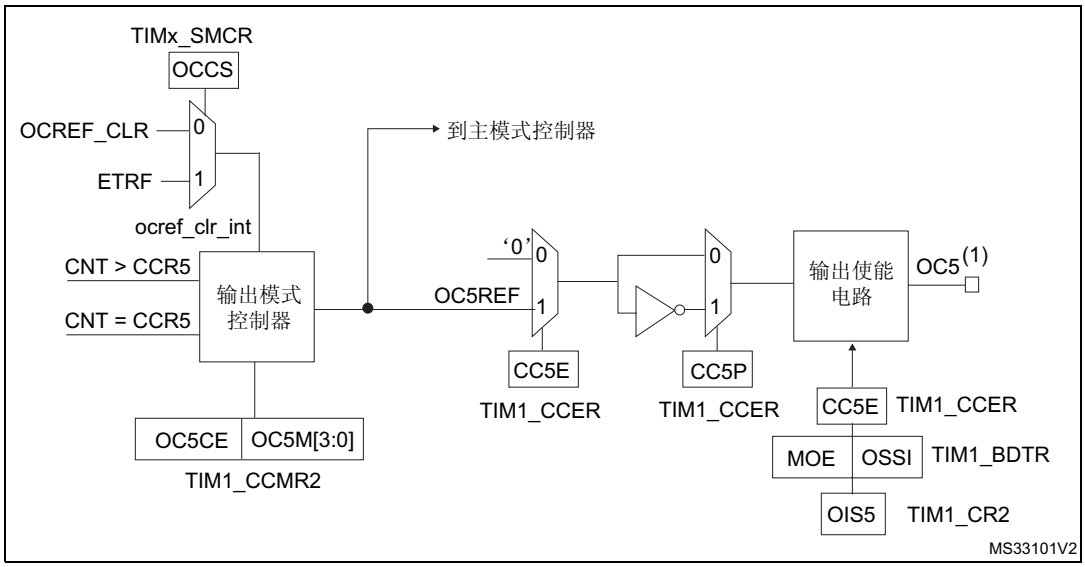


图 87. 捕获/比较通道的输出阶段（通道 5 和通道 6）



1. 不适用于外部。

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。读写访问的始终是预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容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将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

15.3.7 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将过捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件方法向 CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择有效输入：TIMx_CCR1 必须关联到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
3. 根据关联到定时器的信号，对相关输入滤波参数进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设输入信号发生翻转时，信号需要 5 个内部时钟周期才能稳定，则必须设置滤波时间大于 5 个内部时钟周期。若在 TI1 上连续 8 次检测到新电平采样值（以 f_{DTS} 频率采样）判定沿跳变合法，则向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
4. 通过在 TIMx_CCER 寄存器中将 CC1P 位和 CC1NP 位写入 0，选择 TI1 上的有效转换边沿（本例中为上升沿）。

5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
6. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
7. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 过捕获标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理过捕获，建议在读取过捕获标志之前读取数据。这样可避免丢失在读取过捕获标志之后与读取数据之前可能出现的过捕获信息。

注： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

15.3.8 PWM 输入模式

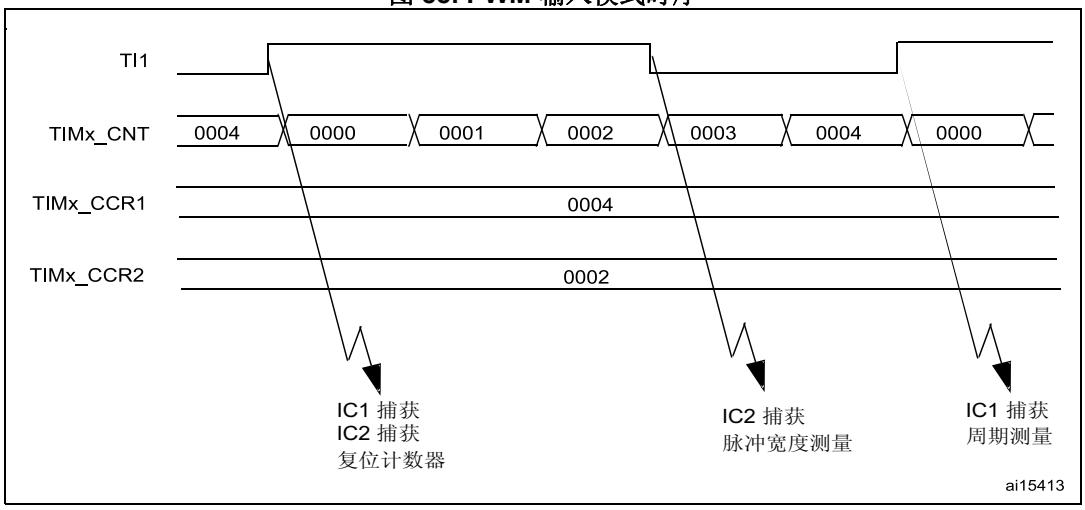
此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，用户可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
3. 选择 TI1FP1 的有效极性（用于在 TIMx_CCR1 中捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
4. 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10（选择 TI1）。
5. 选择 TI1FP2 的有效极性（用于在 TIMx_CCR2 中捕获）：向 CC2P 位和 CC2NP 位写入 CC2P/CC2NP=“10”（下降沿有效）。
6. 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 00101（选择 TI1FP1）。
7. 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 0100。
8. 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 88. PWM 输入模式时序



15.3.9 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（OCxREF 和 OCx/OCxN）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号 (OCxREF/OCx) 强制设置为有效电平，用户只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入 0101。OCxREF 则强制输出高电平，（OCxREF 始终为高电平有效），最终OCx的输出与极性选择位CCxP相反。

例如：CCxP=0（OCx 高电平有效）=> OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 0100，可将 OCxREF 信号强制设置为低电平。

当然，即使强制输出模式下，TIMx_CCRx 影子寄存器与计数器之间的比较仍然会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。下面的输出比较模式一节对此进行了介绍。

15.3.10 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。通道 1 到通道 4 可用作输出，而通道 5 和通道 6 只能在器件内部使用（例如，用于产生混合波形或触发 ADC）。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（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。匹配时，输出引脚根据所选择的模式和极性，存在多种输出可能：保持其原有电平 (OCxM=0000)，输出为有效电平 (OCxM=0001)、输出无效电平(OCxM=0010) 或者进行电平翻转 (OCxM=0011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

使用 TIMx_CCMRx 寄存器中的 OCxPE 位，可将 TIMx_CCRx 寄存器配置为带或不带预装载功能。

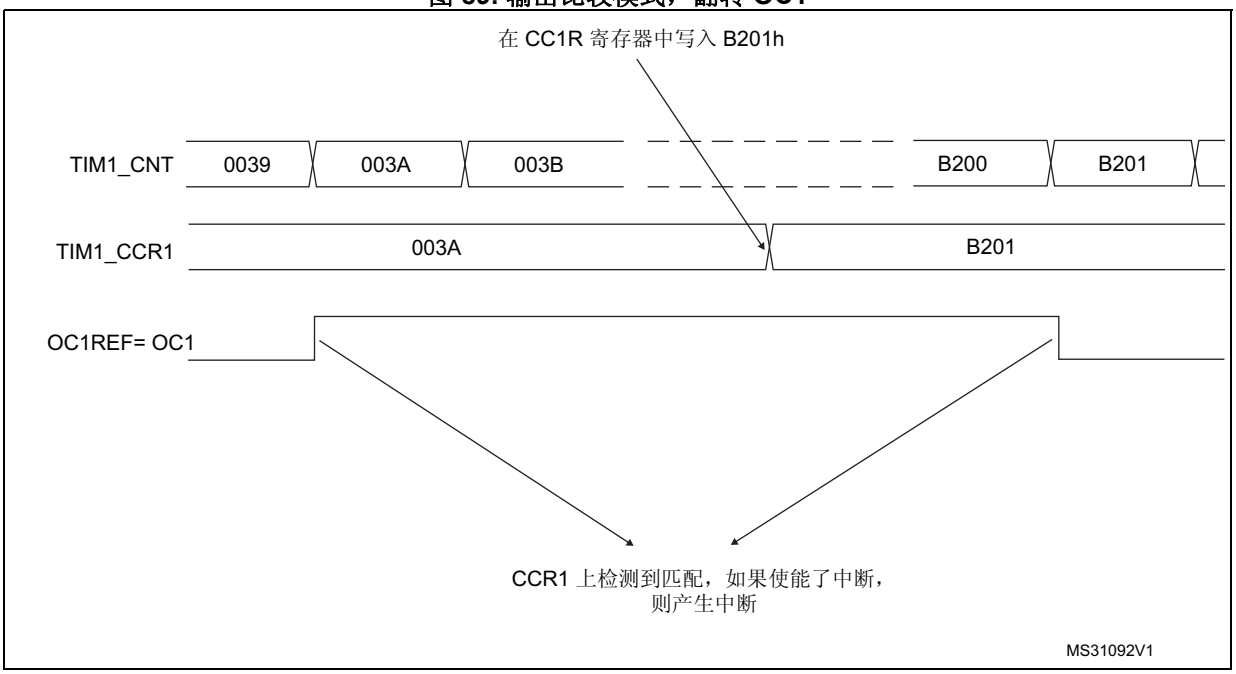
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。时间分辨率可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   - 当 CNT 与 CCRx 匹配时，写入 OCxM = 0011 以翻转 OCx 输出引脚
   - 写入 OCxPE = 0 以禁止预装功能
   - 写入 CCxP = 0 以选择高电平有效极性
   - 写入 CCxE = 1 以使能输出
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装载功能（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIMx_CCRx 影子寄存器）。*图 89* 给出了一个示例。

图 89. 输出比较模式，翻转 OC1



15.3.11 PWM 模式

通过脉冲宽度调整模式可以生成一个信号，该信号频率由 TIMx_ARR 寄存器值决定，其占空比则由 TIMx_CCRx 寄存器值决定。

各通道可以独立选择 PWM 模式（每个 OCx 输出对应一个 PWM），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“0110”（PWM 模式 1）或“0111”（PWM 模式 2）。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

由于只有在发生更新事件时预装载寄存器内容才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。通过 CCxE、CCxNE、MOE、OSSI 和 OSSR 位（TIMx_CCER 和 TIMx_BDTR 寄存器）的组合使能 OCx 输出。有关详细信息，请参见 TIMx_CCER 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 总是与 TIMx_CCRx 进行比较，以确定是 TIMx_CCRx ≤ TIMx_CNT 还是 TIMx_CNT ≤ TIMx_CCRx（取决于计数器计数方向）。

根据 TIMx_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

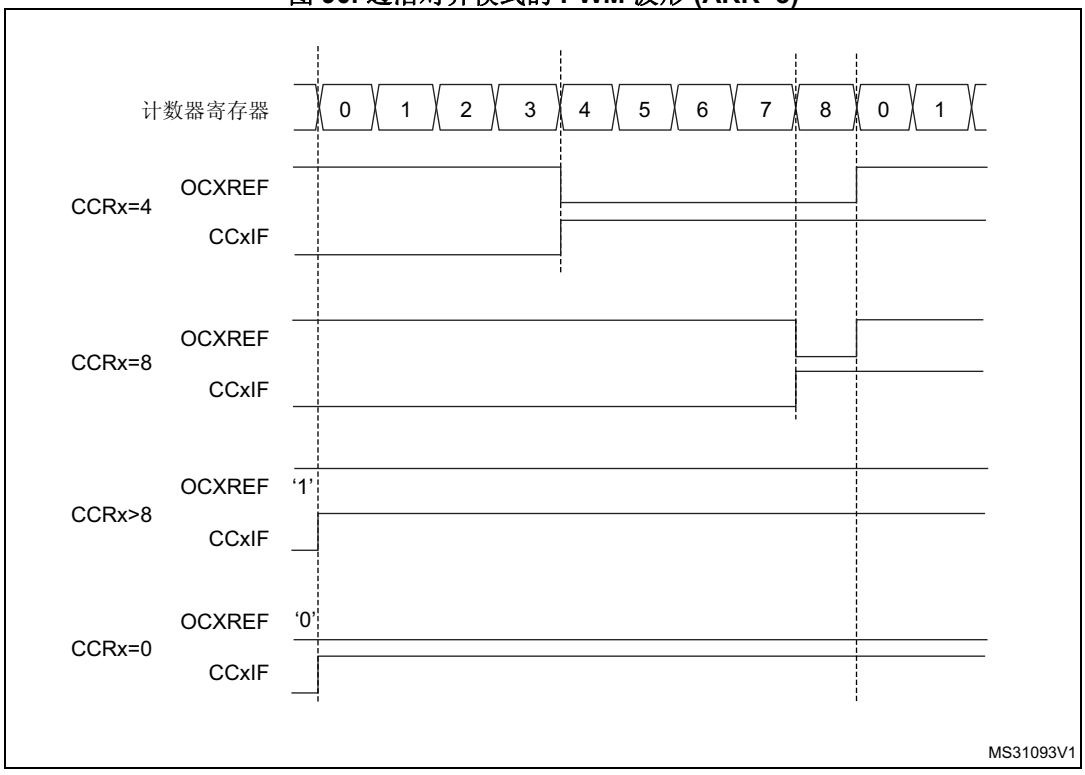
PWM 边沿对齐模式

- 递增计数配置

  当 TIMx_CR1 寄存器中的 DIR 位为低时执行递增计数。请参见*第 277 页的递增计数模式*。

  以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。*图 90* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 90. 边沿对齐模式的 PWM 波形 (ARR=8)



- 递减计数配置

  当 TIMx_CR1 寄存器中的 DIR 位为高时执行递减计数。请参见*第 280 页的递减计数模式*。

  在 PWM 模式 1 下，只要 TIMx_CNT > TIMx_CCRx，参考信号 OCxRef 即为低电平，否则其为高电平。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重载值，则 OCxREF 保持为“1”。此模式下不可能产生 0% 的 PWM 波形。

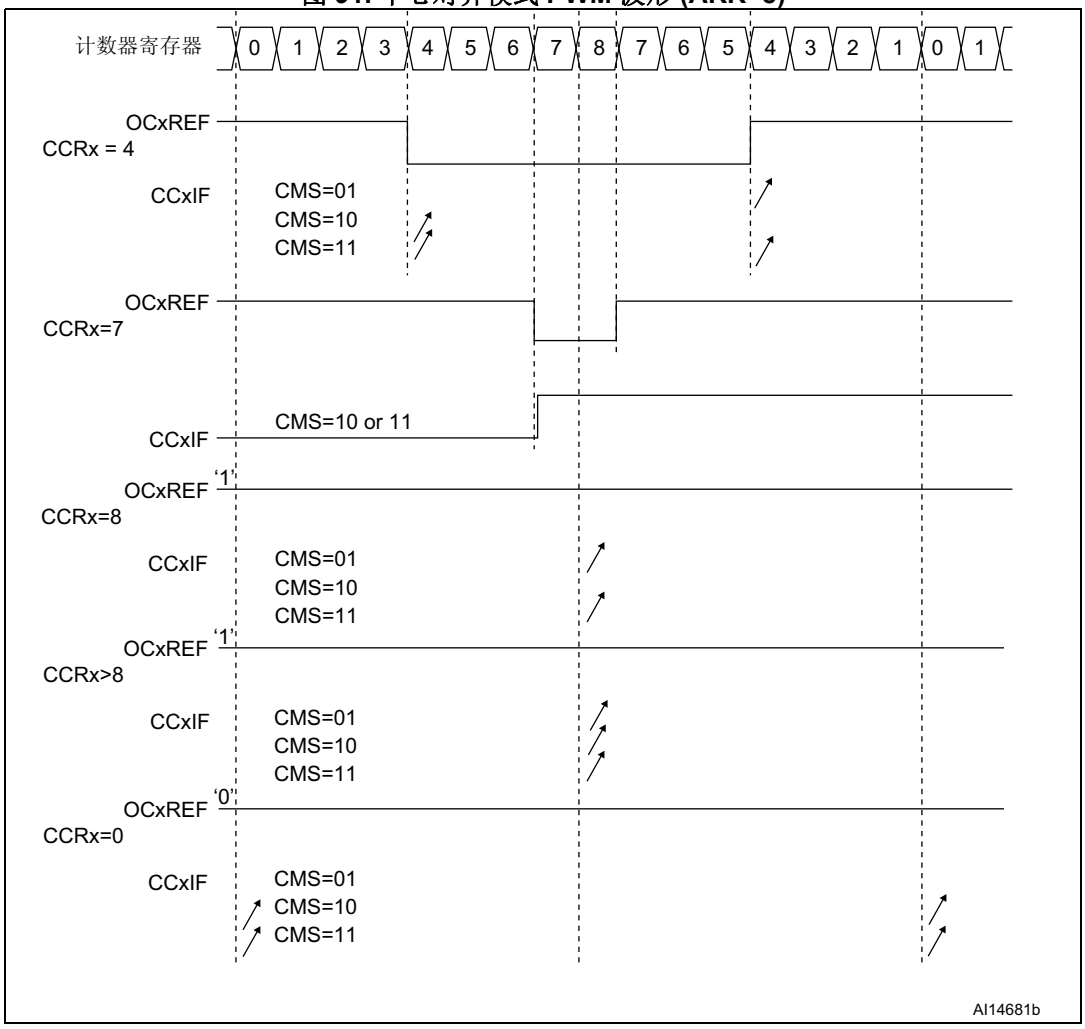
PWM 中心对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为“00”（其余所有配置对 OCxRef/OCx 信号具有相同的作用），中心对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIMx_CR1 寄存器中的方向位 (DIR) 由硬件更新，不得通过软件更改。请参见*第 283 页的中心对齐模式（递增/递减计数）*。

图 91 显示了中心对齐模式的 PWM 波形，在此例中：

- TIMx_ARR=8，
- PWM 模式为 PWM 模式 1，
- 在根据 TIMx_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1。

图 91. 中心对齐模式 PWM 波形 (ARR=8)



中心对齐模式使用建议

- 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得通过软件同时修改 DIR 和 CMS 位。
- 不建议在中心对齐模式下对计数器执行写操作，否则将发生意外错误。尤其是：
  - 如果写入计数器的值大于自动重载值 (TIMx_CNT>TIMx_ARR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
  - 如果向计数器写入 0 或 TIMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

15.3.12 不对称 PWM 模式

在不对称模式下，生成的两个中心对齐 PWM 信号间允许存在可编程相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和相移则由一对 TIMx_CCRx 寄存器确定。两个寄存器分别控制递增计数和递减计数期间的 PWM，这样每半个 PWM 周期便会调节一次 PWM：

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

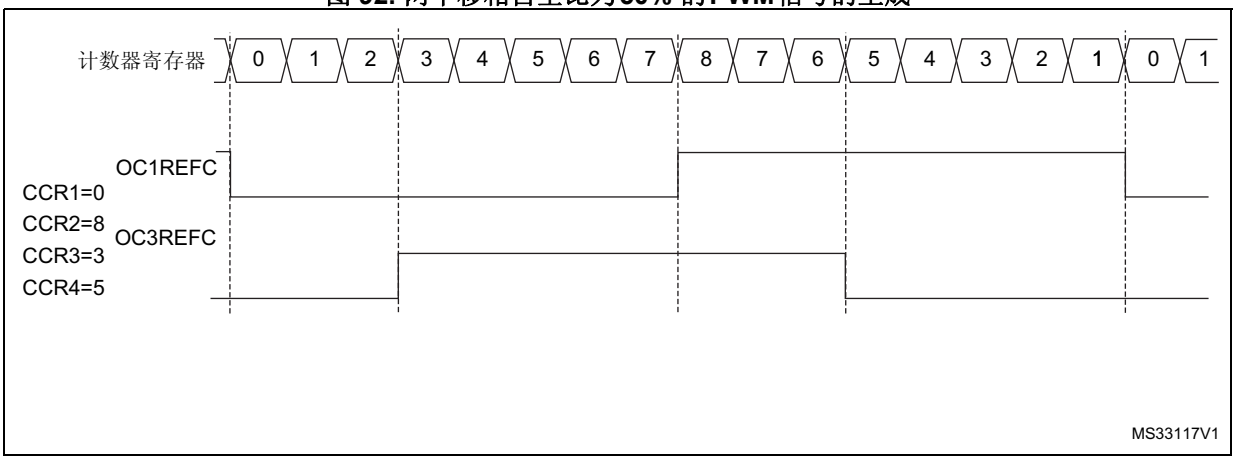
两个通道可以独立选择不对称 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1110”（不对称 PWM 模式 1）或“1111”（不对称 PWM 模式 2）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

给定通道用作不对称 PWM 通道时，也可使用其互补通道。例如，如果通道 1 上产生 OC1REFC 信号（不对称 PWM 模式 1），则由于不对称 PWM 模式 1 的原因，通道 2 上可输出 OC2REF 信号或 OC2REFC 信号。

图 92 显示了不对称 PWM 模式下可以产生的信号示例（通道 1 到通道 4 在不对称 PWM 模式 1 下配置）。与死区发生器配合使用时，这可控制相移全桥直流到直流转换器。

图 92. 两个移相占空比为50% 的PWM信号的生成



15.3.13 组合 PWM 模式

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的“逻辑或”运算或者“逻辑与”运算组合组成。

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

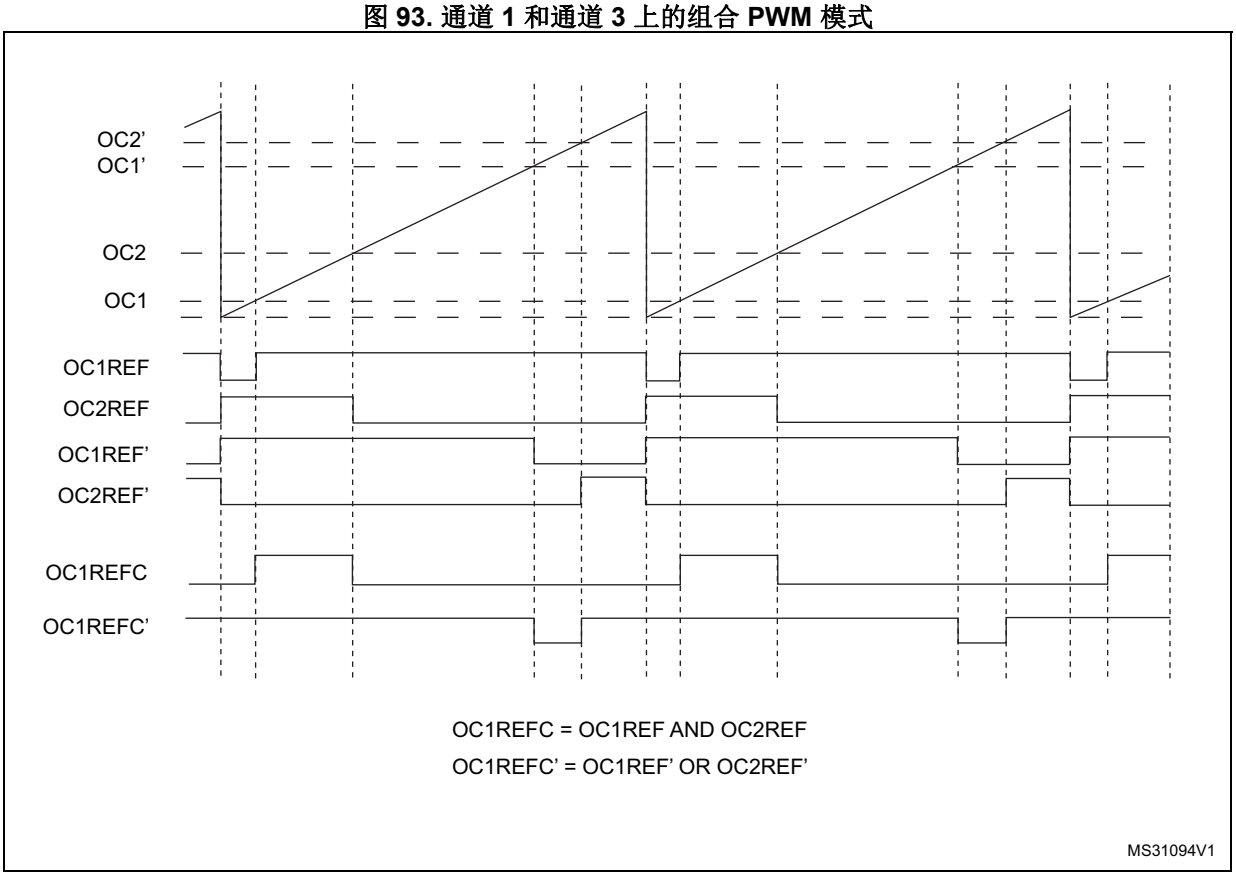
当给定通道用作组合 PWM 通道时，其互补通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 93 显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 配置为组合 PWM 模式 2，
- 通道 2 配置为 PWM 模式 1，
- 通道 3 配置为组合 PWM 模式 2，
- 通道 4 在 PWM 模式 1 下配置。

图 93. 通道 1 和通道 3 上的组合 PWM 模式



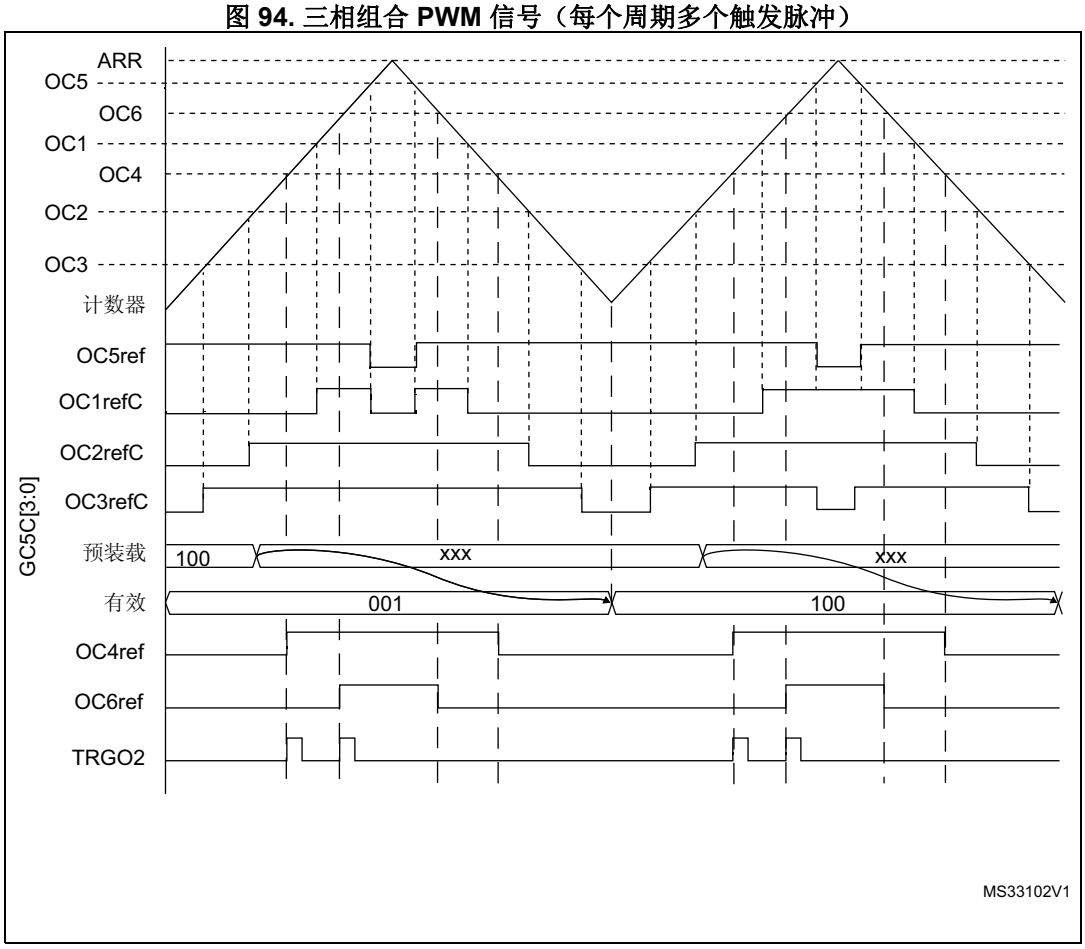
15.3.14 组合三相 PWM 模式

在组合三相 PWM 模式下，产生的一至三个中心对齐 PWM 信号与一个可编程信号间允许在脉冲中间进行逻辑与运算。OC5REF 信号用于定义产生的组合信号。通过 TIMx_CCR5 中的 3 位 GC5C[3:1]，可以选择 OC5REF 与哪个参考信号组合。产生的信号 OCxREFC 由两个参考 PWM 的逻辑与运算组合组成：

- 如果 GC5C1 置 1，则 OC1REFC 由 TIMx_CCR1 和 TIMx_CCR5 控制
- 如果 GC5C2 置 1，则 OC2REFC 由 TIMx_CCR2 和 TIMx_CCR5 控制
- 如果 GC5C3 置 1，则 OC3REFC 由 TIMx_CCR3 和 TIMx_CCR5 控制

通道 1 到通道 3 可独立选择组合三相 PWM 模式，只需将 3 位 GC5C[3:1] 中的至少一位置 1。

图 94. 三相组合 PWM 信号（每个周期多个触发脉冲）



TRGO2 波形说明了如何根据给定的三相 PWM 信号同步 ADC。更多详细信息，请参见 *第 15.3.27 节：ADC 同步*。

15.3.15 互补输出和死区插入

高级控制定时器 (TIM1) 可以输出两路互补信号，并管理输出的关断与接通瞬间。

这段时间通常称为死区，用户必须根据与输出相连接的器件及其特性（电平转换器的固有延迟、开关器件产生的延迟...）来调整死区时间

每路输出可以独立选择输出极性（主输出 OCx 或互补输出 OCxN）。可通过对 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位执行写操作来完成极性选择。

互补信号 OCx 和 OCxN 通过以下多个控制位的组合进行激活：TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位以及 TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。更多详细信息，请参见 *第 347 页的表 66：具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位*。应当注意，切换至空闲状态（MOE 位变为 0）的时刻，死区仍然有效。

CCxE 和 CCxNE 位同时置 1 并且 MOE 位置 1（若存在刹车电路）时，将使能死区插入。每个通道有一个 10 位死区发生器。将基于参考波形 OCxREF 生成 2 个输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高电平有效：

- 输出信号 OCx 与参考信号相同，只是其上升沿相对参考上升沿存在延迟。
- 输出信号 OCxN 与参考信号相反，并且其上升沿相对参考下降沿存在延迟。

如果延迟时间大于有效输出（OCx 或 OCxN）的宽度，则不会产生相应的脉冲。

下图所示为死区发生器的输出信号与参考信号 OCxREF 之间的关系。（在这些示例中，假定 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1）

图 95. 带死区插入的互补输出

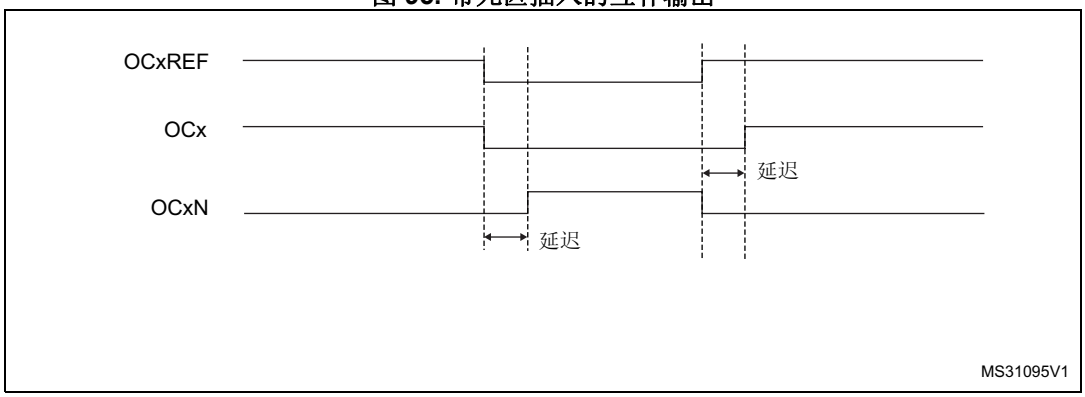


图 96. 延迟时间大于负脉冲宽度的死区波形

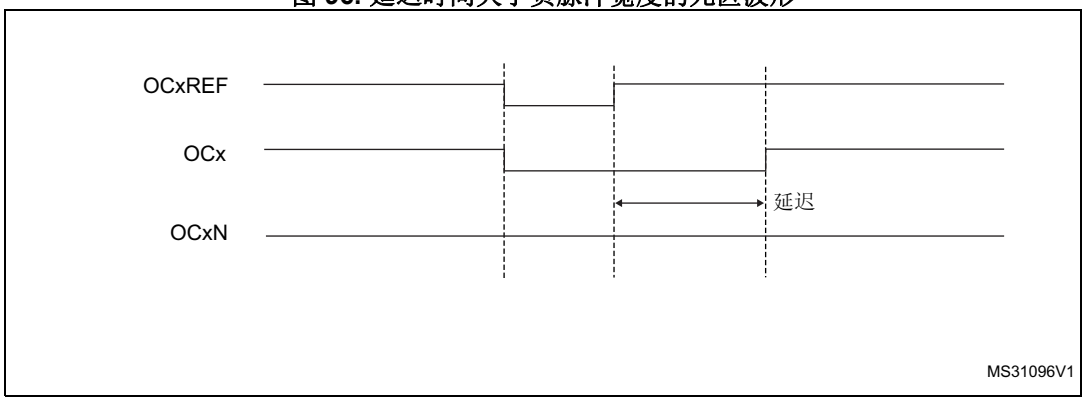
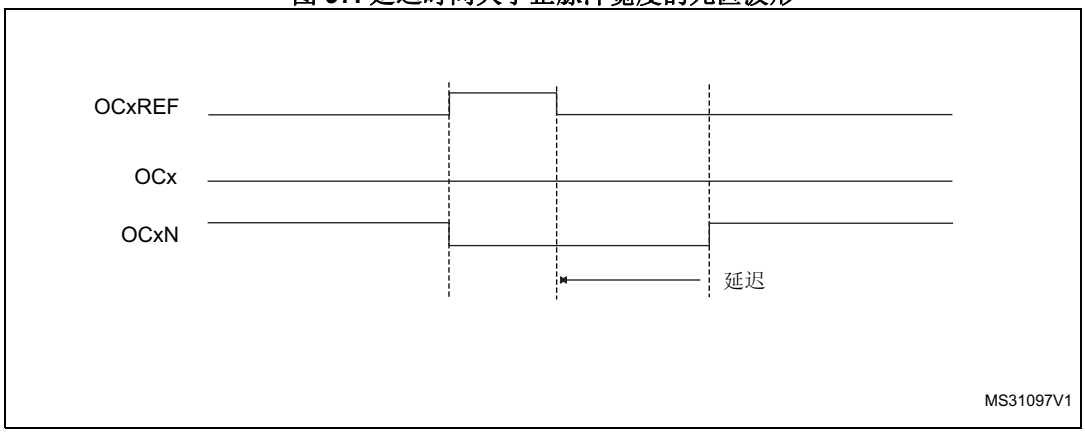


图 97. 延迟时间大于正脉冲宽度的死区波形



死区延迟对于所有通道均相同，可通过 TIMx_BDTR 寄存器中的 DTG 位进行编程。有关延迟时间计算的信息，请参见*第 15.4.20 节：TIM1 刹车和死区寄存器 (TIM1_BDTR)*。

将 OCxREF 重定向到 OCx 或 OCxN

在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位，可将 OCxREF 重定向到 OCx 输出或 OCxN 输出。

通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。或者，使两个输出同时保持无效电平，或者两个输出同时处于有效电平，两者互补并且带死区。

注： *如果仅使能 OCxN (CCxE=0, CCxNE=1)，两者不互补，一旦 OCxREF 为高电平，OCxN 即变为有效。例如，如果 CCxNP=0，则 OCxN=OCxRef。另一方面，如果同时使能 OCx 和 OCxN (CCxE=CCxNE=1)，OCx 在 OCxREF 为高电平时变为有效，而 OCxN 则与之互补，在 OCxREF 为低电平时变为有效。*

15.3.16 使用刹车功能

刹车功能的目的是保护由 TIM1 定时器生成的 PWM 信号所驱动的功率开关。两个刹车输入通常连接到功率级和三相逆变器的故障输出。激活时，刹车电路会关闭 PWM 输出，并将其强制为预定义的安全状态。也可选择一些内部 MCU 事件来触发输出关断。

有两个刹车通道。一个刹车通道收集系统级故障（时钟失效和奇偶校验错误等）和应用故障（来自输入引脚），可以在死区持续时间后将输出强制为预定义的电平（有效或无效）。刹车 2 通道只包括应用故障，能够将输出强制为无效状态。

刹车期间的输出使能信号和输出电平取决于多个控制位：

- TIMx_BDTR 寄存器中的 MOE 位，允许通过软件使能/禁止输出，在发生刹车事件和刹车 2 事件时复位。
- TIMx_BDTR 寄存器中的 OSSI 位，定义定时器将输出控制在无效状态下，还是将输出控制释放给 GPIO 控制器（通常使其处于高阻态模式）
- TIMx_CR2 寄存器中的 OISx 和 OISxN 位，将输出设置为关断电平（有效或无效）。无论 OISx 和 OISxN 的值为何，均无法在给定时间将 OCx 和 OCxN 输出同时设置为有效电平。更多详细信息，请参见*第 347 页的表 66：具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位*。

退出复位状态后，刹车功能处于禁止状态，MOE 位处于低电平。通过设置 TIMx_BDTR 寄存器中的 BKE 位和 BK2E 位来使能刹车功能。可通过配置同一寄存器中的 BKP 位和 BK2P 位来选择刹车输入的极性。BKE/BK2E 和 BKP/BK2P 位可同时修改。对 BKE/BK2E 和 BKP/BK2P 位执行写操作时，写操作会在 1 个 APB 时钟周期的延迟后生效。因此，执行写操作后，需要等待 1 个 APB 时钟周期，才能准确回读该位。

由于 MOE 下降沿可能是异步信号，因此在实际信号（作用于输出）与同步控制位（位于 TIMx_BDTR 寄存器中）之间插入了再同步电路，从而在异步信号与同步信号之间产生延迟。具体而言，如果在 MOE 处于低电平时将其置 1，则必须首先插入延迟（空指令），才能准确进行读取。这是因为写入的是异步信号，而读取的却是同步信号。

可以使用 TIMx_OR2 和 TIMx_OR3 寄存器从多个源产生刹车，这些源可以单独使能并使用可编程边沿有效性。

刹车 (BRK) 通道的源为：

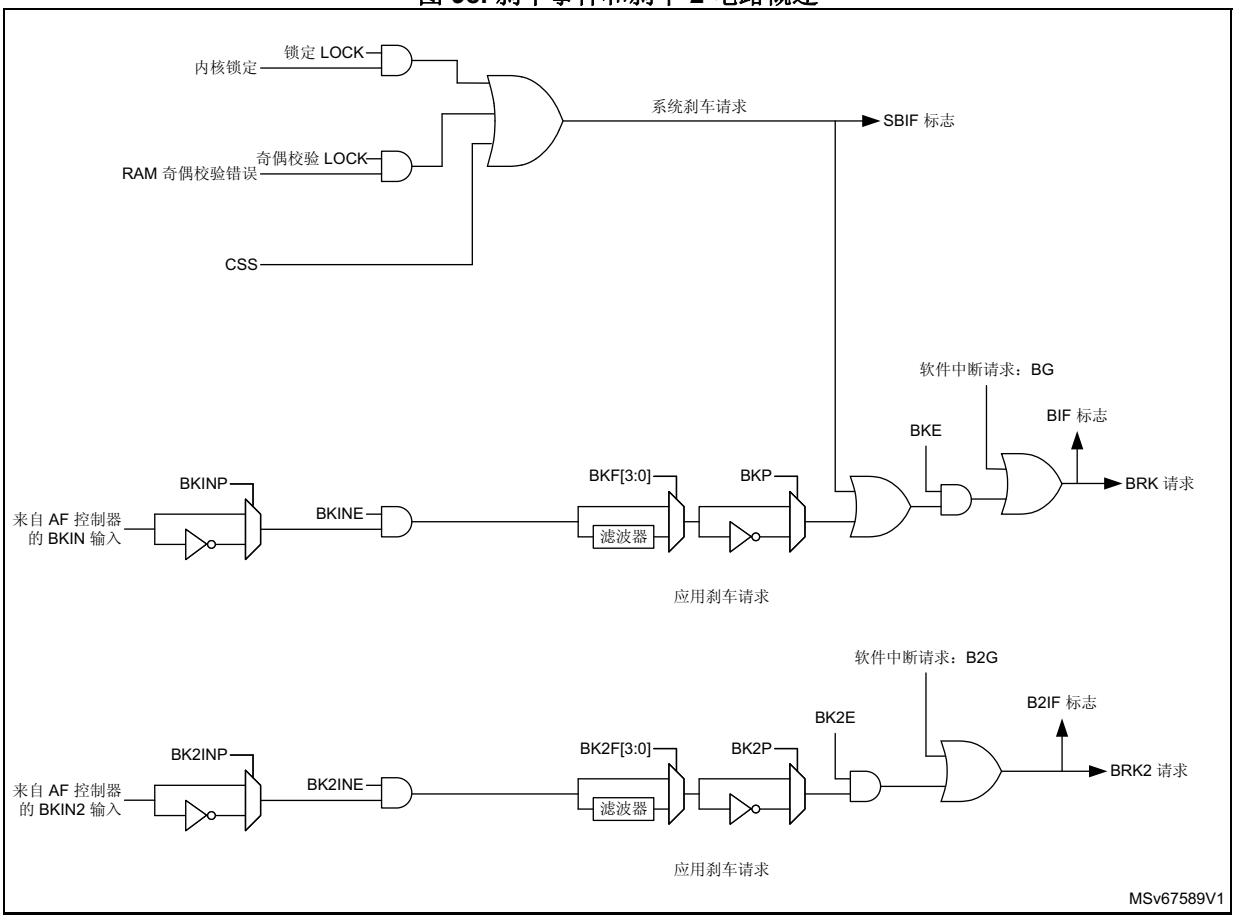
- 连接到 BKIN 引脚的外部源（由 GPIO 复用功能寄存器选定），具有极性选择和可选的数字滤波
- 内部源：
  - Cortex®-M0+ LOCKUP 输出
  - SRAM 奇偶校验错误信号
  - CSS 检测器产生时钟故障事件

刹车 2 (BRK2) 的源是连接到 BKIN 引脚的外部源（由 GPIO 复用功能寄存器选定），具有极性选择和可选的数字滤波。

也可由软件通过 TIMx_EGR 寄存器中的 BG 和 B2G 位产生刹车事件。无论 BKE 和 BK2E 使能位的值如何，都可以使用 BG 和 B2G 通过软件生成刹车。

在所有源进入定时器 BRK 或 BRK2 输入之前，对其进行 OR 运算，如以下*图 98* 所示。

图 98. 刹车事件和刹车 2 电路概述



注： *只有禁止可编程滤波器时才能保证异步（无时钟）操作。如果使能可编程滤波器，必须使用故障安全时钟模式（例如，使用 CSS）来保证能够处理刹车事件。*

发生刹车之一（其中一个刹车输入上出现所选电平）时：

- MOE 位异步清零，使输出处于无效状态、空闲状态甚至释放控制权给 GPIO 控制器（通过 OSSI 位进行选择）。即使 MCU 振荡器关闭，该功能仍然使能。
- MOE=0 时，将以 TIMx_CR2 寄存器 OISx 位中设定的电平驱动每个输出通道。如果 OSSI=0，定时器将释放输出控制（由 GPIO 控制器接管），否则使能输出保持高电平。
- 使用互补输出时：
  - 输出首先置于无效状态（取决于极性）。这是异步操作，因此即使没有为定时器提供时钟，该操作仍有效。
  - 如果定时器时钟仍存在，则将重新激活死区发生器，进而在死区后以 OISx 和 OISxN 位中编程的电平驱动输出。即使在这种情况下，也不能同时将 OCx 和 OCxN 驱动至其有效电平。请注意，MOE 进行再同步，因此死区的持续时间会比通常情况稍长一些（约 2 个 ck_tim 时钟周期）。
  - 如果 OSSI=0，定时器将释放输出控制（由强制高阻态的 GPIO 控制器接管），否则使能输出将保持高电平或在 CCxE 或 CCxNE 位之一为高电平时立即变为高电平。

- 将刹车状态标志（TIMx_SR 寄存器中的 SBIF、BIF 和 B2IF 位）置 1。如果 TIMx_DIER 寄存器中的 BIE 位置 1，则会产生中断。
- 如果 TIMx_BDTR 寄存器中的 AOE 位置 1，则 MOE 位会在发生下一更新事件 (UEV) 时自动再次置 1。例如，这可用于执行调节。否则，MOE 将始终保持低电平，直到应用将其再次置 1。这种情况下，这一特性可用于确保安全。可以将刹车输入连接到功率驱动器的警报、温度传感器或任何安全元件。

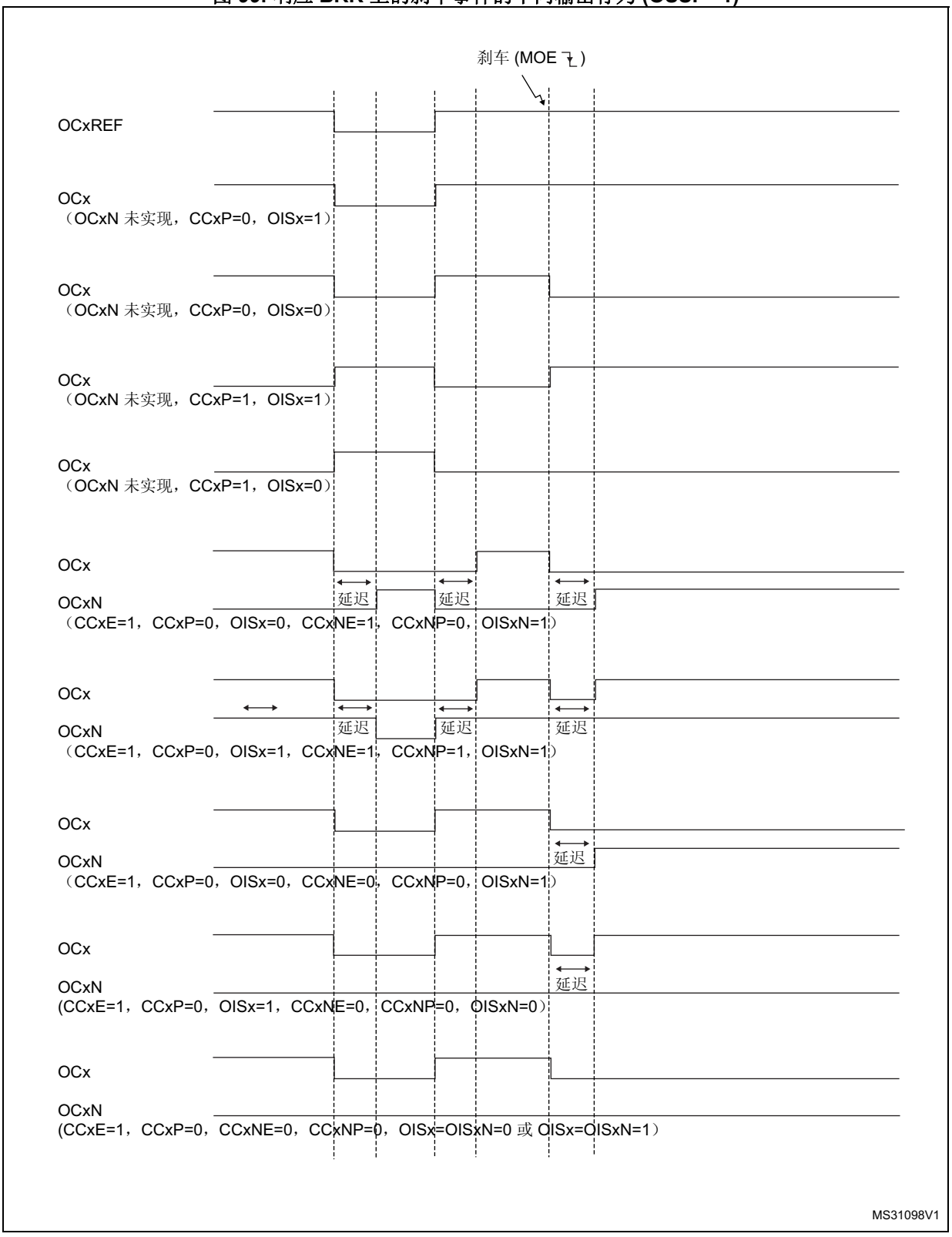
注： *当 AOE 位置 1 时，如果 CPU 将 MOE 复位，则输出处于空闲状态并根据 OSSI 值被强制设为无效电平或高阻态。*
如果 CPU 同时将 MOE 和 AOE 位复位，则输出处于禁止状态并由 TIMx_CR2 寄存器中 OISx 位设定的电平驱动。

注： *刹车输入为电平有效。因此，当刹车输入为有效电平时，不能将 MOE 位置 1（自动或通过软件）。同时，不能将状态标志 BIF 和 B2IF 清零。*

除刹车输入和输出管理外，刹车电路内部还实施了写保护，用以保护应用的安全。通过该功能，用户可冻结多个参数配置（死区持续时间、OCx/OCxN 极性和禁止时的状态、OCxM 配置、刹车使能和极性）。应用可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。请参见*第 15.4.20 节：TIM1 刹车和死区寄存器 (TIM1_BDTR)*。MCU 复位后只能对 LOCK 位执行一次写操作。

图 99 所示为输出对刹车响应行为的示例。

图 99. 响应 BRK 上的刹车事件的不同输出行为 (OSSI = 1)



两个刹车输入针对定时器输出具有不同的行为：

- BRK 输入可禁止（无效状态）PWM 输出，也可将 PWM 输出强制为预定义的安全状态。
- BRK2 只能禁止（无效状态）PWM 输出。

BRK 输入的优先级高于 BRK2 输入，如*表 62* 所示。

注： *BRK2 只能在 OSSR = OSSI = 1 时使用。*

表 62. 定时器输出行为与 BRK/BRK2 输入

BRK	BRK2	定时器输出状态	典型用例	
			OCxN 输出（下桥臂开关）	OCx 输出（上桥臂开关）
有效	X	– 无效，之后强制为输出状态（死区后） – 如果 OSSI = 0，则禁止输出（由 GPIO 逻辑接管控制）	死区插入后开启	关闭
无效	有效	无效	关闭	关闭

图 100 给出了 BRK 和 BRK2 输入上出现有效信号时 OCx 和 OCxN 输出行为示例。在这种情况下，两个输出的极性均为高电平有效（TIMx_CCER 寄存器中的 CCxP = CCxNP = 0）。

图 100. BRK 和 BRK2 引脚使能后的 PWM 输出状态 (OSSI=1)

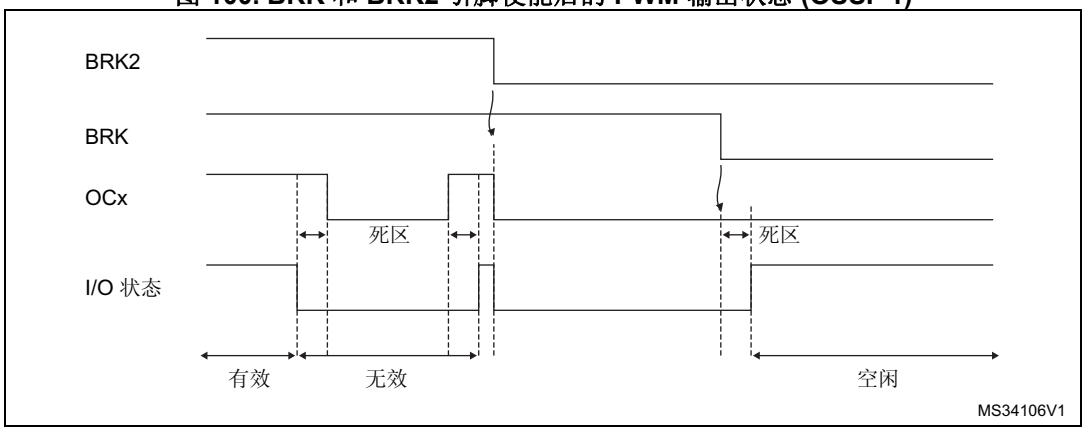
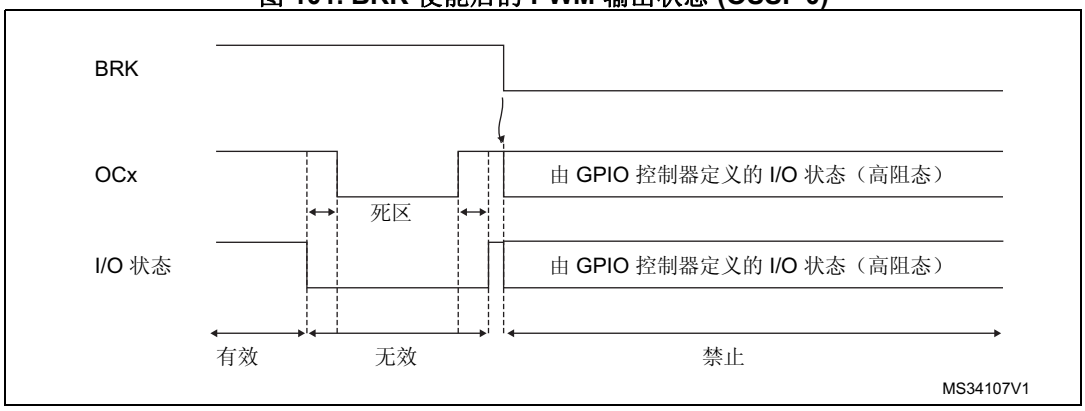


图 101. BRK 使能后的 PWM 输出状态 (OSSI=0)



15.3.17 双向刹车输入

TIM1 具有双向刹车 I/O，如*图 102* 所示。

它们可以实现以下用途：

- 将板级全局刹车信号用于向外部 MCU 或栅极驱动器发送故障信号，唯一的特定引脚作为输入输出双向状态脚。
- 在必须将多个内部和外部刹车输入合并时，将内部刹车源和多个外部开漏比较器输出“或”连接在一起，触发唯一刹车事件。

使用 TIMxBDTR 寄存器的 BKBID 和 BK2BID 位，将刹车事件和刹车 2 输入配置为双向模式。可以使用 TIMxBDTR 寄存器中的 LOCK 位，将 BKBID 编程位锁定在只读模式（锁定级别 1 或更高）。

双向模式可以用于刹车事件和刹车 2 输入，需要将 I/O 配置为低电平有效极性的开漏模式（使用 BKINP、BKP、BK2INP 和 BK2P 位进行配置）。任何来自系统（例如 CSS）、片上外设或刹车输入的刹车请求都会强制将刹车输入置为低电平，以通知发生了故障事件。如果未正确设置极性位（高电平有效极性），则出于安全目的禁止双向模式。

软件刹车事件（BG 和 B2G）也会导致刹车 I/O 被强制置为“0”，从而向外部组件指示定时器已进入刹车状态。但是仅在使能刹车 (BK(2)E = 1) 时有效。当生成软件刹车事件，并且 BK(2)E = 0 时，输出会置于安全状态，并将刹车标志置 1，但对刹车 (2) I/O 无影响。

安全解除机制可防止系统最终锁定（刹车输入上的低电平会触发刹车，进而将相同输入强制置为低电平）。

当 BKDSRM (BK2DSRM) 位置 1 时，会释放刹车输出以清除故障信号，从而使系统能够重新获得保护。

在任何情况下都不能禁止刹车保护电路：

- 刹车输入路径始终有效：即使 BKDSRM (BK2DSRM) 位置 1 且释放开漏控制，刹车事件也仍然有效。这样可以在发生刹车期间防止 PWM 输出重新启动。
- 使能输出（MOE 位置 1）后，BK(2)DSRM 位不能解除刹车保护（请参见*表 63*）

表 63. 刹车保护解除条件

MOE	BKDIR (BK2DIR)	BKDSRM (BK2DSRM)	刹车保护状态
0	0	X	启动
0	1	0	启动
0	1	1	解除
1	X	X	启动

启动和重新启动刹车电路

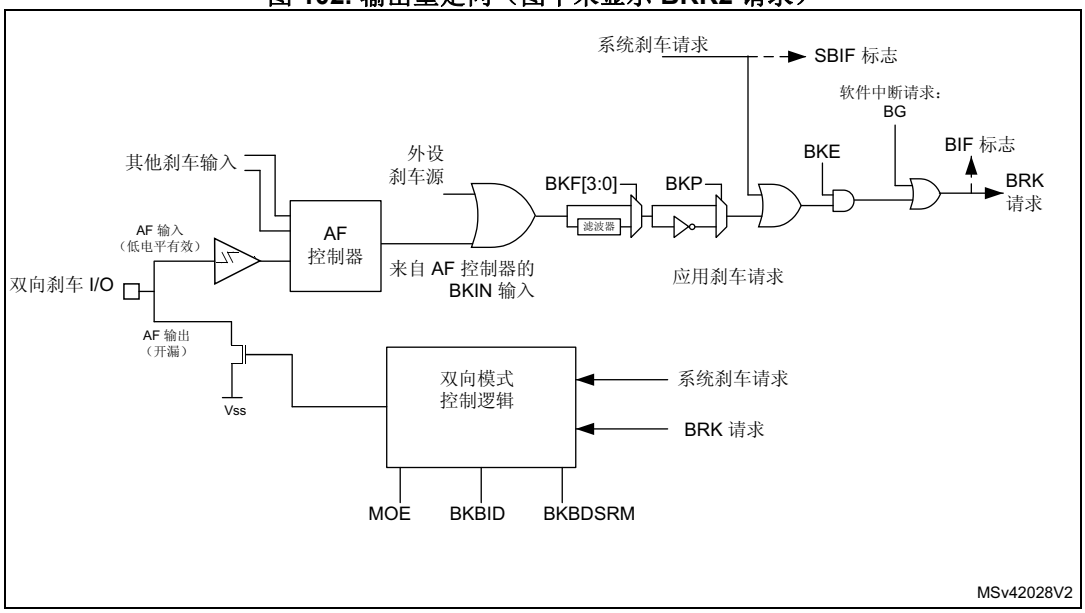
默认情况下（外设复位配置）会启动刹车电路（在输入或双向模式下）。

发生刹车（刹车 2）事件后，必须按照以下步骤重新启动保护：

- 必须将 BKDSRM (BK2DSRM) 位置 1，以释放输出控制
- 软件必须等待系统刹车条件消失（如果有），并清零 SBIF 状态标志（或在重新启动前由系统清零）
- 软件必须轮询 BKDSRM (BK2DSRM) 位，直到该位由硬件清零（当应用程序刹车条件消失时）

此后，刹车电路即启动并激活，可以通过将 MOE 位置 1 来重新使能 PWM 输出。

图 102. 输出重定向（图中未显示 BRK2 请求）



15.3.18 发生外部事件时清除 OCxREF 信号

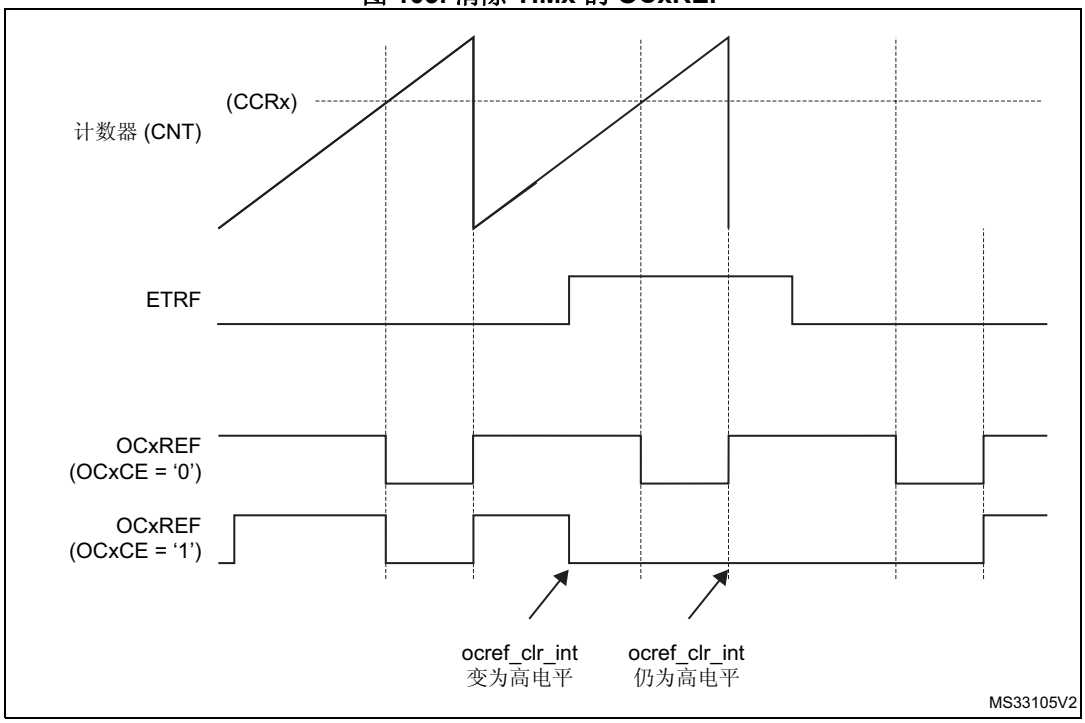
对于给定通道，在 ocref_clr_int 输入上施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号清零。OCxREF 信号将保持低电平，直到在下一 PWM 周期再次变为有效状态。该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。通过配置 TIMx_SMCR 寄存器中的 OCCS 位，可在 OCREF_CLR 输入和 ETRF（滤波器后的 ETR）之间选择 ocref_clr_int 输入。

选择 ETRF 时，ETR 必须配置如下：

1. 必须关闭外部触发预分频器：TIMx_SMCR 寄存器中的 ETPS[1:0] 位置“00”。
2. 必须禁止外部时钟模式 2：TIMx_SMCR 寄存器中的 ECE 位置“0”。
3. 外部触发极性 (ETP) 和外部触发滤波器 (ETF) 可根据用户需要进行配置。

图 103 对比了使能位 OCxCE 在不同值下的情况，显示了当 ETRF 输入变为高电平时 OCxREF 信号的行为。在本例中，定时器 TIMx 编程为 PWM 模式。

图 103. 清除 TIMx 的 OCxREF



注： *如果 PWM 的占空比为 100% (CCRx>ARR)，则下次计数器溢出时会再次使能 OCxREF。*

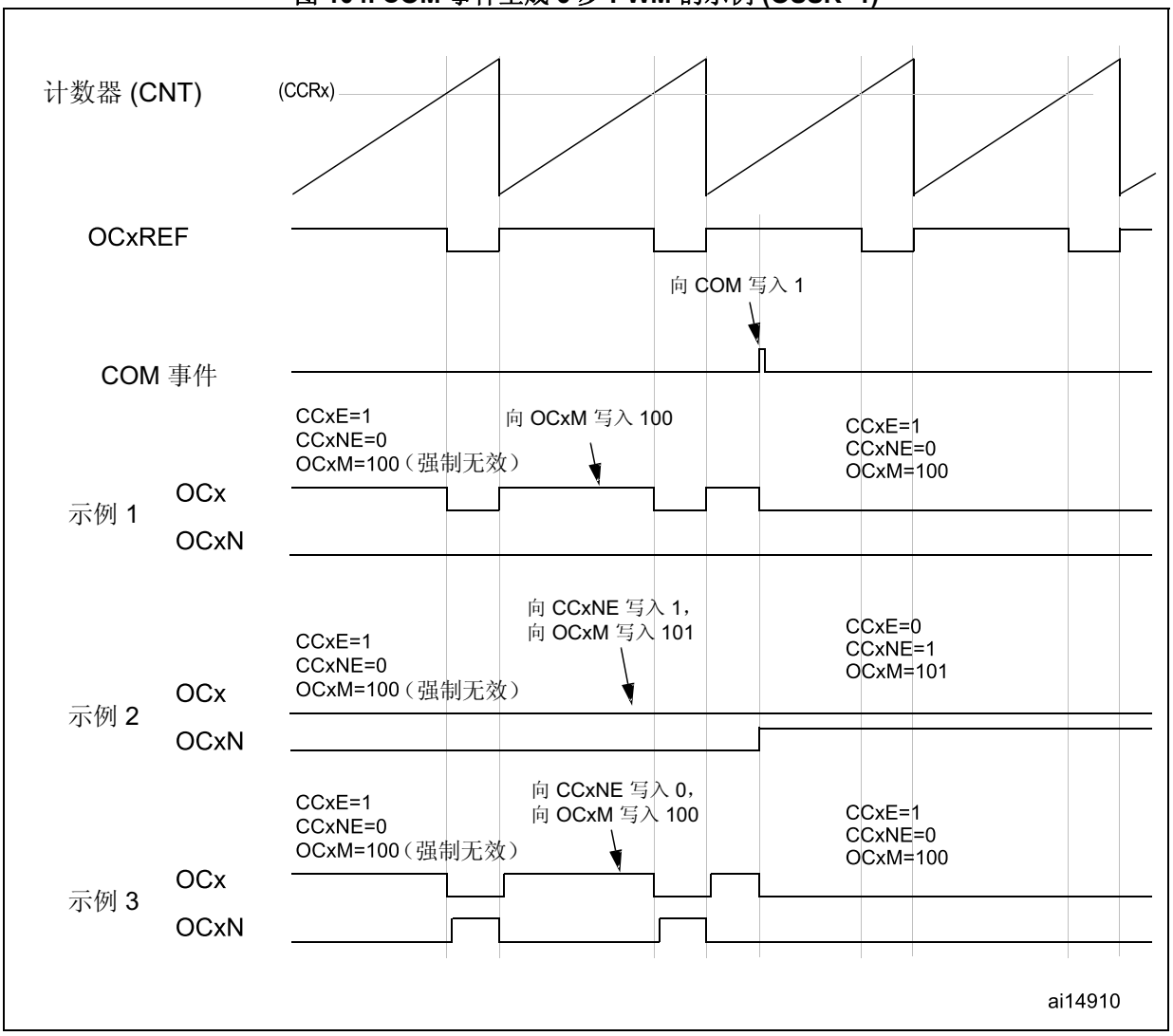
15.3.19 生成 6 步 PWM

当通道使用互补输出时，OCxM、CCxE 和 CCxNE 位上提供预装载位。发生 COM 换向事件时，这些预装载位将传输到影子位。因此，用户可以预先编程下一步骤的配置，并同时更改所有通道的配置。COM 可由软件通过将 TIMx_EGR 寄存器中的 COM 位置 1 而生成，也可以由硬件在 TRGI 上升沿生成。

发生 COM 事件时，某个标志位（TIMx_SR 寄存器中的 COMIF 位）将会置 1。这时，如果 TIMx_DIER 寄存器中的 COMIE 位置 1，将产生中断；如果 TIMx_DIER 寄存器中的 COMDE 位置 1，则将产生 DMA 请求。

图 104 以 3 种不同的编程配置为例，显示了发生 COM 事件时 OCx 和 OCxN 输出的行为。

图 104. COM 事件生成 6 步 PWM 的示例 (OSSR=1)



15.3.20 单脉冲模式

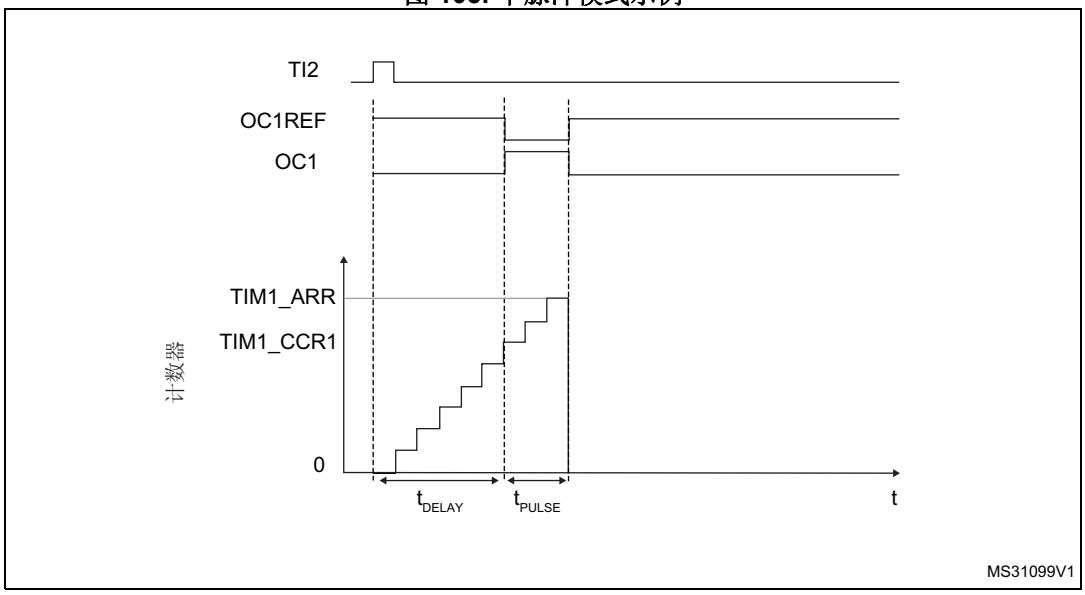
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。通过将 TIMx_CR1 寄存器中的 OPM 位置 1 选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- 递增计数模式下：CNT < CCRx ≤ ARR（特别注意，0 < CCRx）
- 递减计数模式下：CNT > CCRx

图 105. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2x 源（内部或外部）。
2. 在 TIMx_CCMR1 寄存器中写入 CC2S=“01”，以将 TI2FP2 映射到 TI2。
3. 在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
4. 在 TIMx_SMCR 寄存器中写入 TS=00110，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
5. 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），以使用 TI2FP2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，必须通过在 TIMx_CCMR1 寄存器中写入 OC1M=111，来使能 PWM 模式 2。可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

在本例中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此必须向 TIMx_CR1 寄存器的 OPM 位写入 1，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特例：OCx 快速使能：

在单脉冲模式下，TIx 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟（t_{DELAY} 最小值）。

如果要输出延迟时间最短的波形，可以将 TIMx_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 OCxRef（和 OCx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

15.3.21 可再触发单脉冲模式

该模式可以启动计数器来响应激励信号，并且能产生长度可编程的脉冲，但与不可再触发单脉冲模式间存在以下差别，如*第 15.3.20 节*所述：

- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

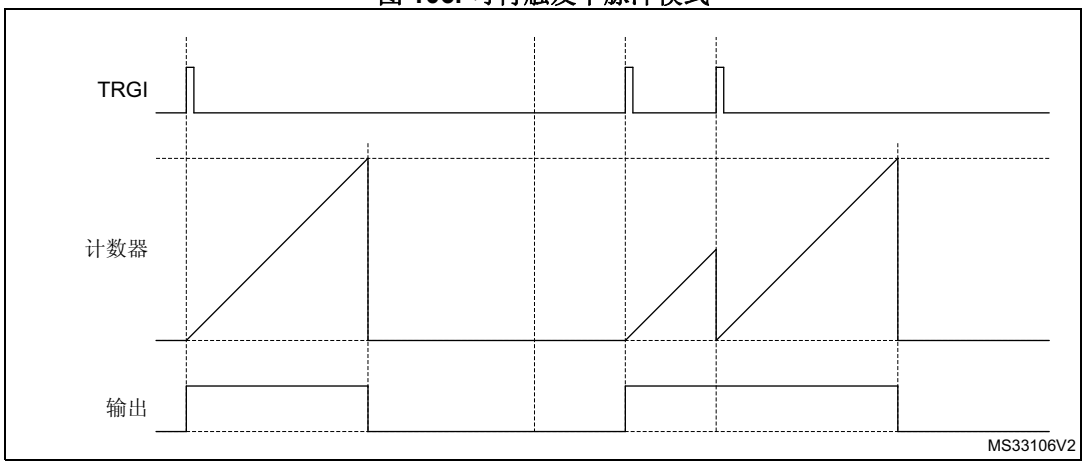
定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] =“1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为“1000”或“1001”。

定时器配置为递增计数模式时，相应的 CCRx 必须置 0（ARR 寄存器设置脉冲长度）。如果定时器配置为递减计数模式，CCRx 必须高于或等于 ARR。

注： *出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

此模式不能与中心对齐 PWM 模式一起使用。在 TIMx_CR1 中必须设置 CMS[1:0] = 00。

图 106. 可再触发单脉冲模式



15.3.22 编码器接口模式

选择编码器接口模式时，如果计数器仅在 TI2 边沿处计数，在 TIMx_SMCR 寄存器中写入 SMS=“001”；如果计数器仅在 TI1 边沿处计数，写入 SMS=“010”；如果计数器在 TI1 和 TI2 边沿处均计数，则写入 SMS=“011”。

通过编程 TIMx_CCER 寄存器的 CC1P 和 CC2P 位，选择 TI1 和 TI2 极性。如果需要，还可对输入滤波器进行编程。CC1NP 和 CC2NP 必须保持低电平。

TI1 和 TI2 两个输入用于连接正交编码器。请参见*表 64*。如果使能计数器（在 TIMx_CR1 寄存器的 CEN 位中写入“1”），则计数器的时钟由 TI1FP1 或 TI2FP2 上的每次有效信号转换提供。TI1FP1 和 TI2FP2 是 TI1 和 TI2 进行输入滤波和极性选择后的信号，如果不进行滤波和反相，则 TI1FP1=TI1，TI2FP2=TI2。将根据两个输入的信号转换序列，产生计数脉冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应修改。任何输入（TI1 或 TI2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI1 或 TI2 边沿处计数，还是同时在 TI1 和 TI2 处计数。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 TIMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIMx_ARR。同样，捕获、比较、重复计数器和触发输出功能继续正常工作。编码器模式和外部时钟模式 2 不兼容，因此不能同时选择。

注： *使能编码器模式时，预分频器必须设置为零*

在此模式下，计数器会根据正交编码器的速度和方向自动进行修改，因此，其内容始终表示编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 TI1 和 TI2 不同时切换）。

表 64. 计数方向与编码器信号的关系

有效边沿	相反信号的电平（TI1FP1 对应 TI2，TI2FP2 对应 TI1）	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
仅在 TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TI2 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 TI1 和 TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

正交编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码器输出可与外部中断输入相连，用以触发计数器复位。

图 107 以计数器工作为例，说明了计数信号的产生和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S=“01”（TIMx_CCMR1 寄存器，TI1FP1 映射到 TI1 上）。
- CC2S=“01”（TIMx_CCMR2 寄存器，TI1FP2 映射到 TI2 上）。
- CC1P=“0”，CC1NP=“0”（TIMx_CCER 寄存器，TI1FP1 未反相，TI1FP1=TI1）。
- CC2P=“0”，CC2NP=“0”（TIMx_CCER 寄存器，TI1FP2 未反相，TI1FP2= TI2）。
- SMS=“011”（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）。
- CEN=“1”（TIMx_CR1 寄存器，使能计数器）。

图 107. 编码器接口模式下的计数器工作示例。

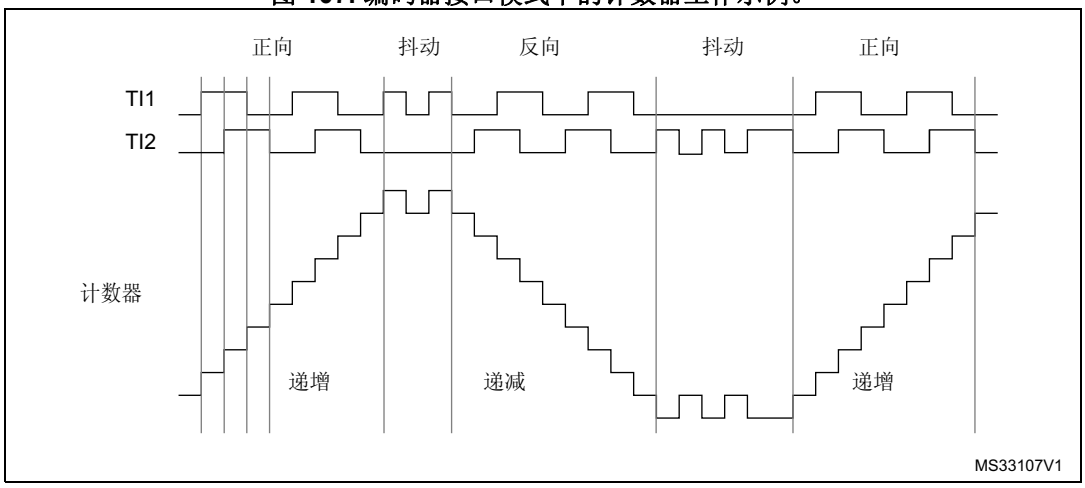
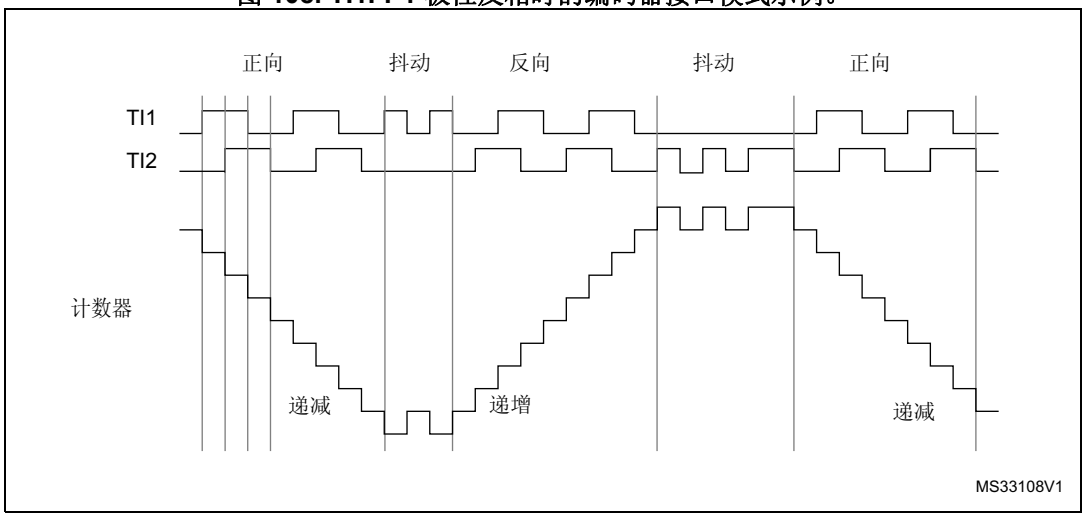


图 108 举例说明 TI1FP1 极性反相时计数器的行为（除 CC1P=“1”外，其他配置与上例相同）。

图 108. TI1FP1 极性反相时的编码器接口模式示例。



定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕获信号必须为周期性信号，可以由另一个定时器产生）；还可以通过 DMA 请求读取计数器值。

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 (UIF) 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可按原子方式读取计数器值以及由 UIFCPY 标志发出的更新事件情况。这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化角速度的计算。

UIF 和 UIFCPY 标志之间的置位没有延迟。

在 32 位定时器实现中，当 IUFREMAP 位置 1 时，计数器的位 31 在读访问时由 UIFCPY 标志覆盖（计数器的最高有效位只能在写模式下访问）。

15.3.23 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时器计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可按原子方式读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

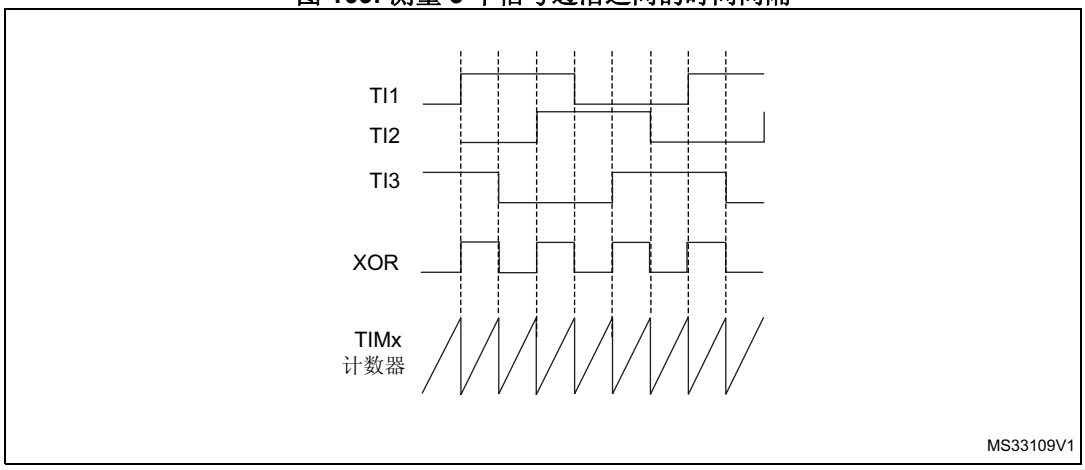
UIF 和 UIFCPY 标志使能之间没有延迟。

15.3.24 定时器输入异或功能

通过 TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 到 TIMx_CH3 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。这样便于测量两个输入信号边沿之间的间隔（如下面的*图 109* 所示）。

图 109. 测量 3 个信号边沿之间的时间间隔



15.3.25 连接霍尔传感器

可通过用于产生电机驱动 PWM 信号的高级控制定时器 (TIM1) 以及*图 110* 中称为“接口定时器”的另一个定时器 TIMx (TIM3)，实现与霍尔传感器的连接。3 个定时器输入引脚（CC1、CC2 和 CC3）通过异或门连接到 TI1 输入通道（通过将 TIMx_CR2 寄存器中的 TI1S 位置 1 来选择），并由“接口定时器”进行捕获。

从模式控制器配置为复位模式；从输入为 TI1F_ED。这样，每当 3 个输入中有一个输入发生切换时，计数器会从 0 开始重新计数。这样将产生由霍尔输入的任何变化而触发的时基。

在“接口定时器”上，捕获/比较通道 1 配置为捕获模式，捕获信号为 TRC（请参见*第 294 页的图 83：捕获/比较通道（例如：通道 1 输入阶段）*）。捕获值对应于输入上两次变化的间隔时间，可提供与电机转速相关的信息。

“接口定时器”可用于在输出模式下产生脉冲，以通过触发 COM 事件更改高级控制定时器 (TIM1) 各个通道的配置。TIM1 定时器用于生成电机驱动 PWM 信号。为此，必须对接口定时器通道进行编程，以便在编程的延迟过后产生正脉冲（在输出比较或 PWM 模式中）。该脉冲通过 TRGO 输出发送到高级控制定时器 (TIM1)。

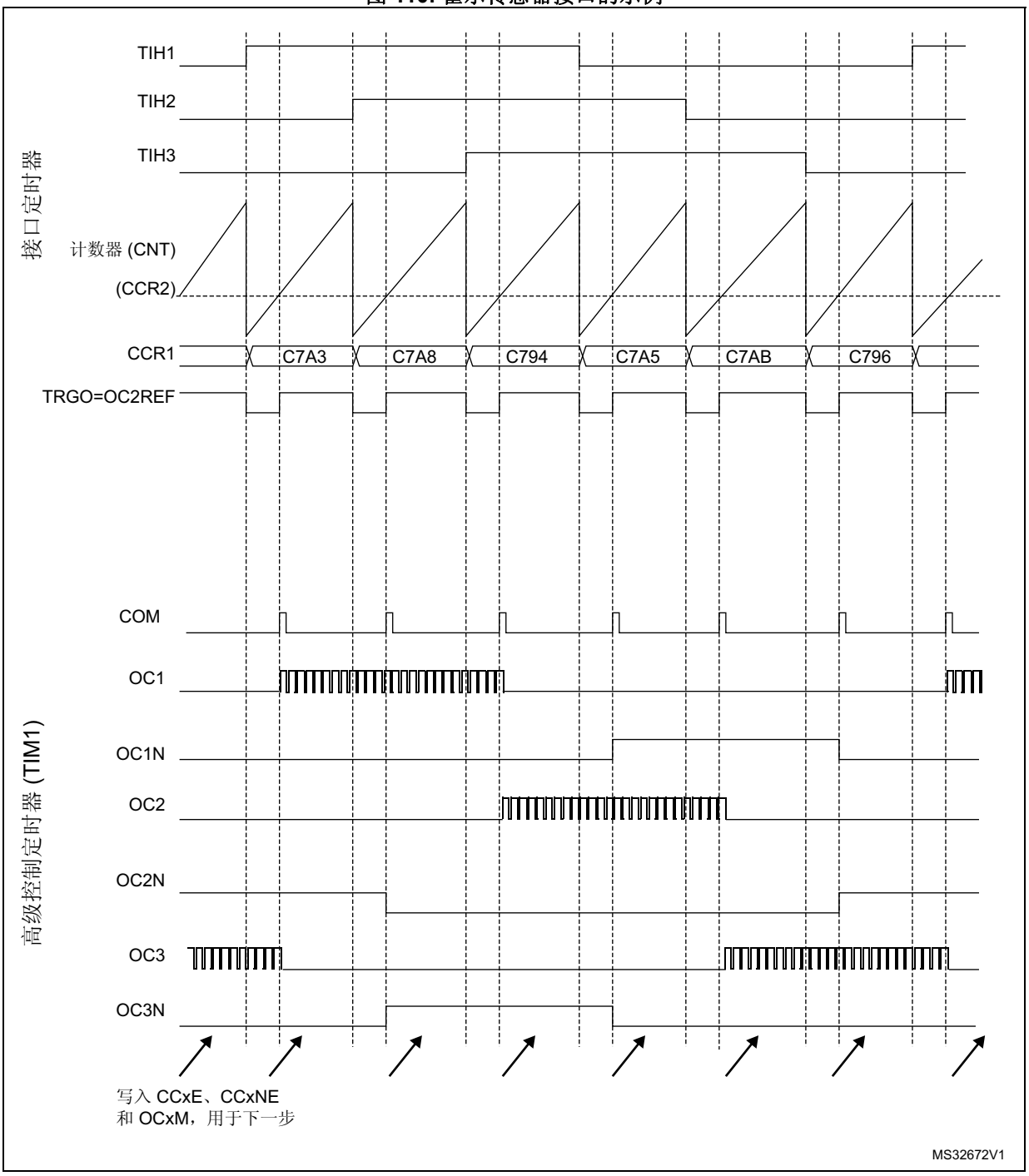
示例：霍尔输入与一个 TIMx 定时器相连接，每当霍尔输入发生更改，需要在所编程的延迟过后更改高级控制定时器 TIM1 的 PWM 配置。

- 向 TIMx_CR2 寄存器的 TI1S 位写入“1”，使 3 个定时器输入经过异或运算后进入 TI1 输入通道。
- 时基编程：向 TIMx_ARR 写入其最大值（计数器必须通过 TI1 的变化清零）。设置预分频器，以得到最大计数器周期，该周期长于传感器上两次变化的间隔时间。
- 将通道 1 编程为捕获模式（选择 TRC）：向 TIMx_CCMR1 寄存器的 CC1S 位写入“01”。如果需要，还可以编程数字滤波器。
- 将通道 2 编程为 PWM 2 模式，并具有所需延迟：向 TIMx_CCMR1 寄存器的 OC2M 位写入“111”，CC2S 位写入“00”。
- 选择 OC2REF 作为 TRGO 上的触发输出：向 TIMx_CR2 寄存器的 MMS 位写入“101”。

在高级控制定时器 TIM1 中，必须选择正确的 ITR 输入作为触发输入，定时器编程为可产生 PWM 信号，捕获/比较控制信号进行预装载（TIMx_CR2 寄存器的 CCPC=1），并且 COM 事件由触发输入控制（TIMx_CR2 寄存器中 CCUS=1）。发生 COM 事件后，在 PWM 控制位（CCxE、OCxM）中写入下一步的配置，此操作可在由 OC2REF 上升沿产生的中断子程序中完成。

图 110 为本示例的示意图。

图 **110.** 霍尔传感器接口的示例



15.3.26 定时器同步

TIMx 定时器从内部连接在一起，以实现定时器同步或链接。有关详细信息，请参见*第 16.3.19 节：定时器同步*。它们可在以下几种模式下实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有带预装特性的寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

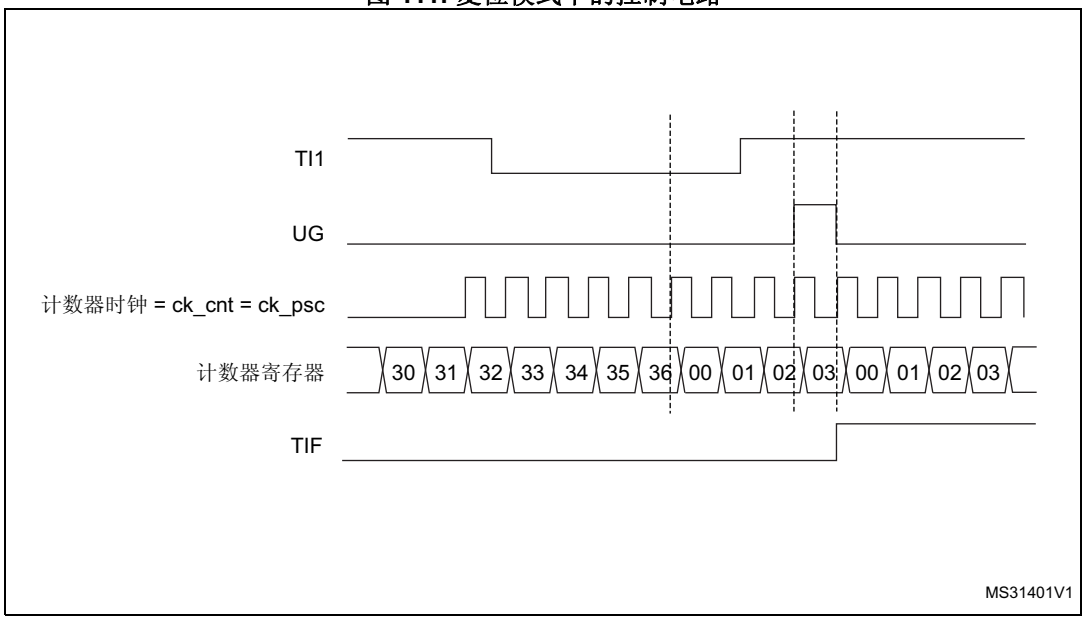
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

- 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S = 01。在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP=“0”，验证极性（仅检测上升沿）。
- 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS= 00101，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 111. 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

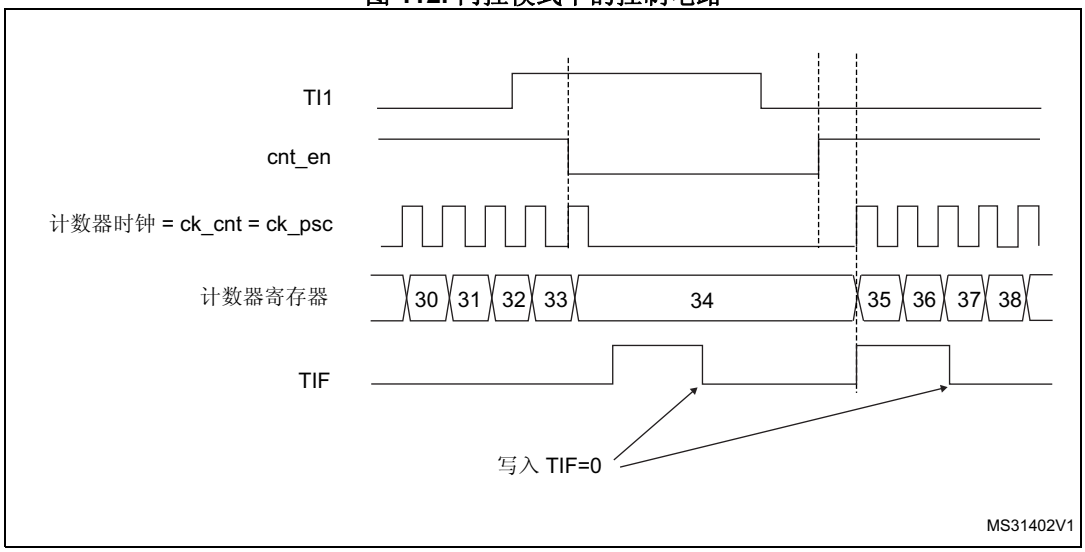
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

- 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=1 和 CC1NP=“0”，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS= 00101，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 112. 门控模式下的控制电路



从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

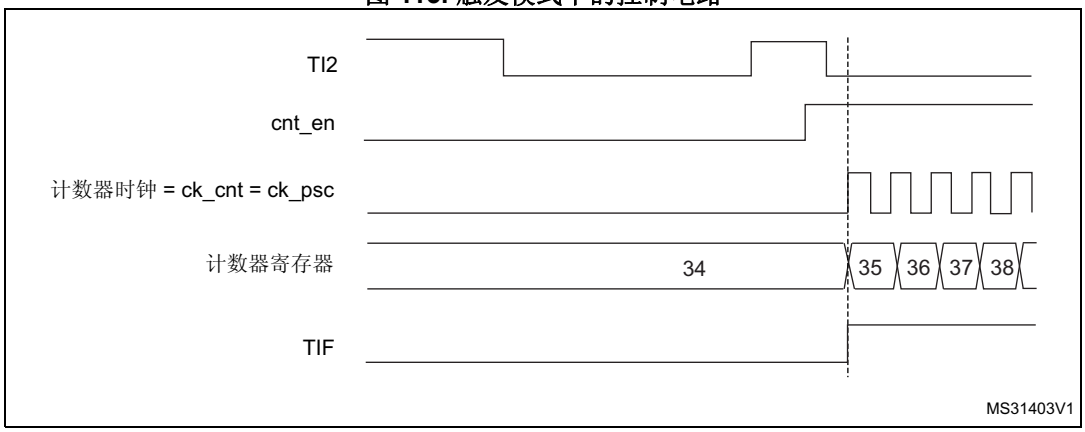
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

- 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=00110，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 113. 触发模式下的控制电路



从模式：复位 + 触发组合模式

在这种情况下，在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个定时器更新事件，并启动计数器。

该模式用于单脉冲模式。

从模式：外部时钟模式 2 + 触发模式

外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下，ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个输入作为触发输入。不建议通过 TIMx_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

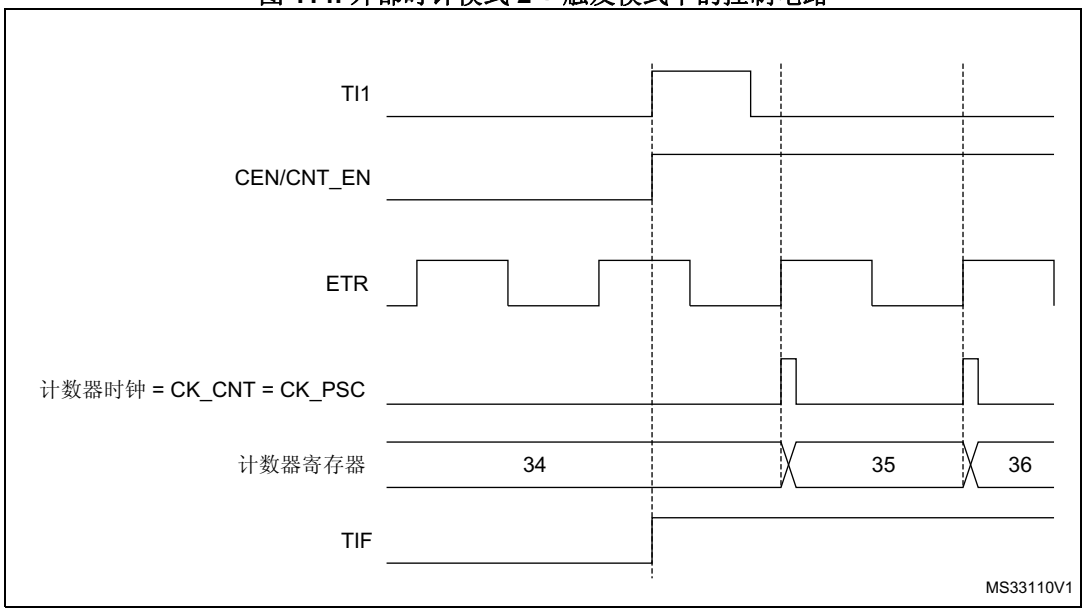
在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

1. 通过对 TIMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：
   - ETF = 0000：无滤波器
   - ETPS = 00：禁止预分频器
   - ETP = 0：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。
2. 如下配置通道 1，以检测 TI 的上升沿：
   - IC1F = 0000：无滤波器
   - 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
   - TIMx_CCMR1 寄存器中 CC1S = 01，选择输入捕获源。
   - TIMx_CCER 寄存器中 CC1P = 0 且 CC1NP = 0，以确定极性（仅检测上升沿）。
3. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS= 00101，选择 TI1 作为输入源。

TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。

ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。

图 114. 外部时钟模式 2 + 触发模式下的控制电路



注： *必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。*

15.3.27 ADC 同步

定时器可通过多种内部信号产生 ADC 触发事件，例如复位、使能或比较事件。也可生成由内部边沿检测器发出的脉冲，例如：

- OC4ref 的上升沿和下降沿
- OC5ref 上的上升沿或 OC6ref 上的下降沿

在重定向到 ADC 的 TRGO2 内部线路上发出触发信号。共有 16 个可能的事件，它们可通过 TIMx_CR2 寄存器中的 MMS2[3:0] 位选择。

第 304 页的图 94 给出了三相电机驱动的应用示例。

注： *必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。*

注： *必须先使能 ADC 时钟，才能从主定时器接收事件；从定时器接收触发信号时，不得实时更改 ADC 时钟。*

15.3.28 DMA 分组传送模式

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的一部分多次重新编程而无需软件开销，但也可以用于定期地一次性地批量读取定时器的多个寄存器的内容。

DMA 控制器访问目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（分组）。每次写入 TIMx_DMAR 寄存器都会重定向到定时器其中的一个寄存器。

TIMx_DCR 寄存器中的 DBL[4:0] 位设置 DMA 分组传送长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次分组传送，DBL 即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA[4:0] 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量：

示例：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

例如，定时器 DMA 分组传送功能用于在发生更新事件后将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   - DMA 通道外设地址为 DMAR 寄存器地址。
   - DMA 通道存储器地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
   - 要传输的数据量 = 3（参见下文注释）。
   - 禁止循环模式。
2. 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器：
   DBL = 3 次传输，DBA = 0xE。
3. 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 启动 TIMx 计数器。
5. 使能 DMA 通道。

本例适用于每个 CCRx 寄存器只更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。

注： *可以将空值写入保留的寄存器中。*

15.3.29 调试模式

当微控制器进入调试模式（Cortex®-M0+ 内核停止）时，TIMx 计数器会根据 DBG 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。

为了安全起见，当计数器停止时，输出被禁止（就像 MOE 位被复位一样）。可以将输出强制变为无效状态（OSSI 位 = 1），或者通过 GPIO 控制器（OSSI 位 = 0）来控制输出，通常将其强制为高阻态。

更多详细信息，请参见调试支持 (DBG) 部分。

15.4 TIM1 寄存器

有关寄存器说明中使用的缩写，请参见相应列表。

外设寄存器可按半字（16 位）或字（32 位）访问。

15.4.1 TIM1 控制寄存器 1 (TIM1_CR1)

TIM1 control register 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
Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
				rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)

0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)

该位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器（ETR、TIx）所使用的采样时钟 (t_{DTS}) 之间的分频比：

00：$t_{DTS}=t_{CK\_INT}$
01：$t_{DTS}=2*t_{CK\_INT}$
10：$t_{DTS}=4*t_{CK\_INT}$
11：保留，不要设置成此值

注： $t_{DTS} = 1/f_{DTS}$，$t_{CK\_INT} = 1/f_{CK\_INT}$。

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)

0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

位 6:5 **CMS[1:0]**：中心对齐模式选择 (Center-aligned mode selection)

00：边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。
01：中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。
10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。
11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志都会置 1。

注： 只要计数器处于使能状态 (CEN=1)，就不得从边沿对齐模式切换为中心对齐模式。

位 4 **DIR**：方向 (Direction)

0：计数器递增计数
1：计数器递减计数

注： 当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。

位 3 **OPM**：单脉冲模式 (One pulse mode)

0：计数器在发生更新事件时不会停止计数

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

位 2 **URS**：更新请求源 (Update request source)

该位由软件置 1 和清零，用以选择 UEV 事件源。

0： 使能时，所有以下事件都会生成更新中断或 DMA 请求。此类事件包括：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。

位 1 **UDIS**：更新禁止 (Update disable)

该位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0： 使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

带缓冲的寄存器被加载为预装值。

1： 禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注： 只有事先通过软件将 CEN 位置 1，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1。

15.4.2 TIM1 控制寄存器 2 (TIM1_CR2)

TIM1 control register 2

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS2[3:0]				Res.	OIS6	Res.	OIS5
								rw	rw	rw	rw		rw		rw

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

位 31:24 保留，必须保持复位值。

位 23:20 **MMS2[3:0]**：主模式选择 2 (Master mode selection 2)

这些位可选择将发送到 ADC 以实现同步的信息 (TRGO2)。这些位的组合如下：

0000： **复位**——对 TIMx_EGR 寄存器中 UG 位的置 1 操作产生触发输出信号 (TRGO2)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO2 上的信号相比实际复位会有延迟。

0001： **使能**——启动计数器时对 CNT_EN 位的置 1 操作产生触发输出信号 (TRGO2）。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。在门控模式下，计数器使能信号由 CEN 控制位与触发输入电平的逻辑与运算结果决定。当计数器使能信号由触发输入控制时，TRGO2 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。

0010： **更新**——选择更新事件作为触发输出 (TRGO2)。例如，主定时器可用作从定时器的预分频器。

0011： **比较脉冲**——CC1IF 标志置 1 时（即使已为高），只要发生捕获或比较匹配，触发输出 (TRGO2) 都会发送一个正脉冲。

0100： **比较**——OC1REFC 信号用作触发输出 (TRGO2)

0101： **比较**——OC2REFC 信号用作触发输出 (TRGO2)

0110： **比较**——OC3REFC 信号用作触发输出 (TRGO2)

0111： **比较**——OC4REFC 信号用作触发输出 (TRGO2)

1000： **比较**——OC5REFC 信号用作触发输出 (TRGO2)

1001： **比较**——OC6REFC 信号用作触发输出 (TRGO2)

1010： **比较脉冲**——OC4REFC 上升沿或下降沿时，TRGO2 上生成脉冲

1011： **比较脉冲**——OC6REFC 上升沿或下降沿时，TRGO2 上生成脉冲

1100： **比较脉冲**——OC4REFC 或 OC6REFC 上升沿时，TRGO2 上生成脉冲

1101： **比较脉冲**——OC4REFC 上升沿或 OC6REFC 下降沿时，TRGO2 上生成脉冲

1110： **比较脉冲**——OC5REFC 或 OC6REFC 上升沿时，TRGO2 上生成脉冲

1111： **比较脉冲**——OC5REFC 上升沿或 OC6REFC 下降沿时，TRGO2 上生成脉冲

注： 必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 19 保留，必须保持复位值。

位 18 **OIS6**：输出空闲状态 6（OC6 输出）(Output Idle state 6 (OC6 output))
请参见 OIS1 位

位 17 保留，必须保持复位值。

位 16 **OIS5**：输出空闲状态 5（OC5 输出）(Output Idle state 5 (OC5 output))
请参见 OIS1 位

位 15 保留，必须保持复位值。

位 14 **OIS4**：输出空闲状态 4（OC4 输出）(Output Idle state 4 (OC4 output))
请参见 OIS1 位

位 13 **OIS3N**：输出空闲状态 3（OC3N 输出）(Output Idle state 3 (OC3N output))
请参见 OIS1N 位

位 12 **OIS3**：输出空闲状态 3（OC3 输出）(Output Idle state 3 (OC3 output))
请参见 OIS1 位

位 11 **OIS2N**：输出空闲状态 2（OC2N 输出）(Output Idle state 2 (OC2N output))
请参见 OIS1N 位

位 10 **OIS2**：输出空闲状态 2（OC2 输出）(Output Idle state 2 (OC2 output))
请参见 OIS1 位

位 9 **OIS1N**：输出空闲状态 1（OC1N 输出）(Output Idle state 1 (OC1N output))
0：当 MOE=0 时，经过死区时间后 OC1N=0
1：当 MOE=0 时，经过死区时间后 OC1N=1
注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位即无法修改。

位 8 **OIS1**：输出空闲状态 1（OC1 输出）(Output Idle state 1 (OC1 output))
0：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0
1：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=1
注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位即无法修改。

位 7 **TI1S**：TI1 选择 (TI1 selection)
0：TIMx_CH1 引脚连接到 TI1 输入
1：TIMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入（异或组合）

位 6:4 **MMS[2:0]**：主模式选择 (Master mode selection)
这些位允许在主模式下将选定的信息发送到从定时器以实现同步 (TRGO)。这些位的组合如下：
000：**复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。
001：**使能**——启动计数器时对 CNT_EN 位的置 1 操作产生触发输出信号 (TRGO2）。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑与运算组合而成。当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。
010：**更新**——选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。
011：**比较脉冲**——CC1IF 标志置 1 时（即使已为高电平），只要发生捕获或比较匹配，触发输出都会发送一个正脉冲。(TRGO)。
100：**比较**——OC1REFC 信号用作触发输出 (TRGO)
101：**比较**——OC2REFC 信号用作触发输出 (TRGO)
110：**比较**——OC3REFC 信号用作触发输出 (TRGO)
111：**比较**——OC4REFC 信号用作触发输出 (TRGO)
注：　必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 3 **CCDS**：捕获/比较 DMA 选择 (Capture/compare DMA selection)
0：发生 CCx 事件时发送 CCx DMA 请求
1：发生更新事件时发送 CCx DMA 请求

位 2 **CCUS**：捕获/比较控制更新选择 (Capture/compare control update selection)
0：如果捕获/比较控制位 (CCPC=1) 进行预装载，仅通过将 COMG 位置 1 来对这些位进行更新
1：如果捕获/比较控制位 (CCPC=1) 进行预装载，可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。
注：　该位仅对具有互补输出的通道有效。

位 1 保留，必须保持复位值。

位 0 **CCPC**：捕获/比较预装载控制 (Capture/compare preloaded control)
0：CCxE、CCxNE 和 OCxM 位未进行预装载
1：CCxE、CCxNE 和 OCxM 位进行了预装载，写入这些位后，仅当发生换向事件 (COM)（COMG 位置 1 或在 TRGI 上检测到上升沿，取决于 CCUS 位）时才会对这些位进行更新。
注：　该位仅对具有互补输出的通道有效。

15.4.3 TIM1 从模式控制寄存器 (TIM1_SMCR)

TIM1 slave mode control register

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS[4:3]		Res.	Res.	Res.	SMS[3]
										rw	rw				rw

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

位 31:22 保留，必须保持复位值。

位 19:17 保留，必须保持复位值。

位 15 **ETP**：外部触发极性 (External trigger polarity)
该位可选择将 ETR 还是 $\overline{ETR}$ 用于触发操作
0：ETR 未反相，高电平或上升沿有效。
1：ETR 反相，低电平或下降沿有效。

位 14 **ECE**：外部时钟使能 (External clock enable)
该位可使能外部时钟模式 2。
0： 禁止外部时钟模式 2
1： 使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。

注： 将 ECE 位置 1 与选择外部时钟模式 1 并将 TRGI 连接到 ETRF（SMS=111 且 TS=00111）具有相同效果。
外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF（TS 位不得为 00111）。
如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。

位 13:12 **ETPS[1:0]**：外部触发预分频器 (External trigger prescaler)
外部触发信号 ETRP 频率不得超过 f_{CK_INT} 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。
00：预分频器关闭
01：2 分频 ETRP 频率
10：4 分频 ETRP 频率
11：8 分频 ETRP 频率

位 11:8 **ETF[3:0]**：外部触发滤波器 (External trigger filter)

该位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 7 **MSM**：主/从模式 (Master/slave mode)

0：无操作
1：当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。

位 21、20、6、5、4 **TS[4:0]**：触发选择 (Trigger selection)

该位域可选择将要用于同步计数器的触发输入。

00000：内部触发 0 (ITR0)
00001：内部触发 1 (ITR1)
00010：内部触发 2 (ITR2)
00011：内部触发 3 (ITR3)
00100：TI1 边沿检测器 (TI1F_ED)
00101：滤波后的定时器输入 1 (TI1FP1)
00110：滤波后的定时器输入 2 (TI2FP2)
00111：外部触发输入 (ETRF)
其他值：Reserved

有关各定时器 ITRx 含义的详细信息，请参见*第 334 页的表 65：TIM1 内部触发连接*。

注：　这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

位 3 **OCCS**：OCREF 清零选择 (OCREF clear selection)

该位用于选择 OCREF 清零源。

0：未连接 OCREF_CLR_INT（保留配置）
1：OCREF_CLR_INT 连接到 ETRF

位 16、2、1、0 **SMS[3:0]**：从模式选择 (Slave mode selection)

选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。

0000：禁止从模式——如果 CEN =“1”，预分频器时钟直接由内部时钟提供。
0001：编码器模式 1——计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。
0010：编码器模式 2——计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。
0011：编码器模式 3——计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个信号的电平。
0100：复位模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。
0101：门控模式——触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。
0110：触发模式——触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。
0111：外部时钟模式 1——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。
1000：复位 + 触发组合型模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。
1000 以上的代码：保留。

注： 如果将 TI1F_ED 选作触发输入 (TS=00100)，则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。

注： 必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

表 65. TIM1 内部触发连接

从 TIM	ITR0 (TS = 00000)	ITR1 (TS = 00001)	ITR2 (TS = 00010)	ITR3 (TS = 00011)
TIM1	未连接	未连接	TIM3	TIM17 OC1

15.4.4 TIM1 DMA/中断使能寄存器 (TIM1_DIER)

TIM1 DMA/interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15 保留，必须保持复位值。

位 14 **TDE**：触发信号 (TGRI) DMA 请求使能 (Trigger DMA request enable)
0：禁止触发信号 (TGRI) DMA 请求
1：使能触发信号 (TGRI) DMA 请求

位 13 **COMDE**：COM DMA 请求使能 (COM DMA request enable)
0：禁止 COM DMA 请求
1：使能 COM DMA 请求

位 12 **CC4DE**：捕获/比较 4 DMA 请求使能 (Capture/Compare 4 DMA request enable)
0：禁止 CC4 DMA 请求
1：使能 CC4 DMA 请求

位 11 **CC3DE**：捕获/比较 3 DMA 请求使能 (Capture/Compare 3 DMA request enable)
0：禁止 CC3 DMA 请求
1：使能 CC3 DMA 请求

位 10 **CC2DE**：捕获/比较 2 DMA 请求使能 (Capture/Compare 2 DMA request enable)
0：禁止 CC2 DMA 请求
1：使能 CC2 DMA 请求

位 9 **CC1DE**：捕获/比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)
0：禁止 CC1 DMA 请求
1：使能 CC1 DMA 请求

位 8 **UDE**：更新 DMA 请求使能 (Update DMA request enable)
0：禁止更新 DMA 请求
1：使能更新 DMA 请求

位 7 **BIE**：刹车中断使能 (Break interrupt enable)
0：禁止刹车中断
1：使能刹车中断

位 6 **TIE**：触发信号 (TGRI) 中断使能 (Trigger interrupt enable)
0：禁止触发信号 (TGRI) 中断
1：使能触发信号 (TGRI) 中断

位 5 **COMIE**：COM 中断使能 (COM interrupt enable)
0：禁止 COM 中断
1：使能 COM 中断

位 4 **CC4IE**：捕获/比较 4 中断使能 (Capture/Compare 4 interrupt enable)
0：禁止 CC4 中断
1：使能 CC4 中断

位 3 **CC3IE**：捕获/比较 3 中断使能 (Capture/Compare 3 interrupt enable)
0：禁止 CC3 中断
1：使能 CC3 中断

位 2 **CC2IE**：捕获/比较 2 中断使能 (Capture/Compare 2 interrupt enable)
0：禁止 CC2 中断
1：使能 CC2 中断

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断
1：使能 CC1 中断

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断
1：使能更新中断

15.4.5 TIM1 状态寄存器 (TIM1_SR)

TIM1 status register

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6IF	CC5IF
														rc_w0	rc_w0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位 31:18 保留，必须保持复位值。

位 17 **CC6IF**：比较 6 中断标志 (Compare 6 interrupt flag)
请参见 CC1IF 说明（注：通道 5 和通道 6 只能配置为输出）

位 16 **CC5IF**：比较 5 中断标志 (Compare 5 interrupt flag)
请参见 CC1IF 说明（注：通道 5 只能配置为输出）

位 15:14 保留，必须保持复位值。

位 13 **SBIF**：系统刹车中断标志 (System Break interrupt flag)
只要系统刹车输入变为有效状态，此标志便由硬件置 1。系统刹车输入无效后可通过软件对其清零。
此标志必须复位以使 PWM 重新开始工作。
0：未发生刹车事件。
1：在系统刹车输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。

位 12 **CC4OF**：捕获/比较 4 过捕获标志 (Capture/Compare 4 overcapture flag)
请参见 CC1OF 说明

位 11 **CC3OF**：捕获/比较 3 过捕获标志 (Capture/Compare 3 overcapture flag)
请参见 CC1OF 说明

位 10 **CC2OF**：捕获/比较 2 过捕获标志 (Capture/compare 2 overcapture flag)
请参见 CC1OF 说明

位 9 **CC1OF**：捕获/比较 1 过捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。
0：未检测到过捕获。
1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8 **B2IF**：刹车 2 中断标志 (Break 2 interrupt flag)
只要刹车 2 输入变为有效状态，此标志便由硬件置 1。刹车 2 输入无效后可通过软件对其清零。
0：未发生刹车事件。
1：在刹车 2 输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。

位 7 **BIF**：刹车中断标志 (Break interrupt flag)
只要刹车输入变为有效状态，此标志便由硬件置 1。刹车输入无效后可通过软件对其清零。
0：未发生刹车事件。
1：在刹车输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。

位 6 **TIF**：触发中断标志 (Trigger interrupt flag)

在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到 TRG 触发事件有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。

0：未发生触发事件。

1：触发中断挂起。

位 5 **COMIF**：COM 中断标志 (COM interrupt flag)

此标志在发生 COM 事件时（捕获/比较控制位 CCxE、CCxNE 和 OCxM 已更新时）由硬件置 1。但需要通过软件清零。

0：未发生 COM 事件。

1：COM 中断挂起。

位 4 **CC4IF**：捕获/比较 4 中断标志 (Capture/Compare 4 interrupt flag)

请参见 CC1IF 说明

位 3 **CC3IF**：捕获/比较 3 中断标志 (Capture/Compare 3 interrupt flag)

请参见 CC1IF 说明

位 2 **CC2IF**：捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)

请参见 CC1IF 说明

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/Compare 1 interrupt flag)

该标志由硬件置 1，但需要通过软件清零（输入捕获或输出比较模式）或通过读取 TIMx_CCR1 寄存器清零（仅限输入捕获模式）。

0：未发生比较匹配/输入捕获

1：发生了比较匹配/输入捕获

如果通道 CC1 配置为输出：当计数器 TIMx_CNT 的值与 TIMx_CCR1 的值匹配时，此标志置 1。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。在中心对齐模式下，可通过 3 种方式将此标志置 1，完整说明请参见 TIMx_CR1 寄存器中的 CMS 位。

如果通道 CC1 配置为输入：当 TIMx_CCR1 寄存器中捕获到计数器值时（如果通过设置 TIMx_CCER 中的 CC1P 和 CC1NP 位定义为边沿有效，则改为在 IC1 上检测到边沿时），该位置 1。

位 0 **UIF**：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

– TIMx_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器 = 0 时更新）。

– TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

– TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（请参见 *第 15.4.3 节：TIM1 从模式控制寄存器 (TIM1_SMCR)*）。

15.4.6 TIM1 事件生成寄存器 (TIM1_EGR)

TIM1 event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
							w	w	w	w	w	w	w	w	w

位 15:9 保留，必须保持复位值。

位 8 **B2G**：刹车 2 生成 (Break 2 generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：无操作
1：生成刹车 2 事件。MOE 位清零且 B2IF 标志置 1。使能后可发生相关中断。

位 7 **BG**：刹车生成 (Break generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：无操作
1：生成刹车事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 6 **TG**：触发生成 (Trigger generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：无操作
1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 5 **COMG**：捕获/比较控制更新生成 (Capture/Compare control update generation)
该位可通过软件置 1，并由硬件自动清零
0：无操作
1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位。
注： 该位仅对具有互补输出的通道有效。

位 4 **CC4G**：捕获/比较 4 生成 (Capture/Compare 4 generation)
请参见 CC1G 说明

位 3 **CC3G**：捕获/比较 3 生成 (Capture/Compare 3 generation)
请参见 CC1G 说明

位 2 **CC2G**：捕获/比较 2 生成 (Capture/Compare 2 generation)
请参见 CC1G 说明

位 1 **CC1G**：捕获/比较 1 生成 (Capture/Compare 1 generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：无操作
1：通道 1 上生成捕获/比较事件：
如果通道 CC1 配置为输出：
使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。
如果通道 CC1 配置为输入：
TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)
该位可通过软件置 1，并由硬件自动清零。
0：无操作
1：重新初始化计数器并生成一个寄存器更新事件。预分频器内部计数器也将清零（预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值 (TIMx_ARR)。

15.4.7 TIM1 捕获/比较模式寄存器 1 (TIM1_CCMR1)

TIM1 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获模式和输出比较模式下的功能均不同。可独立组合两种模式（例如，输入捕获模式下的通道 1 和输出比较模式下的通道 2）。

输入捕获模式：

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:12 **IC2F[3:0]**：输入捕获 2 滤波器 (Input capture 2 filter)
请参见 IC1F[3:0] 说明。

位 11:10 **IC2PSC[1:0]**：输入捕获 2 预分频器 (Input capture 2 prescaler)
请参见 IC1PSC[1:0] 说明。

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)
该位域定义通道方向（输入/输出）以及所使用的输入。
00：CC2 通道配置为输出
01：CC2 通道配置为输入，IC2 映射到 TI2 上
10：CC2 通道配置为输入，IC2 映射到 TI1 上
11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)
该位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：
0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC[1:0]**：输入捕获 1 预分频器 (Input capture 1 prescaler)

该位域定义 CC1 输入 (IC1) 的预分频比。只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获

01：每发生 2 个事件便执行一次捕获

10：每发生 4 个事件便执行一次捕获

11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

该位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

10：CC1 通道配置为输入，IC1 映射到 TI2 上

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

15.4.8 TIM1 捕获/比较模式寄存器 1 [复用] (TIM1_CCMR1)

TIM1 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获模式和输出比较模式下的功能均不同。可独立组合两种模式（例如，输入捕获模式下的通道 1 和输出比较模式下的通道 2）。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]
							rw								rw

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

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 **OC2CE**：输出比较 2 清零使能 (Output Compare 2 clear enable)

请参见 OC1CE 说明。

位 24、14:12 **OC2M[3:0]**：输出比较 2 模式 (Output Compare 2 mode)

请参见 OC1M[3:0] 说明。

位 11 **OC2PE**：输出比较 2 预装载使能 (Output Compare 2 preload enable)

请参见 OC1PE 说明。

位 10 **OC2FE**：输出比较 2 快速使能 (Output Compare 2 fast enable)

请参见 OC1FE 说明。

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)

该位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出

01：CC2 通道配置为输入，IC2 映射到 TI2 上

10：CC2 通道配置为输入，IC2 映射到 TI1 上

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。

位 7 **OC1CE**：输出比较 1 清零使能 (Output Compare 1 clear enable)

0：OC1Ref 不受 ocref_clr_int 信号影响

1：ocref_clr_int 信号（OCREF_CLR 输入或 ETRF 输入）上检测到高电平时，OC1Ref 立即清零

位 16，6:4 **OC1M[3:0]**：输出比较 1 模式 (Output Compare 1 mode)

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。当定时器用作软件时基时，可以使用该模式。在定时器操作期间使能冻结模式时，输出保持进入冻结状态前的状态（有效或无效）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便输出有效电平，否则输出无效电平。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便输出无效电平（OC1REF=“0”），否则输出有效电平（OC1REF=“1”）。

0111：PWM 模式 2——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便输出无效电平，否则输出有效电平。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便输出有效电平，否则输出无效电平。

1000：可再触发 OPM 模式 1——在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。

1001：可再触发 OPM 模式 2——在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。

1010：保留。

1011：保留。

1100：组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。

1101：组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。

1110：不对称 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

1111：不对称 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。

注： 在 PWM 模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。

注： 该位域将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成 COM 事件时，OC1M 有效位才会从预装载位获取新值。

注： OC1M[3] 位不是连续的，而是在位 16 中。

位 3 **OC1PE**：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到影子寄存器中。

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。

位 2 **OC1FE**：输出比较 1 快速使能 (Output Compare 1 fast enable)

该位可降低触发事件与定时器输出跳变之间的延迟。单脉冲模式下必须使用（TIMx_CR1 寄存器中的 OPM 位置 1），以便在启动触发后快速发送输出脉冲。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

该位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

10：CC1 通道配置为输入，IC1 映射到 TI2 上

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

15.4.9 TIM1 捕获/比较模式寄存器 2 (TIM1_CCMR2)

TIM1 capture/compare mode register 2

偏移地址：0x1C

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获模式和输出比较模式下的功能均不同。可独立组合两种模式（例如，输入捕获模式下的通道 1 和输出比较模式下的通道 2）。

输入捕获模式：

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4S[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:12 **IC4F[3:0]**：输入捕获 4 滤波器 (Input capture 4 filter)
请参见 IC1F[3:0] 说明。

位 11:10 **IC4PSC[1:0]**：输入捕获 4 预分频器 (Input capture 4 prescaler)
请参见 IC1PSC[1:0] 说明。

位 9:8 **CC4S[1:0]**：捕获/比较 4 选择 (Capture/Compare 4 selection)
该位域定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC4E = “0”），才可向 CC4S 位写入数据。

位 7:4 **IC3F[3:0]**：输入捕获 3 滤波器 (Input capture 3 filter)
请参见 IC1F[3:0] 说明。

位 3:2 **IC3PSC[1:0]**：输入捕获 3 预分频器 (Input capture 3 prescaler)
请参见 IC1PSC[1:0] 说明。

位 1:0 **CC3S[1:0]**：捕获/比较 3 选择 (Capture/compare 3 selection)
该位域定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC3E = “0”），才可向 CC3S 位写入数据。

15.4.10 TIM1 捕获/比较模式寄存器 2 [复用] (TIM1_CCMR2)

TIM1 capture/compare mode register 2

偏移地址：0x1C

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获模式和输出比较模式下的功能均不同。可独立组合两种模式（例如，输入捕获模式下的通道 1 和输出比较模式下的通道 2）。

输出比较模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4 CE	OC4M[2:0]			OC4 PE	OC4 FE	CC4S[1:0]		OC3 CE	OC3M[2:0]			OC3 PE	OC3 FE	CC3S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 **OC4CE**：输出比较 4 清零使能 (Output compare 4 clear enable)
请参见 OC1CE 说明。

位 24、14:12 **OC4M[3:0]**：输出比较 4 模式
请参见 OC3M[3:0] 说明。

位 11 **OC4PE**：输出比较 4 预装载使能 (Output compare 4 preload enable)
请参见 OC1PE 说明。

位 10 **OC4FE**：输出比较 4 快速使能 (Output compare 4 fast enable)
请参见 OC1FE 说明。

位 9:8 **CC4S[1:0]**：捕获/比较 4 选择 (Capture/Compare 4 selection)
该位域定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC4E = “0”），才可向 CC4S 位写入数据。

位 7 **OC3CE**：输出比较 3 清零使能 (Output compare 3 clear enable)
请参见 OC1CE 说明。

位 16，6:4 **OC3M[3:0]**：输出比较 3 模式 (Output compare 3 mode)
请参见 OC1M[3:0] 说明。

位 3 **OC3PE**：输出比较 3 预装载使能 (Output compare 3 preload enable)
请参见 OC1PE 说明。

位 2 **OC3FE**：输出比较 3 快速使能 (Output compare 3 fast enable)
请参见 OC1FE 说明。

位 1:0 **CC3S[1:0]**：捕获/比较 3 选择 (Capture/Compare 3 selection)
该位域定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC3E = “0”），才可向 CC3S 位写入数据。

15.4.11 TIM1 捕获/比较使能寄存器 (TIM1_CCER)

TIM1 capture/compare enable register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E
										rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21 **CC6P**：捕获/比较 6 输出极性 (Capture/Compare 6 output polarity)
请参见 CC1P 说明

位 20 **CC6E**：捕捉/比较 6 输出使能 (Capture/Compare 6 output enable)
请参见 CC1E 说明

位 19:18 保留，必须保持复位值。

位 17 **CC5P**：捕获/比较 5 输出极性 (Capture/Compare 5 output polarity)
请参见 CC1P 说明

位 16 **CC5E**：捕捉/比较 5 输出使能 (Capture/Compare 5 output enable)
请参见 CC1E 说明

位 15 **CC4NP**：捕获/比较 4 互补输出极性 (Capture/Compare 4 complementary output polarity)
请参见 CC1NP 说明

位 14 保留，必须保持复位值。

位 13 **CC4P**：捕获/比较 4 输出极性 (Capture/Compare 4 output polarity)
请参见 CC1P 说明

位 12 **CC4E**：捕捉/比较 4 输出使能 (Capture/Compare 4 output enable)
请参见 CC1E 说明

位 11 **CC3NP**：捕获/比较 3 互补输出极性 (Capture/Compare 3 complementary output polarity)
请参见 CC1NP 说明

位 10 **CC3NE**：捕捉/比较 3 互补输出使能 (Capture/Compare 3 complementary output enable)
请参见 CC1NE 说明

位 9 **CC3P**：捕获/比较 3 输出极性 (Capture/Compare 3 output polarity)。
请参见 CC1P 说明

位 8 **CC3E**：捕捉/比较 3 输出使能 (Capture/Compare 3 output enable)
请参见 CC1E 说明

位 7 **CC2NP**：捕获/比较 2 互补输出极性 (Capture/Compare 2 complementary output polarity)
请参见 CC1NP 说明

位 6 **CC2NE**：捕捉/比较 2 互补输出使能 (Capture/Compare 2 complementary output enable)
请参见 CC1NE 说明

位 5 **CC2P**：捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)
请参见 CC1P 说明

位 4 **CC2E**：捕捉/比较 2 输出使能 (Capture/Compare 2 output enable)
请参见 CC1E 说明

位 3 **CC1NP**：捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)
CC1 通道配置为输出：
0：OC1N 高电平有效。
1：OC1N 低电平有效。
CC1 通道配置为输入：
该位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3 且 CC1S="00"（通道配置为输出），该位立即变为不可写状态。
该位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NP 有效位才会从预装载位获取新值。

位 2 **CC1NE**：捕捉/比较 1 互补输出使能 (Capture/Compare 1 complementary output enable)
0：关闭——OC1N 未激活。OC1N 电平与 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位存在函数关系。
1：开启——基于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值，在相应输出引脚上输出 OC1N 信号。
该位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NE 有效位才会从预装载位获取新值。

位 1 **CC1P**：捕获/比较 1 输出极性 (Capture/Compare 1 output polarity)
0：OC1 高电平有效（输出模式）/边沿有效性选择（输入模式，见下文）
1：OC1 低电平有效（输出模式）/边沿有效性选择（输入模式，见下文）
CC1 通道配置为输入时，CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。
CC1NP=0，CC1P=0： 非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。
CC1NP=0，CC1P=1： 反相/下降沿触发。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。
CC1NP=1，CC1P=1： 非反相/上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。
CC1NP=1，CC1P=0： 该配置保留，不得使用。
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，该位立即变为不可写状态。
该位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。

位 0 **CC1E**：捕捉/比较 1 输出使能 (Capture/Compare 1 output enable)
0：禁止捕获模式/OC1 无效（见下文）
1：使能捕获模式/在相应输出引脚上输出 OC1 信号
当 CC1 通道配置为输出时，OC1 电平取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位，与 CC1E 位的状态无关。有关详细信息，请参见*表 66*。
注： 该位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1E 有效位才会从预装载位获取新值。

表 66. 具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	X	0	0	禁止输出（不由定时器驱动：高阻态） OCx=0、OCxN=0	
		0	0	1	禁止输出（不由定时器驱动：高阻态） OCx=0	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	禁止输出（不由定时器驱动：高阻态 OCxN=0
		X	1	1	OCREF + 极性 + 死区	OCREF 互补项（对 OCREF 进行“非”运算）+ 极性 + 死区
		1	0	1	关闭状态（输出使能为无效状态） OCx=CCxP	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		1	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	关闭状态（输出使能为无效状态） OCxN=CCxNP
0	0	X	X	X	禁止输出（不由定时器驱动：高阻态）。	
	1		0	0		
			0	1	关闭状态（输出使能为无效状态） 异步：OCx=CCxP、OCxN=CCxNP（如果触发 BRK 或 BRK2）。 随后（仅当触发 BRK 时才有效），如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，假定 OISx 和 OISxN 并没有都设置成 OCx 及 OCxN 的有效电平（否则在半桥配置下驱动开关时可能导致短路）。 **注意**：BRK2 只能在 OSSI = OSSR = 1 时使用。	
			1	0		
			1	1		

1. 如果一个通道的两个输出均未使用（由 GPIO 接管控制控制），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注： *与互补通道 OCx 和 OCxN 相连的外部 I/O 引脚的状态取决于通道 OCx 和 OCxN 的状态以及 GPIO 寄存器。*

15.4.12 TIM1 计数器 (TIM1_CNT)

TIM1 counter

偏移地址：0x24

复位值：0x0000 0000

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

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

位 31 **UIFCPY**：UIF 副本 (UIF copy)

该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。如果 TIMxCR1 中的 UIFREMAP 位复位，则位 31 保留，读数为 0。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

15.4.13 TIM1 预分频器 (TIM1_PSC)

TIM1 prescaler

偏移地址：0x28

复位值：0x0000

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

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)

计数器时钟频率 (CK_CNT) 等于 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到影子预分频器寄存器的值。

15.4.14 TIM1 自动重载寄存器 (TIM1_ARR)

TIM1 auto-reload register

偏移地址：0x2C

复位值：0xFFFF

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

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的更多详细信息，请参见*第 275 页的第 15.3.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

15.4.15 TIM1 重复计数器寄存器 (TIM1_RCR)

TIM1 repetition counter register

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[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **REP[15:0]**：重复计数器值 (Repetition counter value)

使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器向影子寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。

与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，向 TIMx_RCR 寄存器写入任何值都无影响。

这意味着 PWM 模式下 (REP+1) 相当于：

边沿对齐模式下的 PWM 周期数

中心对齐模式下的 PWM 半周期数。

15.4.16 TIM1 捕获/比较寄存器 1 (TIM1_CCR1)

TIM1 capture/compare register 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[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR1[15:0]**：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：CCR1 是捕获/比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际的捕获/比较寄存器 1）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出相应电平的值。

如果通道 CC1 配置为输入：CR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。只能读取 TIMx_CCR1 寄存器，无法对其进行编程。

15.4.17 TIM1 捕获/比较寄存器 2 (TIM1_CCR2)

TIM1 capture/compare register 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[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR2[15:0]**：捕获/比较 2 值 (Capture/Compare 2 value)

如果通道 CC2 配置为输出：CCR2 是捕获/比较寄存器 2 的预装载值。
如果没有通过 TIMx_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 2）。
有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。
如果通道 CC2 配置为输入：CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。只能读取 TIMx_CCR2 寄存器，无法对其进行编程。

15.4.18 TIM1 捕获/比较模式寄存器 3 (TIM1_CCR3)

TIM1 capture/compare register 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[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR3[15:0]**：捕获/比较值 (Capture/Compare value)

如果通道 CC3 配置为输出：CCR3 是捕获/比较寄存器 3 的预装载值。
如果没有通过 TIMx_CCMR2 寄存器中的 OC3PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 3）。
有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。

如果通道 CC3 配置为输入：CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。只能读取 TIMx_CCR3 寄存器，无法对其进行编程。

15.4.19 TIM1 捕获/比较寄存器 4 (TIM1_CCR4)

TIM1 capture/compare register 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[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR4[15:0]**：捕获/比较值 (Capture/Compare value)

如果通道 CC4 配置为输出：CCR4 是捕获/比较寄存器 4 的预装载值。
如果没有通过 TIMx_CCMR2 寄存器中的 OC4PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 4）。
有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。
如果通道 CC4 配置为输入：CCR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值。只能读取 TIMx_CCR4 寄存器，无法对其进行编程。

15.4.20 TIM1 刹车和死区寄存器 (TIM1_BDTR)

TIM1 break and dead-time register

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.	Res.	BK2BID	BKBID	BK2 DSRM	BK DSRM	BK2P	BK2E	BK2F[3:0]				BKF[3:0]			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注： *由于可以根据 LOCK 配置锁定位 BK2BID、BKBID、BK2DSRM、BKDSRM、BK2P、BK2E、BK2F[3:0]、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 的写操作，因此必须在第一次对 TIMx_BDTR 寄存器执行写访问时对这些位进行配置。*

位 31:30 保留，必须保持复位值。

位 29 **BK2BID**：刹车 2 双向 (Break2 bidirectional)
请参见 BKBID 说明

位 28 **BKBID**：双向刹车 (Break bidirectional)
0：刹车输入 BRK 为输入模式
1：刹车输入 BRK 为双向模式
在双向模式下（BKBID 位置 1），刹车输入配置为输入模式和开漏输出模式。任何激活的刹车事件都将使刹车输入上呈逻辑低电平，以向外部器件指示发生了内部刹车事件。
注： *只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。*
注： *对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 27 **BK2DSRM**：刹车 2 解除 (Break2 Disarm)
请参见 BKDSRM 说明

位 26 **BKDSRM**：刹车解除 (Break Disarm)
0：启动刹车输入 BRK
1：解除刹车输入 BRK
当没有刹车源被激活时，该位由硬件清零。
必须通过软件将 BKDSRM 位置 1 以释放双向输出控制（开漏输出处于高阻态），然后不断轮询该位，直到其由硬件复位，指示故障条件已消失。
注： *对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 25 **BK2P**：刹车 2 极性 (Break2 polarity)
0：刹车输入 BRK2 为低电平有效
1：刹车输入 BRK2 为高电平有效
注： *只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。*
注： *对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 24 **BK2E**：刹车 2 使能 (Break2 enable)
0：禁止刹车输入 BRK2
1：使能刹车输入 BRK2
注： *BRK2 只能在 OSSR = OSSI = 1 时使用。*
注： *编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，该位即无法修改。*
注： *对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 23:20 **BK2F[3:0]**：刹车 2 滤波器 (Break 2 filter)

该位域可定义 BRK2 输入的采样频率和适用于 BRK2 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，BRK2 异步工作
0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

注： 编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，该位即无法修改。

位 19:16 **BKF[3:0]**：刹车滤波器 (Break filter)

该位域可定义 BRK 输入的采样频率和适用于 BRK 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，BRK 异步工作
0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

注： 编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，该位即无法修改。

位 15 **MOE**：主输出使能 (Main output enable)

只要刹车输入（BRK 或 BRK2）为有效状态，该位便由硬件异步清零。该位由软件置 1，也可根据 AOE 位状态自动置 1。该位仅对配置为输出的通道有效。

0： 响应刹车事件（2 个）。禁止 OC 和 OCN 输出
响应刹车事件或向 MOE 写入 0 时：OC 和 OCN 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。

1： 如果 OC 和 OCN 输出的相应使能位（TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位）均置 1，则使能 OC 和 OCN 输出。

有关详细信息，请参见 OC/OCN 使能说明（*第 15.4.11 节：TIM1 捕获/比较使能寄存器 (TIM1_CCER)*）。

位 14 **AOE**：自动输出使能 (Automatic output enable)

0：MOE 只能由软件置 1

1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1（如果刹车输入 BRK 和 BRK2 均无效）

注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 13 **BKP**：刹车极性 (Break polarity)

0：刹车输入 BRK 为低电平有效

1：刹车输入 BRK 为高电平有效

注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

注：　对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 12 **BKE**：刹车使能 (Break enable)

该位可使能完整的刹车保护（包括连接到 bk_acth 的所有源和相应的 BKIN 源，如*图 98: 刹车事件和刹车 2 电路概述*所示）。

0：禁止刹车功能

1：使能刹车功能

注：　编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，该位即无法修改。

注：　对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 11 **OSSR**：运行模式下的关闭状态选择 (Off-state selection for Run mode)

该位在 MOE=1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。

有关详细信息，请参见 OC/OCN 使能说明（*第 15.4.11 节：TIM1 捕获/比较使能寄存器 (TIM1_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（输出控制器呈高阻态，相应输出由 GPIO 接管）。

1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平（输出仍由定时器控制）。

注：　编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，该位即无法修改。

位 10 **OSSI**：空闲模式下的关闭状态选择 (Off-state selection for Idle mode)

当由于刹车事件或软件写操作而使 MOE=0 时，该位作用于配置为输出的通道。

有关详细信息，请参见 OC/OCN 使能说明（*第 15.4.11 节：TIM1 捕获/比较使能寄存器 (TIM1_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 GPIO 逻辑接管）。

1：处于无效状态时，首先将 OC/OCN 输出强制为其无效电平，然后在死区后将其强制为空闲电平。定时器始终控制输出。

注：　编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，该位即无法修改。

位 9:8 **LOCK[1:0]**：锁定配置 (Lock configuration)

这些位用于针对软件错误提供写保护。

00：关闭锁定——不对任何位提供写保护。

01：锁定级别 1，此时无法对 TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_BDTR 寄存器中的 *BK2BID*、*BKBID*、*BK2DSRM*、*BKDSRM*、*BK2P*、*BK2E*、*BK2F[3:0]*、*BKF[3:0]*、*AOE*、*BKP*、*BKE*、*OSSI*、*OSSR* 和 *DTG[7:0]* 位执行写操作。

10：锁定级别 2，此时无法对锁定级别 1 中适用的各位、CC 极性位（TIMx_CCER 寄存器中的 CCxP/CCxNP 位，只要通过 CCxS 位将相关通道配置为输出）以及 OSSR 和 OSSI 位执行写操作。

11：锁定级别 3，此时无法对锁定级别 2 中适用的各位、CC 控制位（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

注：　复位后只能对 LOCK 位执行一次写操作。对 TIMx_BDTR 寄存器执行写操作后其中的内容将冻结，直到下一次复位。

位 7:0 **DTG[7:0]**：配置死区发生器 (Dead-time generator setup)

该位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。

DTG[7:5] = 0xx => DT = DTG[7:0] x t_{DTG}，其中 $t_{DTG} = t_{DTS}$。
DTG[7:5] = 10x => DT = (64 + DTG[5:0]) x t_{DTG}，其中 $t_{DTG} = 2 \times t_{DTS}$。
DTG[7:5] = 110 => DT = (32 + DTG[4:0]) x t_{DTG}，其中 $t_{DTG} = 8 \times t_{DTS}$。
DTG[7:5] = 111 => DT = (32 + DTG[4:0]) x t_{DTG}，其中 $t_{DTG} = 16 \times t_{DTS}$。
示例：如果 t_{DTS} = 125 ns (8 MHz)，则可能的死区值为：
0 到 15875 ns（步长为 125 ns），
16 us 到 31750 ns（步长为 250 ns），
32 µs 到 63 µs（步长为 1 µs），
64 us 到 126 us（步长为 2 us）

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位域即无法修改。

15.4.21 TIM1DMA 控制寄存器 (TIM1_DCR)

TIM1 DMA control register

偏移地址：0x48

复位值：0x0000

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

位 15:13 保留，必须保持复位值。

位 12:8 **DBL[4:0]**：DMA 分组传送长度 (DMA burst length)

该 5 位向量定义了 DMA 的传送长度（当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次分组传送），即传送次数。可按半字或字节进行传送（请参见下面的示例）。

00000：1 次传输
00001：2 次传输
00010：3 次传输
...
10001：18 次传输

示例：考虑以下传输：DBL = 7 字节，DBA = TIMx_CR1。

– 如果 DBL = 7 字节且 DBA = TIMx_CR1 表示待传送字节的地址，应通过以下公式给出传送的地址：

(TIMx_CR1 地址) + DBA + (DMA 索引)，其中 DMA 索引 = DBL

在本例中，将为 (TIMx_CR1 地址) + DBA 加上 7 个字节，得到将要复制数据的源/目标地址。这种情况下将向自以下地址开始的 7 个寄存器传输数据：(TIMx_CR1 地址) + DBA

根据 DMA 数据大小的配置，可能发生下面几种情况：

– 如果按半字配置 DMA 数据大小，则将向 7 个寄存器中的每一个传送 16 位数据。
– 如果按字节配置 DMA 数据大小，也将向 7 个寄存器传送数据：第一个寄存器包含第一个 MSB 字节，第二个寄存器包含第一个 LSB 字节，依此类推。因此，使用传送定时器时，还必须指定 DMA 传送的数据大小。

位 7:5 保留，必须保持复位值。

位 4:0 **DBA[4:0]**：DMA 基址 (DMA base address)

该 5 位向量定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：

00000：TIMx_CR1，
00001：TIMx_CR2，
00010：TIMx_SMCR，
...

15.4.22 TIM1 全传输 DMA 地址 (TIM1_DMAR)

TIM1 DMA address for full transfer

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
DMAB[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DMAB[31:0]**：DMA 分组传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器：(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

15.4.23 TIM1 捕获/比较模式寄存器 3 (TIM1_CCMR3)

TIM1 capture/compare mode register 3

偏移地址：0x54

复位值：0x0000 0000

通道 5 和通道 6 只能配置为输出。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC6 CE	OC6M[2:0]			OC6 PE	OC6FE	Res.	Res.	OC5 CE	OC5M[2:0]			OC5PE	OC5FE	Res.	Res.
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 **OC6CE**：输出比较 6 清零使能 (Output compare 6 clear enable)

请参见 OC1CE 说明。

位 24、14、13、12 **OC6M[3:0]**：输出比较 6 模式 (Output compare 6 mode)

请参见 OC1M 说明。

位 11 **OC6PE**：输出比较 6 预装载使能 (Output compare 6 preload enable)

请参见 OC1PE 说明。

位 10 **OC6FE**：输出比较 6 快速使能 (Output compare 6 fast enable)
请参见 OC1FE 说明。

位 9:8 保留，必须保持复位值。

位 7 **OC5CE**：输出比较 5 清零使能 (Output compare 5 clear enable)
请参见 OC1CE 说明。

位 16、6、5、4 **OC5M[3:0]**：输出比较 5 模式 (Output compare 5 mode)
请参见 OC1M 说明。

位 3 **OC5PE**：输出比较 5 预装载使能 (Output compare 5 preload enable)
请参见 OC1PE 说明。

位 2 **OC5FE**：输出比较 5 快速使能 (Output compare 5 fast enable)
请参见 OC1FE 说明。

位 1:0 保留，必须保持复位值。

15.4.24 TIM1 捕获/比较寄存器 5 (TIM1_CCR5)

TIM1 capture/compare register 5

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
GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR5[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **GC5C3**：通道 5 和通道 3 组合 (Group Channel 5 and Channel 3)
通道 3 输出上失真：
0：OC5REF 对 OC3REFC 无影响
1：OC3REFC 是 OC3REFC 和 OC5REF 的逻辑与运算结果
该位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR2 中选择了预装载功能）。
注： 也可在组合 PWM 信号上应用此失真。

位 30 **GC5C2**：通道 5 和通道 2 组合 (Group Channel 5 and Channel 2)
通道 2 输出上失真：
0：OC5REF 对 OC2REFC 无影响
1：OC2REFC 是 OC2REFC 和 OC5REF 的逻辑与运算结果
该位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR1 中选择了预装载功能）。
注： 也可在组合 PWM 信号上应用此失真。

位 29 **GC5C1**：通道 5 和通道 1 组合 (Group Channel 5 and Channel 1)

通道 1 输出上失真：

0：OC5REF 对 OC1REFC5 无影响

1：OC1REFC 是 OC1REFC 和 OC5REF 的逻辑与运算结果

该位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR1 中选择了预装载功能）。

注： 也可在组合 PWM 信号上应用此失真。

位 28:16 保留，必须保持复位值。

位 15:0 **CCR5[15:0]**：捕获/比较 5 值 (Capture/Compare 5 value)

CCR5 是捕获/比较寄存器 5 的预装载值。

如果没有通过 TIMx_CCMR3 寄存器中的 OC5PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 5）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC5 输出上发出信号的值。

15.4.25 TIM1 捕获/比较寄存器 6 (TIM1_CCR6)

TIM1 capture/compare register 6

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
CCR6[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR6[15:0]**：捕获/比较 6 值 (Capture/Compare 6 value)

CCR6 是捕获/比较寄存器 6 的预装载值。

如果没有通过 TIMx_CCMR3 寄存器中的 OC6PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 6）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC6 输出上发出信号的值。

15.4.26 TIM1 复用功能选项寄存器 1 (TIM1_AF1)

TIM1 alternate function option register 1

偏移地址：0x60

复位值：0x0000 0001

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

位 31:18 保留，必须保持复位值。

位 17:14 **ETRSEL[3:0]**：ETR 源选择 (ETR source selection)
这些位选择 ETR 输入源。
0000：ETR legacy mode
0011：ADC1 AWD1
0100：ADC1 AWD2
0101：ADC1 AWD3
其他值：Reserved
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，这些位即无法修改。

位 13:10 保留，必须保持复位值。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
该位选择 BKIN 复用功能输入有效电平，必须与 BKP 极性位一起编程。
0：BKIN 输入极性不反相（BKP=0 时低电平有效，BKP=1 时高电平有效）
1：BKIN 输入极性反相（BKP=0 时高电平有效，BKP=1 时低电平有效）
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 8:1 保留，必须保持复位值。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
该位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

注： 请参见图 77：TIM1 ETR 输入电路和图 98：刹车事件和刹车 2 电路概述。

15.4.27 TIM1 复用功能寄存器 2 (TIM1_AF2)

TIM1 Alternate function register 2

偏移地址：0x64

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	BK2 INP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BK2INE
						rw									rw

位 31:10 保留，必须保持复位值。

位 9 **BK2INP**：BRK2 BKIN2 输入极性 (BRK2 BKIN2 input polarity)
该位选择 BKIN2 复用功能输入有效电平，必须与 BK2P 极性位一起编程。
0：BKIN2 输入极性不反相（BK2P=0 时低电平有效，BK2P=1 时高电平有效）
1：BKIN2 输入极性反相（BK2P=0 时高电平有效，BK2P=1 时低电平有效）
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 8:1 保留，必须保持复位值。

位 0 **BK2INE**：BRK2 BKIN 输入使能 (BRK2 BKIN input enable)
该位使能定时器 BRK2 输入的 BKIN2 复用功能。BKIN2 输入与其他 BRK2 源进行“或”运算。
0：禁止 BKIN2 输入
1：使能 BKIN2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

注： 请参见图 98：刹车事件和刹车 2 电路概述。

15.4.28 TIM1 定时器输入选择寄存器 (TIM1_TISEL)

TIM1 timer input selection register

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.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **TI4SEL[3:0]**：选择 TI4[0] 到 TI4[15] 输入 (selects TI4[0] to TI4[15] input)
0000：TIM1_CH4 输入
其他值：Reserved

位 23:20 保留，必须保持复位值。

位 19:16 **TI3SEL[3:0]**：选择 TI3[0] 到 TI3[15] 输入 (selects TI3[0] to TI3[15] input)
0000：TIM1_CH3 输入
其他值：Reserved

位 15:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：选择 TI2[0] 到 TI2[15] 输入 (selects TI2[0] to TI2[15] input)
0000：TIM1_CH2 输入
其他值：Reserved

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM1_CH1 输入
其他值：Reserved

15.4.29 TIM1 寄存器映射

TIM1 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 67. TIM1 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIM1_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0	0	0	0	0	0	0	0
0x04	**TIM1_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS2[3:0]				Res.	OIS6	Res.	OIS5	Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS [2:0]			CCDS	CCUS	Res.	CCPC
	Reset value									0	0	0	0		0		0		0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x08	**TIM1_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS [4:3]		Res.	Res.	Res.	SMS[3]	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			OCCS	SMS[2:0]		
	Reset value											0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**TIM1_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**TIM1_SR**	Res.	Res.	Res	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6IF	CC5IF	Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value															0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**TIM1_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																								0	0	0	0	0	0	0	0	0
0x18	**TIM1_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	OC2CE	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM1_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**TIM1_CCMR2** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]	OC4CE	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM1_CCMR2 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]				IC4PSC [1:0]		CC4S [1:0]		IC3F[3:0]				IC3PSC [1:0]		CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**TIM1_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E	CC4NP	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	Reset value											0	0			0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 67. TIM1 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x24	**TIM1_CNT**	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIM1_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM1_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	**TIM1_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	**TIM1_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIM1_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**TIM1_CCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	**TIM1_CCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIM1_BDTR**	Res.	Res.	BK2BID	BKBID	BK2DSRM	BKDSRM	BK2P	BK2E	BK2F[3:0]				BKF[3:0]				MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DT[7:0]							
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**TIM1_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIM1_DMAR**	DMAB[31:0]																															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	**Reserved**	Res.																															
0x54	**TIM1_CCMR3** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]	OC6CE	OC6M [2:0]			OC6PE	OC6FE	Res.	Res.	OC5CE	OC5M [2:0]			OC5PE	OC5FE	Res.	Res.
	Reset value								0								0	0	0	0	0	0	0			0	0	0	0	0	0		
0x58	**TIM1_CCR5**	GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR5[15:0]															
	Reset value	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 67. TIM1 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x5C	**TIM1_CCR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR6[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM1_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res ETRSEL [3:0]								BKINP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINE
	Reset value															0	0	0	0					0									1
0x64	**TIM1_AF2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BK2INP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BK2INE
	Reset value																							0									1
0x68	**TIM1_TISEL**	Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]				Res.	Res.	Res.	Res.	TI2SEL[3:0]			Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

16 通用定时器 (TIM3)

16.1 TIM3 简介

通用定时器 TIM3 包含一个 16-bit 自动重载计数器，该计数器由可编程预分频器驱动。

该定时器用途广泛，其中包括测量输入信号的脉冲宽度（*输入捕获*）或生成输出波形（*输出比较和 PWM*）。

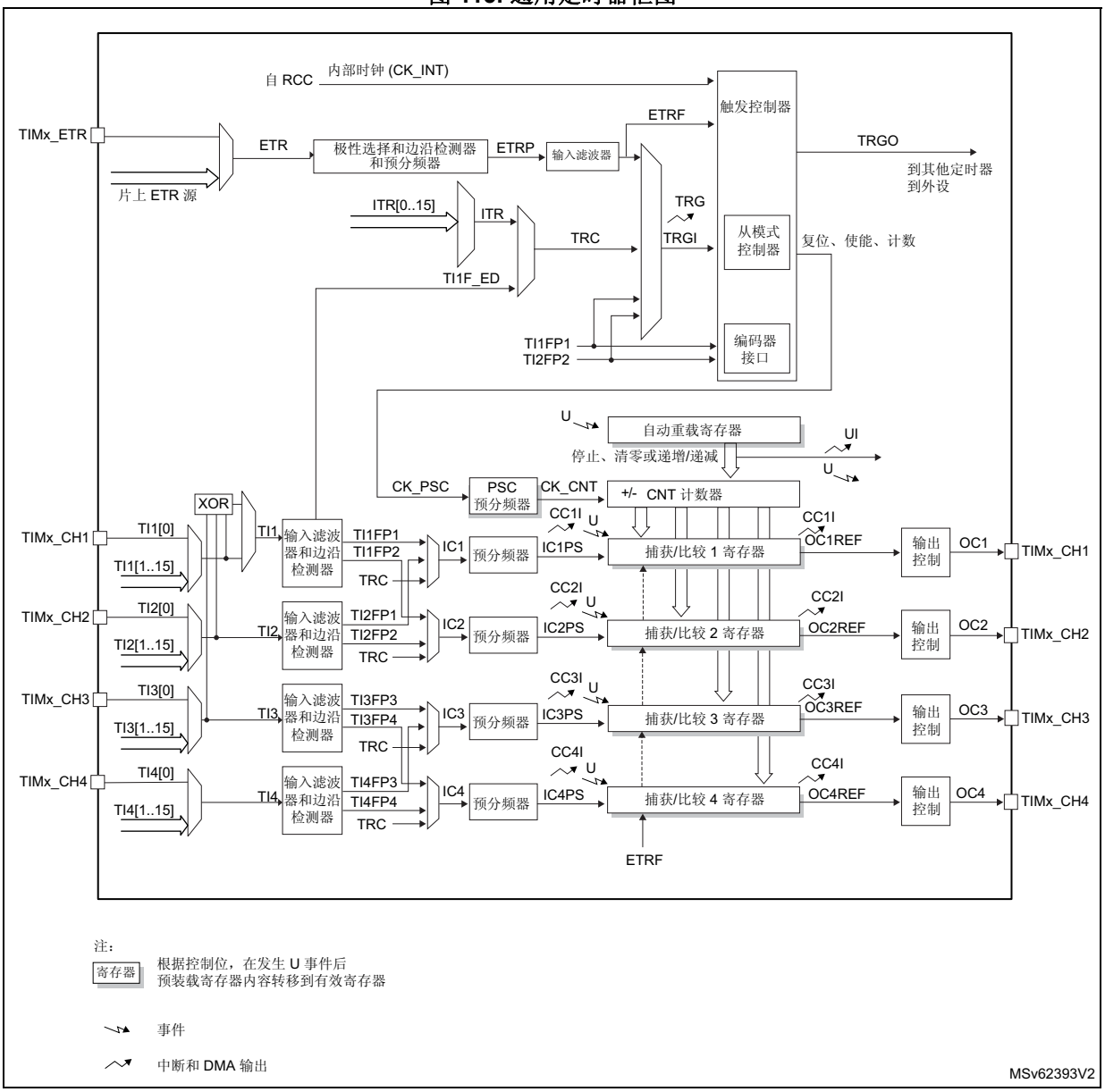
使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

16.2 TIM3 主要特性

通用 TIMx 定时器具有以下特性：

- 16 位 TIM3 递增、递减、递增/递减自动重载计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65535 之间。
- 多达 4 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿对齐模式和中心对齐模式）
  - 单脉冲模式输出
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路。
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较
- 支持定位用增量（正交）编码器和霍尔传感器电路
- 触发输入可用作外部时钟或用于逐周期电流管理

图 115. 通用定时器框图



16.3 TIM3 功能描述

16.3.1 时基单元

可编程定时器的主要模块由一个 16-bit 计数器及其相关的自动重载寄存器组成。计数器可递增计数、递减计数或进行递增递减计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以随时直接写入影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将产生更新事件。该更新事件也可由软件产生。下文将详细介绍各配置的更新事件的产生过程。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

请注意，实际的计数器使能信号 CNT_EN 在 CEN 置 1 的一个时钟周期后被置 1。

预分频器说明

预分频器可对计数器时钟进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 16 位/32 位寄存器（TIMx_PSC 寄存器）所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 116 和*图 117* 以一些示例说明在预分频比实时变化时计数器的行为：

图 116. 预分频器分频由 1 变为 2 时的计数器时序图

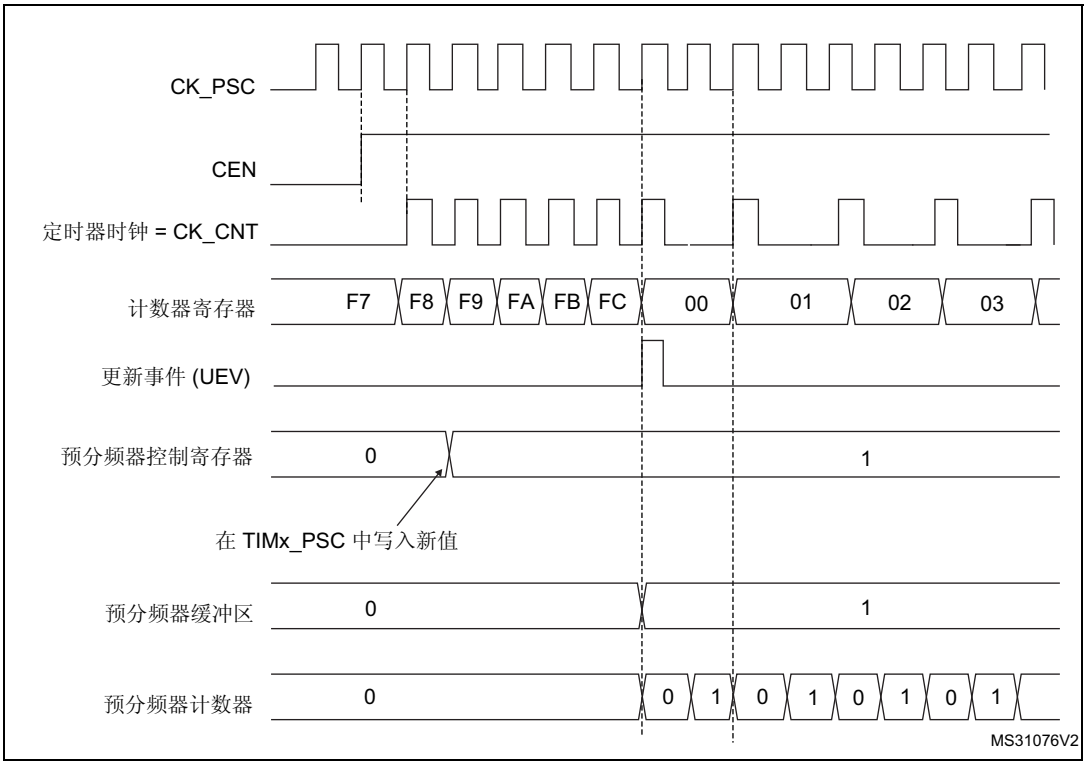
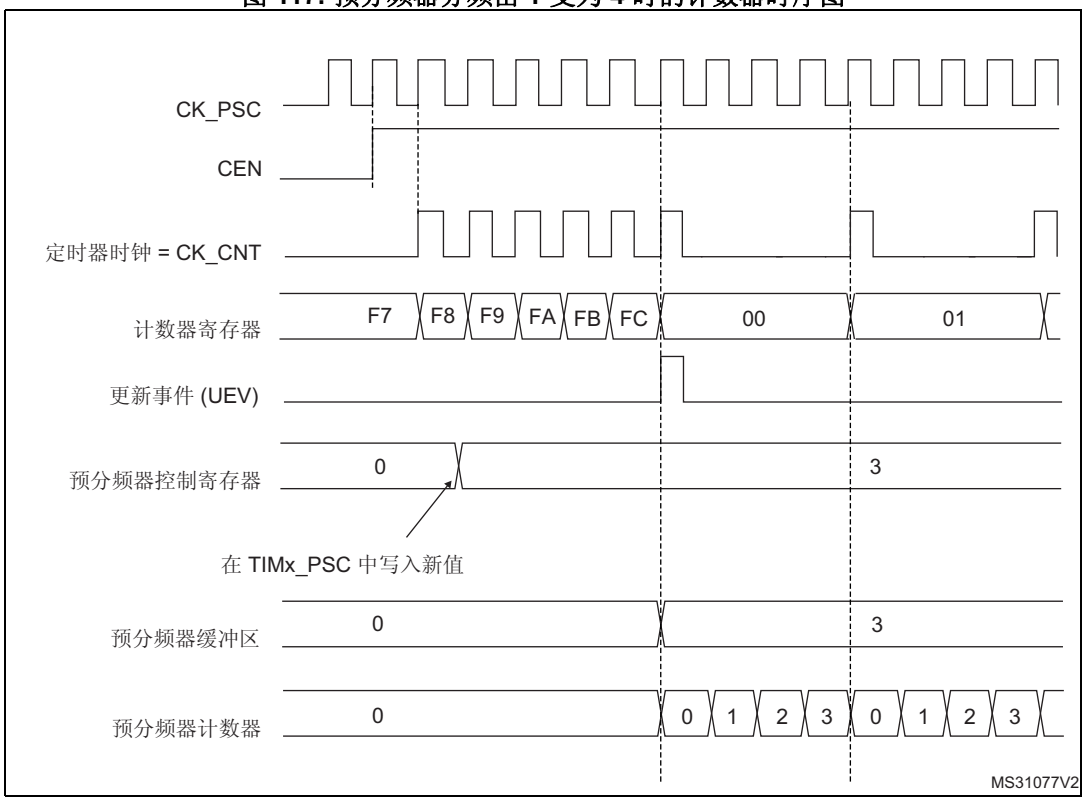


图 117. 预分频器分频由 1 变为 4 时的计数器时序图



16.3.2 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

每次发生计数器上溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。利用这一特性，在定时器的输入捕获操作时可以通过对 UG 位置 1 让计数器清零而不至于触发更新中断，也就避免了在捕获中断里触发更新中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）
- 使用预装载值 (TIMx_ARR) 更新自动重载影子寄存器

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 118. 计数器时序图，1 分频内部时钟

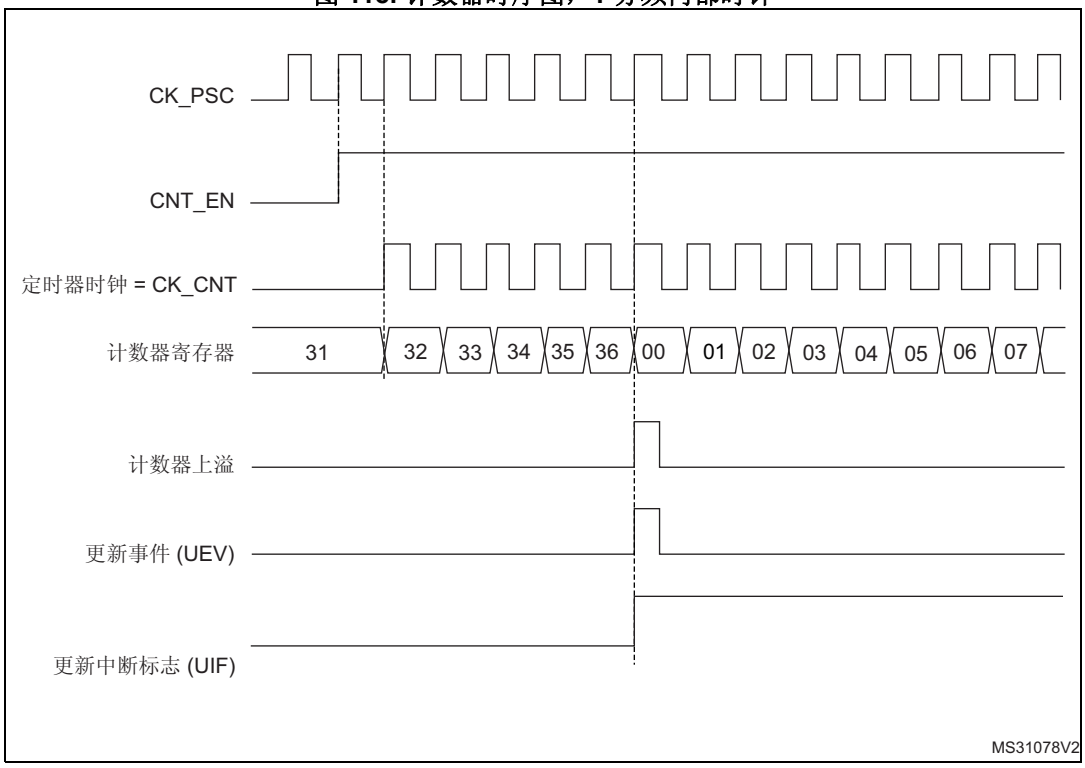


图 119. 计数器时序图，2 分频内部时钟

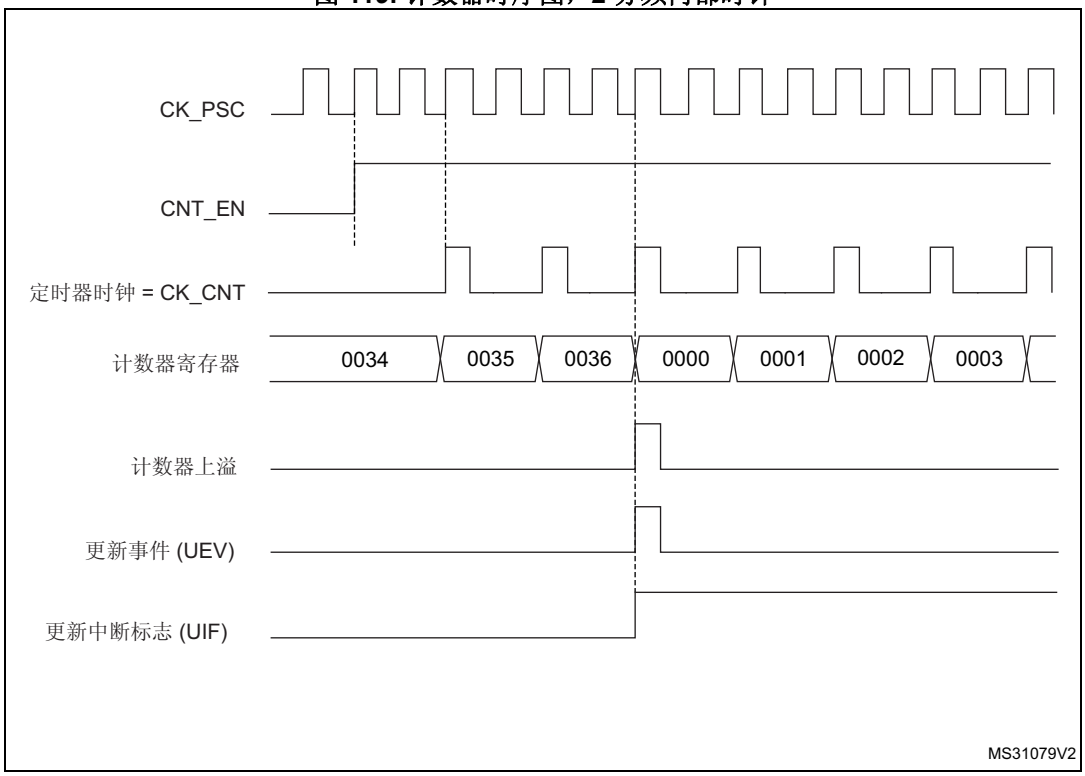


图 120. 计数器时序图，4 分频内部时钟

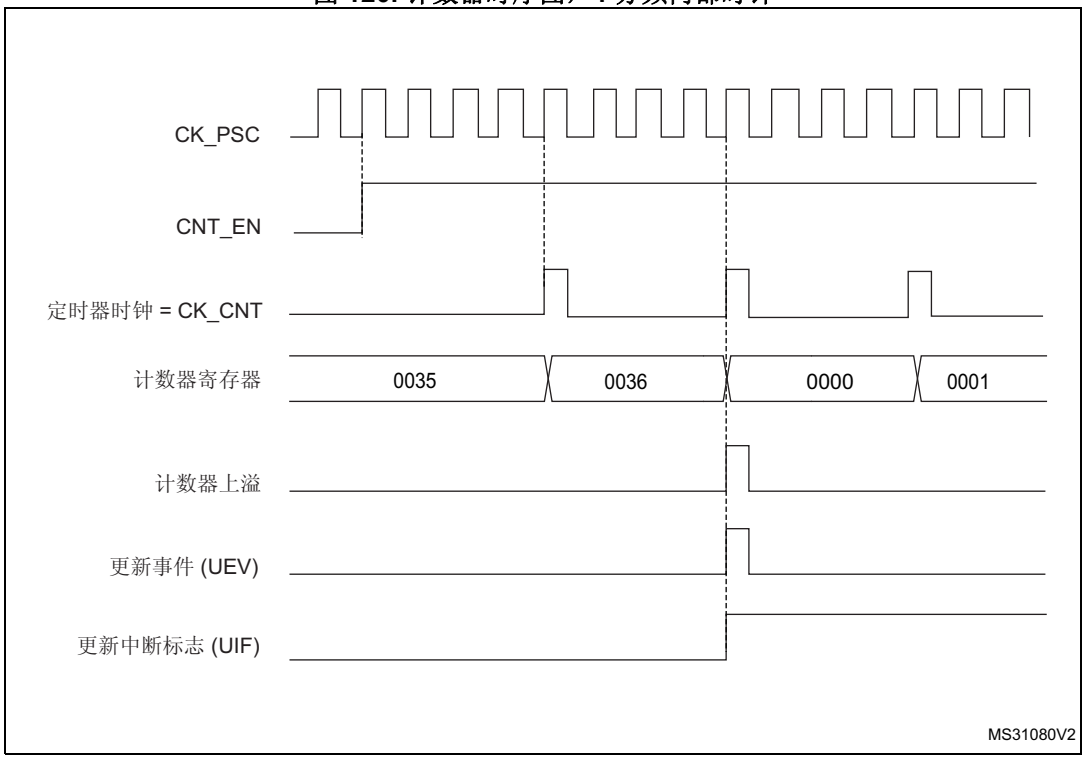


图 121. 计数器时序图，N 分频内部时钟

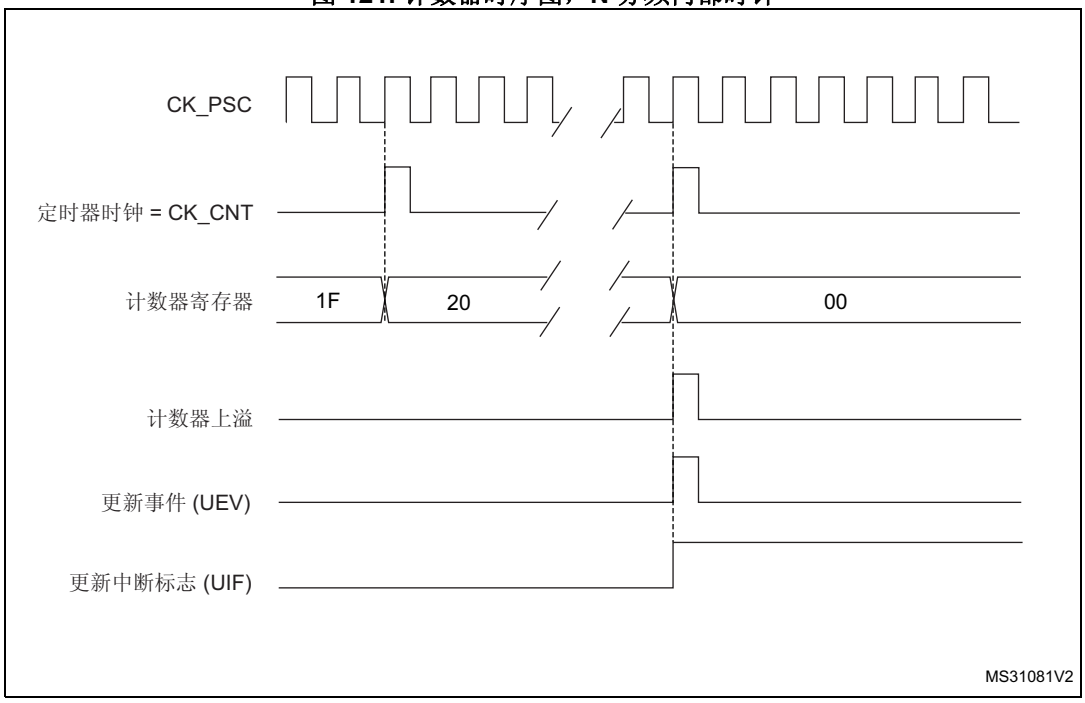


图 122. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

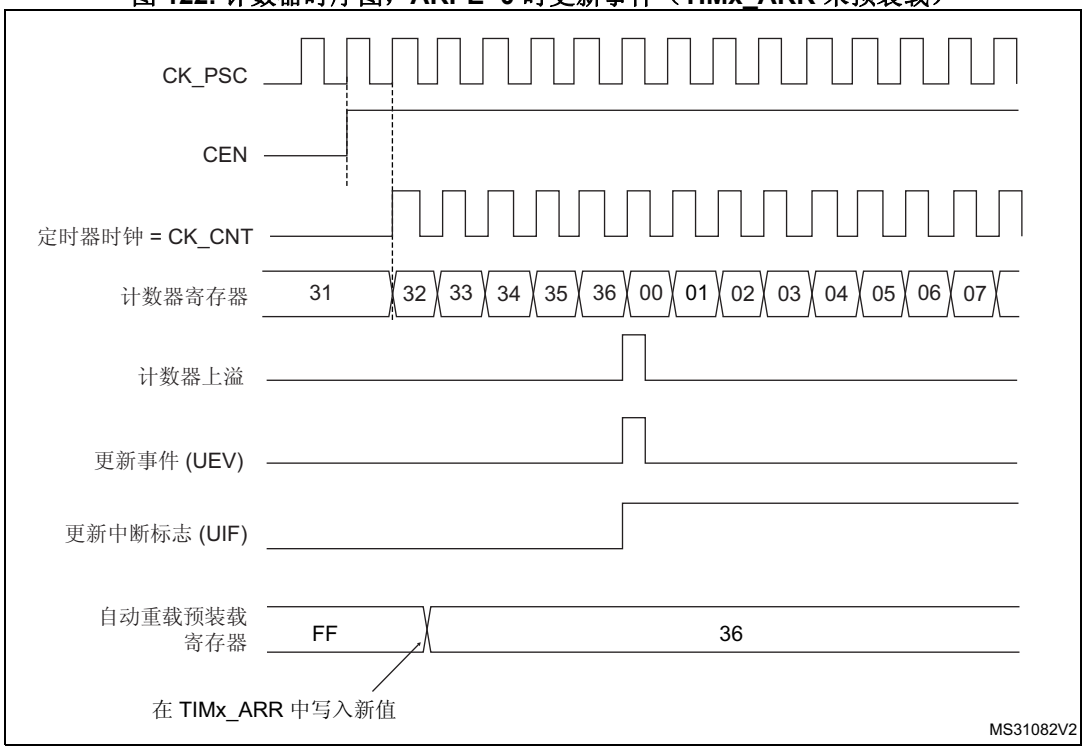
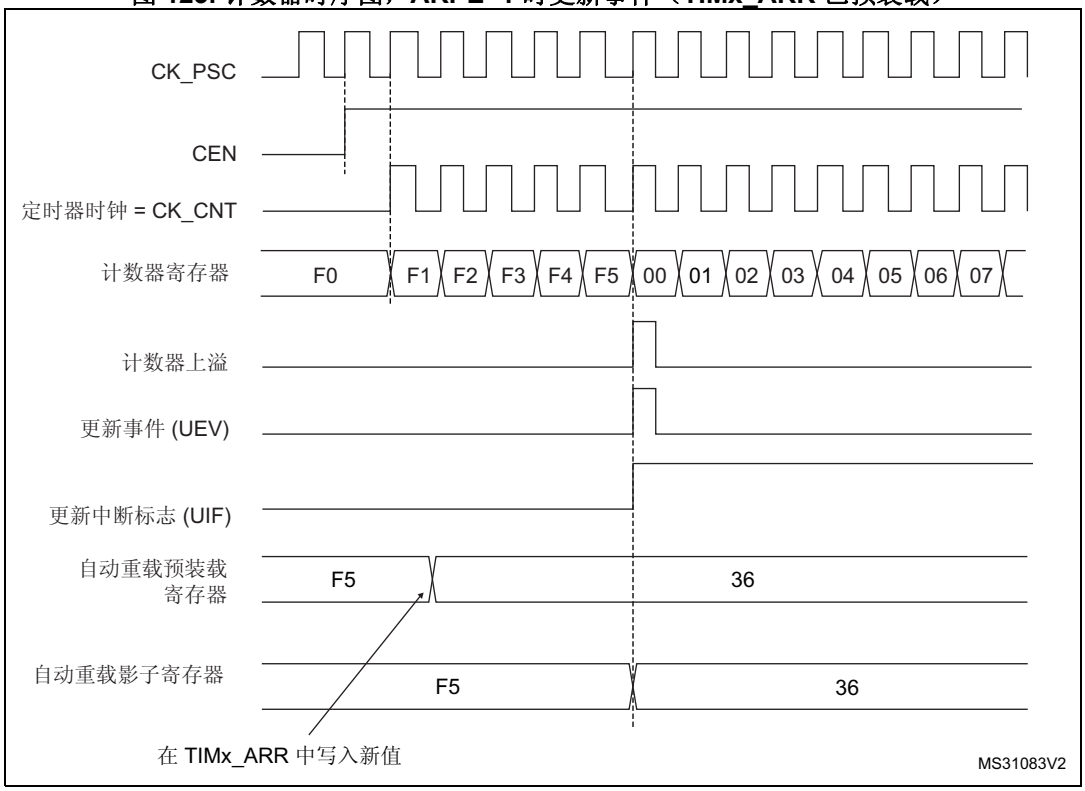


图 123. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载）



递减计数模式

在递减计数模式下，计数器从自动重载值（TIMx_ARR 寄存器的内容）开始递减计数到 0，然后重新从自动重载值开始计数并生成计数器下溢事件。

每次发生计数器下溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会从当前自动重载值开始重新计数，而预分频器计数器则从 0 开始重新计数（但预分频比保持不变）。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载影子寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，ARR 寄存器更新在计数器重载之前被更新，因此下一个周期就是预期的值。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 124. 计数器时序图，1 分频内部时钟

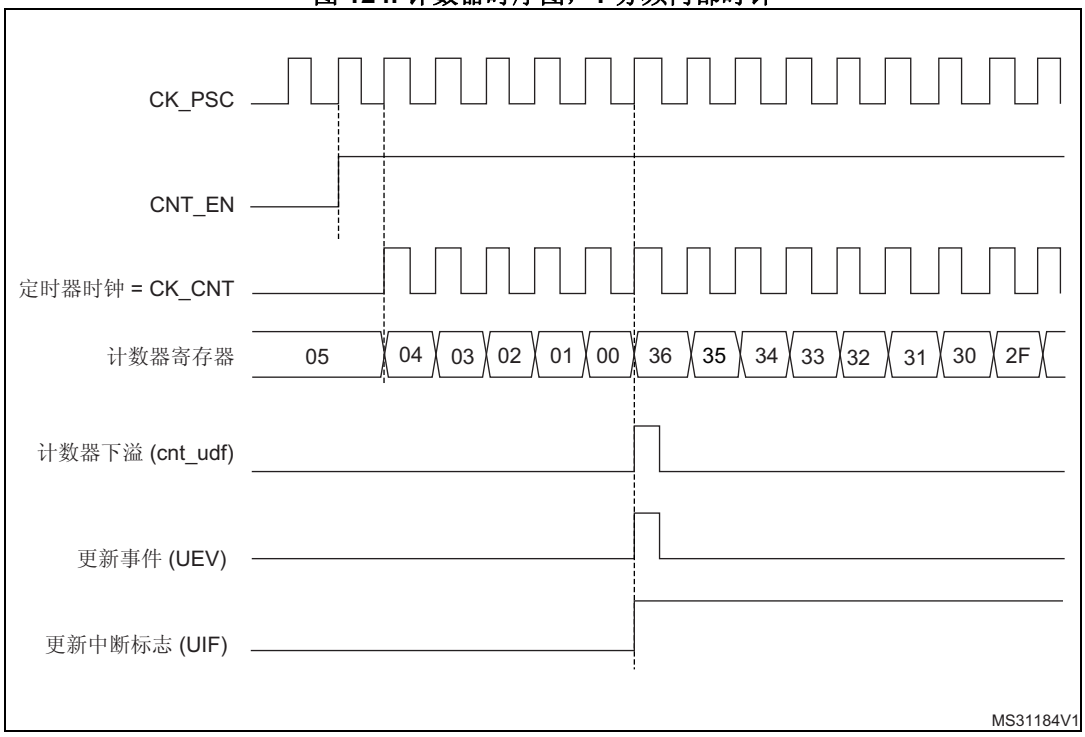


图 125. 计数器时序图，2 分频内部时钟

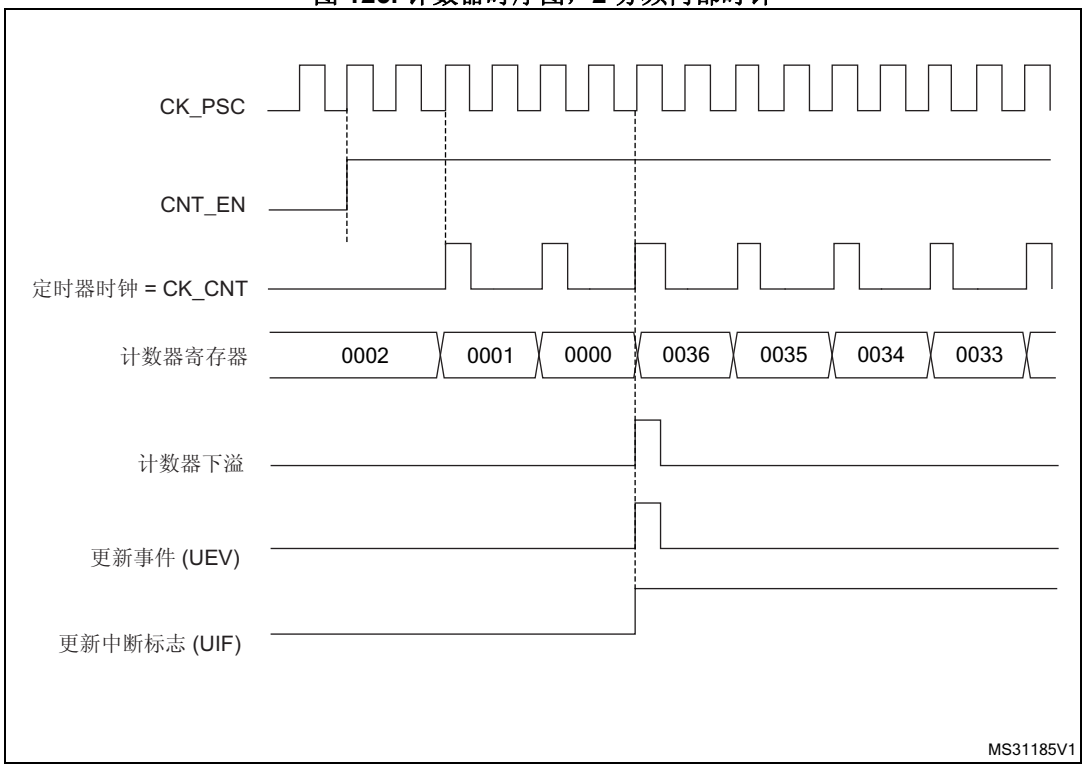


图 126. 计数器时序图，4 分频内部时钟

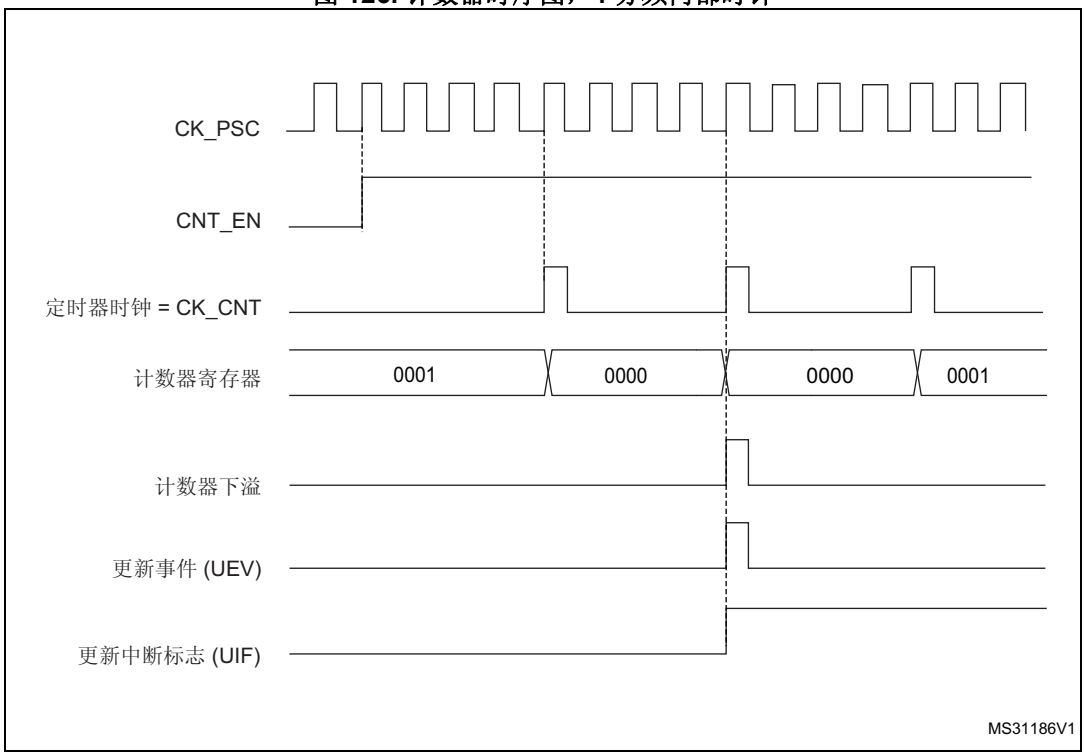


图 127. 计数器时序图，N 分频内部时钟

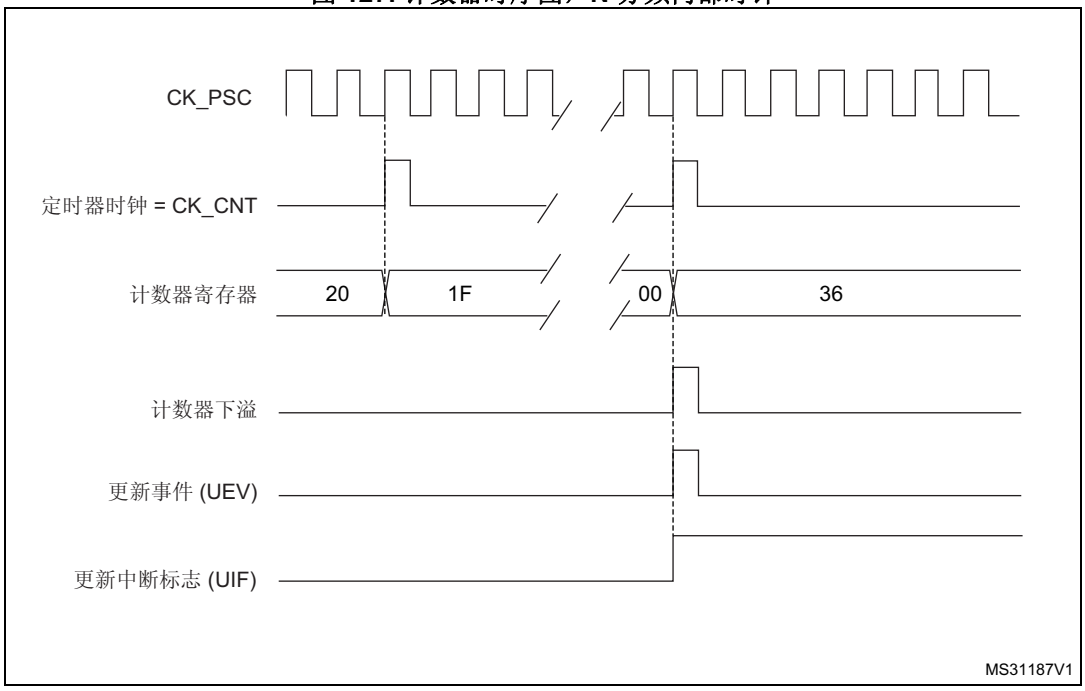
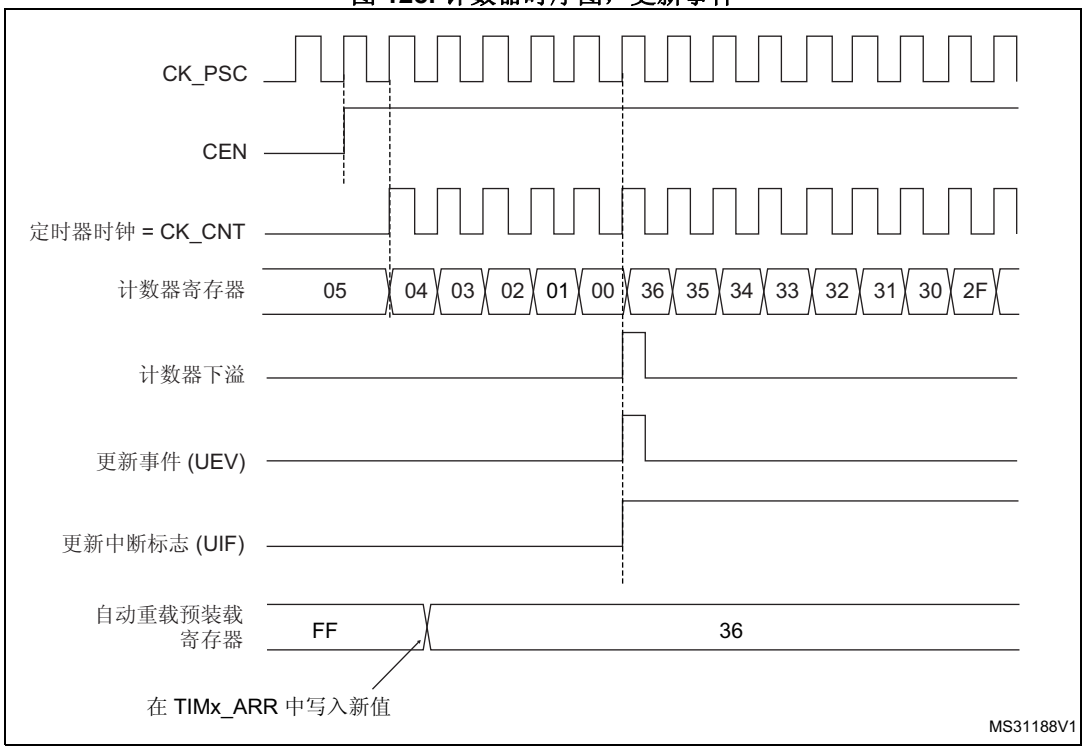


图 128. 计数器时序图，更新事件



中心对齐模式（递增/递减计数）

在中心对齐模式下，计数器从 0 开始计数到自动重载值（TIMx_ARR 寄存器的内容）– 1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中心对齐模式 1，CMS =“01”）、计数器递增计数（中心对齐模式 2，CMS =“10”）以及计数器递增/递减计数（中心对齐模式 3，CMS =“11”）。

此模式下无法写入方向位（TIMx_CR1 寄存器中的 DIR 位）。而是由硬件更新并指示当前计数器方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

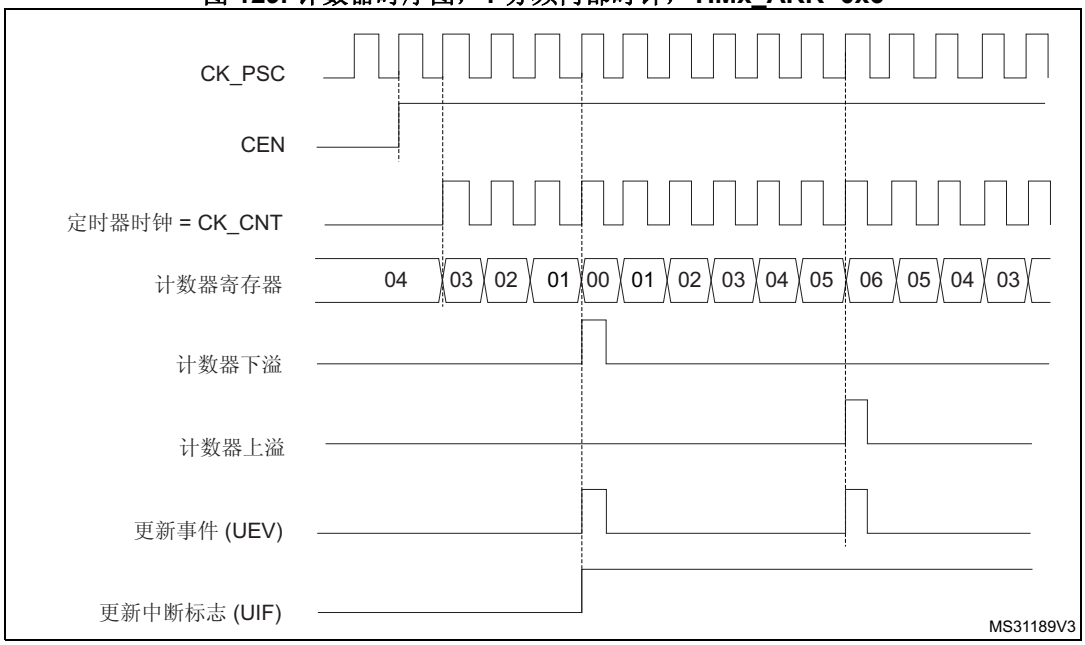
此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载影子寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，如果更新操作是由计数器上溢触发的，则 ARR 寄存器在计数器重载之前更新，因此，下一个计数周期就是我们所希望的新的周期长度（计数器被重载新的值）。

以下各图以一些示例说明不同时钟频率下计数器的行为。

图 129. 计数器时序图，1 分频内部时钟，TIMx_ARR=0x6



1. 此处使用中心对齐模式 1（有关详细信息，请参见*第 405 页的第 16.4.1 节：TIM3 控制寄存器 1 (TIM3_CR1)*）。

图 130. 计数器时序图，2 分频内部时钟

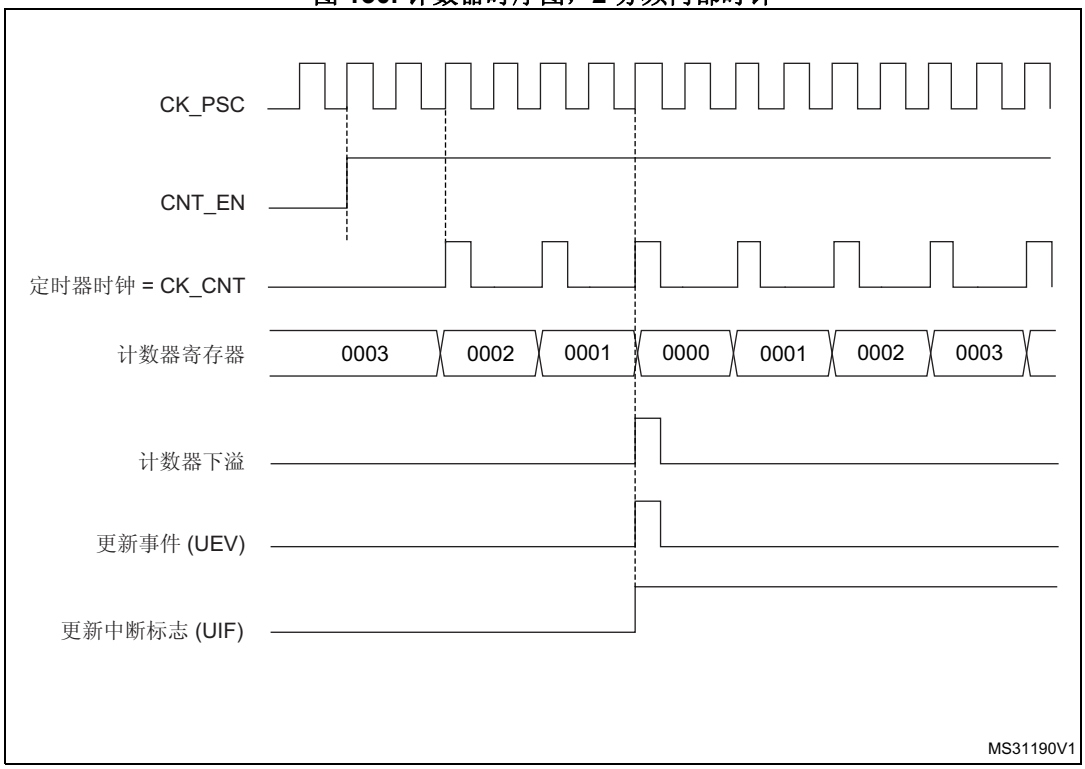
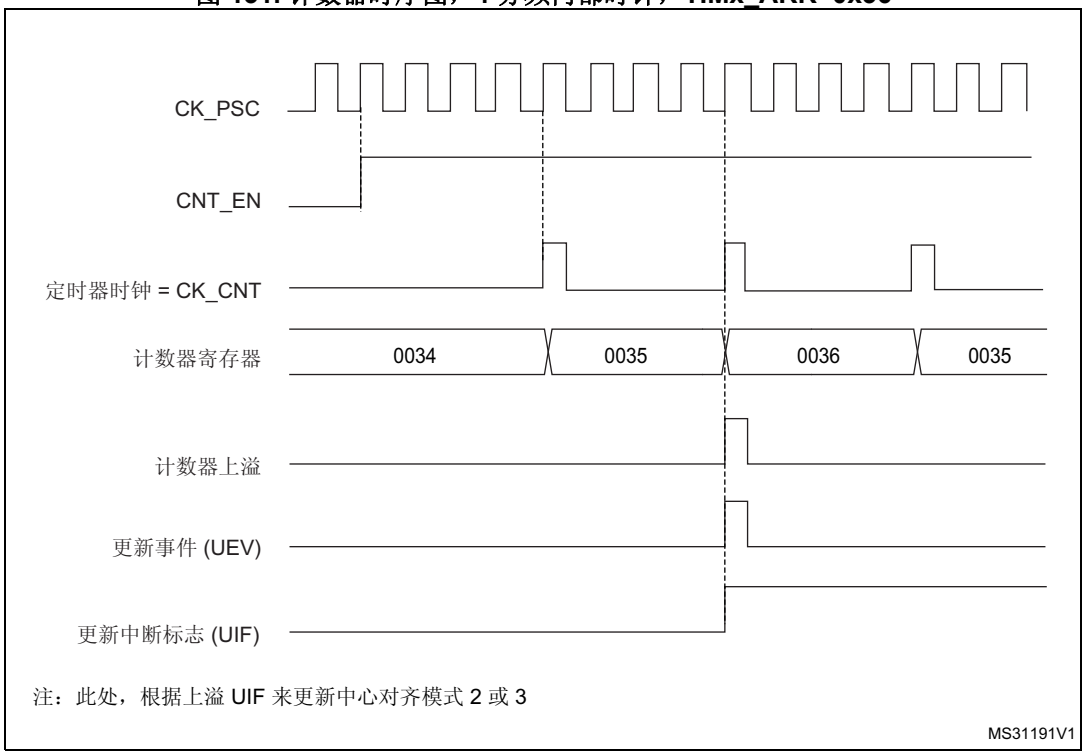


图 131. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36



1. 中心对齐模式 2 或模式 3 与上溢 UIF 结合使用。

图 132. 计数器时序图，N 分频内部时钟

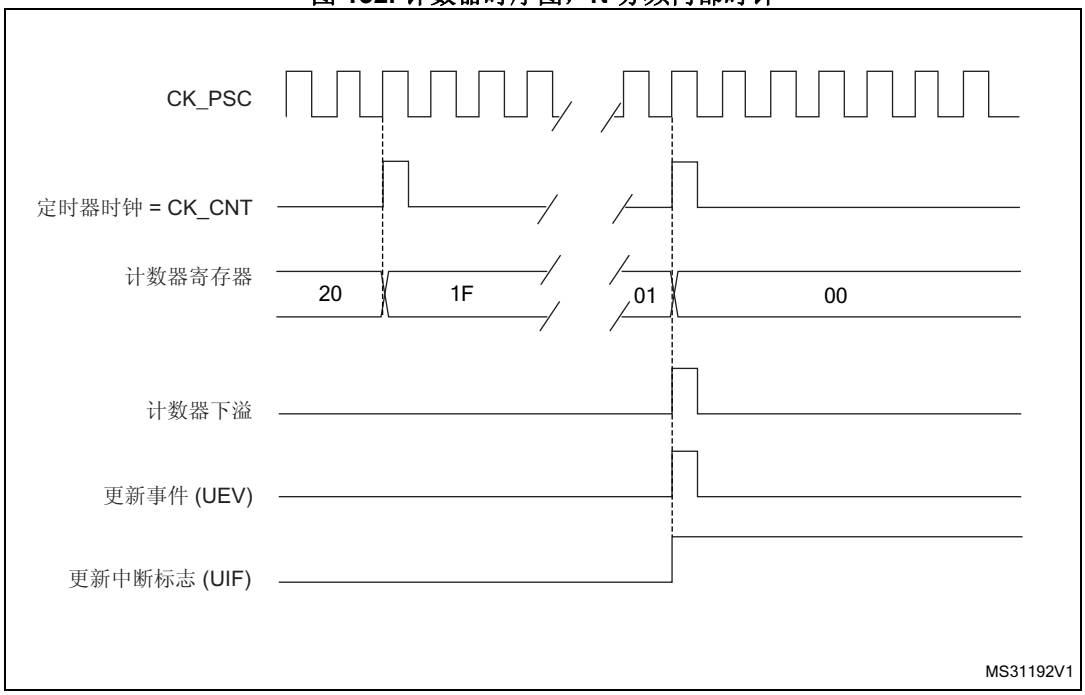


图 133. 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

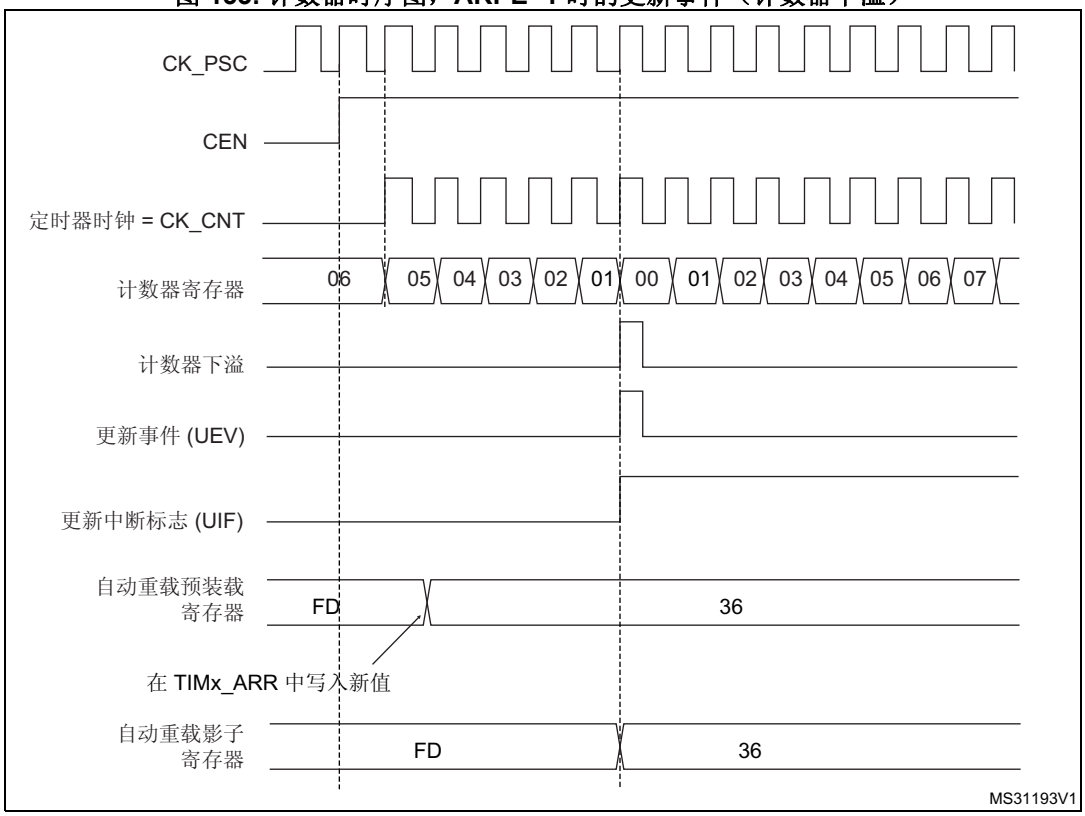
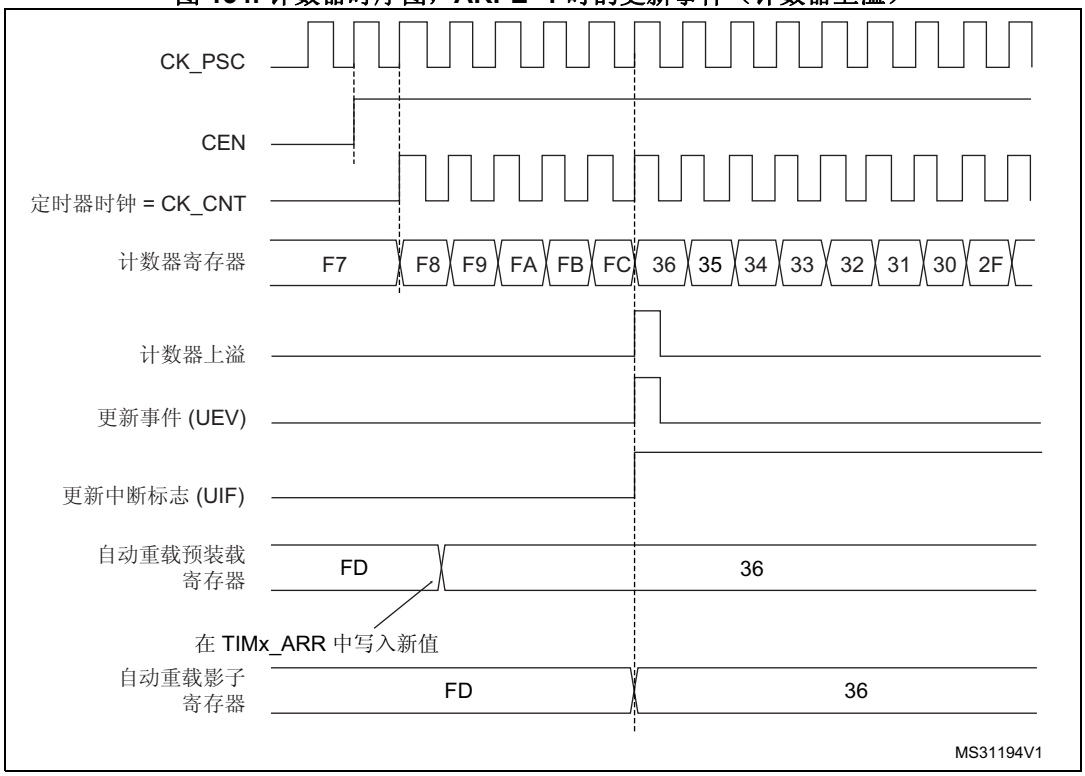


图 134. 计数器时序图，ARPE=1 时的更新事件（计数器上溢）



16.3.3 时钟选择

计数器时钟可由下列时钟源提供：

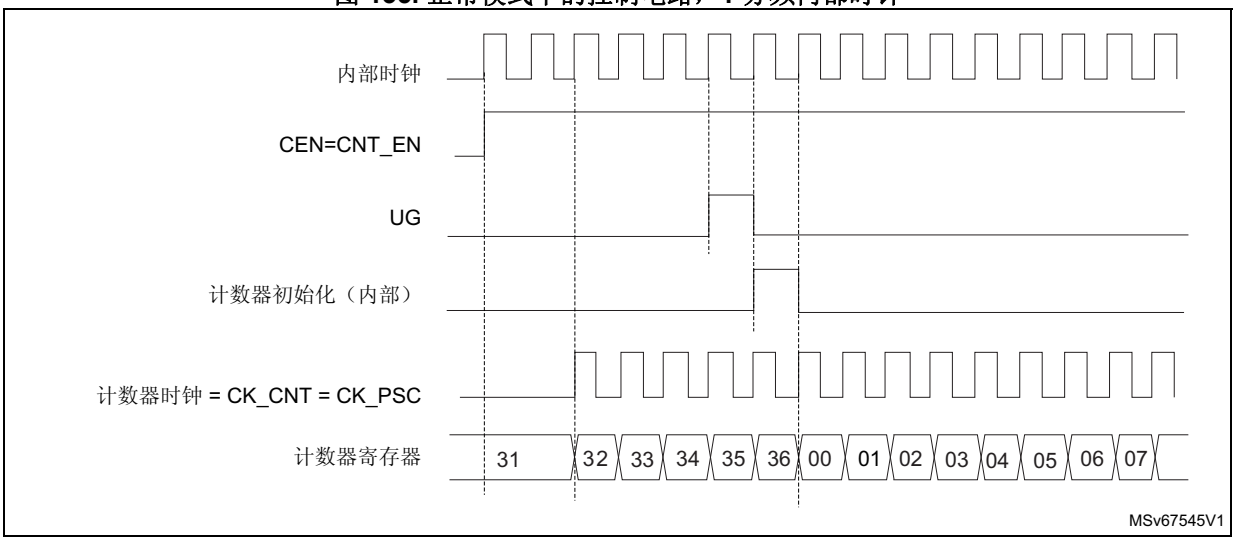
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚 (TIx)
- 外部时钟模式 2：外部触发输入 (ETR)
- 内部触发输入 (ITRx)：使用一个定时器作为另一个定时器的预分频器，例如可以将定时器 X 配置为定时器 Y 的预分频器。更多详细信息，请参见*第 400 页的将一个定时器用作另一个定时器的预分频器*一节。

内部时钟源 (CK_INT)

如果禁止从模式控制器（TIMx_SMCR 寄存器中 SMS=000），则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 135 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

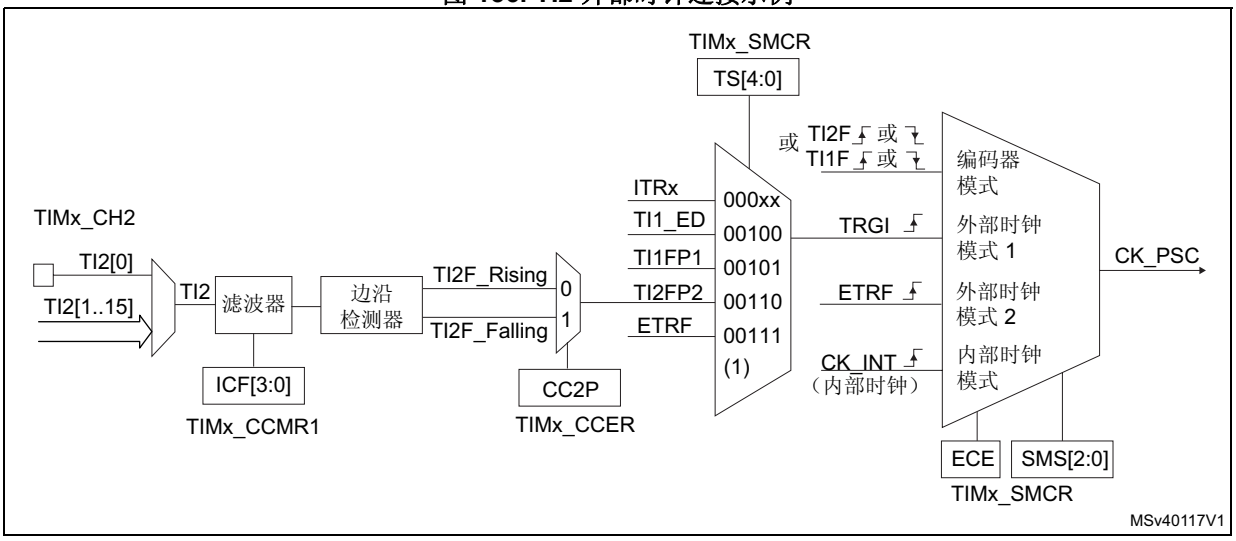
图 135. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 136. TI2 外部时钟连接示例



1. 从 01000 到 11111 的代码：ITRy。

例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2x 源（内部或外部）。
2. 通过在 TIMx_CCMR1 寄存器中写入 CC2S=“01”来配置通道 2，使其能够检测 TI2 输入的上升沿。
3. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。

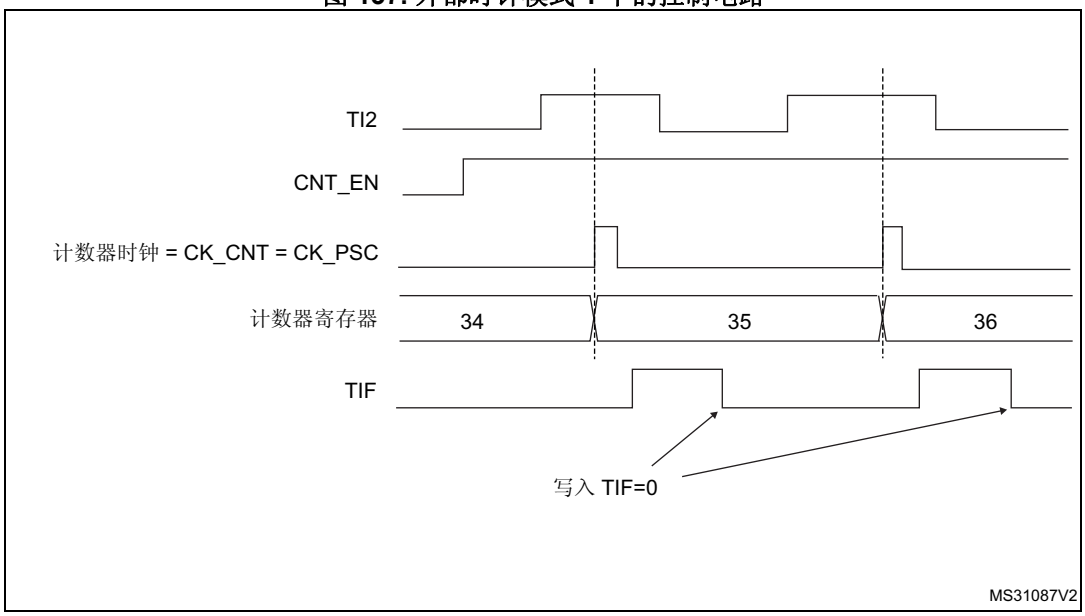
注意： *由于捕获预分频器不用于触发操作，因此无需对其进行配置。*

4. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
5. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
6. 通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为输入源。
7. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 137. 外部时钟模式 1 下的控制电路



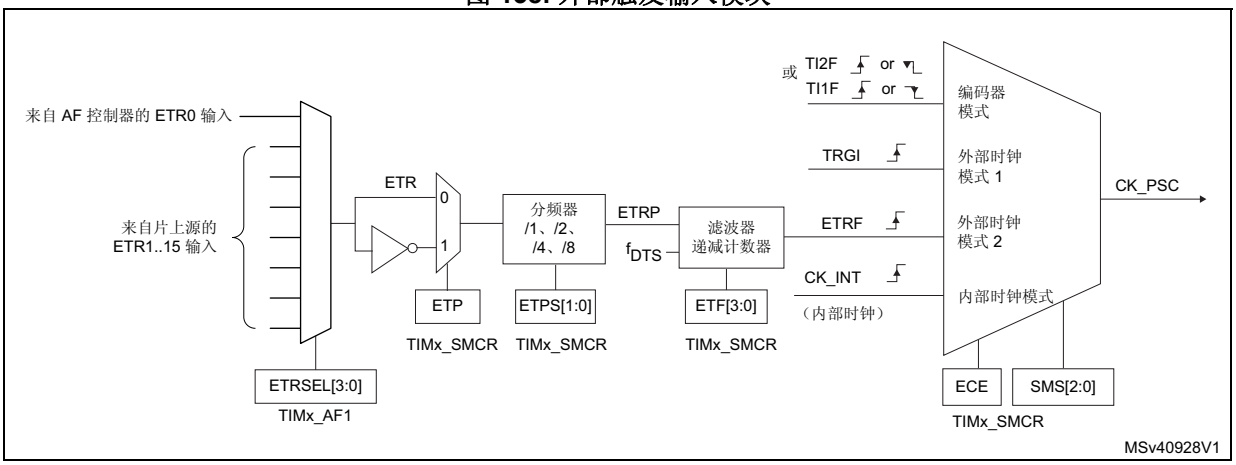
外部时钟源模式 2

通过在 TIMx_SMCR 寄存器中写入 ECE=1 可选择此模式。

计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

图 138 简要介绍了外部触发输入模块。

图 138. 外部触发输入模块



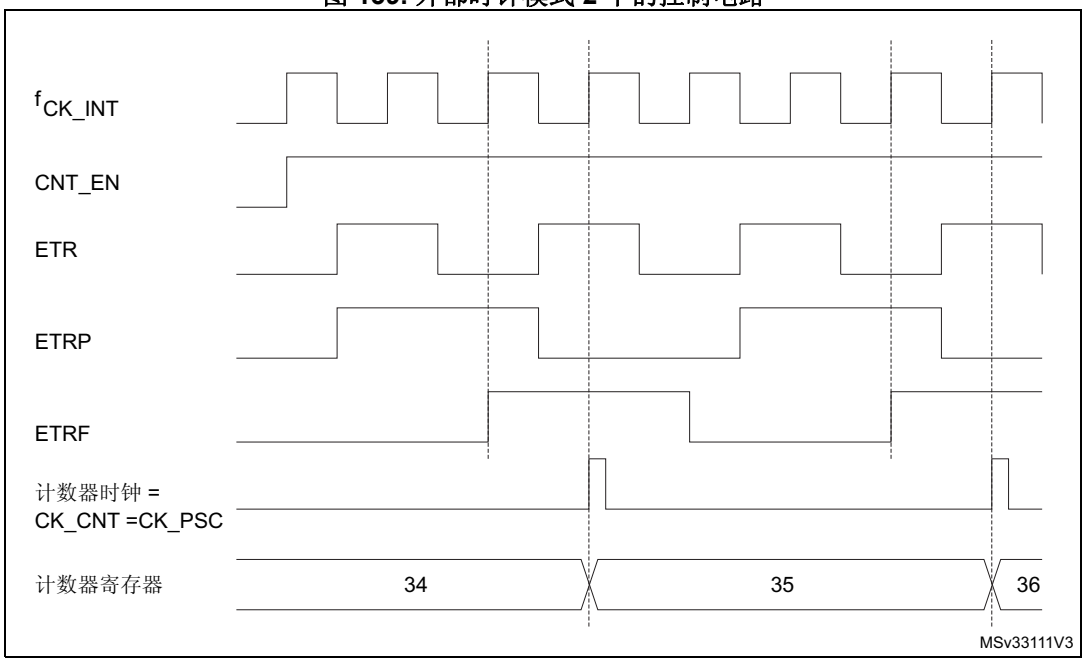
例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 使用 TIMx_AF1 寄存器中的 ETRSEL[3:0] 位选择正确的 ETR 源（内部或外部）。
2. 由于此例中不需滤波器，因此在 TIMx_SMCR 寄存器中写入 ETF[3:0]=0000。
3. 通过在 TIMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
4. 通过在 TIMx_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
5. 通过在 TIMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
6. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

ETR 的上升沿与实际计数器时钟之间的延迟是由于 ETRP 信号的重新同步电路引起的。因此，计数器能够正确捕获的最大频率不超过 TIMxCLK 频率的 ¼。当 ETRP 信号较快时，用户应通过适当的 ETPS 预分频器设置对外部信号进行分频。

图 139. 外部时钟模式 2 下的控制电路



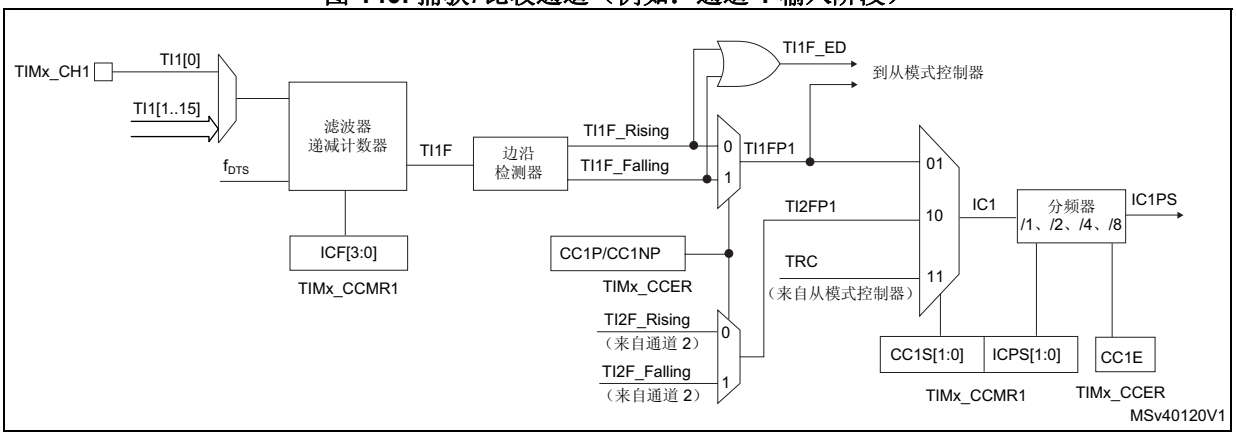
16.3.4 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入单元（数字滤波、多路复用和预分频器）和一个输出单元（比较器和输出控制）构建而成。

下图简要介绍了一路捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 140. 捕获/比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 141. 捕获/比较通道 1 主电路

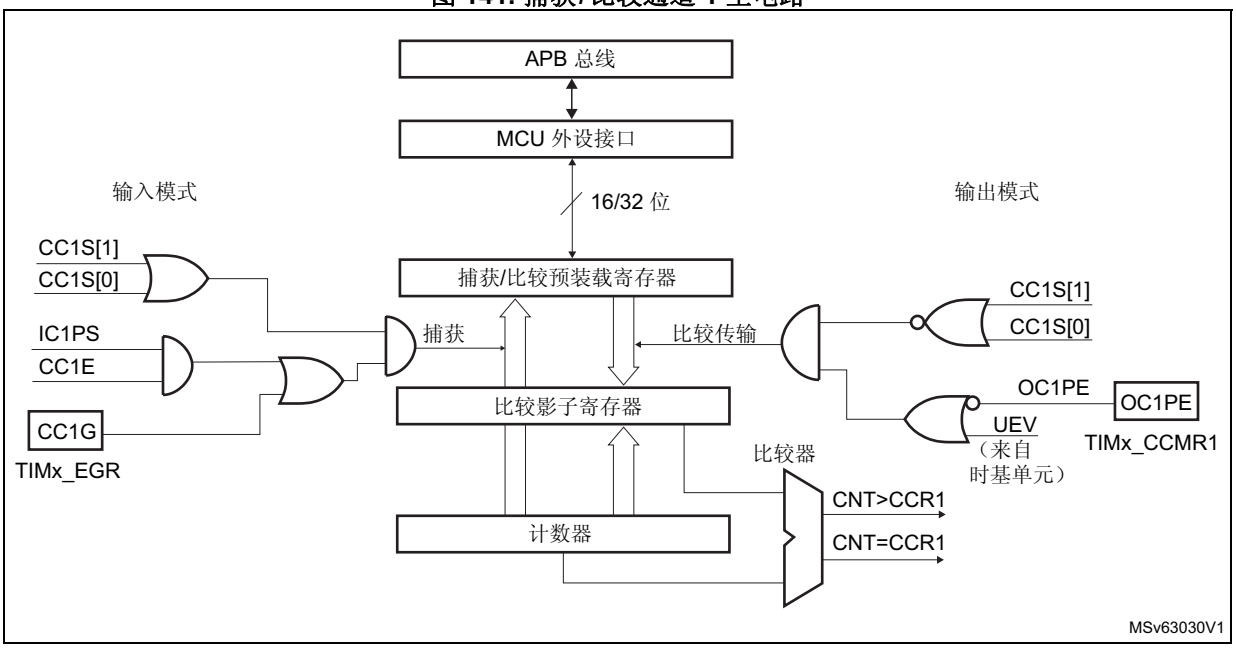
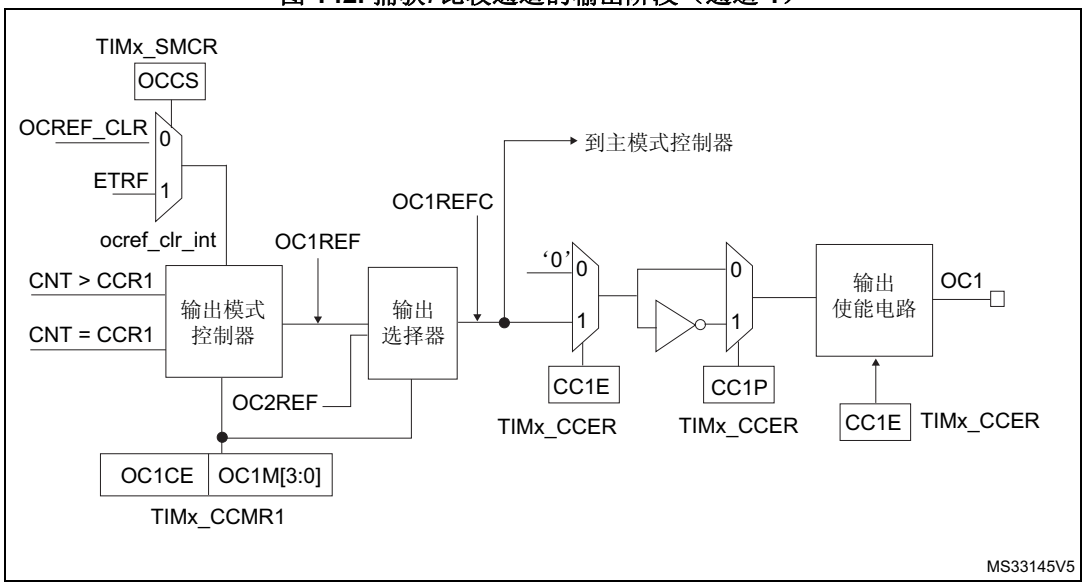


图 142. 捕获/比较通道的输出阶段（通道 1）



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。读写访问的始终是预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容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将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

16.3.5 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将过捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件向 CCxIF 写入 0 使其清零，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择有效输入：TIMx_CCR1 必须关联到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
3. 根据关联到定时器的信号，对相关输入滤波参数进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设输入信号发生翻转时，信号需要 5 个内部时钟周期才能稳定，则必须设置滤波时间大于 5 个内部时钟周期。若在 TI1 上连续 8 次检测到新电平采样值（以 f_{DTS} 频率采样）判定沿跳变合法，则向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
4. 通过向 TIMx_CCER 寄存器中的 CC1P 位和 CC1NP 位写入 000，选择 TI1 通道的有效跳变沿（本例中为上升沿）。
5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入 00）。

6. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
7. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理过捕获，建议在读取捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的过捕获信息。

注意： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

16.3.6 PWM 输入模式

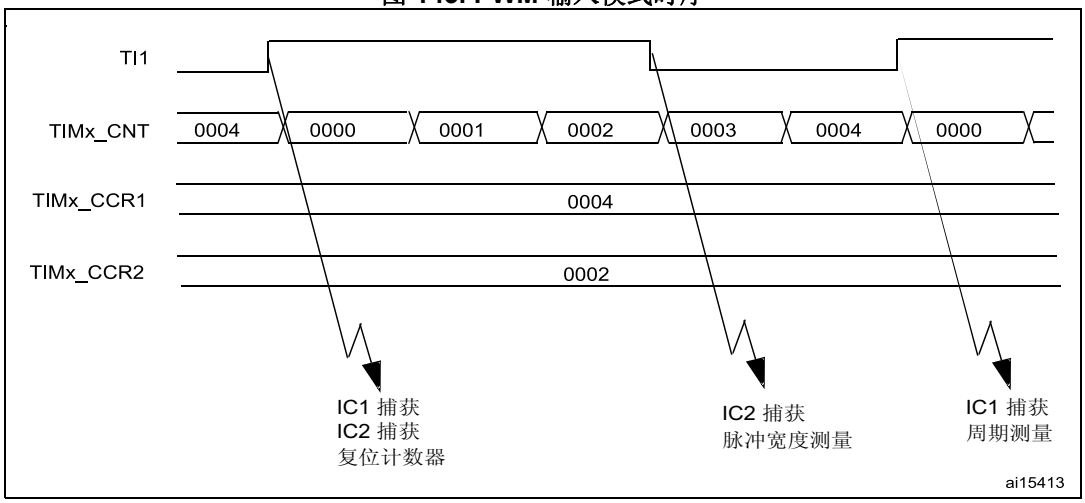
此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
3. 选择 TI1FP1 的有效极性（同时用于 TIMx_CCR1 中的捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
4. 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10（选择 TI1）。
5. 选择 TI1FP2 的有效极性（用于 TIMx_CCR2 中的捕获）：向 CC2P 位写入“1”，向 CC2NP 位写入“0”（下降沿有效）。
6. 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 00101（选择 TI1FP1）。
7. 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 100。
8. 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 143. PWM 输入模式时序



1. PWM 输入模式只能与 TIMx_CH1/TIMx_CH2 信号配合使用，因为只有 TI1FP1 和 TI2FP2 与从模式控制器相连。

16.3.7 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（OCxREF 和 OCx）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号 (ocxref/OCx) 强制设置为有效电平，只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入 101。ocxref 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=0（OCx 高电平有效）=> OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 100，可将 ocxref 信号强制设置为低电平。

当然，即使强制输出模式下，TIMx_CCRx 影子寄存器与计数器之间的比较仍然会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。输出比较模式一节对此进行了介绍。

16.3.8 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（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。匹配时，输出引脚既可保持其电平 (OCXM=000)，也可设置为有效电平 (OCXM=001)、无效电平 (OCXM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

使用 TIMx_CCMRx 寄存器中的 OCxPE 位，可将 TIMx_CCRx 寄存器配置为带或不带预装功能。

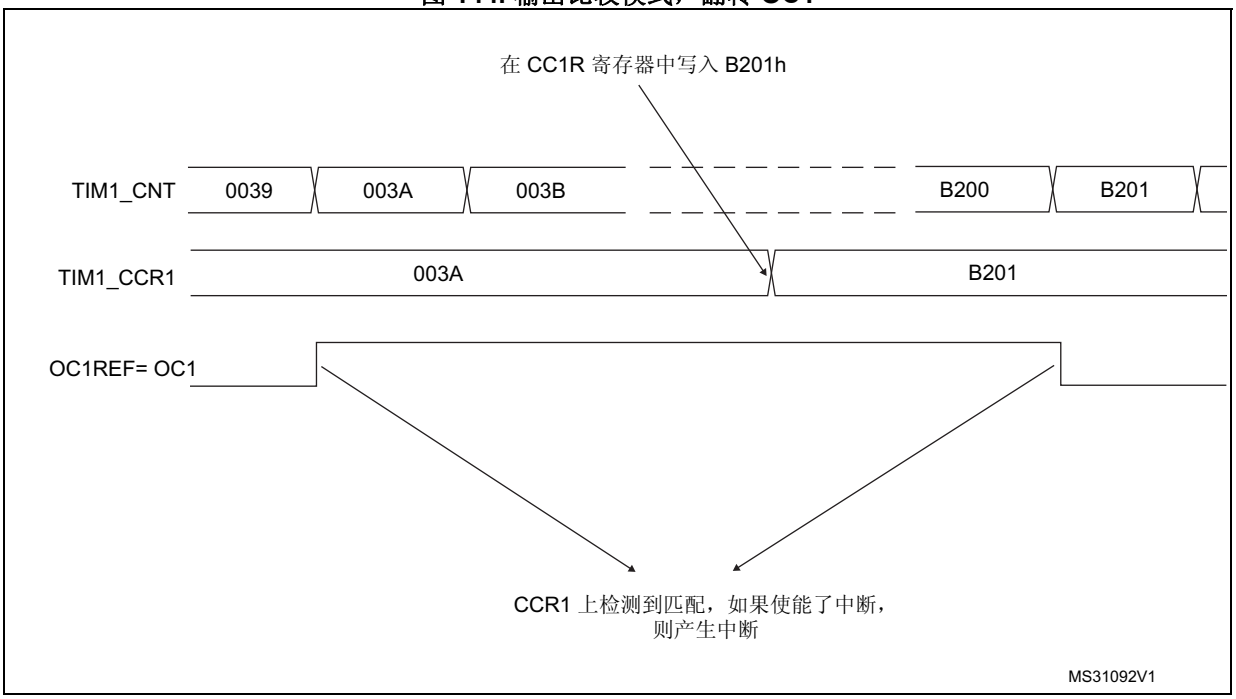
在输出比较模式下，更新事件 UEV 对 ocxref 和 OCx 输出毫无影响。时间分辨率可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断和/或 DMA 请求，将 CCxIE 位和/或 CCxDE 位置 1。
4. 选择输出模式。例如，当 CNT 与 CCRx 匹配、未使用预装载 CCRx 并且 OCx 使能且为高电平有效时，必须写入 OCxM=011、OCxPE=0、CCxP=0 和 CCxE=1 来翻转 OCx 输出引脚。
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装功能（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIMx_CCRx 影子寄存器）。*图 144* 给出了一个示例。

图 144. 输出比较模式，翻转 OC1



16.3.9 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIMx_ARR 寄存器值决定，其占空比则由 TIMx_CCRx 寄存器值决定。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 110（PWM 模式 1）或 111（PWM 模式 2），可以独立选择各通道（每个 OCx 输出对应一个 PWM）的 PWM 模式。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

由于只有在发生更新事件时预装载寄存器内容才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。OCx 输出通过将 TIMx_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息，请参见 TIMx_CCERx 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 与 TIMx_CCRx 进行实时比较，以确定是 TIMx_CCRx ≤ TIMx_CNT 还是 TIMx_CNT ≤ TIMx_CCRx（取决于计数器计数方向）。不过，为符合 OCREF_CLR 功能（在下一个 PWM 周期之前，ETR 信号上的一个外部事件能够清除 OCREF），OCREF 信号仅在以下情况下变为有效状态：

- 比较结果发生改变，或
- 输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）从“冻结”配置（不进行比较，OCxM=“000”）切换为任一 PWM 模式（OCxM=“110”或“111”）。

定时器运行期间，可以通过软件强制 PWM 输出。

根据 TIMx_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

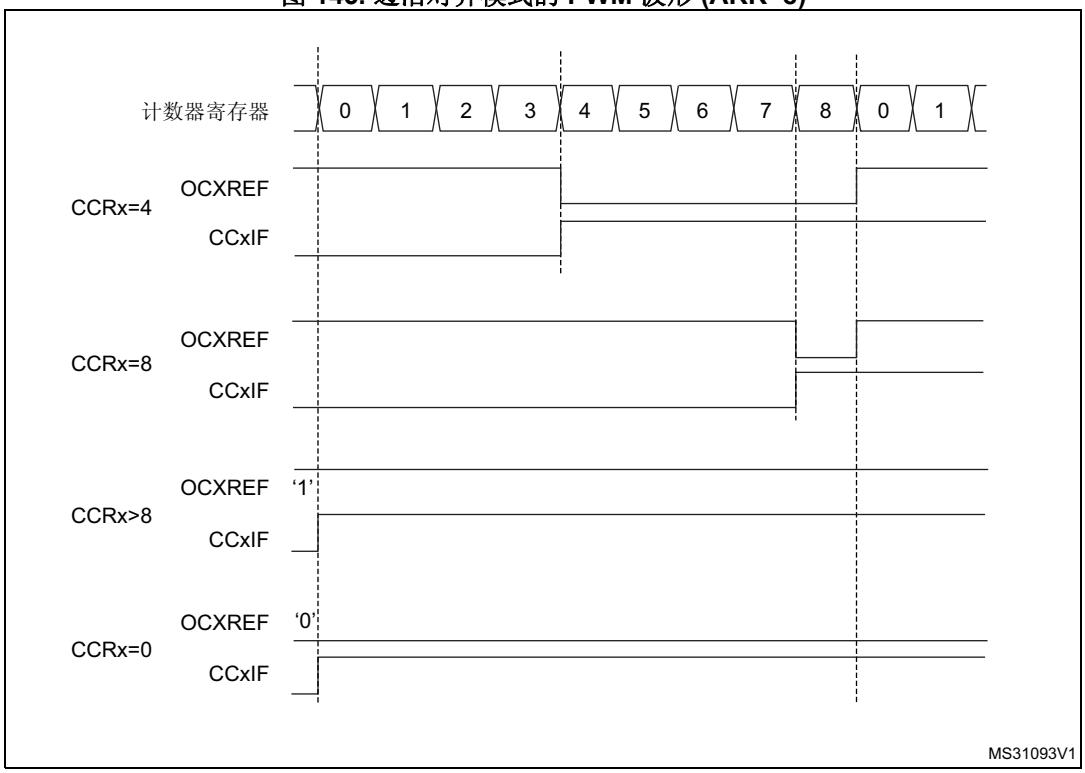
PWM 边沿对齐模式

递增计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低时执行递增计数。请参见*第 367 页的递增计数模式*。

以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。*图 145* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 145. 边沿对齐模式的 PWM 波形 (ARR=8)



递减计数配置

当 TIMx_CR1 寄存器中的 DIR 位为高时执行递减计数。请参见*第 370 页的递减计数模式*。

在 PWM 模式 1 下，只要 TIMx_CNT>TIMx_CCRx，参考信号 ocxref 便为低电平，否则为高电平。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重载值，则 ocxref 保持为 100%。此模式下不可能产生 PWM。

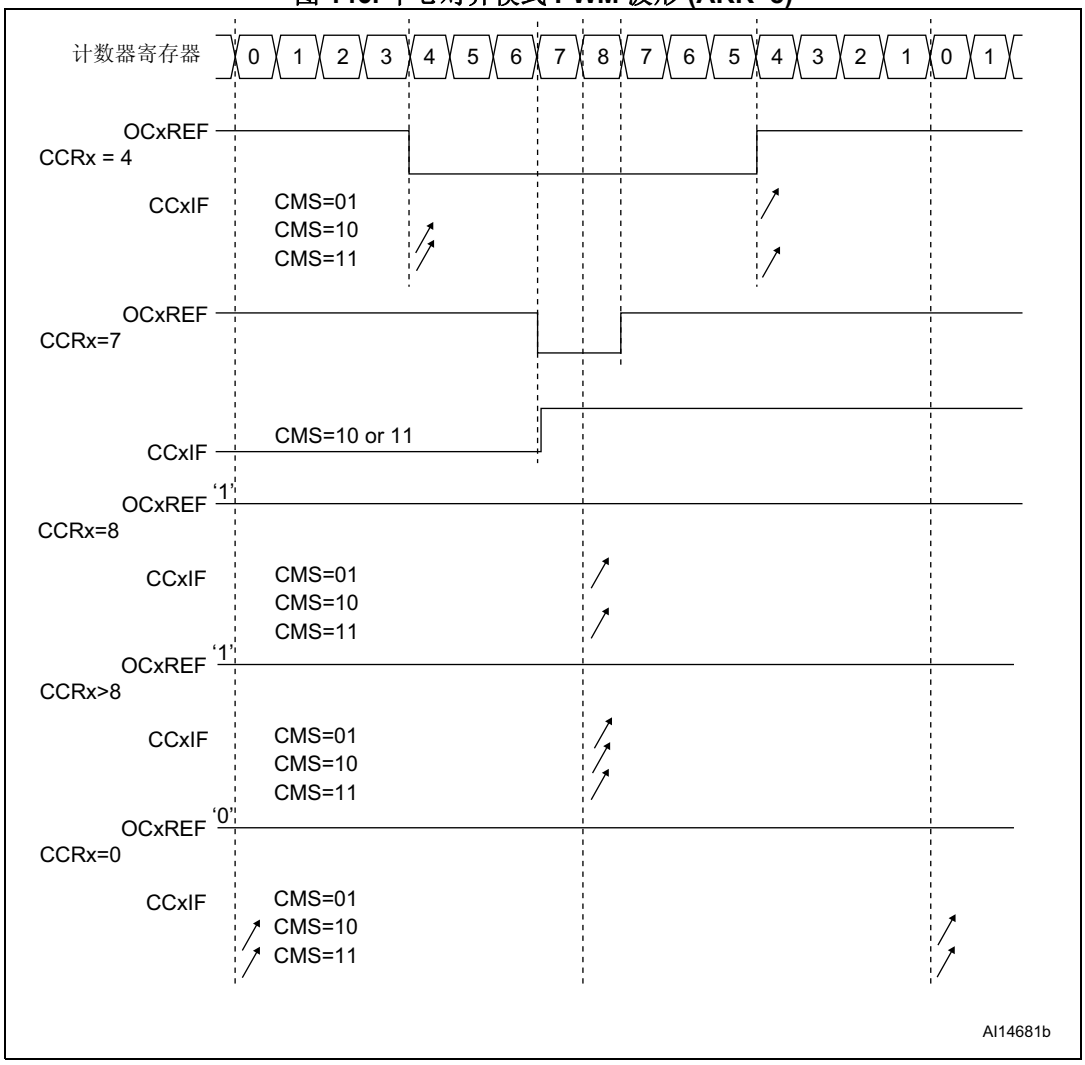
PWM 中心对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时（其余所有配置对 ocxref/OCx 信号具有相同的作用），中心对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIMx_CR1 寄存器中的方向位 (DIR) 由硬件更新，不得通过软件更改。请参见*第 373 页的中心对齐模式（递增/递减计数）*。

图 146 显示了中心对齐模式的 PWM 波形，在此例中：

- TIMx_ARR=8，
- PWM 模式为 PWM 模式 1，
- 在根据 TIMx_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1。

图 146. 中心对齐模式 PWM 波形 (ARR=8)



中心对齐模式使用建议：

- 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得通过软件同时修改 DIR 和 CMS 位。
- 不建议在中心对齐模式下对计数器执行写操作，否则将发生意外错误。尤其是：
  - 如果写入计数器的值大于自动重载值 (TIMx_CNT>TIMx_ARR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
  - 如果向计数器写入 0 或 TIMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

16.3.10 不对称 PWM 模式

在不对称模式下，生成的两个中心对齐 PWM 信号间允许存在可编程相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和相移则由一对 TIMx_CCRx 寄存器确定。两个寄存器分别控制递增计数和递减计数期间的 PWM，这样每半个 PWM 周期便会调节一次 PWM：

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

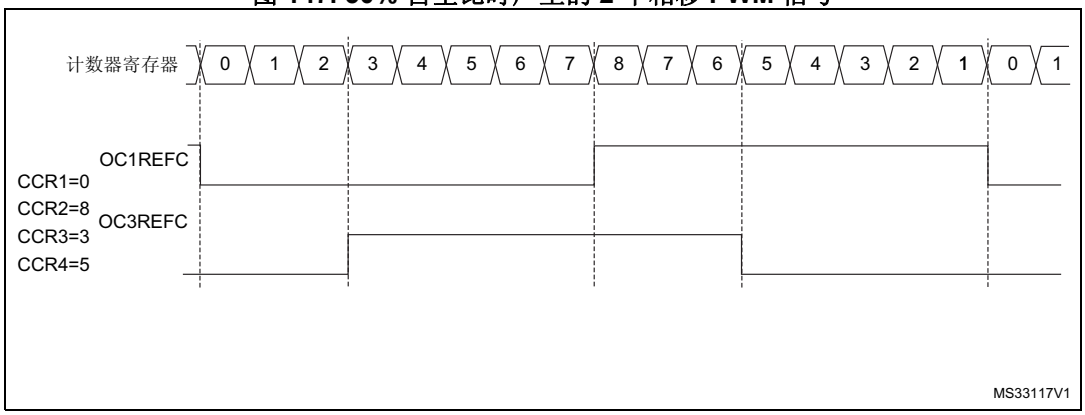
两个通道可以独立选择不对称 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1110”（不对称 PWM 模式 1）或“1111”（不对称 PWM 模式 2）。

注意： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

给定通道用作不对称 PWM 通道时，也可使用其辅助通道。例如，如果通道 1 上产生 OC1REFC 信号（不对称 PWM 模式 1），则由于不对称 PWM 模式 2 的原因，通道 2 上可输出 OC2REF 信号或 OC2REFC 信号。

图 147 显示了不对称 PWM 模式下可以产生的信号示例（通道 1 到通道 4 在不对称 PWM 模式 1 下配置）。

图 147. 50% 占空比时产生的 2 个相移 PWM 信号



16.3.11 组合 PWM 模式

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的“逻辑或”运算或者“逻辑与”运算组合组成。

——OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制

——OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

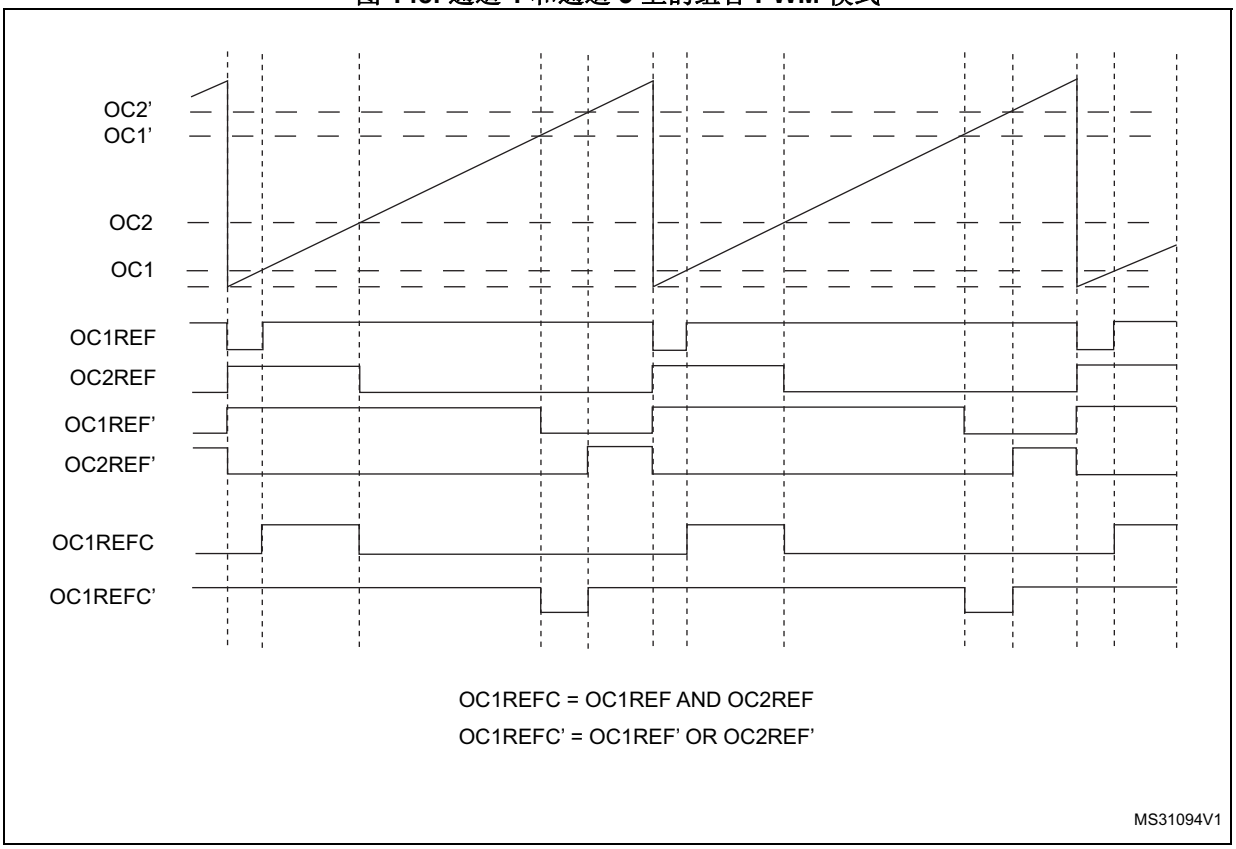
当给定通道用作组合 PWM 通道时，其辅助通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注意： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 148 显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 配置为组合 PWM 模式 2，
- 通道 2 配置为 PWM 模式 1，
- 通道 3 配置为组合 PWM 模式 2，
- 通道 4 配置为 PWM 模式 1

图 148. 通道 1 和通道 3 上的组合 PWM 模式



16.3.12 发生外部事件时清除 OCxREF 信号

对于给定通道，在 ocref_clr_int 输入上施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号清零。OCxREF 信号将保持低电平，直到在下一 PWM 周期再次变为有效状态。该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。

通过配置 TIMx_SMCR 寄存器中的 OCCS 位，可在 OCREF_CLR 输入和 ETRF（滤波器后的 ETR）之间选择 OCREF_CLR_INPUT。

对于给定通道，在 ETRF 输入施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号复位。OCxREF 信号将保持低电平，直到在下一 PWM 周期再次变为有效状态。

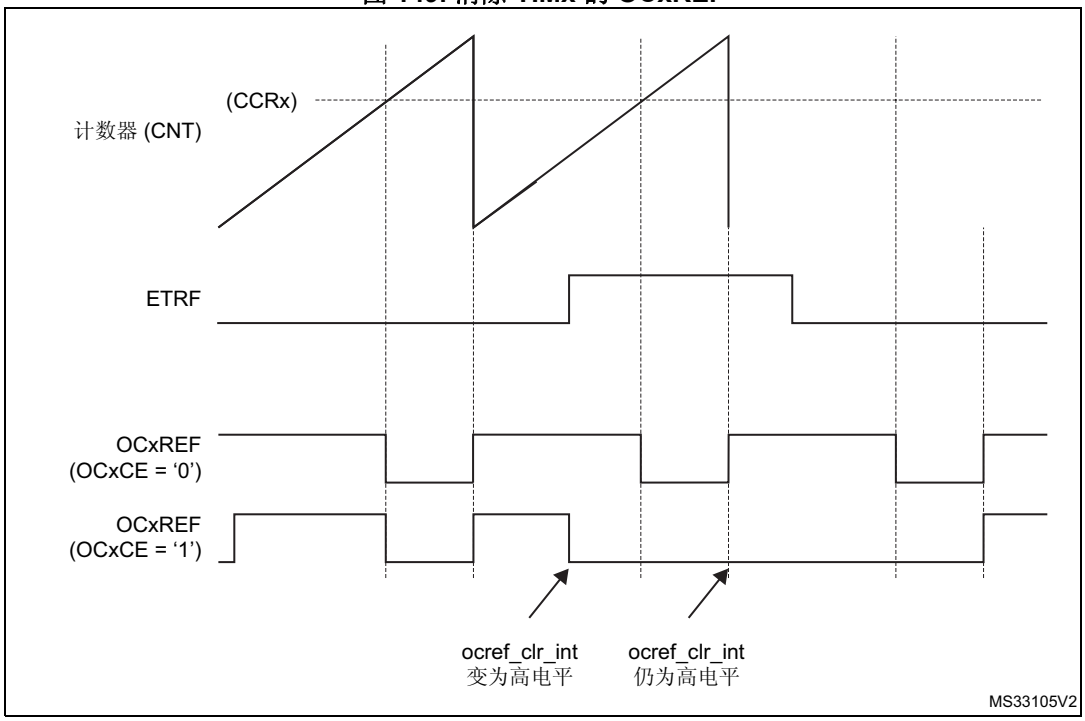
该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。

例如，OCxREF 信号可以连接到比较器的输出，用于控制电流。此时，ETR 必须配置如下：

1. 必须关闭外部触发预分频器：TIMx_SMCR 寄存器中的 ETPS[1:0] 位清为 00。
2. 必须禁止外部时钟模式 2：TIM1_SMCR 寄存器中的 ECE 位清为 0。
3. 外部触发极性 (ETP) 和外部触发滤波器 (ETF) 可根据应用需要进行配置。

图 149 对比了不同使能位 OCxCE 值的情况，显示了当 ETRF 输入变为高电平时 OCxREF 信号的行为。在本例中，定时器 TIMx 编程为 PWM 模式。

图 149. 清除 TIMx 的 OCxREF



注意： *如果 PWM 的占空比为 100% (CCRx>ARR)，则下次计数器上溢时会再次使能 OCxREF。*

16.3.13 单脉冲模式

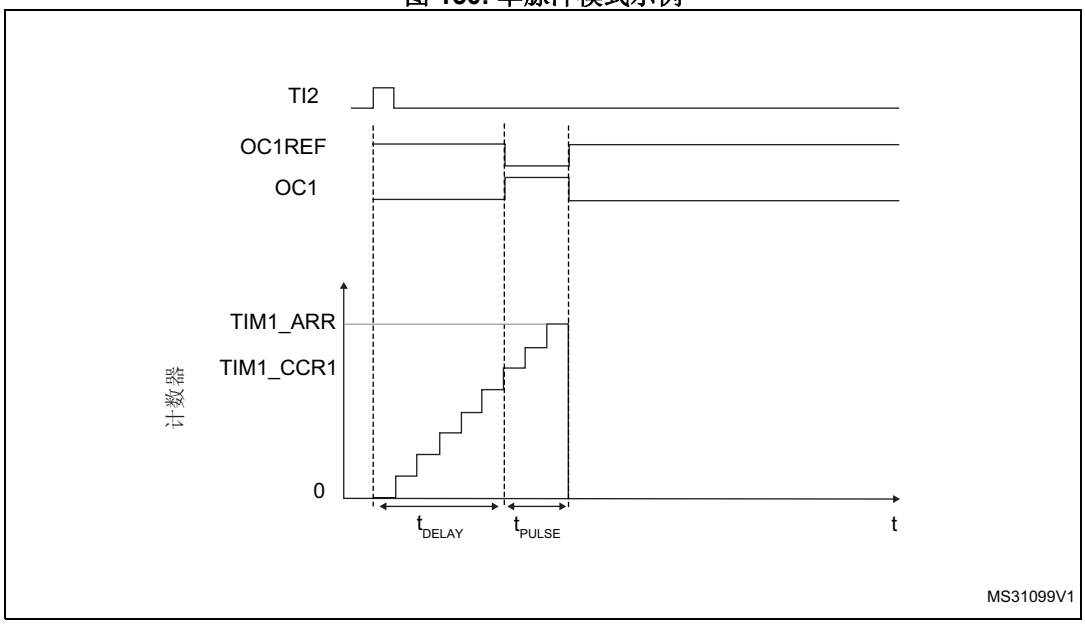
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。通过将 TIMx_CR1 寄存器中的 OPM 位置 1 选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- CNT < CCRx ≤ ARR（特别注意，0 < CCRx），

图 150. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2x 源（内部或外部）。
2. 在 TIMx_CCMR1 寄存器中写入 CC2S=01，将 TI2FP2 映射到 TI2。
3. 在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
4. 在 TIMx_SMCR 寄存器中写入 TS=00110，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
5. 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），使用 TI2FP2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，必须通过在 TIMx_CCMR1 寄存器中写入 OC1M=111，来使能 PWM 模式 2。可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

在本例中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此必须向 TIMx_CR1 寄存器的 OPM 位写入 1，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特例：OCx 快速使能：

在单脉冲模式下，TIx 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟（t_{DELAY} 最小值）。

如果要输出延迟时间最短的波形，可以将 TIMx_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 OCxRef（和 OCx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

16.3.14 可再触发单脉冲模式

该模式可以启动计数器来响应激励信号，并且能产生长度可编程的脉冲，但与不可再触发单脉冲模式间存在以下差别，如*第 16.3.13 节*所述：

- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] = “1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为“1000”或“1001”。

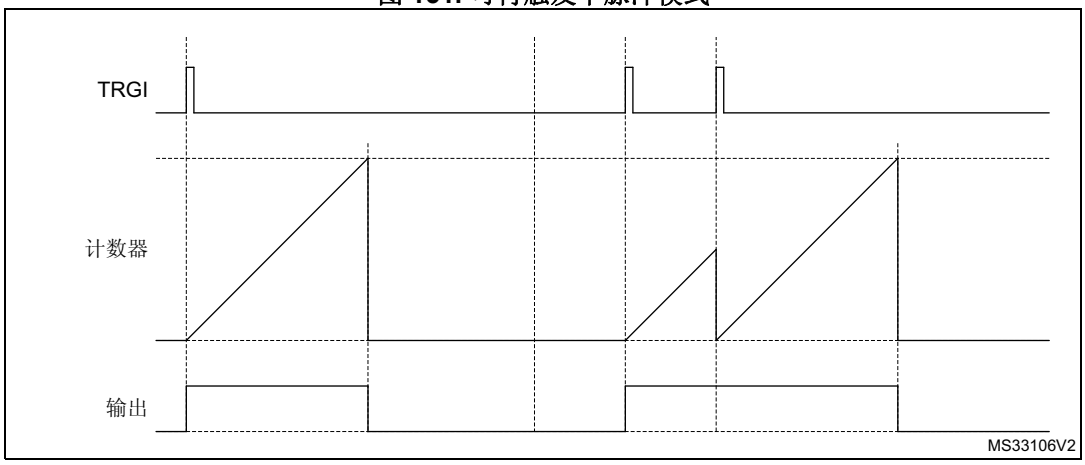
定时器配置为递增计数模式时，相应的 CCRx 必须置 0（ARR 寄存器设置脉冲长度）。如果定时器配置为递减计数模式，CCRx 必须高于或等于 ARR。

注意： *在可再触发单脉冲模式下，CCxIF 标志没有意义。*

出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。

此模式不能与中心对齐 PWM 模式一起使用。在 TIMx_CR1 中必须设置 CMS[1:0] = 00。

图 151. 可再触发单脉冲模式



16.3.15 编码器接口模式

选择编码器接口模式时，如果计数器仅在 TI2 边沿处计数，在 TIMx_SMCR 寄存器中写入 SMS=001；如果计数器仅在 TI1 边沿处计数，写入 SMS=010；如果计数器在 TI1 和 TI2 边沿处均计数，则写入 SMS=011。

通过编程 TIMx_CCER 寄存器的 CC1P 和 CC2P 位，选择 TI1 和 TI2 极性。CC1NP 和 CC2NP 必须保持清零。如果需要，还可对输入滤波器进行编程。CC1NP 和 CC2NP 必须保持低电平。

TI1 和 TI2 两个输入用于连接增量编码器。请参见*表 68*。如果使能计数器（在 TIMx_CR1 寄存器的 CEN 位中写入“1”），则计数器的时钟由 TI1FP1 或 TI2FP2 上的每次有效信号转换提供。TI1FP1 和 TI2FP2 是 TI1 和 TI2 进行输入滤波和极性选择后的信号，如果不进行滤波和反相，则 TI1FP1=TI1，TI2FP2=TI2。将根据两个输入的信号转换序列，产生计数脉冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应修改。任何输入（TI1 或 TI2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI1 或 TI2 边沿处计数，还是同时在 TI1 和 TI2 处计数。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 TIMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIMx_ARR。同样，捕获、比较、预分频器、触发输出功能继续正常工作。

在此模式下，计数器会根据正交编码器的速度和方向自动进行修改，因此，其内容始终表示编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 TI1 和 TI2 不同时切换）。

表 68. 计数方向与编码器信号的关系

有效边沿	相反信号的电平（TI1FP1 对应 TI2，TI2FP2 对应 TI1）	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
仅在 TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TI2 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 TI1 和 TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

外部增量编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码器输出可与外部中断输入相连，用以触发计数器复位。

图 152 以计数器工作为例，说明了计数信号的生成和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S=01（TIMx_CCMR1 寄存器，TI1FP1 映射到 TI1 上）
- CC2S=01（TIMx_CCMR2 寄存器，TI2FP2 映射到 TI2 上）
- CC1P 和 CC1NP = “0”（TIMx_CCER 寄存器，TI1FP1 未反相，TI1FP1=TI1）
- CC2P 和 CC2NP = “0”（TIMx_CCER 寄存器，TI2FP2 未反相，TI2FP2= TI2）
- SMS=011（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）
- CEN = 1（TIMx_CR1 寄存器，使能计数器）

图 152. 编码器接口模式下的计数器工作示例

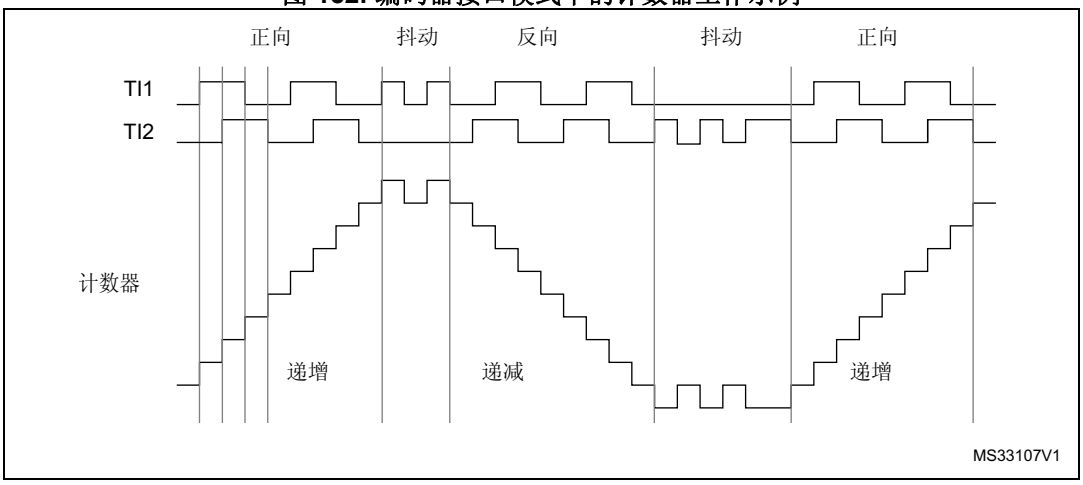
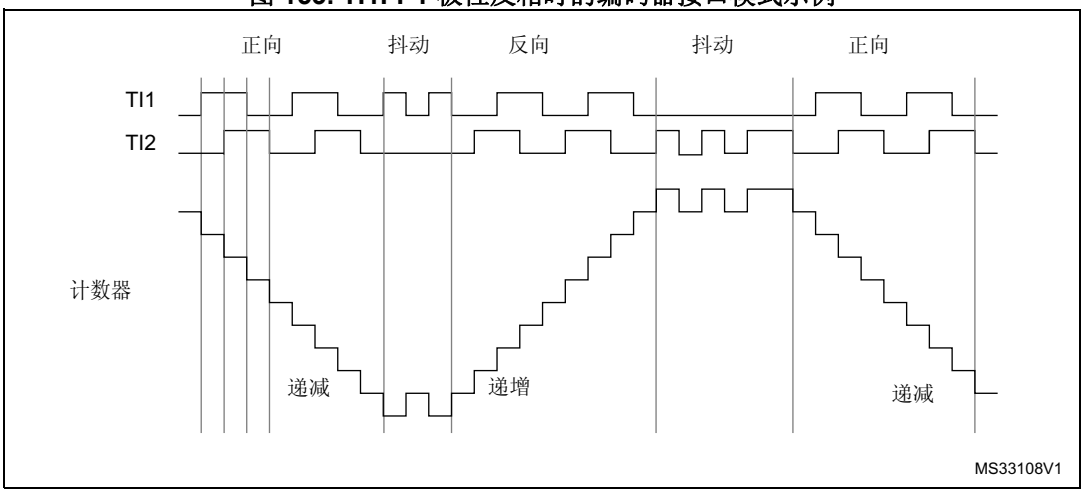


图 153 举例说明了 TI1FP1 极性反相时的计数器行为（除 CC1P=1 外，其他配置与上例相同）。

图 153. TI1FP1 极性反相时的编码器接口模式示例



定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕获信号必须为周期性信号，可以由另一个定时器产生）；还可以通过由实时时钟生成的 DMA 请求读取计数器值。

16.3.16 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 (UIF) 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可按原子方式读取计数器值以及由 UIFCPY 标志发出的更新事件情况。这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化角速度的计算。

UIF 和 UIFCPY 标志之间的置位没有延迟。

16.3.17 定时器输入异或功能

借助 TIM1xx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 到 TIMx_CH3 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。

*第 321 页的第 15.3.25 节：连接霍尔传感器*以连接霍尔传感器为例介绍了此功能。

16.3.18 定时器与外部触发同步

TIMx 定时器可在多种模式下与外部触发实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

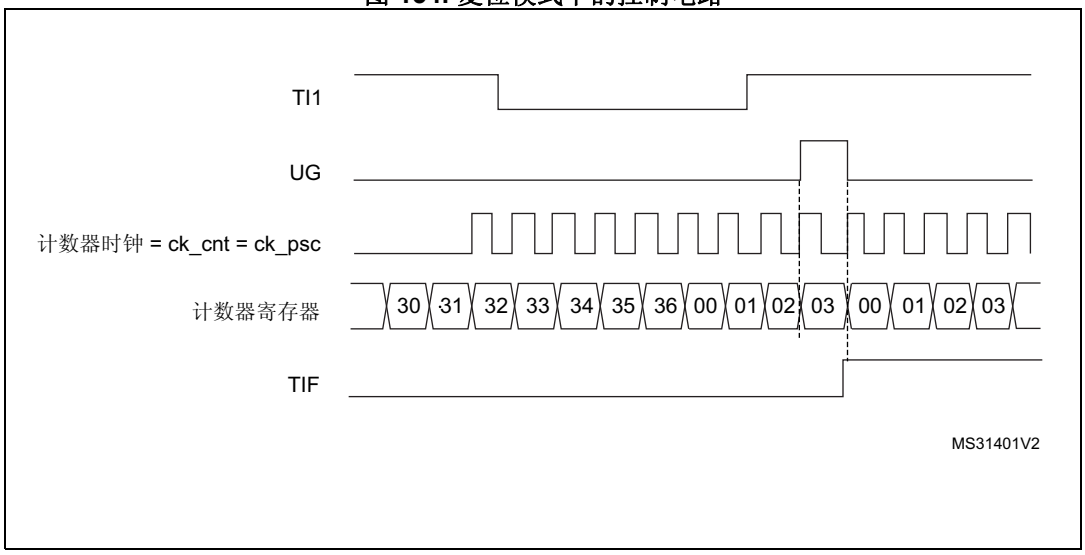
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

1. 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S = 01。
   在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，验证极性（仅检测上升沿）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS= 00101，选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 154. 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

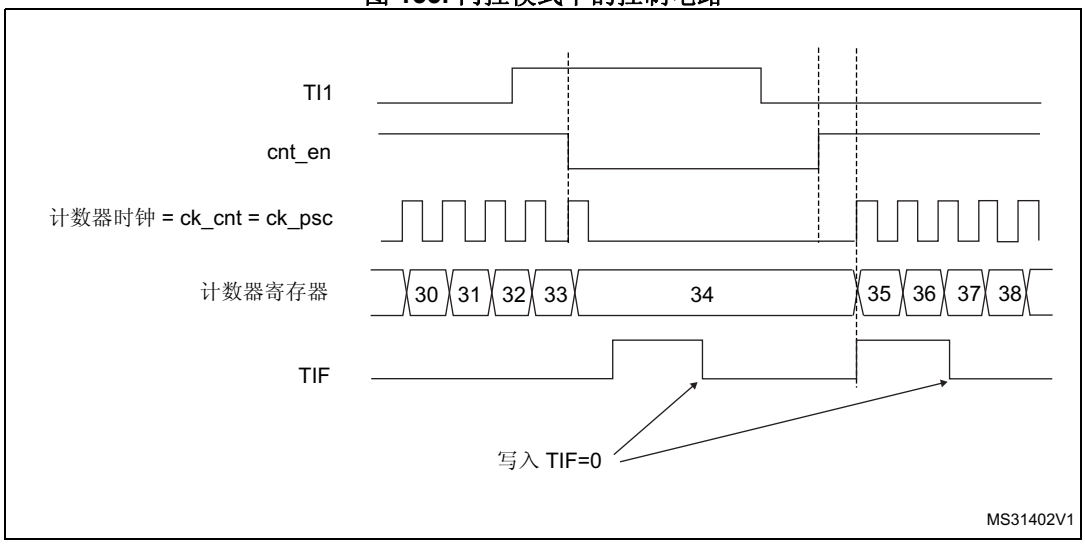
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

1. 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=1 和 CC1NP=0，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS= 00101，选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 155. 门控模式下的控制电路



1. 由于门控模式作用于电平而非边沿，因此在门控模式下，“CCxP=CCxNP=1”（同时检测上升沿和下降沿）不发挥任何作用。

注意： *由于门控模式作用于电平而非边沿，因此在门控模式下，“CCxP=CCxNP=1”（同时检测上升沿和下降沿）不发挥任何作用。*

从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

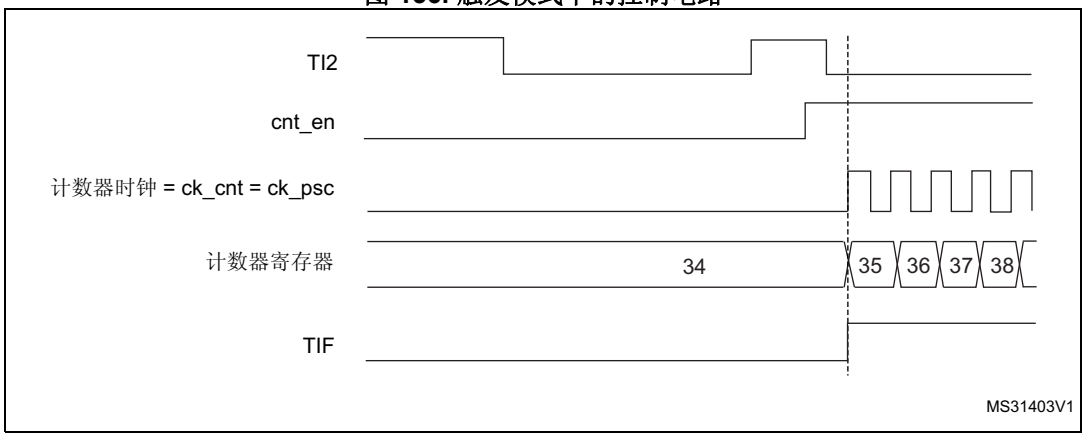
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

1. 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=00110，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 156. 触发模式下的控制电路



从模式：外部时钟模式 2 + 触发模式

外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下，ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个输入作为触发输入。不建议通过 TIMx_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

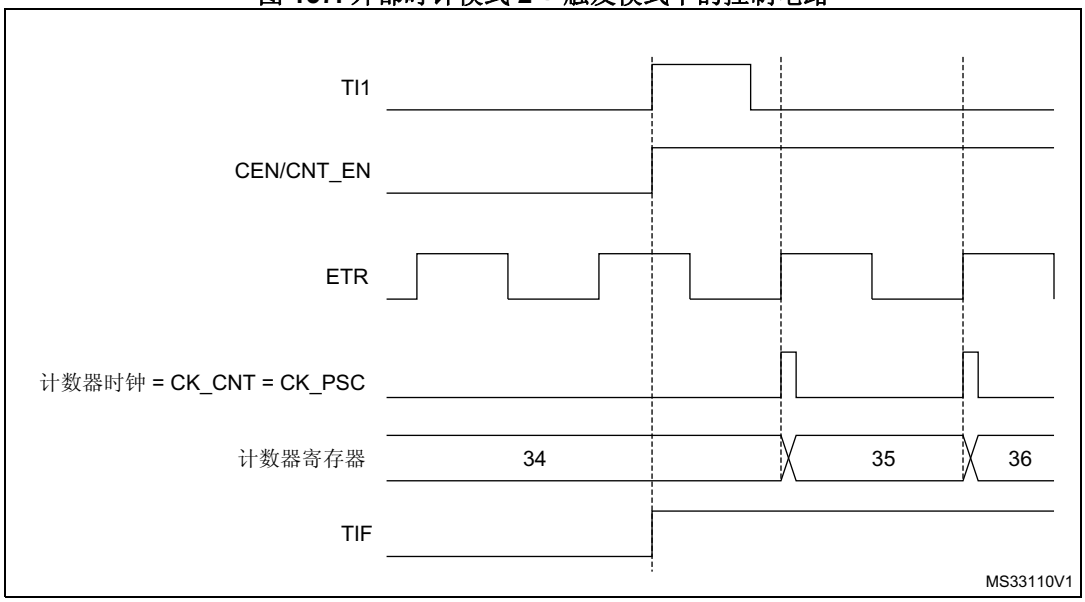
在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

1. 通过对 TIMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：
   - ETF = 0000：无滤波器
   - ETPS = 00：禁止预分频器
   - ETP = 0：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。
2. 如下配置通道 1，以检测 TI 的上升沿：
   - IC1F=0000：无滤波器。
   - 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
   - TIMx_CCMR1 寄存器中 CC1S=01，只选择输入捕获源。
   - TIMx_CCER 寄存器中 CC1P=0 且 CC1NP= 0，以确定极性（仅检测上升沿）。
3. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS= 00101，选择 TI1 作为输入源。

TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。

ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。

图 157. 外部时钟模式 2 + 触发模式下的控制电路



16.3.19 定时器同步

TIMx 定时器从内部连接在一起，以实现定时器同步或链接。当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。

*图 158：主/从定时器示例*和*图 159：单通道定时器的主/从连接示例*简要介绍了触发选择和主模式选择框图。

图 158. 主/从定时器示例

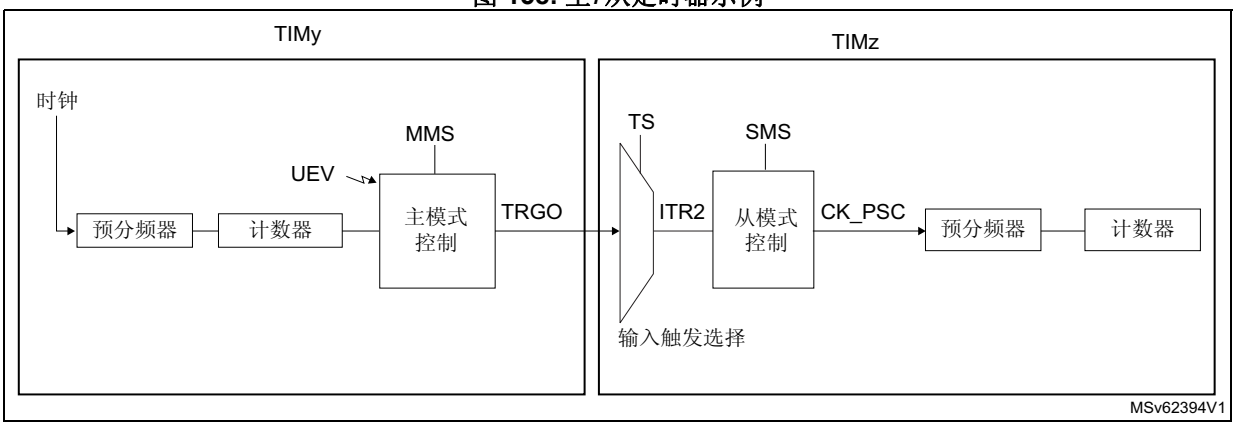
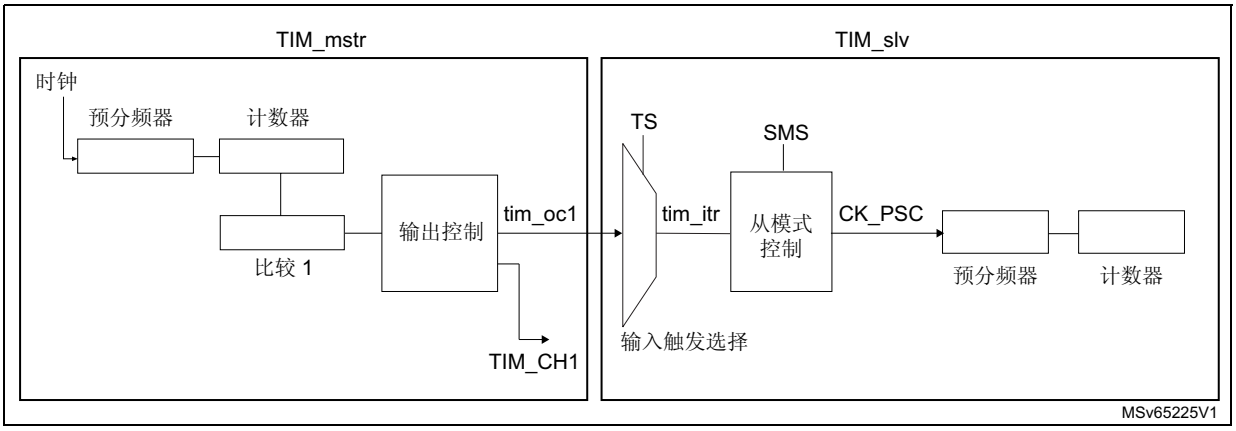


图 159. 单通道定时器的主/从连接示例



注意： *单通道定时器（请参见图 159）没有主模式。但是，可以使用 OC1 输出信号来触发其他定时器（包括本文档其他部分所述的定时器）。通过查看关于器件上任何 TIMx_SMCR 寄存器的“TIMx 内部触发连接”表，可以确定哪些定时器可以作为目标从定时器。*
OC1 信号的脉冲宽度必须至少设为目标定时器时钟周期的 2 倍，以便确保从定时器能够检测到触发。
例如，如果目标定时器 CK_INT 时钟频率是源定时器的 1/4，则 OC1 的脉冲宽度必须至少是 8 个源定时器时钟周期。

将一个定时器用作另一个定时器的预分频器

例如，可以将 TIMy 配置为 TIMz 的预分频器。请参见图 158。为此：

1. 将定时器 TIMy 配置为主模式，以便每次发生更新事件 UEV 时都输出一个周期性触发信号。如果在 TIMy_CR2 寄存器中写入 MMS=010，则每次生成更新事件时，TRGO 都会输出一个上升沿。
2. 要将 TIMy 的 TRGO 输出连接到 TIMz，必须将 TIMz 配置为从模式，使用 ITR 作为内部触发。通过 TIMz_SMCR 寄存器中的 TS 位（写入 TS=00）可对此进行选择。
3. 然后必须将从模式控制器设为外部时钟模式 1（在 TIMz_SMCR 寄存器中写入 SMS=111）。这样一来，TIMz 的时钟将由 TIMy 周期性触发信号的上升沿（与 TIMy 的计数器上溢对应）提供。
4. 最后必须通过将这两个定时器的相应 CEN 位（TIMx_CR1 寄存器）置 1 同时使能二者。

注意： *如果选择 TIMy 的 OCx 信号作为触发输出 (MMS=1xx)，该信号的上升沿将用于驱动 TIMz 的计数器。*

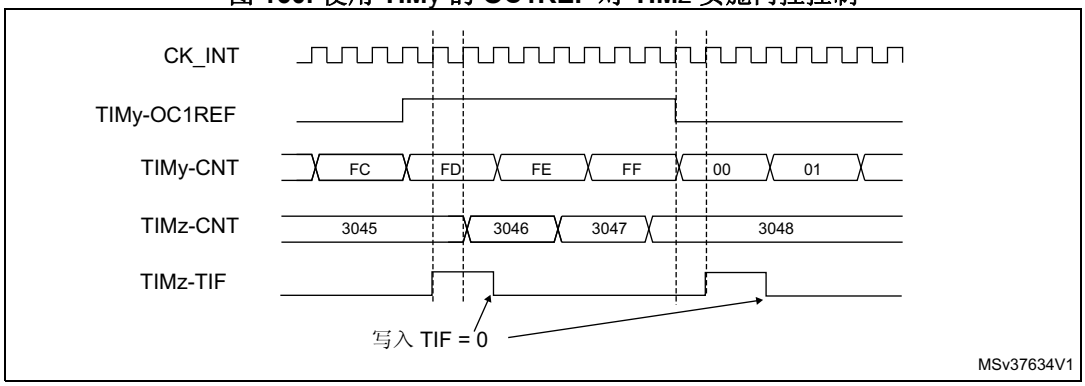
使用一个定时器使能另一个定时器

本例中通过定时器 y 的输出比较 1 来使能 TIMz。相关连接图，请参见图 158。仅当 TIMy 的 OC1REF 为高电平时，TIMz 才根据分频后的内部时钟进行计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 ($f_{CK\_CNT} = f_{CK\_INT}/3$)。

1. 将 TIMy 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出（TIMy_CR2 寄存器中的 MMS=100）。
2. 配置 TIMy 的 OC1REF 波形（TIMy_CCMR1 寄存器）。
3. 配置 z 以接收来自 TIMy 的输入触发（TIMz_SMCR 寄存器中的 TS=00）。
4. 将 TIMz 配置为门控模式（TIMz_SMCR 寄存器中的 SMS=101）。
5. 通过向 CEN 位（TIMz_CR1 寄存器）写入“1”使能 TIMz。
6. 通过向 CEN 位（TIMy_CR1 寄存器）写入“1”启动 TIMy。

注意： *计数器 z 的时钟与计数器 1 不同步，此模式仅影响 TIMz 的计数器使能信号。*

图 160. 使用 TIMy 的 OC1REF 对 TIMz 实施门控控制

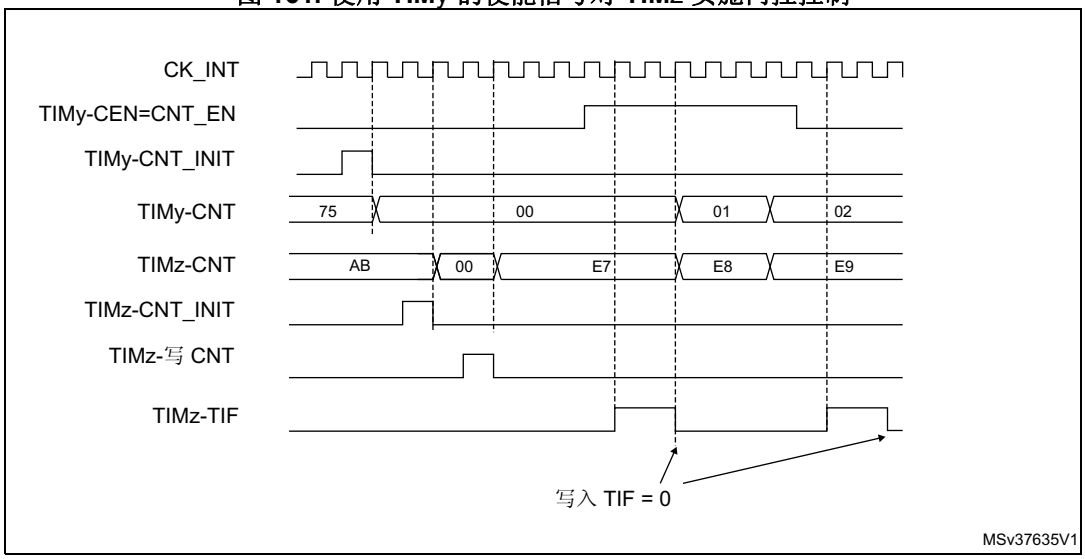


在*图 160* 的示例中，TIMz 的计数器和预分频器在启动前未进行初始化。因此从各自的当前值开始计数。启动定时器 TIMy 之前，通过复位这两个定时器可以从指定值开始计数。然后便可以在定时器计数器中写入任意值。两个定时器都可通过软件使用 TIMx_EGR 寄存器中的 UG 位轻松复位。

在下一示例中（请参见*图 161*），TIMy 与 TIMz 同步。TIMy 为主模式，从 0 开始计数。TIMz 为从模式，从 0xE7 开始计数。两个定时器的预分频比相同。通过向 TIMy_CR1 寄存器中的 CEN 位写入“0”来禁止 TIMy 时，TIMz 将停止：

1. 将 TIMy 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出（TIMy_CR2 寄存器中的 MMS=100）。
2. 配置 TIMy 的 OC1REF 波形（TIMy_CCMR1 寄存器）。
3. 配置 z 以接收来自 TIMy 的输入触发（TIMz_SMCR 寄存器中的 TS=00）。
4. 将 TIMz 配置为门控模式（TIMz_SMCR 寄存器中的 SMS=101）。
5. 通过向 UG 位（TIMy_EGR 寄存器）写入“1”复位 TIMy。
6. 通过向 UG 位（TIMz_EGR 寄存器）写入“1”复位 TIMz。
7. 通过在 TIMz 的计数器 (TIMz_CNTL) 中写入“0xE7”使 TIMz 初始化为 0xE7。
8. 通过向 CEN 位（TIMz_CR1 寄存器）写入“1”使能 TIMz。
9. 通过向 CEN 位（TIMy_CR1 寄存器）写入“1”启动 TIMy。
10. 通过向 CEN 位（TIMy_CR1 寄存器）写入“0”停止 TIMy。

图 161. 使用 TIMy 的使能信号对 TIMz 实施门控控制

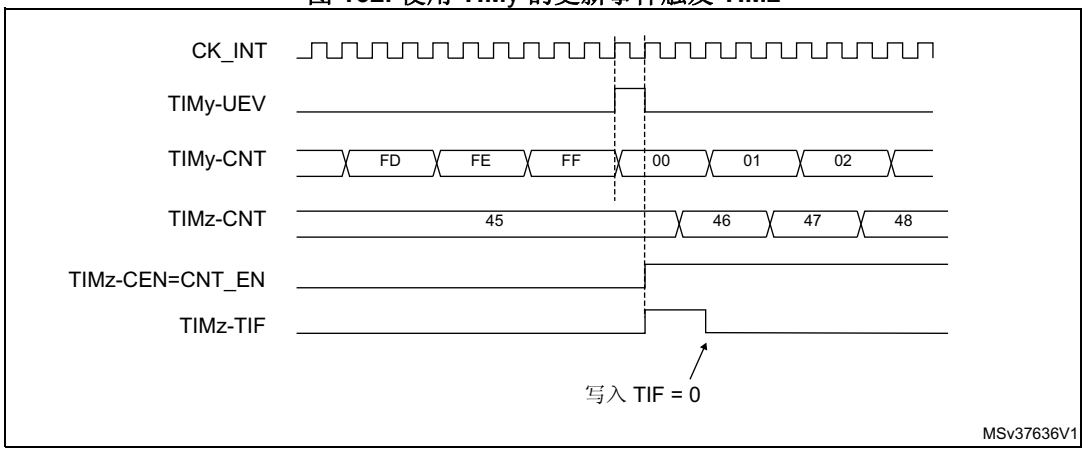


使用一个定时器启动另一个定时器

本例中，使用定时器 y 的更新事件使能定时器 z。相关连接图，请参见*图 158*。只要定时器 1 生成更新事件，定时器 z 便根据分频后的内部时钟从当前值（可以不为 0）开始计数。定时器 z 收到触发信号时，其 CEN 位自动置 1，并且计数器开始计数，直到向 TIMz_CR1 寄存器的 CEN 位写入“0”后停止计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 ($f_{CK\_CNT} = f_{CK\_INT}/3$)。

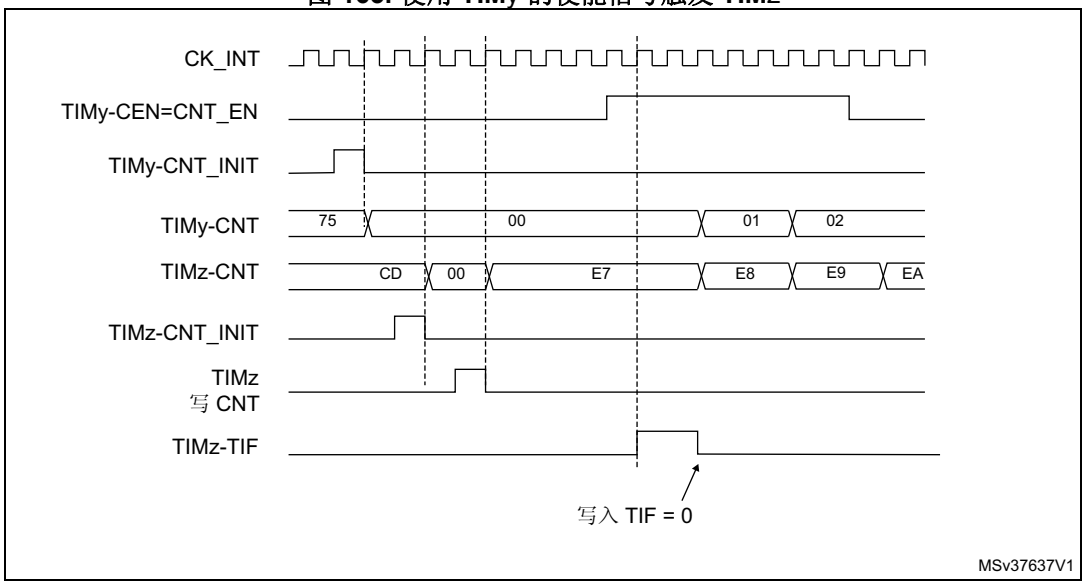
1. 将 TIMy 配置为主模式，发送其更新事件 (UEV) 作为触发输出（TIMy_CR2 寄存器中的 MMS=010）。
2. 配置 TIMy 的周期（TIMy_ARR 寄存器）。
3. 配置 z 以接收来自 TIMy 的输入触发（TIMz_SMCR 寄存器中的 TS=00）。
4. 将 TIMz 配置为触发模式（TIMz_SMCR 寄存器中的 SMS=110）。
5. 通过向 CEN 位（TIMy_CR1 寄存器）写入“1”启动 TIMy。

图 162. 使用 TIMy 的更新事件触发 TIMz



如上述示例所示，用户可以在开始计数之前初始化两个计数器。*图 163* 显示了与*图 162* 具有相同配置，只不过处于触发模式（TIMz_SMCR 寄存器中的 SMS=110）而非门控模式的计数行为。

图 163. 使用 TIMy 的使能信号触发 TIMz



使用一个外部触发同步启动 2 个定时器

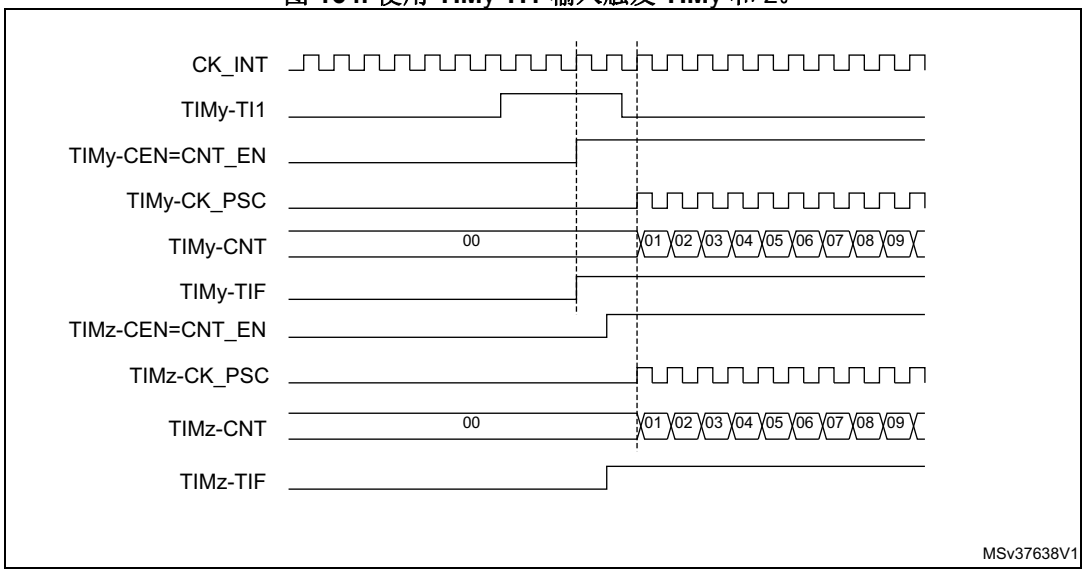
本例中，TIMy 的 TI1 输入出现上升沿时使能 TIMy，使能 TIMy 的同时使能 TIMz。相关连接图，请参见*图 158*。要确保计数器对齐，TIMy 必须配置为主/从模式（对于 TI1 而言，TIMy 为从；对于 TIMz 而言，TIMy 为主）：

1. 将 TIMy 配置为主模式，发送其使能信号作为触发输出（TIMy_CR2 寄存器中的 MMS=001）。
2. 将 TIMy 配置为从模式以接收来自 TI1 的输入触发（TIMy_SMCR 寄存器中的 TS=00100）。
3. 将 TIMy 配置为触发模式（TIMy_SMCR 寄存器中的 SMS=110）。
4. 通过写入 MSM=1（TIMy_SMCR 寄存器）将 TIMy 配置为主/从模式。
5. 配置 TIMz 以接收来自 TIMy 的输入触发（TIMz_SMCR 寄存器中的 TS=00_）。
6. 将 TIMz 配置为触发模式（TIMz_SMCR 寄存器中的 SMS=110）。

当 TI1 (TIMy) 出现上升沿时，两个计数器开始根据内部时钟同步计数，并且两个 TIF 标志都置 1。

注意： *本例中，两个定时器都在启动之前进行了初始化（通过将各自的 UG 位置 1）。两个计数器都从 0 开始计数，但可以通过对任意一个计数器寄存器 (TIMx_CNT) 进行写操作，在二者之间轻松插入一个偏移量。可注意到主/从模式在 TIMy 的 CNT_EN 与 CK_PSC 之间产生了延迟。*

图 164. 使用 TIMy TI1 输入触发 TIMy 和 z。



注意： *必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。*

16.3.20 DMA 分组传送模式

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的一部分多次重新编程而无需软件开销，但也可以用于定期地一次性地批量读取定时器的多个寄存器的内容。

DMA 控制器访问目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（分组）。每次写入 TIMx_DMAR 寄存器都会重定向到定时器其中的一个寄存器。

TIMx_DCR 寄存器中的 DBL[4:0] 位设置 DMA 分组传送长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次分组传送，DBL 即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA[4:0] 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量：

示例：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

例如，定时器 DMA 分组传送功能用于在发生更新事件后将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   - DMA 通道外设地址为 DMAR 寄存器地址。
   - DMA 通道存储器地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
   - 要传输的数据量 = 3（参见下文注释）。
   - 禁止循环模式。

2. 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器：
   DBL = 3 次传输，DBA = 0xE。
3. 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx
5. 使能 DMA 通道

本例适用于每个 CCRx 寄存器必须更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。

注意： *可以将空值写入保留的寄存器中。*

16.3.21 调试模式

当微控制器进入调试模式时（Cortex®-M0+ 内核停止），TIMx 计数器会根据 DBGMCU 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。有关详细信息，请参见 *第 26.9.2 节：对定时器、看门狗和 I2C 的调试支持*。

16.4 TIM3 寄存器

在本章中，应将“TIMx”理解为“TIM3”，因为本参考手册所适用的产品中只有一个此类定时器的实例。

有关寄存器说明中使用的缩写，请参见*第 1.2 节*。

外设寄存器可按半字（16 位）或字（32 位）访问。

16.4.1 TIM3 控制寄存器 1 (TIM3_CR1)

TIM3 control register 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
Res.	Res.	Res.	Res.	UIFRE MAP	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
				rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)
该位域指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟（ETR、TIx）之间的分频比，
00：$t_{DTS} = t_{CK\_INT}$
01：$t_{DTS} = 2 \times t_{CK\_INT}$
10：$t_{DTS} = 4 \times t_{CK\_INT}$
11：Reserved

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)

0：TIMx_ARR 寄存器不进行缓冲

1：TIMx_ARR 寄存器进行缓冲

位 6:5 **CMS[1:0]**：中心对齐模式选择 (Center-aligned mode selection)

00：边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。

01：中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。

10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。

11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志都会置 1。

注意：只要计数器处于使能状态 (CEN=1)，就不得从边沿对齐模式切换为中心对齐模式。

位 4 **DIR**：方向 (Direction)

0：计数器递增计数

1：计数器递减计数

注意：当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。

位 3 **OPM**：单脉冲模式

0：计数器在发生更新事件时不会停止计数

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

位 2 **URS**：更新请求源 (Update request source)

该位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能时，所有以下事件都会生成更新中断或 DMA 请求。此类事件包括：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。

位 1 **UDIS**：更新禁止 (Update disable)

该位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

带缓冲的寄存器被加载为预加载数值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注意：只有事先通过软件将 CEN 位置 1，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1。

在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。

16.4.2 TIM3 控制寄存器 2 (TIM3_CR2)

TIM3 control register 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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			CCDS	Res.	Res.	Res.
								rw	rw	rw	rw	rw			

位 15:8 保留，必须保持复位值。

位 7 **TI1S**：TI1 选择 (TI1 selection)

0：TIMx_CH1 引脚连接到 TI1 输入

1：TIMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入（异或组合）。另请参见*第 321 页的第 15.3.25 节：连接霍尔传感器*

位 6:4 **MMS[2:0]**：主模式选择 (Master mode selection)

这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下：

000：**复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。

001：**使能**——计数器使能信号 (CNT_EN) 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑与运算组合而成。

当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。

010：**更新**——选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。

011：**比较脉冲**——CC1IF 标志置 1 时（即使已为高电平），只要发生捕获或比较匹配，触发输出都会发送一个正脉冲。(TRGO)

100：**比较**——OC1REFC 信号用作触发输出 (TRGO)

101：**比较**——OC2REFC 信号用作触发输出 (TRGO)

110：**比较**——OC3REFC 信号用作触发输出 (TRGO)

111：**比较**——OC4REFC 信号用作触发输出 (TRGO)

注意：必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 3 **CCDS**：捕获/比较 DMA 选择 (Capture/compare DMA selection)

0：发生 CCx 事件时发送 CCx DMA 请求

1：发生更新事件时发送 CCx DMA 请求

位 2:0 保留，必须保持复位值。

16.4.3 TIM3 从模式控制寄存器 (TIM3_SMCR)

TIM3 slave mode control register

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

位 31:22 保留，必须保持复位值。

位 19:17 保留，必须保持复位值。

位 15 **ETP**：外部触发极性 (External trigger polarity)
该位可选择将 ETR 还是 $\overline{\text{ETR}}$ 用于触发操作
0：ETR 未反相，高电平或上升沿有效
1：ETR 反相，低电平或下降沿有效

位 14 **ECE**：外部时钟使能 (External clock enable)
该位可使能外部时钟模式 2。
0：禁止外部时钟模式 2
1：使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。
注意：将 ECE 位置 1 与选择外部时钟模式 1 并将 TRGI 连接到 ETRF（SMS=111 且 TS=00111）具有相同效果。
外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF（TS 位不得为 00111）。
如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。

位 13:12 **ETPS[1:0]**：外部触发预分频器 (External trigger prescaler)
外部触发信号 ETRP 频率不得超过 CK_INT 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。
00：预分频器关闭
01：2 分频 ETRP 频率
10：4 分频 ETRP 频率
11：8 分频 ETRP 频率

位 11:8 **ETF[3:0]**：外部触发滤波器 (External trigger filter)

该位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 7 **MSM**：主/从模式 (Master/Slave mode)

0：无操作
1：当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。

位 21、20、6、5、4 **TS[4:0]**：触发选择 (Trigger selection)

该位域可选择将要用于同步计数器的触发输入。

00000：内部触发 0 (ITR0)
00001：内部触发 1 (ITR1)
00010：内部触发 2 (ITR2)
00011：内部触发 3 (ITR3)
00100：TI1 边沿检测器 (TI1F_ED)
00101：滤波后的定时器输入 1 (TI1FP1)
00110：滤波后的定时器输入 2 (TI2FP2)
00111：外部触发输入 (ETRF)
01000：内部触发 4 (ITR4)
01001：内部触发 5 (ITR5)
01010：内部触发 6 (ITR6)
01011：内部触发 7 (ITR7)
01100：内部触发 8 (ITR8)
其他值：Reserved

有关各定时器 ITRx 含义的详细信息，请参见*第 410 页的表 69：TIM3 内部触发连接*。

注意：这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

位 3 **OCCS**：OCREF 清零选择 (OCREF clear selection)

该位用于选择 OCREF 清零源
0：OCREF_CLR_INT 无连接。
1：OCREF_CLR_INT 连接到 ETRF

位 16、2、1、0 **SMS[3:0]**：从模式选择 (Slave mode selection)

选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。

0000：禁止从模式——如果 CEN = “1”，预分频器时钟直接由内部时钟提供。
0001：编码器模式 1——计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。
0010：编码器模式 2——计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。
0011：编码器模式 3——计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个信号的电平。
0100：复位模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。
0101：门控模式——触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。
0110：触发模式——触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。
0111：外部时钟模式 1——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。
1000：复位 + 触发组合模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个定时器更新事件并启动计数器。

注意：如果将 TI1F_ED 选作触发输入 (TS=00100)，则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。

注意：必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

表 69. TIM3 内部触发连接

从 TIM	ITR0	ITR1	ITR2	ITR3
TIM3	TIM1	-	-	TIM14_OC1

16.4.4 TIM3 DMA/中断使能寄存器 (TIM3_DIER)

TIM3 DMA/Interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

位 15 保留，必须保持复位值。

位 14 **TDE**：触发信号 (TGRI) DMA 请求使能 (Trigger DMA request enable)
0：禁止触发信号 (TGRI) DMA 请求。
1：使能触发信号 (TGRI) DMA 请求。

位 13 保留，必须保持复位值。

位 12 **CC4DE**：捕获/比较 4 DMA 请求使能 (Capture/Compare 4 DMA request enable)
0：禁止 CC4 DMA 请求。
1：使能 CC4 DMA 请求。

位 11 **CC3DE**：捕获/比较 3 DMA 请求使能 (Capture/Compare 3 DMA request enable)
0：禁止 CC3 DMA 请求。
1：使能 CC3 DMA 请求。

位 10 **CC2DE**：捕获/比较 2 DMA 请求使能 (Capture/Compare 2 DMA request enable)
0：禁止 CC2 DMA 请求。
1：使能 CC2 DMA 请求。

位 9 **CC1DE**：捕获/比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)
0：禁止 CC1 DMA 请求。
1：使能 CC1 DMA 请求。

位 8 **UDE**：更新 DMA 请求使能 (Update DMA request enable)
0：禁止更新 DMA 请求。
1：使能更新 DMA 请求。

位 7 保留，必须保持复位值。

位 6 **TIE**：触发信号 (TGRI) 中断使能 (Trigger interrupt enable)
0：禁止触发信号 (TGRI) 中断。
1：使能触发信号 (TGRI) 中断。

位 5 保留，必须保持复位值。

位 4 **CC4IE**：捕获/比较 4 中断使能 (Capture/Compare 4 interrupt enable)
0：禁止 CC4 中断。
1：使能 CC4 中断。

位 3 **CC3IE**：捕获/比较 3 中断使能 (Capture/Compare 3 interrupt enable)
0：禁止 CC3 中断。
1：使能 CC3 中断。

位 2 **CC2IE**：捕获/比较 2 中断使能 (Capture/Compare 2 interrupt enable)
0：禁止 CC2 中断。
1：使能 CC2 中断。

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断。
1：使能 CC1 中断。

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断。
1：使能更新中断。

16.4.5 TIM3 状态寄存器 (TIM3_SR)

TIM3 status register

偏移地址：0x10

复位值：0x0000

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

位 15:13 保留，必须保持复位值。

位 12 **CC4OF**：捕获/比较 4 过捕获标志 (Capture/Compare 4 overcapture flag)
请参见 CC1OF 说明

位 11 **CC3OF**：捕获/比较 3 过捕获标志 (Capture/Compare 3 overcapture flag)
请参见 CC1OF 说明

位 10 **CC2OF**：捕获/比较 2 过捕获标志 (Capture/compare 2 overcapture flag)
请参见 CC1OF 说明

位 9 **CC1OF**：捕获/比较 1 过捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。
0：未检测到过捕获。
1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8:7 保留，必须保持复位值。

位 6 **TIF**：触发中断标志 (Trigger interrupt flag)
在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到 TRG 触发事件有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。
0：未发生触发事件。
1：触发中断挂起。

位 5 保留，必须保持复位值。

位 4 **CC4IF**：捕获/比较 4 中断标志 (Capture/Compare 4 interrupt flag)
请参见 CC1IF 说明

位 3 **CC3IF**：捕获/比较 3 中断标志 (Capture/Compare 3 interrupt flag)
请参见 CC1IF 说明

位 2 **CC2IF**：捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)
请参见 CC1IF 说明

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/compare 1 interrupt flag)
该标志由硬件置 1，但需要通过软件清零（输入捕获或输出比较模式）或通过读取 TIMx_CCR1 寄存器清零（仅限输入捕获模式）。
0：未发生比较匹配/输入捕获
1：发生了比较匹配/输入捕获
如果通道 CC1 配置为输出：当计数器 TIMx_CNT 的值与 TIMx_CCR1 的值匹配时，此标志置 1。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。在中心对齐模式下，可通过 3 种方式将此标志置 1，完整说明请参见 TIMx_CR1 寄存器中的 CMS 位。
如果通道 CC1 配置为输入：当 TIMx_CCR1 寄存器中捕获到计数器值时（如果通过设置 TIMx_CCER 中的 CC1P 和 CC1NP 位定义为边沿有效，则改为在 IC1 上检测到边沿时），该位置 1。

位 0 **UIF**：更新中断标志 (Update interrupt flag)
该位在发生更新事件时通过硬件置 1。但需要通过软件清零。
0：未发生更新
1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：
上溢或下溢并且当 TIMx_CR1 寄存器中 UDIS = 0 时。
TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。
TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（参见同步控制寄存器说明）。

16.4.6 TIM3 事件生成寄存器 (TIM3_EGR)

TIM3 event generation register

偏移地址：0x14

复位值：0x0000

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

位 15:7 保留，必须保持复位值。

位 6 **TG**：触发生成 (Trigger generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：无操作
1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 5 保留，必须保持复位值。

位 4 **CC4G**：捕获/比较 4 生成 (Capture/compare 4 generation)
请参见 CC1G 说明

位 3 **CC3G**：捕获/比较 3 生成 (Capture/compare 3 generation)
请参见 CC1G 说明

位 2 **CC2G**：捕获/比较 2 生成 (Capture/compare 2 generation)
请参见 CC1G 说明

位 1 **CC1G**：捕获/比较 1 生成 (Capture/compare 1 generation)
该位由软件置 1 以生成事件，并由硬件自动清零。
0：无操作
1：通道 1 上生成捕获/比较事件：
如果通道 CC1 配置为输出：
使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。
如果通道 CC1 配置为输入：
TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)
该位可通过软件置 1，并由硬件自动清零。
0：无操作
1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值 (TIMx_ARR)。

16.4.7 TIM3 捕获/比较模式寄存器 1 (TIM3_CCMR1)

TIM3 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:12 **IC2F[3:0]**：输入捕获 2 滤波器 (Input capture 2 filter)

位 11:10 **IC2PSC[1:0]**：输入捕获 2 预分频器 (Input capture 2 prescaler)

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/compare 2 selection)

该位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出。

01：CC2 通道配置为输入，IC2 映射到 TI2 上。

10：CC2 通道配置为输入，IC2 映射到 TI1 上。

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)

该位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样

0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC[1:0]**：输入捕获 1 预分频器 (Input capture 1 prescaler)

该位域定义 CC1 输入 (IC1) 的预分频比。只要 CC1E=0（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

该位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：CC1 通道配置为输入，IC1 映射到 TI2 上
11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

16.4.8 TIM3 捕获/比较模式寄存器 1 [复用] (TIM3_CCMR1)

TIM3 capture/compare mode register 1 [alternate]

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
							rw								rw

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

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 **OC2CE**：输出比较 2 清零使能 (Output compare 2 clear enable)

位 24、14:12 **OC2M[3:0]**：输出比较 2 模式 (Output compare 2 mode)

请参见 OC1M 说明——位 6:4

位 11 **OC2PE**：输出比较 2 预装载使能 (Output compare 2 preload enable)

位 10 **OC2FE**：输出比较 2 快速使能 (Output compare 2 fast enable)

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)

该位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出

01：CC2 通道配置为输入，IC2 映射到 TI2 上

10：CC2 通道配置为输入，IC2 映射到 TI1 上

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

位 7 **OC1CE**：输出比较 1 清零使能 (Output compare 1 clear enable)

0：OC1Ref 不受 ETRF 输入影响

1：ETRF 输入上检测到高电平时，OC1Ref 立即清零。

位 16，6:4 **OC1M[3:0]**：输出比较 1 模式 (Output compare 1 mode)

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。当定时器用作软件时基时，可以使用该模式。在定时器操作期间使能冻结模式时，输出保持进入冻结状态前的状态（有效或无效）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便输出有效电平，否则输出无效电平。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便输出无效电平 (OC1REF=0)，否则输出有效电平 (OC1REF=1)。

0111：PWM 模式 2——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便输出无效电平，否则输出有效电平。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 输出有效电平，否则输出无效电平。

1000：可再触发 OPM 模式 1——在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。

1001：可再触发 OPM 模式 2——在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。

1010：保留。

1011：保留。

1100：组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。

1101：组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。

1110：不对称 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

1111：不对称 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

注意：在 PWM 模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。

注意：OC1M[3] 位不是连续的，而是在位 16 中。

位 3 **OC1PE**：输出比较 1 预装载使能 (Output compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到影子寄存器中。

位 2 **OC1FE**：输出比较 1 快速使能 (Output compare 1 fast enable)

该位可降低触发事件与定时器输出跳变之间的延迟。单脉冲模式下必须使用（TIMx_CR1 寄存器中的 OPM 位置 1），以便在启动触发后快速发送输出脉冲。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

该位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出。

01：CC1 通道配置为输入，IC1 映射到 TI1 上。

10：CC1 通道配置为输入，IC1 映射到 TI2 上。

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

16.4.9 TIM3 捕获/比较模式寄存器 2 (TIM3_CCMR2)

TIM3 capture/compare mode register 2

偏移地址：0x1C

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4S[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:12 **IC4F[3:0]**：输入捕获 4 滤波器 (Input capture 4 filter)

位 11:10 **IC4PSC[1:0]**：输入捕获 4 预分频器 (Input capture 4 prescaler)

位 9:8 **CC4S[1:0]**：捕获/比较 4 选择 (Capture/Compare 4 selection)
该位域定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIMx_CCER 中的 CC4E = 0），才可向 CC4S 位写入数据。

位 7:4 **IC3F[3:0]**：输入捕获 3 滤波器 (Input capture 3 filter)

位 3:2 **IC3PSC[1:0]**：输入捕获 3 预分频器 (Input capture 3 prescaler)

位 1:0 **CC3S[1:0]**：捕获/比较 3 选择 (Capture/Compare 3 selection)
该位域定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIMx_CCER 中的 CC3E = 0），才可向 CC3S 位写入数据。

16.4.10 TIM3 捕获/比较模式寄存器 2 [复用] (TIM3_CCMR2)

TIM3 capture/compare mode register 2 [alternate]

偏移地址：0x1C

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M [3]
							rw								rw

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

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 **OC4CE**：输出比较 4 清零使能 (Output compare 4 clear enable)

位 24、14:12 **OC4M[3:0]**：输出比较 4 模式
请参见 OC1M 说明（TIMx_CCMR1 寄存器中的位 6:4）

位 11 **OC4PE**：输出比较 4 预装载使能 (Output compare 4 preload enable)

位 10 **OC4FE**：输出比较 4 快速使能 (Output compare 4 fast enable)

位 9:8 **CC4S[1:0]**：捕获/比较 4 选择 (Capture/Compare 4 selection)
该位域定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIMx_CCER 中的 CC4E = 0），才可向 CC4S 位写入数据。

位 7 **OC3CE**：输出比较 3 清零使能 (Output compare 3 clear enable)

位 16，6:4 **OC3M[3:0]**：输出比较 3 模式 (Output compare 3 mode)
请参见 OC1M 说明（TIMx_CCMR1 寄存器中的位 6:4）

位 3 **OC3PE**：输出比较 3 预装载使能 (Output compare 3 preload enable)

位 2 **OC3FE**：输出比较 3 快速使能 (Output compare 3 fast enable)

位 1:0 **CC3S[1:0]**：捕获/比较 3 选择 (Capture/Compare 3 selection)
该位域定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（TIMx_CCER 中的 CC3E = 0），才可向 CC3S 位写入数据。

16.4.11 TIM3 捕获/比较使能寄存器 (TIM3_CCER)

TIM3 capture/compare enable register

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

位 15 **CC4NP**：*捕获/比较 4 输出极性 (Capture/Compare 4 output Polarity)*。
请参见 CC1NP 说明

位 14 保留，必须保持复位值。

位 13 **CC4P**：*捕获/比较 4 输出极性 (Capture/Compare 4 output Polarity)*。
请参见 CC1P 说明

位 12 **CC4E**：*捕捉/比较 4 输出使能 (Capture/Compare 4 output enable)*。
请参见 CC1E 说明

位 11 **CC3NP**：*捕获/比较 3 输出极性 (Capture/Compare 3 output polarity)*。
请参见 CC1NP 说明

位 10 保留，必须保持复位值。

位 9 **CC3P**：*捕获/比较 3 输出极性 (Capture/Compare 3 output polarity)*。
请参见 CC1P 说明

位 8 **CC3E**：*捕捉/比较 3 输出使能 (Capture/Compare 3 output enable)*。
请参见 CC1E 说明

位 7 **CC2NP**：*捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)*。
请参见 CC1NP 说明

位 6 保留，必须保持复位值。

位 5 **CC2P**：*捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)*。
请参见 CC1P 说明

位 4 **CC2E**：*捕捉/比较 2 输出使能 (Capture/Compare 2 output enable)*。
请参见 CC1E 说明

位 3 **CC1NP**：*捕获/比较 1 输出极性 (Capture/Compare 1 output polarity)*。
CC1 通道配置为输出：在这种情况下，CC1NP 必须保持清零。
CC1 通道配置为输入：该位与 CC1P 配合使用，用以定义 TI1FP1/TI2FP1 的极性。请参见 CC1P 说明。

位 2 保留，必须保持复位值。

位 1 **CC1P**：*捕获/比较 1 输出极性 (Capture/Compare 1 output polarity)*。
0：OC1 高电平有效（输出模式）/边沿有效性选择（输入模式，见下文）
1：OC1 低电平有效（输出模式）/边沿有效性选择（输入模式，见下文）
CC1 通道配置为输入时，CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。
CC1NP=0，CC1P=0：非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。
CC1NP=0，CC1P=1：反相/下降沿触发。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。
CC1NP=1，CC1P=1：非反相/上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。
CC1NP=1，CC1P=0：该配置保留，不得使用。

位 0 **CC1E**：*捕获/比较 1 输出使能 (Capture/Compare 1 output enable)*。
0：禁止捕获模式/OC1 无效
1：使能捕获模式/在相应输出引脚上输出 OC1 信号

表 70. 标准 OCx 通道的输出控制位

CCxE 位	OCx 输出状态
0	禁止输出（不由定时器驱动：高阻态
1	使能输出（tim_ocx = tim_ocxref + 极性）

注意： *与标准 OCx 通道相连的外部 IO 引脚的状态取决于 OCx 通道的状态以及 GPIO 控制和复用功能寄存器。*

16.4.12 TIM3 计数器 (TIM3_CNT)

TIM3 counter

该寄存器的位 31 有两种可能的定义，具体取决于 TIMx_CR1 寄存器中 UIFREMAP 的值：

- 本节内容对应于 UIFREMAP = 0 的情况
- 下一节内容对应于 UIFREMAP = 1 的情况

偏移地址：0x24

复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值。

位 15:0 **CNT**[15:0]：计数器值 (counter value)

16.4.13 TIM3 计数器 [复用] (TIM3_CNT)

TIM3 counter [alternate]

该寄存器的位 31 有两种可能的定义，具体取决于 TIMx_CR1 寄存器中 UIFREMAP 的值：

- 上一节内容对应于 UIFREMAP = 0 的情况
- 本节内容对应于 UIFREMAP = 1 的情况

偏移地址：0x24

复位值：0x0000 0000

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

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

位 31 **UIFCPY**：UIF 副本 (UIF Copy)
该位是 TIMx_ISR 寄存器中 UIF 位的只读副本

位 30:16 保留，必须保持复位值。

位 15:0 **CNT**[15:0]：计数器值 (counter value)

16.4.14 TIM3 预分频器 (TIM3_PSC)

TIM3 prescaler

偏移地址：0x28

复位值：0x0000

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

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)

计数器时钟频率 CK_CNT 等于 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。

16.4.15 TIM3 自动重载寄存器 (TIM3_ARR)

TIM3 auto-reload register

偏移地址：0x2C

复位值：0xFFFF FFFF

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

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

位 31:16 保留，必须保持复位值。

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的更多详细信息，请参见*第 365 页的第 16.3.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

16.4.16 TIM3 捕获/比较寄存器 1 (TIM3_CCR1)

TIM3 capture/compare register 1

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **CCR1[15:0]**：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：

CCR1 是捕获/比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际的捕获/比较寄存器 1）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出相应电平的值。

如果通道 CC1 配置为输入：

CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。只能读取 TIMx_CCR1 寄存器，无法对其进行编程。

16.4.17 TIM3 捕获/比较寄存器 2 (TIM3_CCR2)

TIM3 capture/compare register 2

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **CCR2[15:0]**：捕获/比较 2 值 (Capture/Compare 2 value)

如果通道 CC2 配置为输出：

CCR2 是捕获/比较寄存器 2 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 2）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。

如果通道 CC2 配置为输入：

CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。只能读取 TIMx_CCR2 寄存器，无法对其进行编程。

16.4.18 TIM3 捕获/比较寄存器 3 (TIM3_CCR3)

TIM3 capture/compare register 3

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **CCR3[15:0]**：捕获/比较值 (Capture/Compare value)

如果通道 CC3 配置为输出：

CCR3 是捕获/比较寄存器 3 的预装载值。

如果没有通过 TIMx_CCMR2 寄存器中的 OC3PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 3）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。

如果通道 CC3 配置为输入：

CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。只能读取 TIMx_CCR3 寄存器，无法对其进行编程。

16.4.19 TIM3 捕获/比较寄存器 4 (TIM3_CCR4)

TIM3 capture/compare register 4

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **CCR4[15:0]**：捕获/比较值 (Capture/Compare value)

1. 如果 CC4 通道配置为输出（CC4S 位）：
   CCR4 是捕获/比较寄存器 4 的预装载值。
   如果没有通过 TIMx_CCMR2 寄存器中的 OC4PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 4）。
   有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。
2. 如果 CC4 通道配置为输入（TIMx_CCMR4 寄存器中的 CC4S 位）：
   CCR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值。只能读取 TIMx_CCR4 寄存器，无法对其进行编程。

16.4.20 TIM3 DMA 控制寄存器 (TIM3_DCR)

TIM3 DMA control register

偏移地址：0x48

复位值：0x0000

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

位 15:13 保留，必须保持复位值。

位 12:8 **DBL[4:0]**：DMA 分组传送长度 (DMA burst length)

该 5 位向量定义了 DMA 的传送次数（当对 TIMx_DMAR 寄存器进行读或写时，定时器进行一次分组传送）。

00000：1 次传输，
00001：2 次传输，
00010：3 次传输，
...
10001：18 次传输。

位 7:5 保留，必须保持复位值。

位 4:0 **DBA[4:0]**：DMA 基址 (DMA base address)

该 5 位向量定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：
00000：TIMx_CR1
00001：TIMx_CR2
00010：TIMx_SMCR
...

示例：考虑以下传输：DBL = 7 次传输，DBA = TIMx_CR1。这种情况下将向/从自 TIMx_CR1 地址开始的 7 个寄存器传输数据。

16.4.21 TIM3 全传输 DMA 地址(TIM3_DMAR)

TIM3 DMA address for full transfer

偏移地址：0x4C

复位值：0x0000

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

位 15:0 **DMAB[15:0]**：DMA 分组传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器

(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

16.4.22 TIM3 复用功能选项寄存器 1 (TIM3_AF1)

TIM3 alternate function option register 1

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

位 31:18 保留，必须保持复位值。

位 17:14 **ETRSEL[3:0]**：ETR 源选择 (ETR source selection)
这些位选择 ETR 输入源。
0000：ETR legacy mode
其他值：Reserved

位 13:0 保留，必须保持复位值。

16.4.23 TIM3 定时器输入选择寄存器 (TIM3_TISEL)

TIM3 timer input selection register

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI3SEL[3:0]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:20 保留，必须保持复位值。

位 19:16 **TI3SEL[3:0]**：TI3[0] 到 TI3[15] 输入选择 (TI3[0] to TI3[15] input selection)
这些位选择 TI3[0] 到 TI3[15] 输入源。
0000：TIM3_CH3 输入
其他值：Reserved

位 15:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：TI2[0] 到 TI2[15] 输入选择 (TI2[0] to TI2[15] input selection)
这些位选择 TI2[0] 到 TI2[15] 输入源。
0000：TIM3_CH2 输入
其他值：Reserved

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：TI1[0] 到 TI1[15] 输入选择 (TI1[0] to TI1[15] input selection)
这些位选择 TI1[0] 到 TI1[15] 输入源。
0000：TIM3_CH1 输入
其他值：Reserved

16.4.24 TIMx 寄存器映射

TIMx 寄存器按照下表所述方式映射：

表 71. TIM3 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0	0	0	0	0	0	0	0
0x04	**TIMx_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			CCDS	Res.	Res.	Res.
	Reset value																									0	0	0	0	0			
0x08	**TIMx_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS [4:3]		Res.	Res.	Res.	SMS[3]	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value											0	0				0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																		0		0	0	0	0	0		0		0	0	0	0	0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value																				0	0	0	0			0		0	0	0	0	0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																										0		0	0	0	0	0
0x18	**TIMx_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	OC2CE	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**TIMx_CCMR2** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]	O24CE	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR2 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]				IC4 PSC [1:0]		CC4S [1:0]		IC3F[3:0]				IC3 PSC [1:0]		CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**TIMx_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
	Reset value																	0		0	0	0		0	0	0		0	0	0		0	0

表 71. TIM3 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x24	**TIMx_CNT**	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIMx_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIMx_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	Reserved																																
0x34	**TIMx_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIMx_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**TIMx_CCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	**TIMx_CCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	Reserved																																
0x48	**TIMx_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIMx_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM3_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value															0	0	0	0														
0x68	**TIM3_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI3SEL[3:0]				Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value													0	0	0	0					0	0	0	0					0	0	0	0

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

17 通用定时器 (TIM14)

17.1 TIM14 简介

TIM14 通用定时器包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

它可用于多种用途，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输出比较、PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

TIM14 定时器完全独立，不共享任何资源。

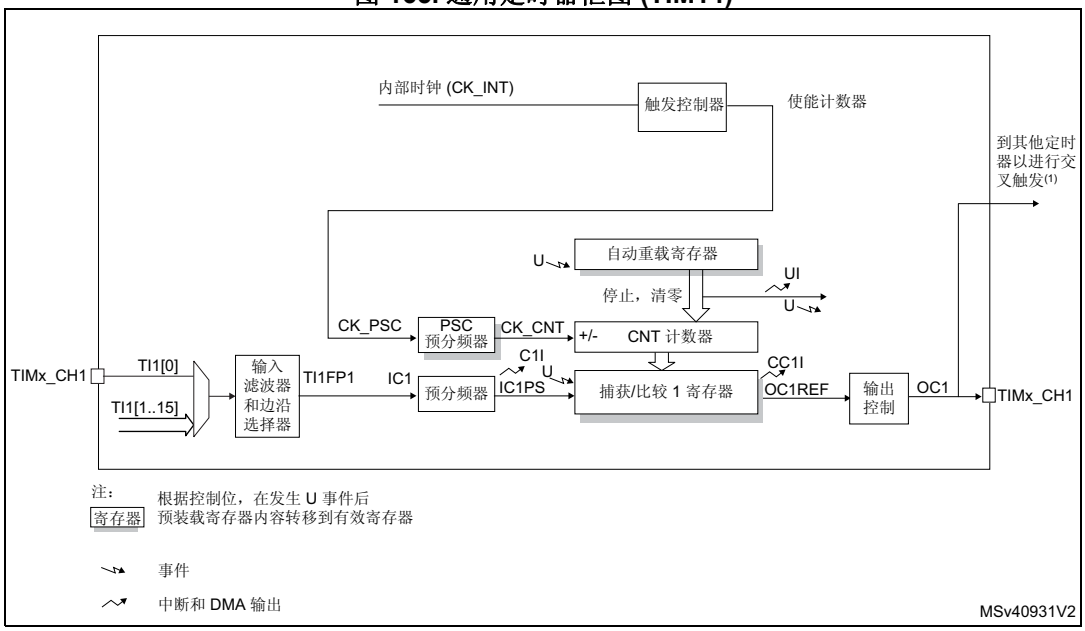
17.2 TIM14 主要特性

17.2.1 TIM14 主要特性

通用定时器 TIM14 具有以下特性：

- 16 位自动重载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65536 之间
- 独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿对齐模式）
  - 单脉冲模式输出
- 发生如下事件时生成中断：
  - 更新：计数器上溢、计数器初始化（通过软件）
  - 输入捕获
  - 输出比较

图 165. 通用定时器框图 (TIM14)



1. 该信号可用于触发一些从定时器，请参见*第 17.3.11 节：使用定时器输出作为其他定时器的触发 (TIM14)*。

17.3 TIM14 功能描述

17.3.1 时基单元

定时器的主要模块由一个 16 位递增计数器及其相关的自动重载寄存器组成。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以随时直接写入影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将产生更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器。

注意，计数器将在 TIMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器说明

预分频器可对计数器时钟进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 TIMx_PSC 寄存器中的 16 位寄存器所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 166 和*图 167* 以一些示例说明在预分频比实时变化时计数器的行为。

图 166. 预分频器分频由 1 变为 2 时的计数器时序图

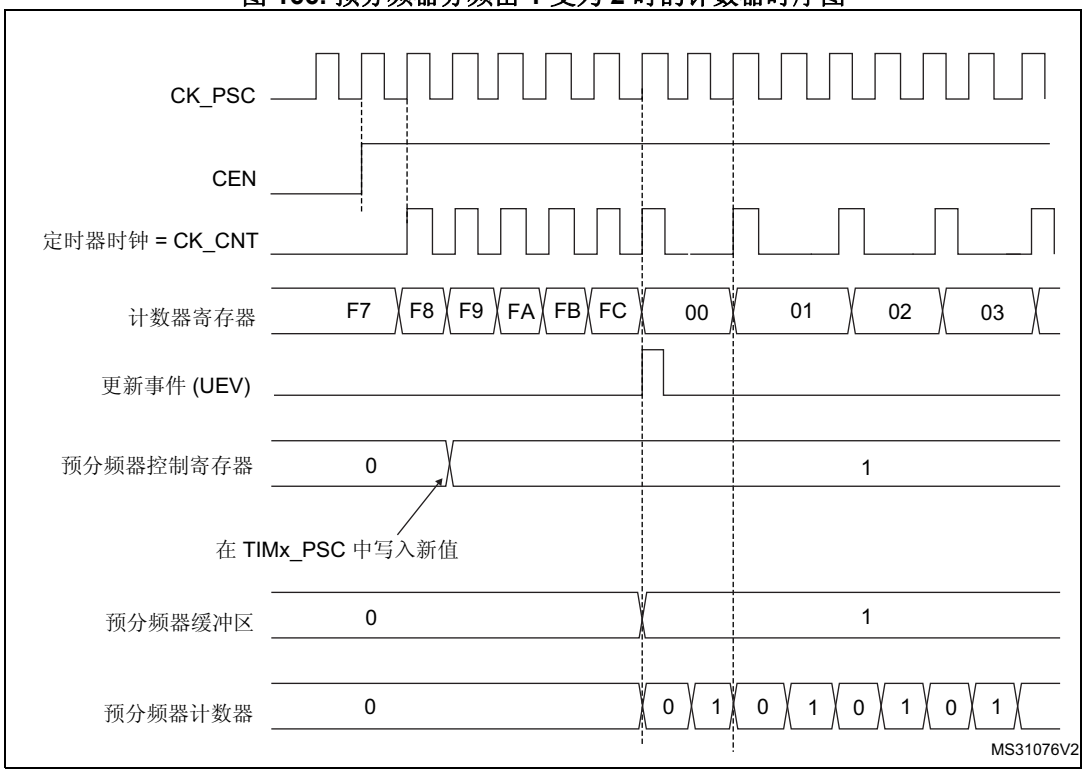
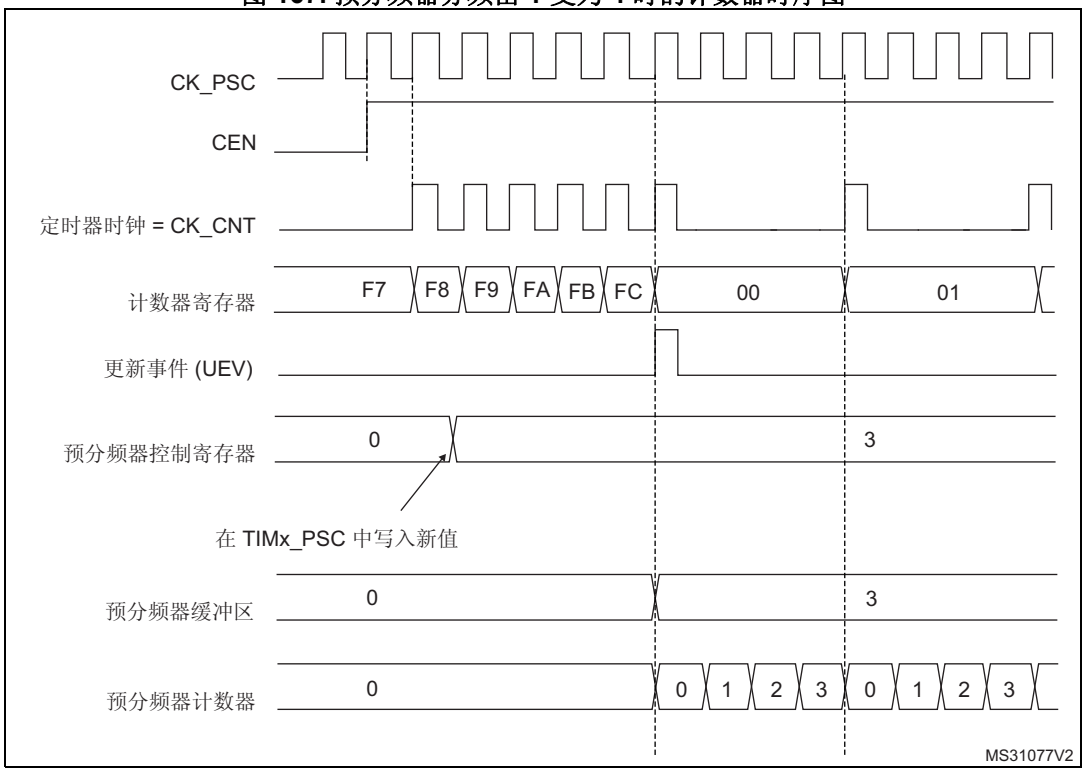


图 167. 预分频器分频由 1 变为 4 时的计数器时序图



17.3.2 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

每次发生计数器上溢时会生成更新事件。TIMx_EGR 寄存器中的 UG 位置 1（通过软件）时，也将产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断）。利用这一特性，在定时器的输入捕获操作时可以通过对 UG 位置 1 让计数器清零而不至于触发更新中断，也就避免了在捕获中断里触发更新中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 自动重载影子寄存器将以预装载值 (TIMx_ARR) 进行更新，
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 168. 计数器时序图，1 分频内部时钟

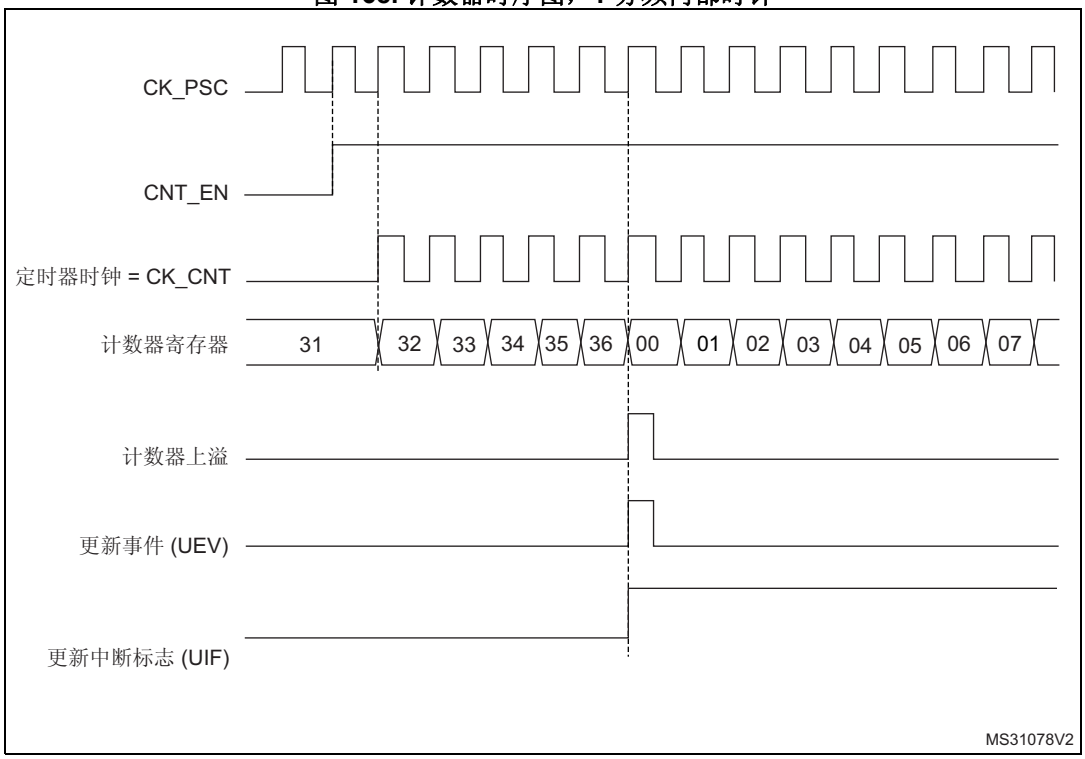


图 169. 计数器时序图，2 分频内部时钟

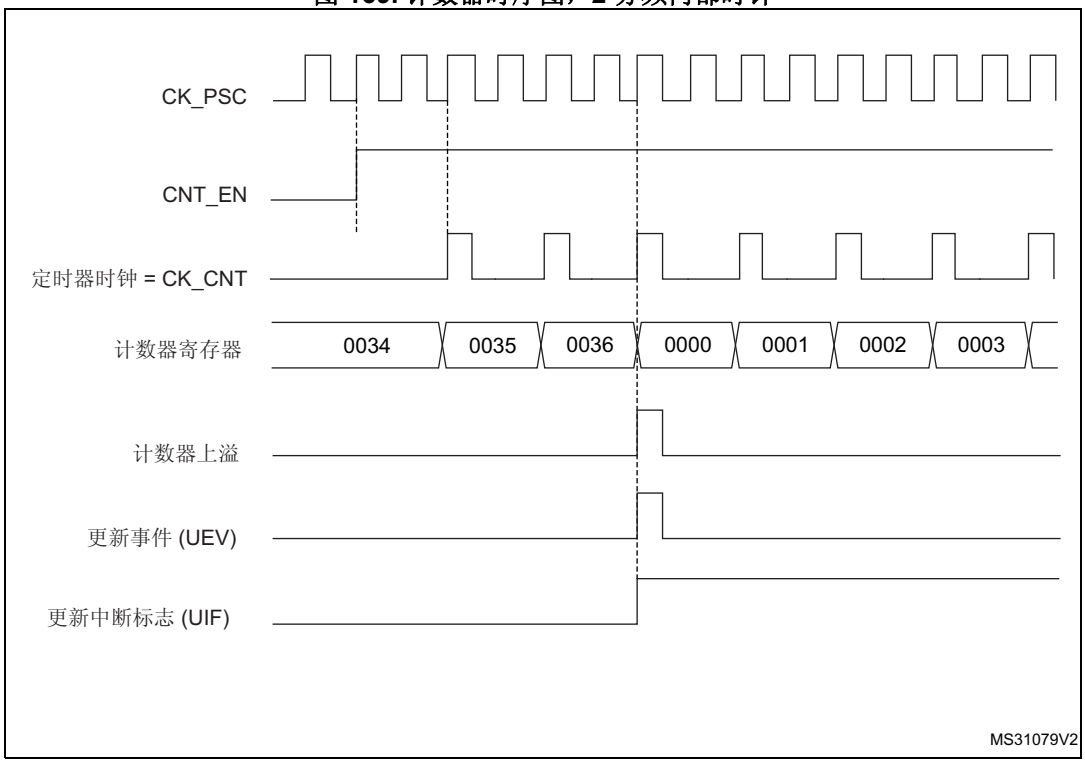


图 170. 计数器时序图，4 分频内部时钟

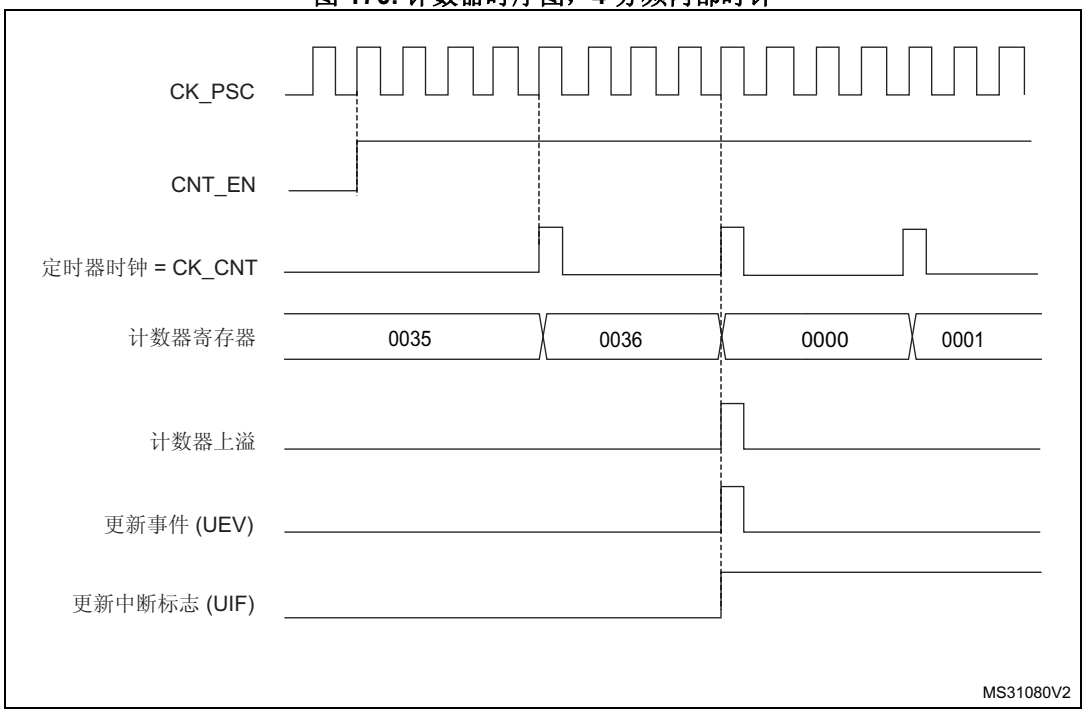


图 171. 计数器时序图，N 分频内部时钟

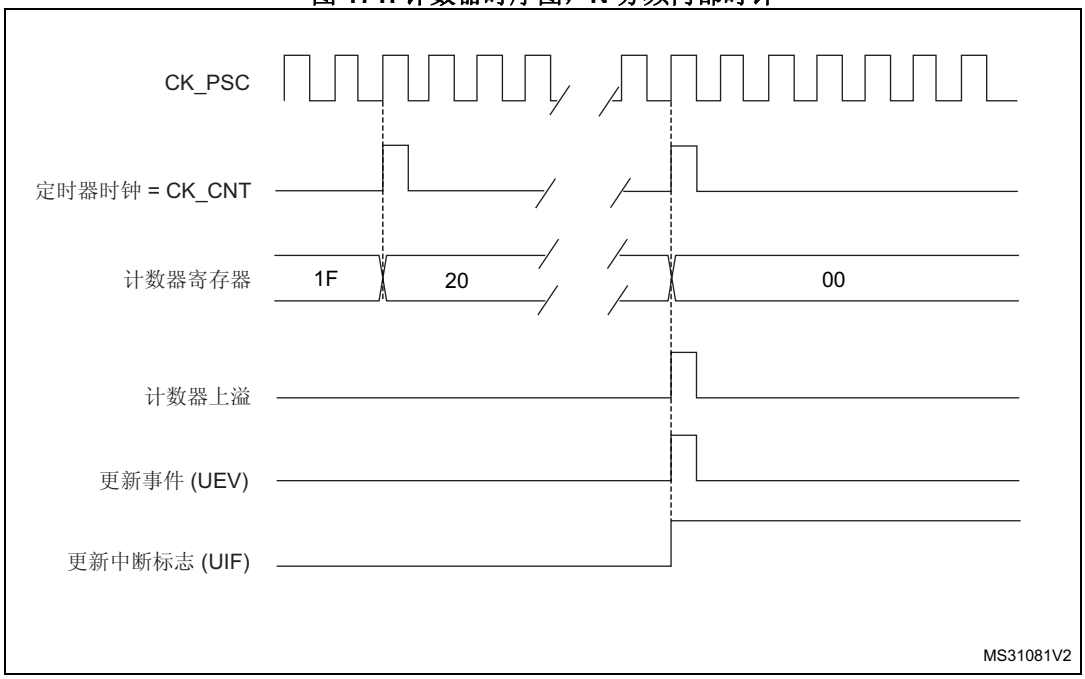


图 172. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

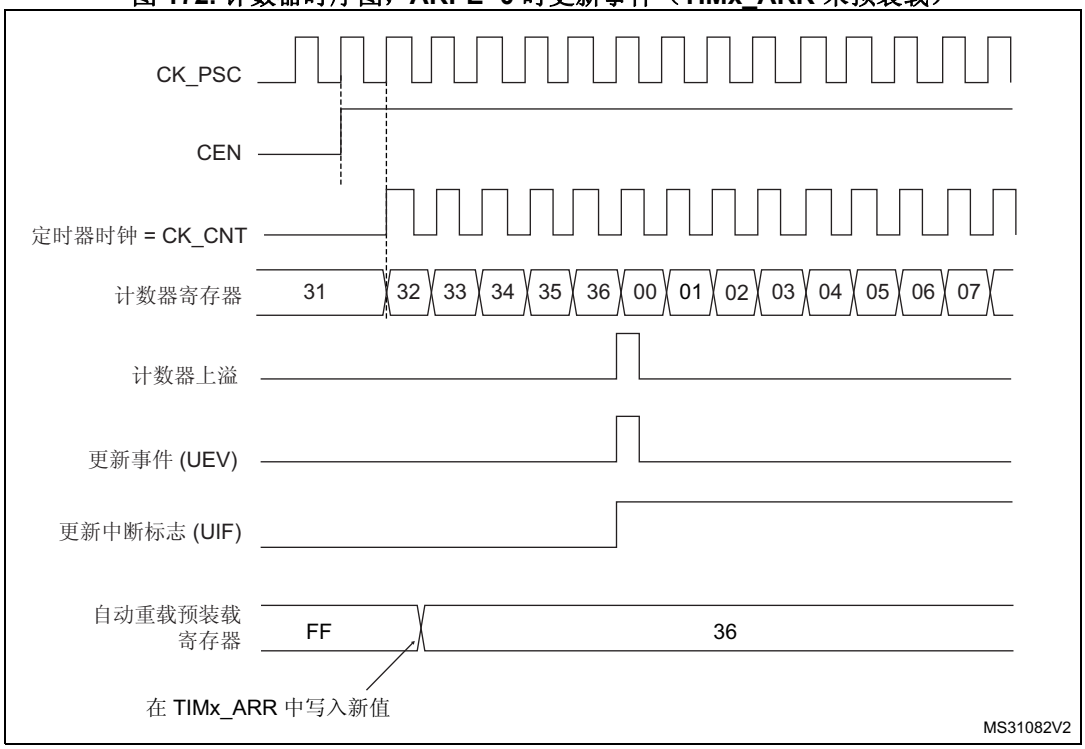
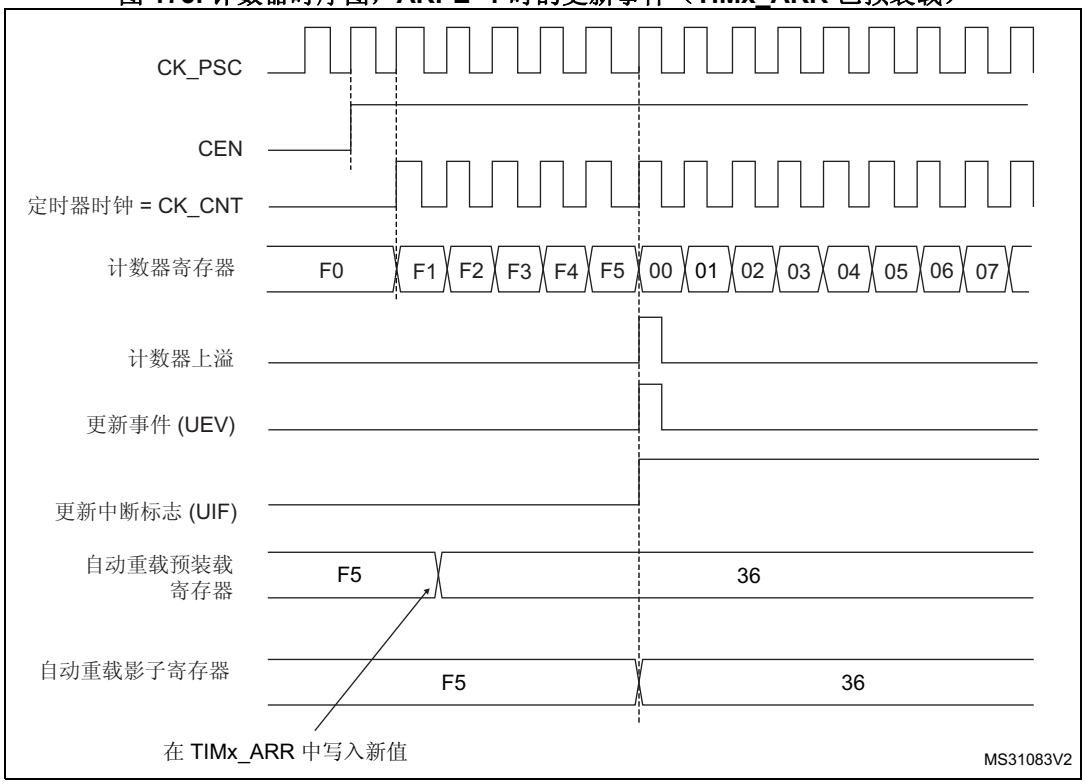


图 173. 计数器时序图，ARPE=1 时的更新事件（TIMx_ARR 已预装载）



17.3.3 时钟选择

计数器时钟可由下列时钟源提供：

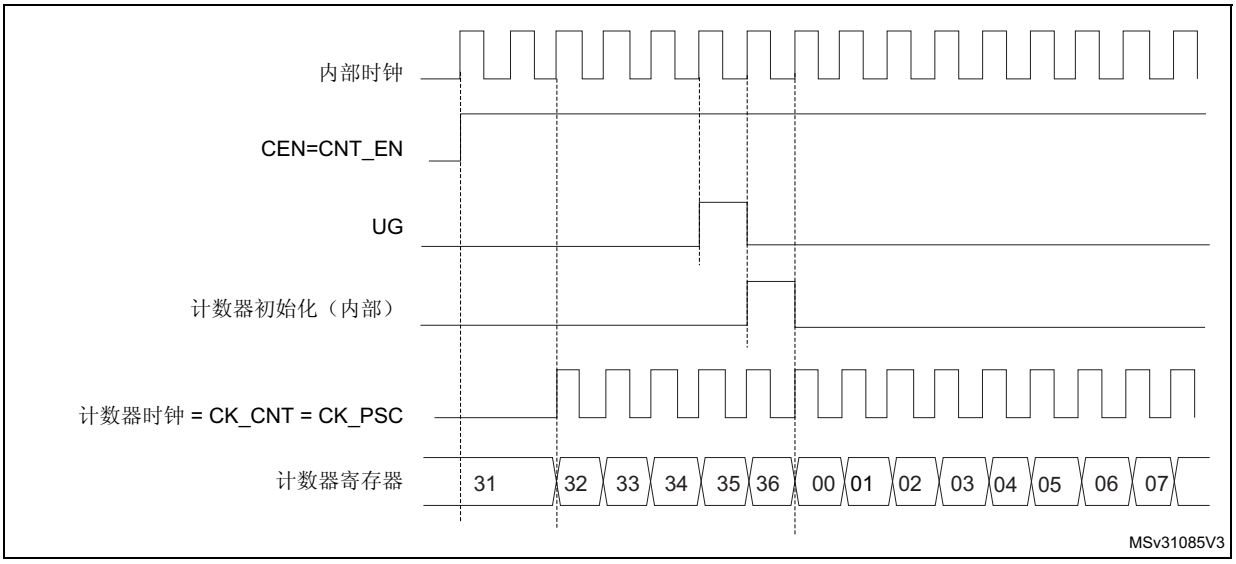
- 内部时钟 (CK_INT)

内部时钟源 (CK_INT)

内部时钟源是 TIM14 的默认时钟源。

图 174 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

图 174. 正常模式下的控制电路，1 分频内部时钟



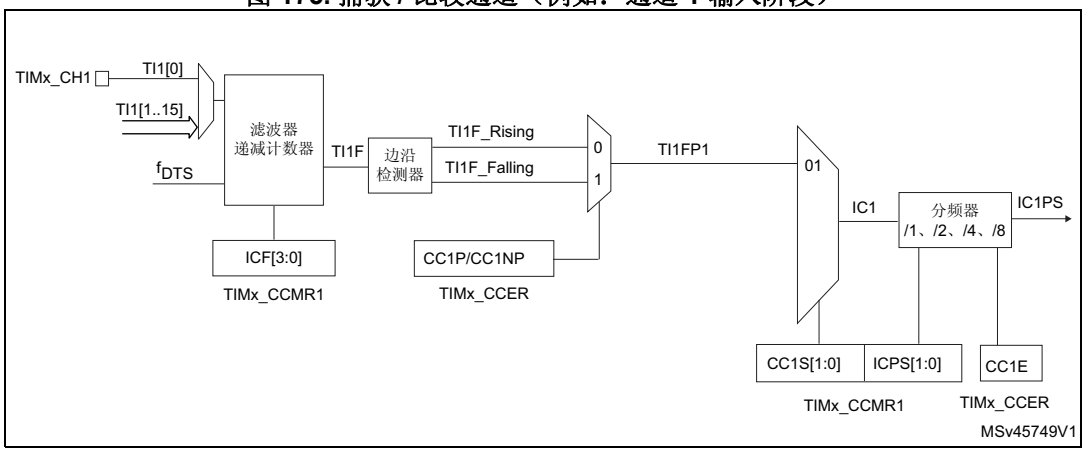
17.3.4 捕获 / 比较通道

每个捕获 / 比较通道均围绕一个捕获 / 比较寄存器（包括一个影子寄存器）、一个捕获输入单元（数字滤波、多路复用和预分频器）和一个输出单元（比较器和输出控制）构建而成。

图 175 到*图 177* 简要介绍了一个捕获 / 比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 175. 捕获 / 比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 176. 捕获 / 比较通道 1 主电路

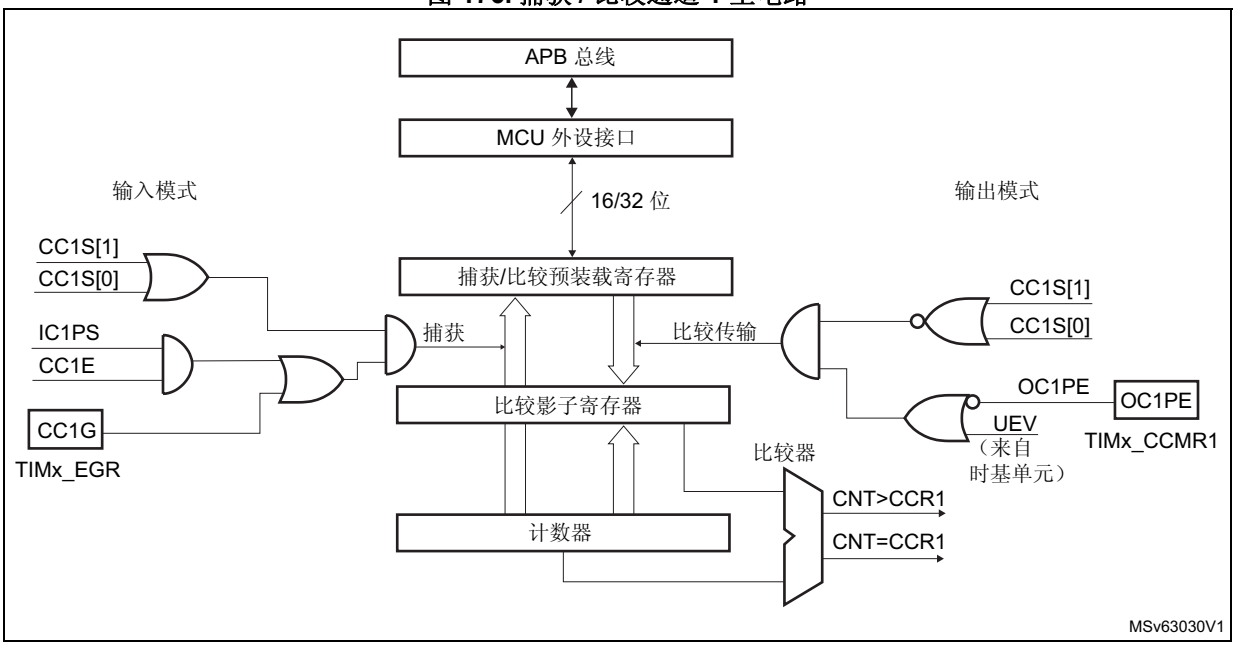
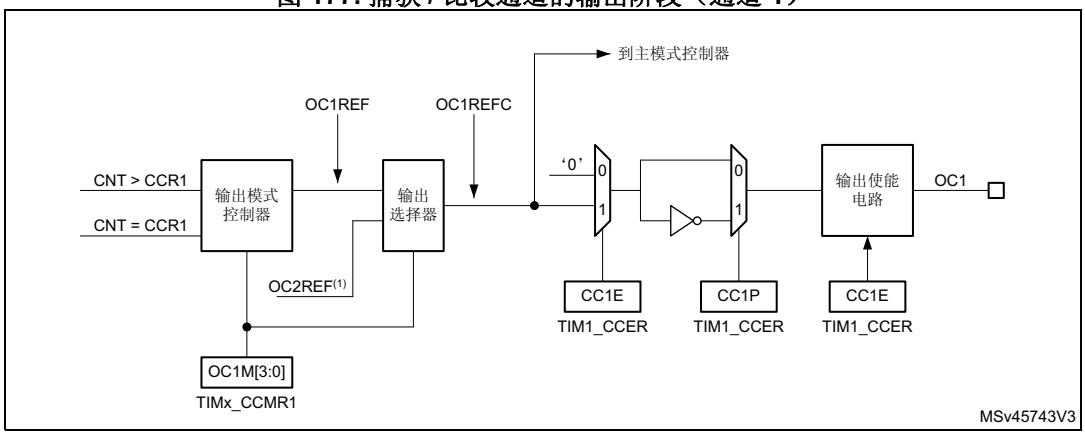


图 177. 捕获 / 比较通道的输出阶段（通道 1）



1. 只有 TIM12 提供。

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。读写访问的始终是预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容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将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

17.3.5 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获 / 比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将过捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件方法向 CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1[x] 源（内部或外部）。
2. 选择有效输入：TIMx_CCR1 必须关联到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入“01”。只要 CC1S 不等于“00”，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
3. 根据关联到定时器的信号，对相关输入滤波参数进行编程（如果输入为 TIx 输入之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设输入信号发生翻转时，信号需要 5 个内部时钟周期才能稳定，则必须设置滤波时间大于 5 个内部时钟周期。若在 TI1 上连续 8 次检测到新电平采样值（以 f_{DTS} 频率采样）判定沿跳变合法，然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入“0011”。
4. 通过在 TIMx_CCER 寄存器中将 CC1P 位和 CC1NP 位编程为“00”，选择 TI1 上的有效转换边沿（本例中为上升沿）。
5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
6. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
7. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。

要处理过捕获，建议在读取捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的过捕获信息。

注意： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断请求。*

17.3.6 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = ‘00’）下，可直接由软件将每个输出比较信号（OCxREF 和 OCx）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号 (OCxREF/OCx) 强制设置为有效电平，用户只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入“0101”。OCxREF 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=“0”（OCx 高电平有效）=> 将 OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入“0100”，可将 OCxREF 信号强制设置为低电平。

当然，即使强制输出模式下，TIMx_CCRx 影子寄存器与计数器之间的比较仍然会执行，而且允许将标志置 1。因此可相应发送中断请求。下面的输出比较模式一节对此进行了介绍。

17.3.7 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获 / 比较寄存器与计数器之间相匹配时，输出比较功能：

1. 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（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。匹配时，输出引脚既可保持其电平（OCxM=“0000”），也可设置为有效电平（OCxM=“0001”）、无效电平（OCxM=“0010”）或进行翻转（OCxM=“0011”）。
2. 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
3. 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。

使用 TIMx_CCMRx 寄存器中的 OCxPE 位，可将 TIMx_CCRx 寄存器配置为带或不带预装功能。

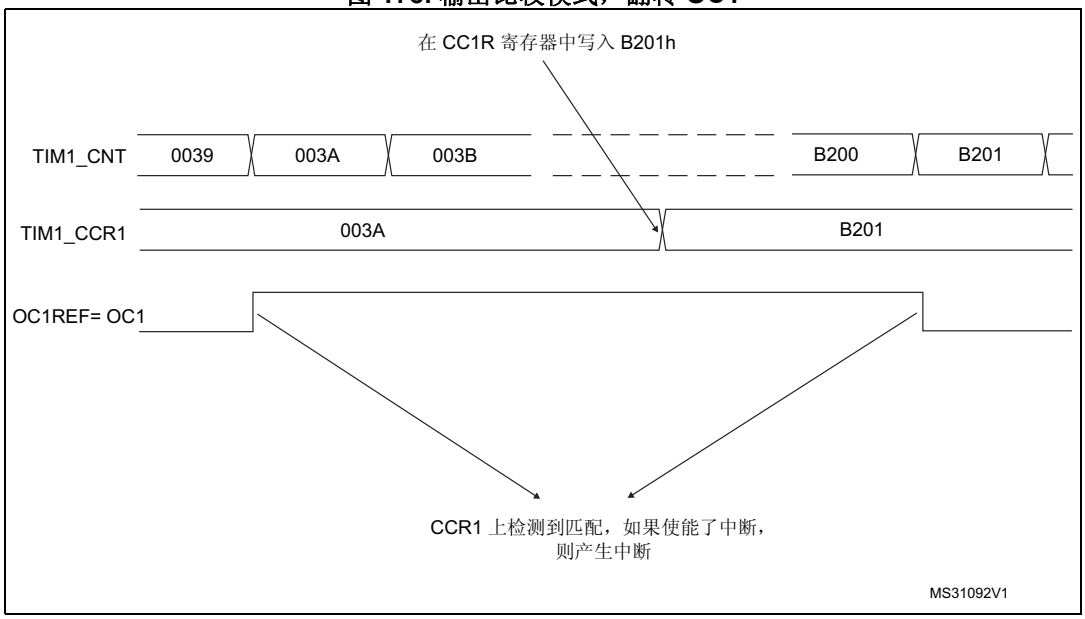
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。时间分辨率可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤：

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   - 当 CNT 与 CCRx 匹配时，写入 OCxM =“0011”以翻转 OCx 输出引脚。
   - 写入 OCxPE =“0”以禁止预装功能
   - 写入 CCxP =“0”以选择高电平有效极性
   - 写入 CCxE =“1”以使能输出
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装功能（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIMx_CCRx 影子寄存器）。*图 178* 给出了一个示例。

图 178. 输出比较模式，翻转 OC1



17.3.8 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIMx_ARR 寄存器值决定，其占空比则由 TIMx_CCRx 寄存器值决定。

各通道可以独立选择 PWM 模式（每个 OCx 输出对应一个 PWM），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“0110”（PWM 模式 1）或“0111”（PWM 模式 2）。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

由于只有在发生更新事件时预装载寄存器内容才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

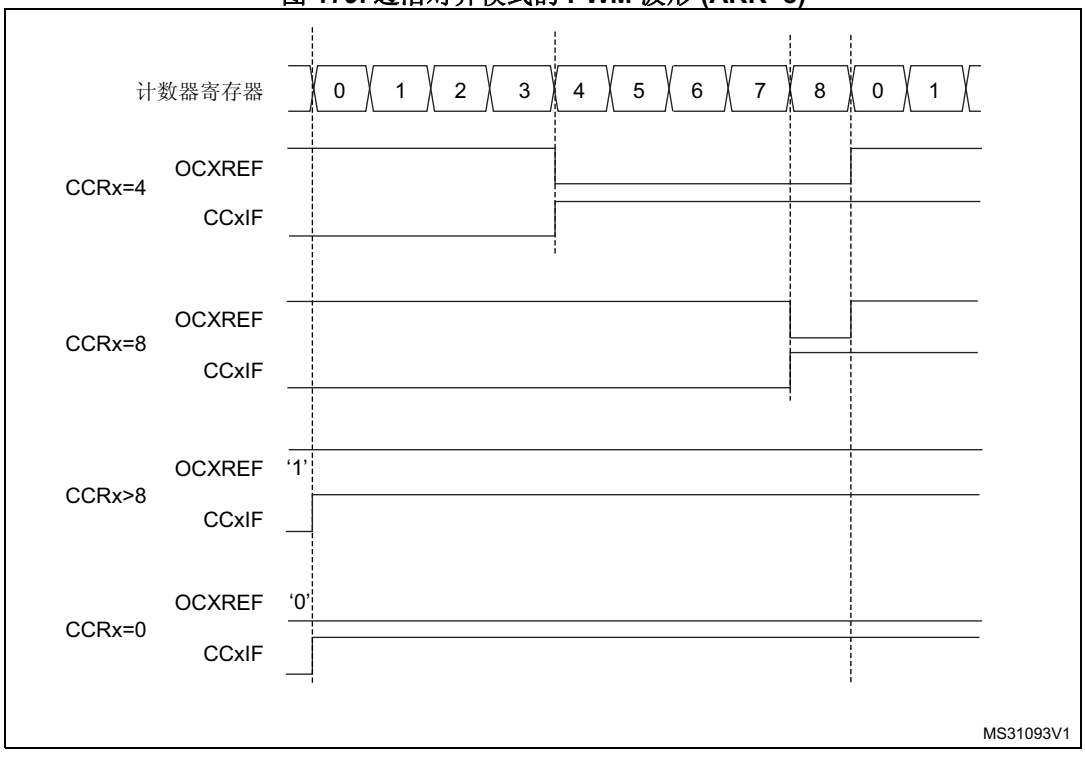
OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。OCx 输出通过将 TIMx_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息，请参见 TIMx_CCERx 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 始终与 TIMx_CCRx 进行比较，以确定 TIMx_CNT ≤ TIMx_CCRx 是否成立。

只有计数器采用递增方式计数后，定时器才能够在边沿对齐模式下生成 PWM。

以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。*图 179* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 179. 边沿对齐模式的 PWM 波形 (ARR=8)



17.3.9 单脉冲模式

单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以使用 CEN 位来控制启动计数器。可以在输出比较模式或 PWM 模式下生成波形。通过将 TIMx_CR1 寄存器中的 OPM 位置 1 选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

CNT < CCRx = ARR（特别注意，0 < CCRx）

17.3.10 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的更新事件情况。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

UIF 和 UIFCPY 标志之间的置位没有延迟。

17.3.11 使用定时器输出作为其他定时器的触发 (TIM14)

单通道定时器没有主模式。但是，可以使用 OC1 输出信号来触发其他定时器（包括本文档其他部分所述的定时器）。通过查看关于器件上任何 TIMx_SMCR 寄存器的“TIMx 内部触发连接”表，可以确定哪些定时器可以作为目标从定时器。

OC1 信号的脉冲宽度必须至少设为目标定时器时钟周期的 2 倍，以便确保从定时器会检测到触发。

例如，如果目标定时器 CK_INT 时钟频率是源定时器的 1/4，则 OC1 的脉冲宽度必须至少是 8 个源定时器时钟周期。

17.3.12 调试模式

当微控制器进入调试模式（Cortex®-M0+ 内核停止）时，TIMx 计数器会根据 DBG 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。更多详细信息，请参见*第 26.9.2 节：对定时器、看门狗和 I2C 的调试支持*。

17.4 TIM14 寄存器

外设寄存器的写访问仅支持半字（16 位）或字（32 位）。而读访问可支持字节（8 位）、半字（16 位）或字（32 位）。

17.4.1 TIM14 控制寄存器 1 (TIM14_CR1)

TIM14 control register 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
Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
				rw		rw	rw	rw				rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)

0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。

1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)

该位域指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟 (TIx) 之间的分频比。

00：$t_{DTS} = t_{CK\_INT}$

01：$t_{DTS} = 2 \times t_{CK\_INT}$

10：$t_{DTS} = 4 \times t_{CK\_INT}$

11：保留

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)

0：TIMx_ARR 寄存器不进行缓冲

1：TIMx_ARR 寄存器进行缓冲

位 6:4 保留，必须保持复位值。

位 3 **OPM**：单脉冲模式

0：计数器在发生更新事件时不会停止计数

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）。

位 2 **URS**：更新请求源 (Update request source)

该位由软件置 1 和清零，用以选择更新中断 (UEV) 源。

0：使能后，所有以下事件都会生成 UEV：

– 计数器上溢

– 将 UG 位置 1

1：使能后，只有计数器上溢会生成 UEV。

位 1 **UDIS**：更新禁止 (Update disable)

该位由软件置 1 和清零，用以使能 / 禁止更新中断 (UEV) 事件生成。

0：使能 UEV。UEV 可通过以下事件之一生成：

– 计数器上溢

– 将 UG 位置 1。

带缓冲的寄存器被加载为预加载数值。

1：禁止 UEV。不会生成 UEV，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。如果 UG 位置 1，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)
0：禁止计数器
1：使能计数器
注意：只有事先通过软件将 CEN 位置 1，才可以使用外部时钟和门控
模式。而触发模式可通过硬件自动将 CEN 位置 1。

17.4.2 TIM14 中断使能寄存器 (TIM14_DIER)

TIM14 Interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IE	UIE
														rw	rw

位 15:2 保留，必须保持复位值。

位 1 **CC1IE**：捕获 / 比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断
1：使能 CC1 中断

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断
1：使能更新中断

17.4.3 TIM14 状态寄存器 (TIM14_SR)

TIM14 status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IF	UIF
						rc_w0								rc_w0	rc_w0

位 15:10 保留，必须保持复位值。

位 9 **CC1OF**：捕获 / 比较 1 过捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。
0：未检测到过捕获。
1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8:2 保留，必须保持复位值。

位 1 **CC1IF**：捕获 / 比较 1 中断标志 (Capture/compare 1 interrupt flag)

该标志由硬件置 1，但需要通过软件清零（输入捕获或输出比较模式）或通过读取 TIMx_CCR1 寄存器清零（仅限输入捕获模式）。

0：未发生比较匹配 / 输入捕获

1：发生了比较匹配 / 输入捕获

如果通道 CC1 配置为输出：当计数器 TIMx_CNT 的值与 TIMx_CCR1 的值匹配时，此标志置 1。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。在中心对齐模式下，可通过 3 种方式将此标志置 1，完整说明请参见 TIMx_CR1 寄存器中的 CMS 位。

如果通道 CC1 配置为输入：当 TIMx_CCR1 寄存器中捕获到计数器值时（如果通过设置 TIMx_CCER 中的 CC1P 和 CC1NP 位定义为边沿有效，则改为在 IC1 上检测到边沿时），该位置 1。

位 0 **UIF**：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

– 上溢且当 TIMx_CR1 寄存器中 UDIS =“0”时。

– 当由于 TIMx_CR1 寄存器中 URS =“0”且 UDIS =“0”而通过软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

17.4.4 TIM14 事件生成寄存器 (TIM14_EGR)

TIM14 event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1G	UG
														w	w

位 15:2 保留，必须保持复位值。

位 1 **CC1G**：捕获 / 比较 1 生成 (Capture/compare 1 generation)

该位由软件置 1 以生成事件，并由硬件自动清零。

0：无操作

1：通道 1 上生成捕获 / 比较事件：

如果通道 CC1 配置为输出：

使能后，CC1IF 标志置 1 并发送相应的中断。

如果通道 CC1 配置为输入：

TIMx_CCR1 寄存器中将捕获到计数器当前值。使能后，CC1IF 标志置 1 并发送相应中断。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)

该位可通过软件置 1，并由硬件自动清零。

0：无操作

1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。计数器清零。

17.4.5 TIM14 捕获 / 比较模式寄存器 1 (TIM14_CCMR1)

TIM14 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)

该位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC[1:0]**：输入捕获 1 预分频器 (Input capture 1 prescaler)

该位域定义 CC1 输入 (IC1) 的预分频比。
只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。
00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S[1:0]**：捕获 / 比较 1 选择 (Capture/Compare 1 selection)

该位域定义通道方向（输入 / 输出）以及所使用的输入。
00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：保留
11：保留

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

17.4.6 TIM14 捕获 / 比较模式寄存器 1 [复用] (TIM14_CCMR1)

TIM14 capture/compare mode register 1 [alternate]

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
									rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 15:7 保留，必须保持复位值。

位 16，6:4 **OC1M[3:0]**：输出比较 1 模式 (Output compare 1 mode)（请参见 OC1M[3] 的位 16）

这些位定义提供 OC1 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 的有效电平则取决于 CC1P 位。

0000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。当定时器用作软件时基时，可以使用该模式。在定时器操作期间使能冻结模式时，输出保持进入冻结状态前的状态（有效或无效）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获 / 比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获 / 比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转 ——TIMx_CNT = TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平 ——OC1REF 强制变为低电平。

0101：强制变为有效电平 ——OC1REF 强制变为高电平。

0110：PWM 模式 1—— 只要 TIMx_CNT < TIMx_CCR1，通道 1 便输出有效电平，否则输出无效电平。

0111：PWM 模式 2—— 只要 TIMx_CNT < TIMx_CCR1，通道 1 便输出无效电平，否则输出有效电平。

其他值：保留

注意：在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。

注意：OC1M[3] 位不是连续的，而是在位 16 中。

位 3 **OC1PE**：输出比较 1 预装载使能 (Output compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读 / 写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到影子寄存器中。

位 2 **OC1FE**：输出比较 1 快速使能 (Output compare 1 fast enable)

该位可降低触发事件与定时器输出跳变之间的延迟。单脉冲模式下必须使用（TIMx_CR1 寄存器中的 OPM 位置 1），以便在启动触发后快速发送输出脉冲。

0： 即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1： 触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OC1FE 才会起作用。

位 1:0 **CC1S[1:0]**：捕获 / 比较 1 选择 (Capture/Compare 1 selection)

该位域定义通道方向（输入 / 输出）以及所使用的输入。

00：CC1 通道配置为输出。

01：CC1 通道配置为输入，IC1 映射到 TI1 上。

10：保留。

11：保留。

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

17.4.7 TIM14 捕获 / 比较使能寄存器 (TIM14_CCER)

TIM14 capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	Res.	CC1P	CC1E
												rw		rw	rw

位 15:4 保留，必须保持复位值。

位 3 **CC1NP**：捕获 / 比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)。

CC1 通道配置为输出：CC1NP 必须保持清零。

CC1 通道配置为输入：CC1NP 位与 CC1P 配合使用可定义 TI1FP1 的极性（请参见 CC1P 说明）。

位 2 保留，必须保持复位值。

位 1 **CC1P**：捕获 / 比较 1 输出极性 (Capture/Compare 1 output polarity)。

0：OC1 高电平有效（输出模式）/ 边沿有效性选择（输入模式，见下文）

1：OC1 低电平有效（输出模式）/ 边沿有效性选择（输入模式，见下文）

CC1 通道配置为输入时，CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。

CC1NP=0，CC1P=0： 非反相 / 上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=0，CC1P=1： 反相 / 下降沿触发。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=1，CC1P=1： 非反相 / 上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。

CC1NP=1，CC1P=0：该配置保留，不得使用。

位 0 **CC1E**：捕获 / 比较 1 输出使能 (Capture/Compare 1 output enable)。

0：禁止捕获模式 /OC1 无效

1：使能捕获模式 / 在相应输出引脚上输出 OC1 信号

表 72. 标准 OCx 通道的输出控制位

CCxE 位	OCx 输出状态
0	禁止输出（不由定时器驱动：高阻态
1	使能输出（tim_ocx = tim_ocxref + 极性）

注意： *与标准 OCx 通道相连的外部 I/O 引脚的状态取决于通道 OCx 的状态以及 GPIO 寄存器。*

17.4.8 TIM14 计数器 (TIM14_CNT)

TIM14 counter

偏移地址：0x24

复位值：0x0000 0000

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

位 31 **UIFCPY**：UIF 副本 (UIF Copy)
该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

17.4.9 TIM14 预分频器 (TIM14_PSC)

TIM14 prescaler

偏移地址：0x28

复位值：0x0000

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

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)
计数器时钟频率 CK_CNT 等于 f_{CK_PSC} / (PSC[15:0] + 1)。
PSC 包含在每次发生更新事件时要装载到有效预分频器寄存器的值。
（包括当计数器通过 TIMx_EGR 寄存器的 UG 位或在“复位模式”下通过触发控制器清零时）。

17.4.10 TIM14 自动重载寄存器 (TIM14_ARR)

TIM14 auto-reload register

偏移地址：0x2C

复位值：0xFFFF

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

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的详细信息，请参见*第 17.3.1 节：第 430 页的时基单元*。

当自动重载值为空时，计数器不工作。

17.4.11 TIM14 捕获 / 比较寄存器 1 (TIM14_CCR1)

TIM14 capture/compare register 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[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR1[15:0]**：捕获 / 比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：

CCR1 是捕获 / 比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际的捕获 / 比较寄存器 1）。

有效捕获 / 比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出相应电平的值。

如果通道 CC1 配置为输入：

CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。

17.4.12 TIM14 定时器输入选择寄存器 (TIM14_TISEL)

TIM14 timer input selection register

偏移地址：0x68

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
												rw	rw	rw	rw

位 15:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)

0000：TIM14_CH1 输入
0001：RTC CLK
0010：HSE/32
0011：MCO
0100：MCO2
其他值：保留

17.4.13 TIM14 寄存器映射

TIMx 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 73. TIM14 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0				0	0	0	0
0x04 to 0x08	Reserved	Res.																															
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IE	UIE
	Reset value																															0	0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IF	UIF
	Reset value																							0								0	0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1G	UG
	Reset value																															0	0
0x18	**TIMx_CCMR1** Output compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																0										0	0	0	0	0	0	0
	TIMx_CCMR1 Input capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0x1C	Reserved	Res.																															
0x20	**TIMx_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	Res.	CC1P	CC1E
	Reset value																													0		0	0

表 73. TIM14 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x24	**TIMx_CNT**	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIMx_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIMx_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	Reserved	Res.																															
0x34	**TIMx_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38 to 0x64	Reserved	Res.																															
0x68	**TIM14_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

18 通用定时器 (TIM16/TIM17)

18.1 TIM16/TIM17 简介

TIM16/TIM17 定时器包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

此类定时器用途广泛，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输出比较、PWM 和带死区插入的互补 PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

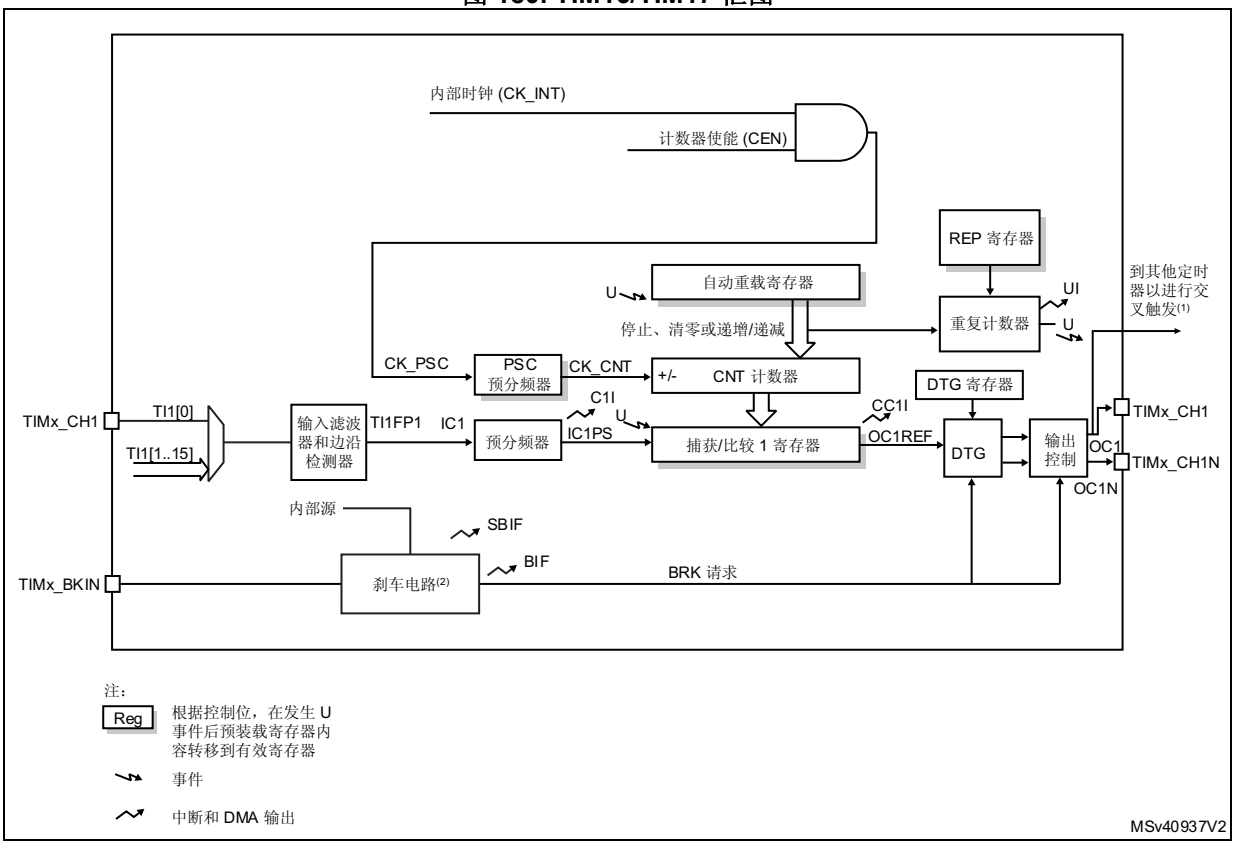
TIM16/TIM17 定时器彼此完全独立，不共享任何资源。

18.2 TIM16/TIM17 主要特性

TIM16/TIM17 定时器包括下列特性：

- 16 位自动重载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65535 之间
- 一个具有以下用途的通道：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿对齐模式）
  - 单脉冲模式输出
- 带可编程死区的互补输出
- 重复计数器，用于仅在给定数目的计数周期后更新定时器寄存器。
- 用于将定时器的输出信号置于复位状态或已知状态的刹车输入。
- 发生如下事件时生成中断 /DMA 请求：
  - 更新：计数器上溢
  - 输入捕获
  - 输出比较
  - 刹车输入

图 180. TIM16/TIM17 框图



1. 该信号可用于触发一些从定时器，请参见*第 18.3.18 节：使用定时器输出作为其他定时器的触发 (TIM16/TIM17)*。
2. 内部刹车事件源可以是：
   - CSS 生成的时钟故障事件。有关 CSS 的详细信息，请参见*第 5.2.6 节：时钟安全系统 (CSS)*
   - SRAM 奇偶校验错误信号
   - Cortex®-M0+LOCKUP (Hardfault) 输出

18.3 TIM16/TIM17 功能描述

18.3.1 时基单元

可编程高级控制定时器的主要模块是一个 16 位递增计数器及其相关的自动重载寄存器。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)
- 重复计数器寄存器 (TIMx_RCR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以随时直接写入影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将产生更新事件。该更新事件也可由软件产生。下文将详细介绍每个配置中更新事件的产生方式。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意，计数器将在 TIMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器说明

预分频器可对计数器时钟进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 TIMx_PSC 寄存器中的 16 位寄存器所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 181 和*图 182* 以一些示例说明在预分频比实时变化时计数器的行为：

图 181. 预分频器分频由 1 变为 2 时的计数器时序图

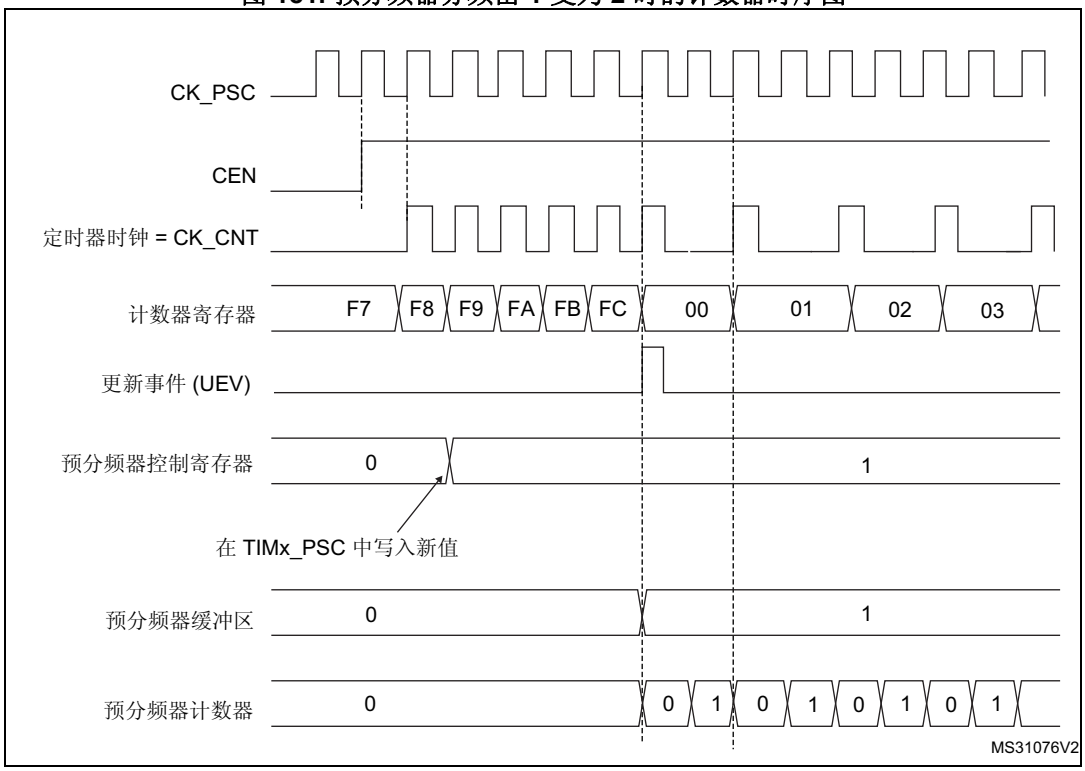
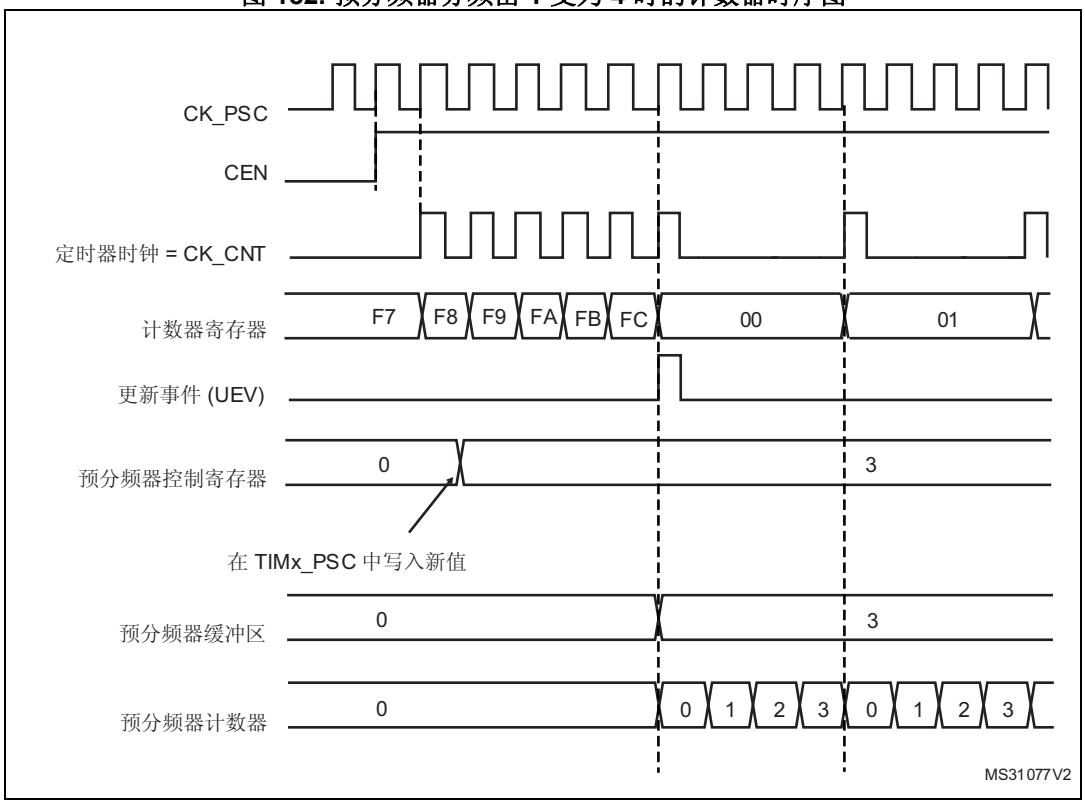


图 182. 预分频器分频由 1 变为 4 时的计数器时序图



18.3.2 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中编程的次数 (TIMx_RCR) 后，将生成更新事件 (UEV)。否则，将在每次计数器上溢时产生更新事件。

将 TIMx_EGR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。利用这一特性，在定时器的输入捕获操作时可以通过对 UG 位置 1 让计数器清零而不至于触发更新中断，也就避免了在捕获中断里触发更新中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容，
- 自动重载影子寄存器将以预装载值 (TIMx_ARR) 进行更新，
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 183. 计数器时序图，1 分频内部时钟

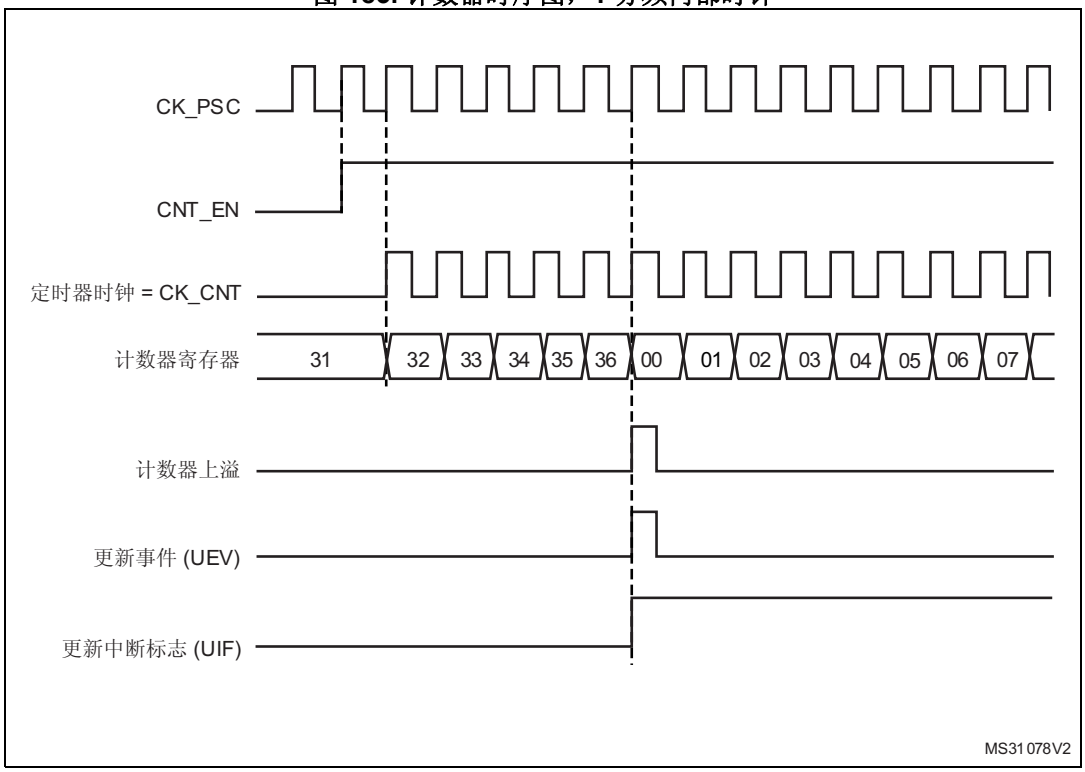


图 184. 计数器时序图，2 分频内部时钟

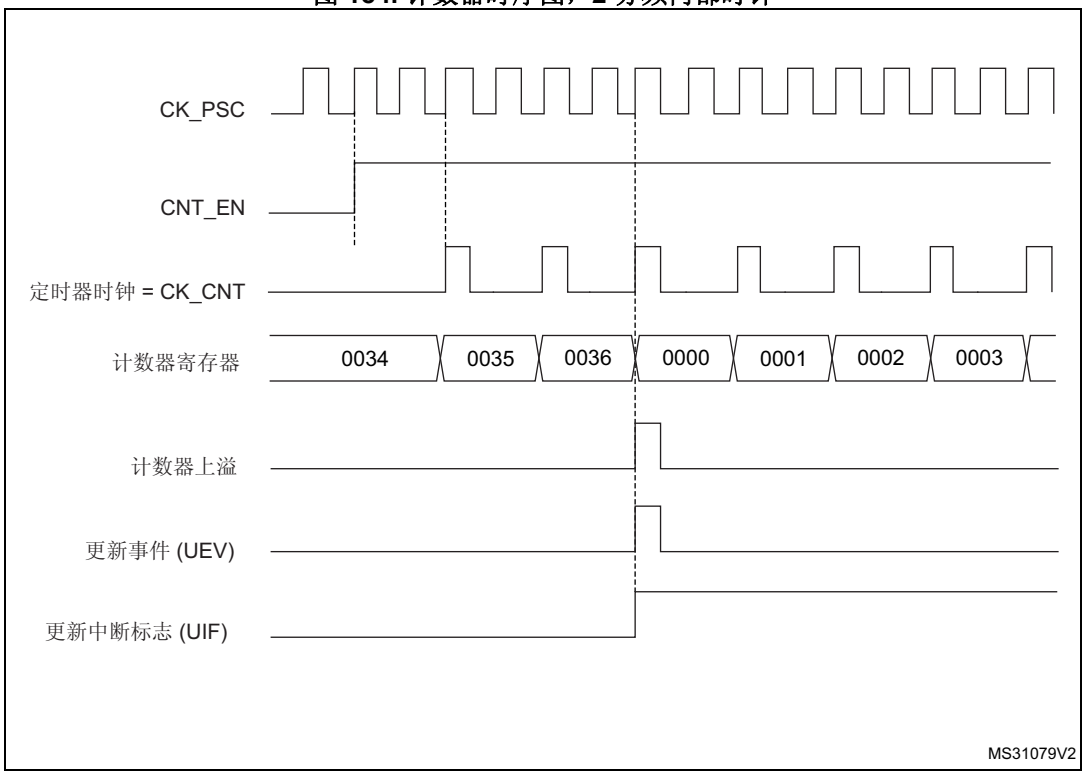


图 185. 计数器时序图，4 分频内部时钟

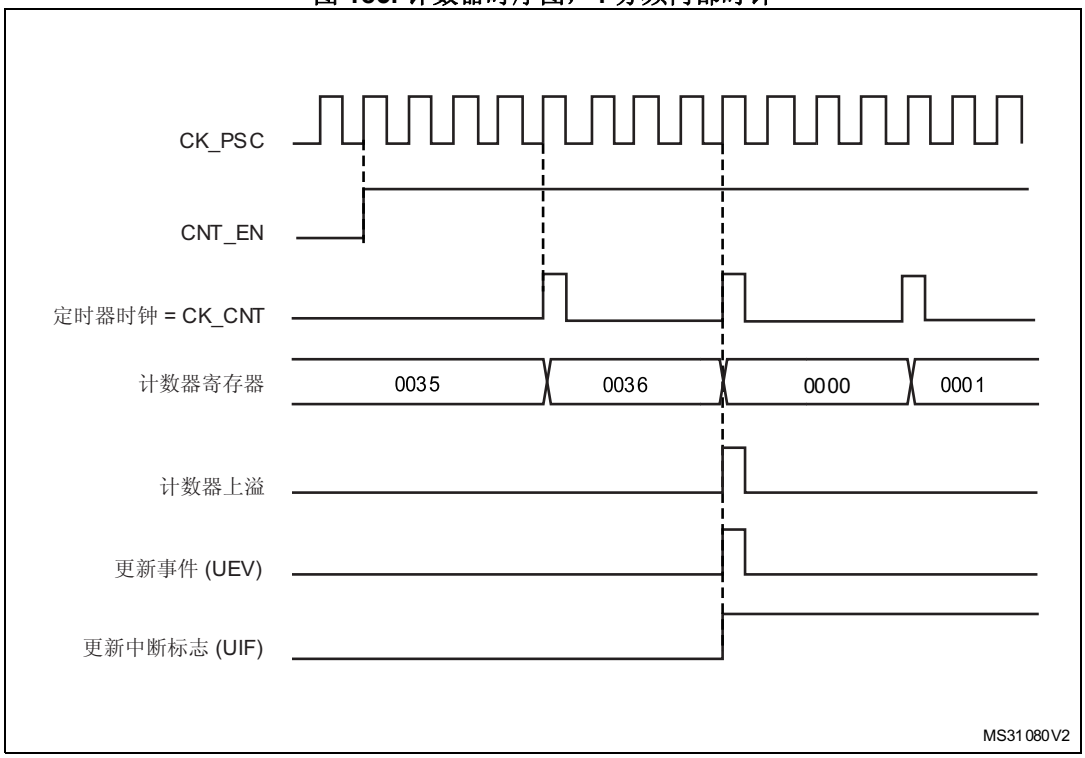


图 186. 计数器时序图，N 分频内部时钟

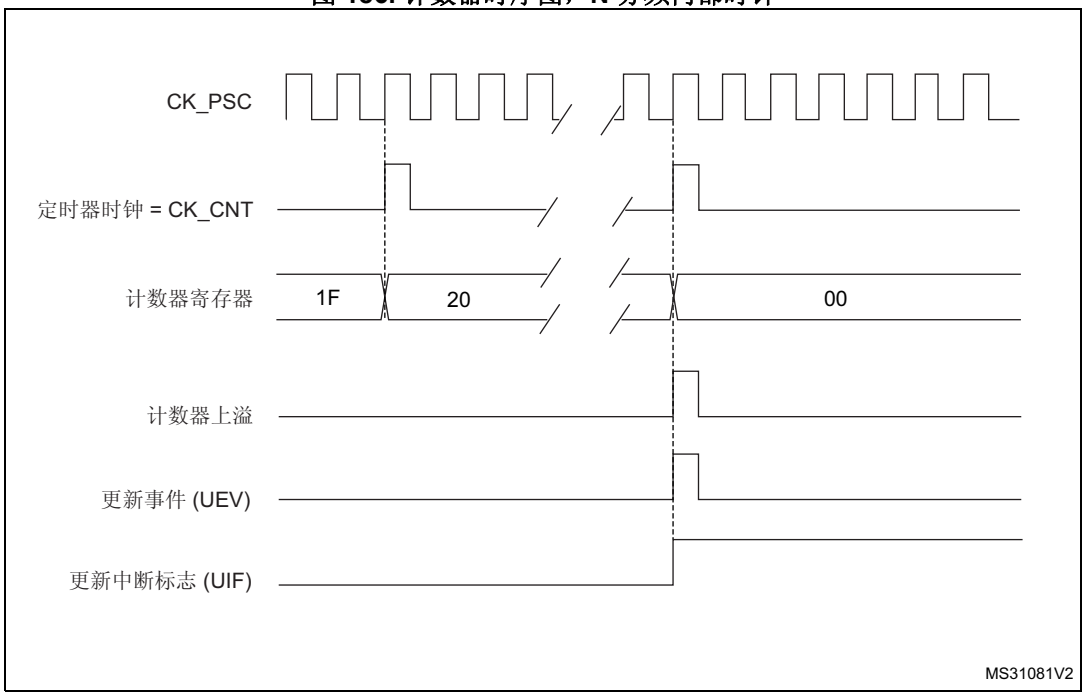


图 187. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

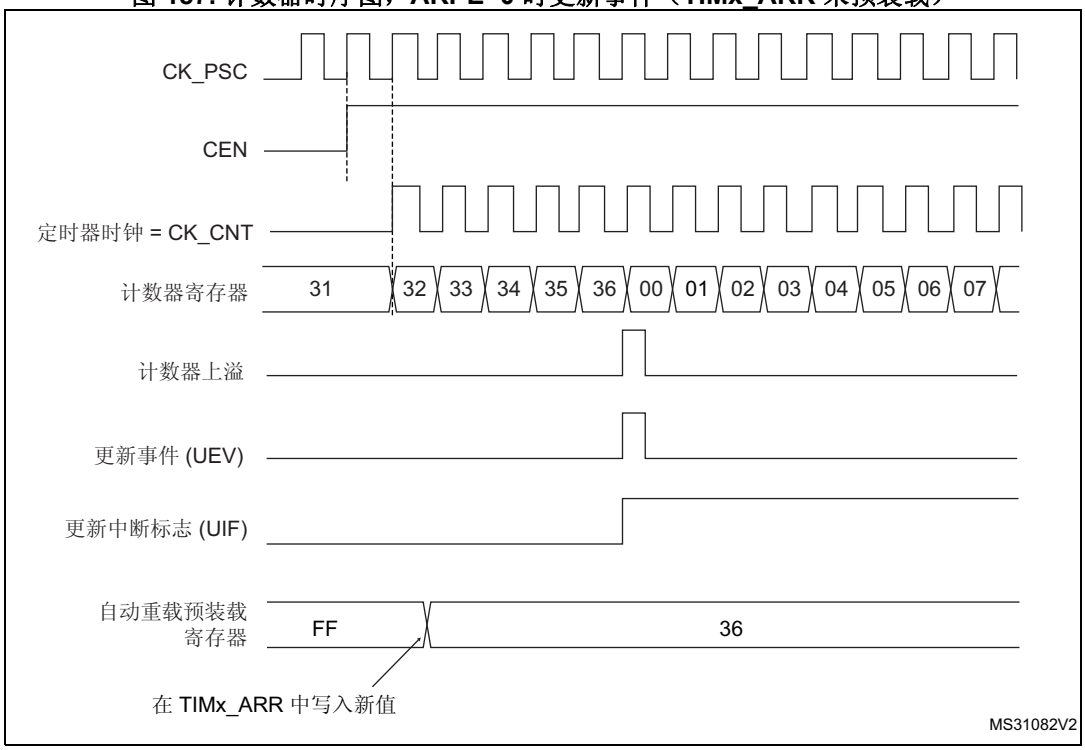
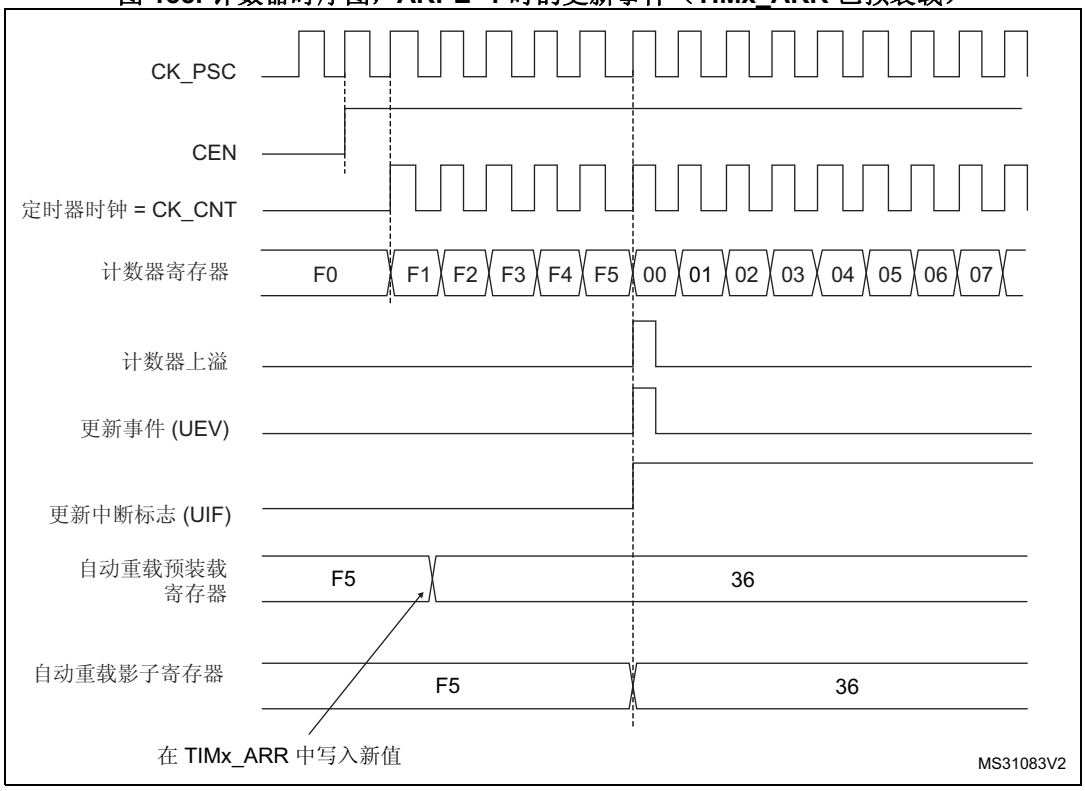


图 188. 计数器时序图，ARPE=1 时的更新事件（TIMx_ARR 已预装载）



18.3.3 重复计数器

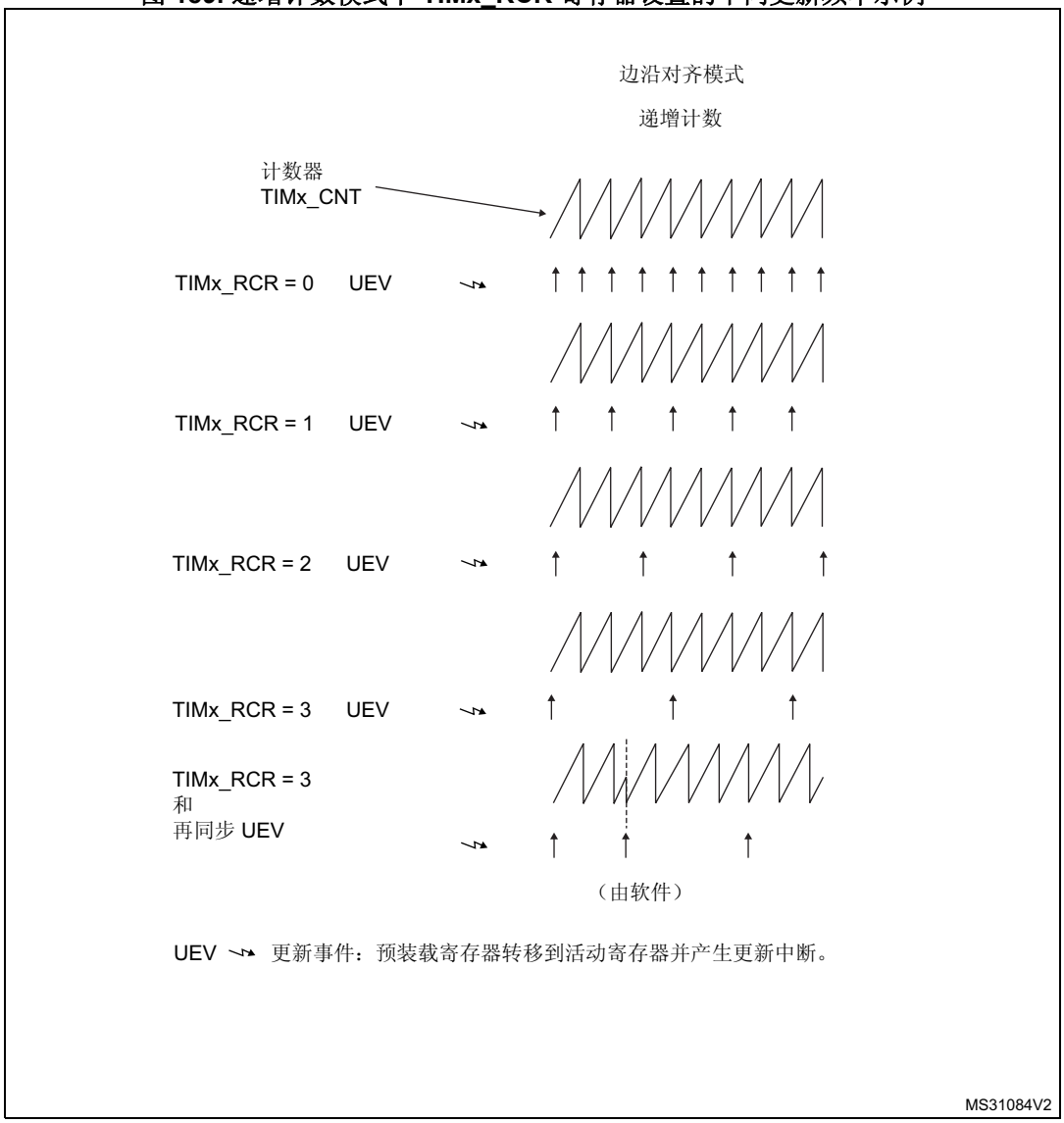
*第 18.3.1 节：时基单元*介绍如何因计数器上溢而生成更新事件 (UEV)。实际上，只有当重复计数器达到零时，才会生成更新事件。这在生成 PWM 信号时很有用。

这意味着，每当发生 N+1 次计数器上溢（其中，N 是 TIMx_RCR 重复计数器寄存器中的值），数据就将从预装载寄存器转移到影子寄存器（TIMx_ARR 自动重载寄存器、TIMx_PSC 预分频器寄存器以及比较模式下的 TIMx_CCRx 捕获 / 比较寄存器）。

重复计数器在每个计数器上溢时递减。

重复计数器是自动重载类型；其重复率为 TIMx_RCR 寄存器所定义的值（请参见*图 189*）。当更新事件由软件（通过将 TIMx_EGR 寄存器的 UG 位置 1）或硬件（通过从模式控制器）生成时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 TIMx_RCR 寄存器的内容。

图 189. 递增计数模式下 TIMx_RCR 寄存器设置的不同更新频率示例



18.3.4 时钟选择

计数器时钟可由下列时钟源提供：

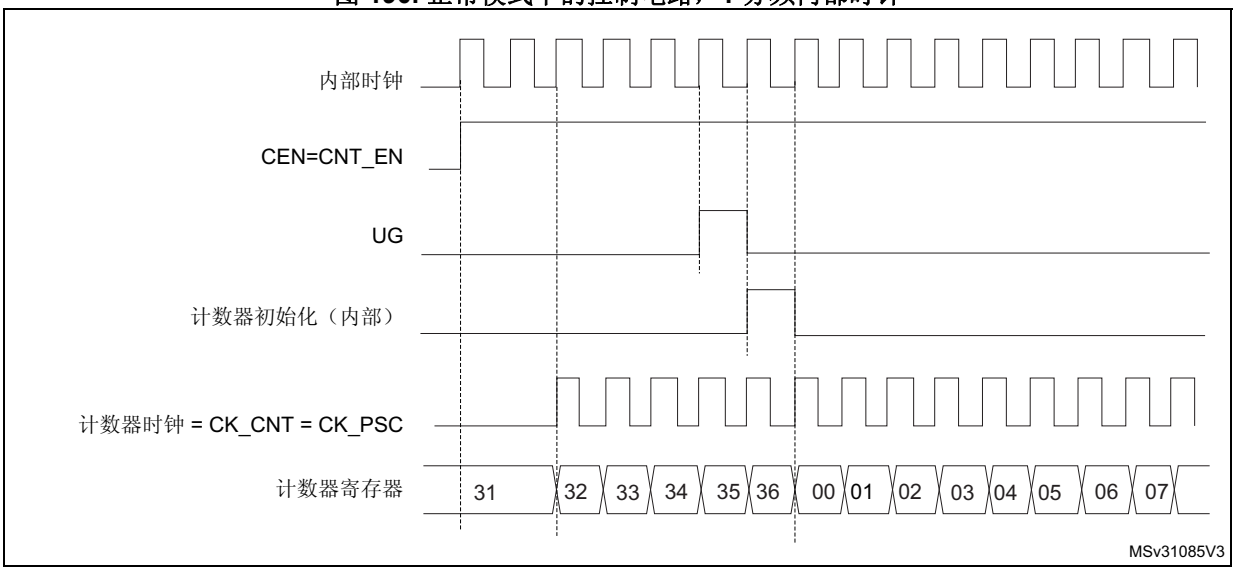
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚

内部时钟源 (CK_INT)

如果禁止从模式控制器 (SMS=000)，则 CEN 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍保持自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 190 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

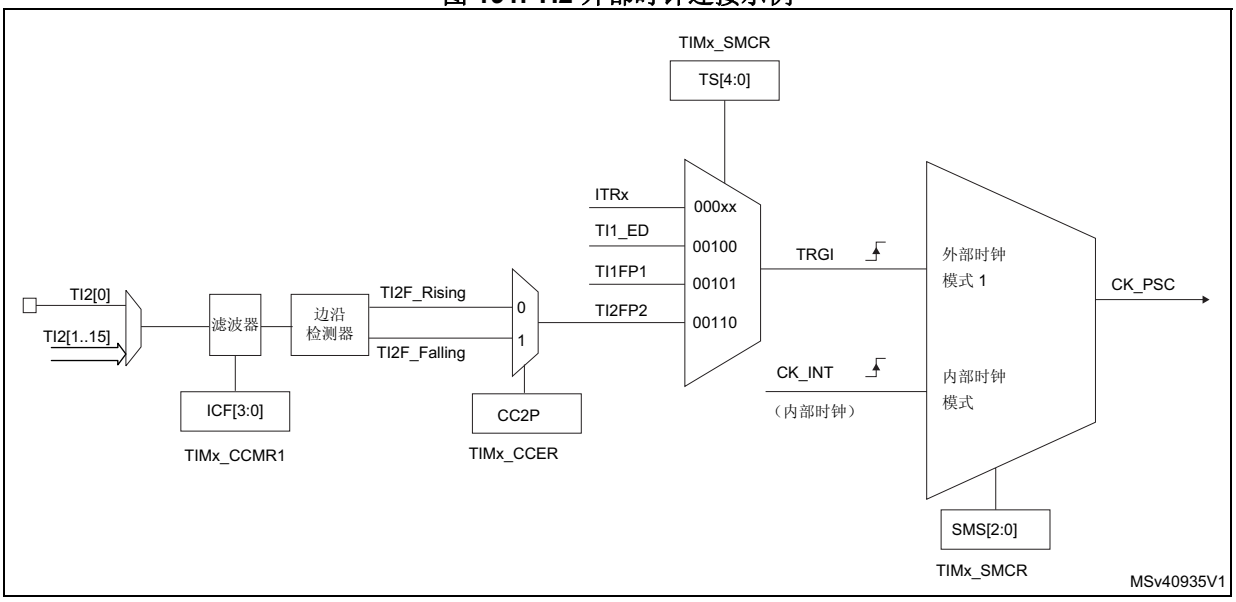
图 190. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 191. TI2 外部时钟连接示例



例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

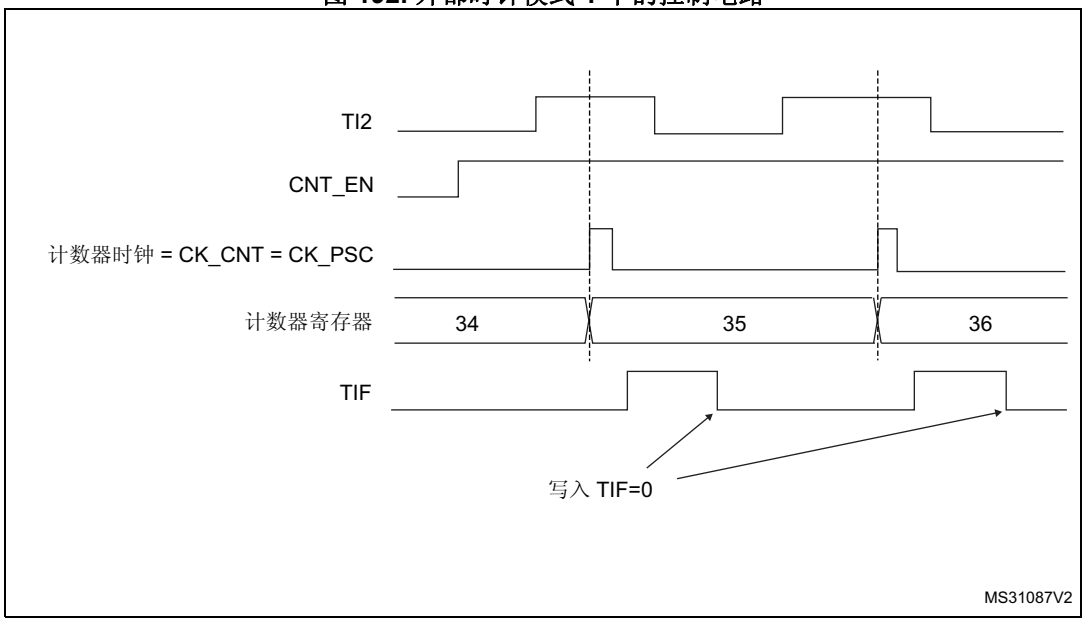
1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2[x] 源（内部或外部）。
2. 通过在 TIMx_CCMR1 寄存器中写入 CC2S = “01” 来配置通道 2，使其能够检测 TI2 输入的上升沿。
3. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。
4. 通过在 TIMx_CCER 寄存器中写入 CC2P = 0 来选择上升沿极性。
5. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
6. 通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为触发输入源。
7. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

注意： *由于捕获预分频器不用于触发操作，因此无需对其进行配置。*

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 192. 外部时钟模式 1 下的控制电路



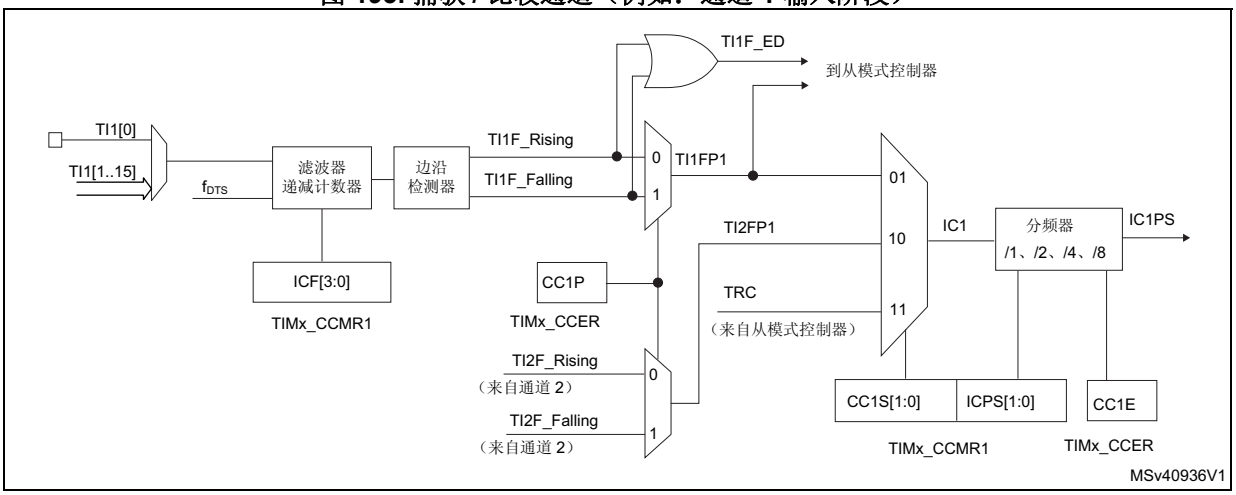
18.3.5 捕获 / 比较通道

每个捕获 / 比较通道均围绕一个捕获 / 比较寄存器（包括一个影子寄存器）、一个捕获输入单元（数字滤波、多路复用和预分频器）和一个输出单元（比较器和输出控制）构建而成。

图 193 到*图 195* 简要介绍了一个捕获 / 比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 193. 捕获 / 比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 194. 捕获 / 比较通道 1 主电路

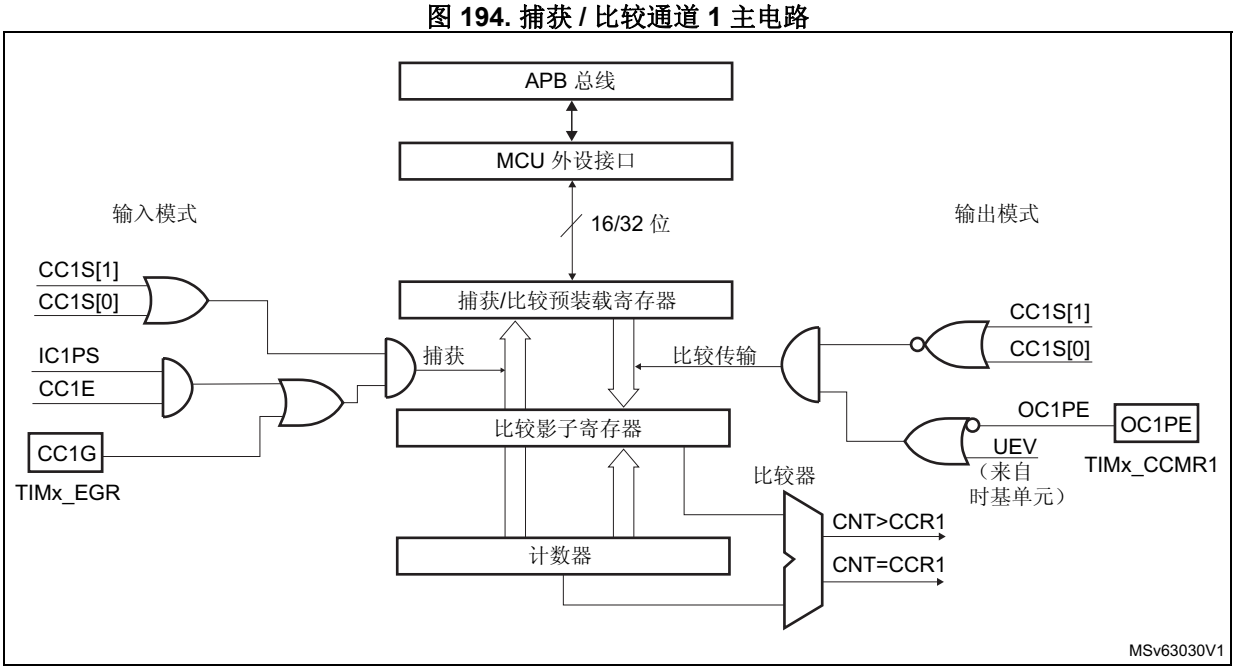
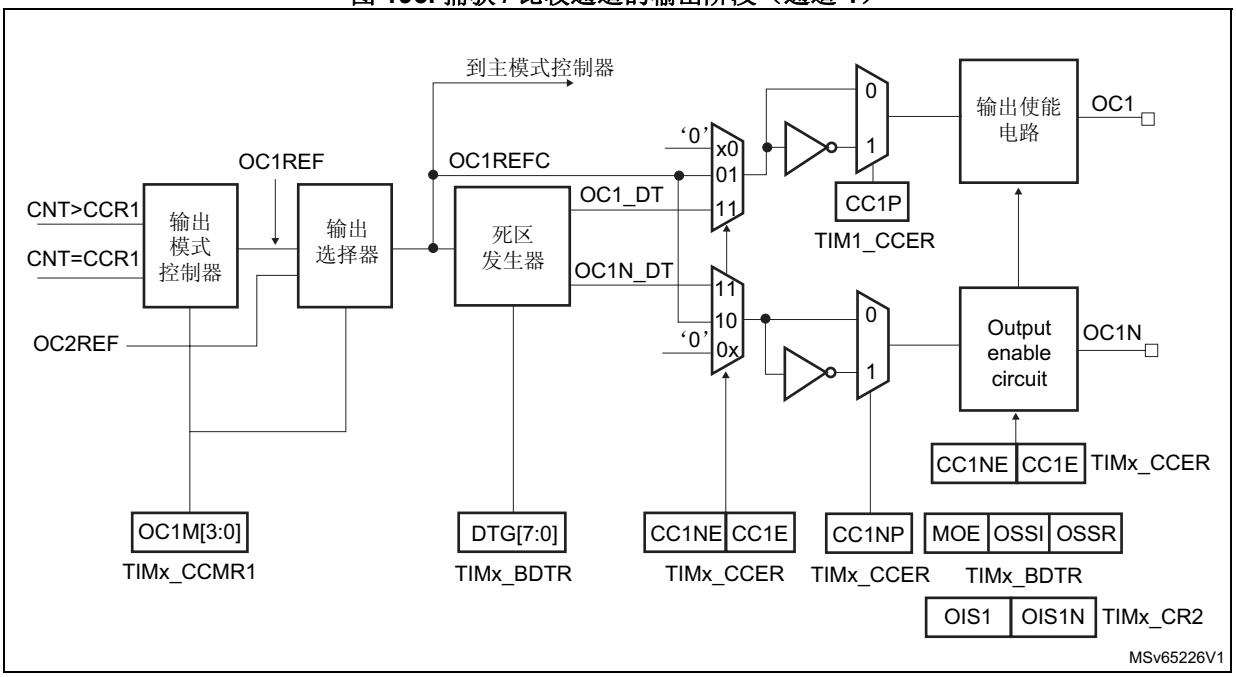
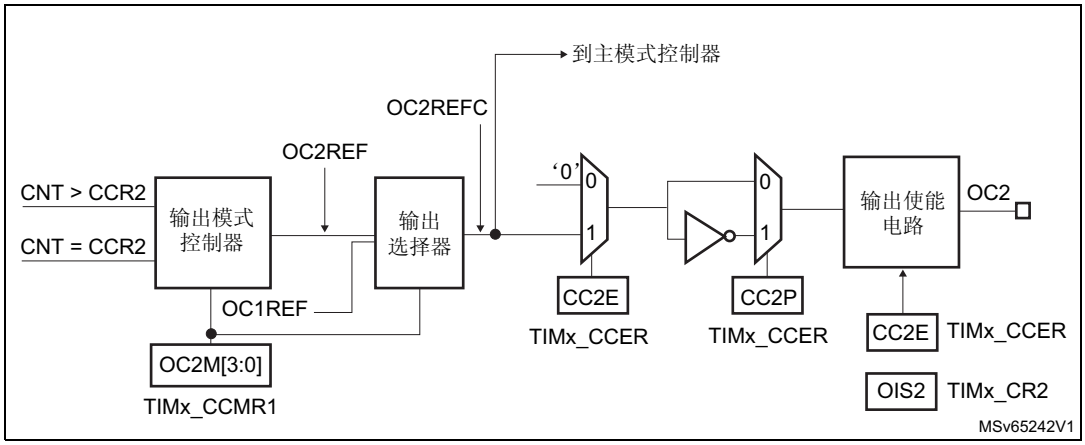


图 195. 捕获 / 比较通道的输出阶段（通道 1）





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。读写访问的始终是预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容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将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

18.3.6 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获 / 比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将过捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件方法向

CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择有效输入：TIMx_CCR1 必须关联到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
3. 根据关联到定时器的信号，对相关输入滤波参数进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设输入信号发生翻转时，信号需要 5 个内部时钟周期才能稳定，则必须设置滤波时间大于 5 个内部时钟周期。若在 TI1 上连续 8 次检测到新电平采样值（以 f_{DTS} 频率采样）判定沿跳变合法，则向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
4. 通过在 TIMx_CCER 寄存器中将 CC1P 位写入 0，选择 TI1 上的有效转换边沿（本例中为上升沿）。
5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
6. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
7. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且 / 或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理过捕获，建议在读取捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的过捕获信息。

注意： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和 / 或 DMA 请求。*

1. 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。

18.3.7 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（OCxREF 和 OCx/OCxN）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号 (OCXREF/OCx) 强制设置为有效电平，只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入 101。OCxREF 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=0（OCx 高电平有效）=> OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 100，可将 OCxREF 信号强制设置为低电平。

当然，即使强制输出模式下，TIMx_CCRx 影子寄存器与计数器之间的比较仍然会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。下面的输出比较模式一节对此进行了介绍。

18.3.8 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获 / 比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（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。匹配时，输出引脚既可保持其电平 (OCXM=000)，也可设置为有效电平 (OCXM=001)、无效电平 (OCXM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

使用 TIMx_CCMRx 寄存器中的 OCxPE 位，可将 TIMx_CCRx 寄存器配置为带或不带预装功能。

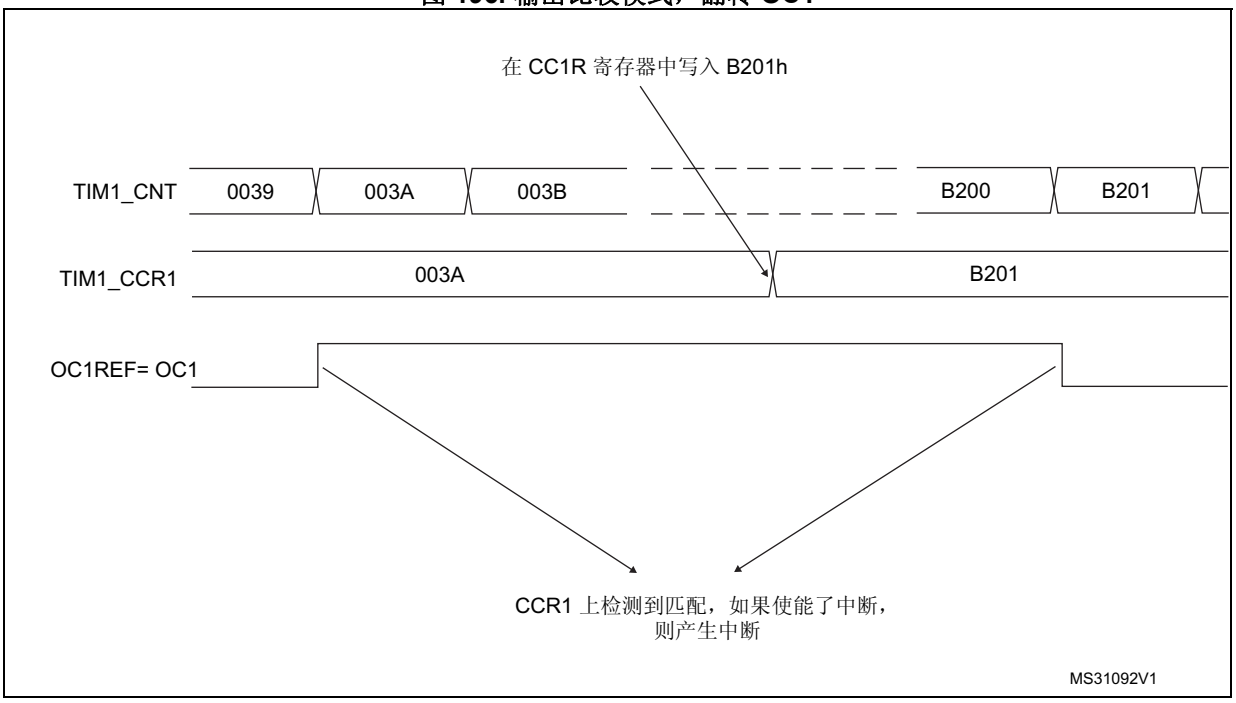
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。时间分辨率可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   - 当 CNT 与 CCRx 匹配时，写入 OCxM = 011 以翻转 OCx 输出引脚
   - 写入 OCxPE = 0 以禁止预装功能
   - 写入 CCxP = 0 以选择高电平有效极性
   - 写入 CCxE = 1 以使能输出
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装功能（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIMx_CCRx 影子寄存器）。*图 196* 给出了一个示例。

图 196. 输出比较模式，翻转 OC1



18.3.9 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIMx_ARR 寄存器值决定，其占空比则由 TIMx_CCRx 寄存器值决定。

各通道可以独立选择 PWM 模式（每个 OCx 输出对应一个 PWM），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“110”（PWM 模式 1）或“111”（PWM 模式 2）。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

由于只有在发生更新事件时预装载寄存器内容才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

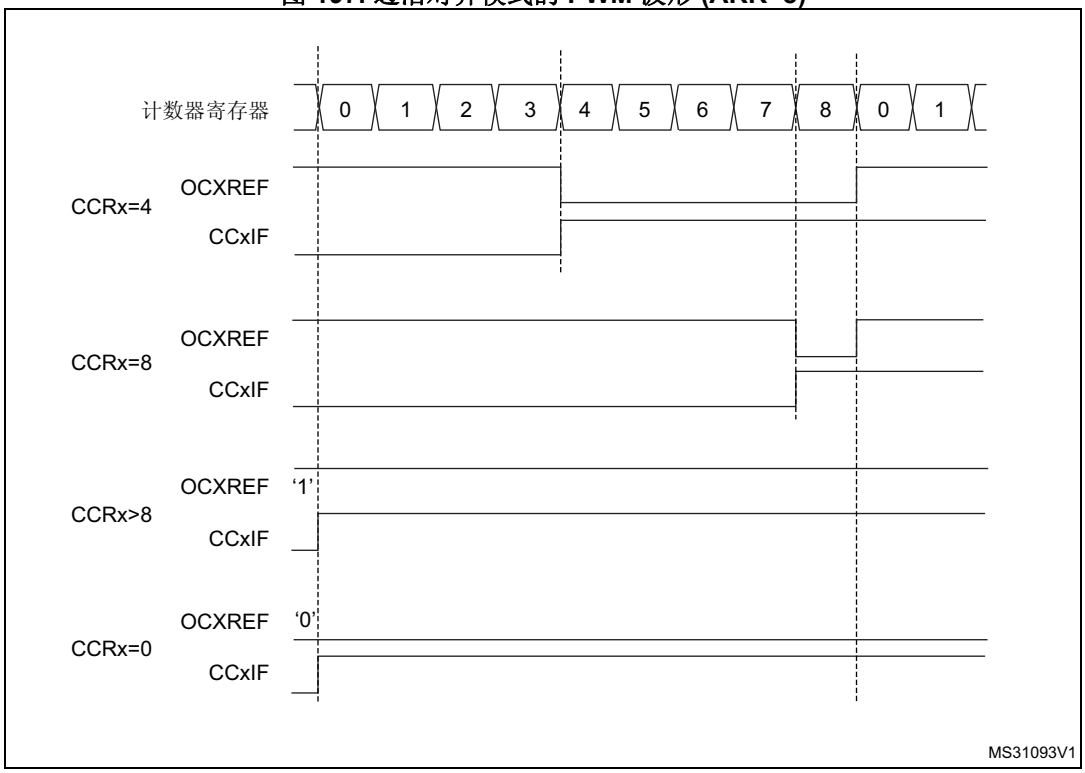
OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。通过 CCxE、CCxNE、MOE、OSSI 和 OSSR 位（TIMx_CCER 和 TIMx_BDTR 寄存器）的组合使能 OCx 输出。有关详细信息，请参见 TIMx_CCER 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 总是与 TIMx_CCRx 进行比较，以确定是 TIMx_CCRx ≤ TIMx_CNT 还是 TIMx_CNT ≤ TIMx_CCRx（取决于计数器计数方向）。

TIM16/TIM17 只能递增计数。请参见*第 456 页的递增计数模式*。

以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。*图 197* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 197. 边沿对齐模式的 PWM 波形 (ARR=8)



18.3.10 互补输出和死区插入

TIM16/TIM17 通用定时器可以输出一路互补信号，并管理输出的关断和接通。

这段时间通常称为死区，用户必须根据与输出相连接的器件及其特性（电平转换器的固有延迟、开关器件产生的延迟 ...）来调整死区时间。

每路输出可以独立选择输出极性（主输出 OCx 或互补输出 OCxN）。可通过对 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位执行写操作来完成极性选择。

互补信号 OCx 和 OCxN 通过以下多个控制位的组合进行激活：TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位以及 TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。更多详细信息，请参见 *第 488 页的表 75：具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17)*。应当注意，切换至空闲状态（MOE 位变为 0）的时刻，死区仍然有效。

CCxE 和 CCxNE 位同时置 1 并且 MOE 位置 1（若存在刹车电路）时，将使能死区插入。每个通道有一个 10 位死区发生器。将基于参考波形 OCxREF 生成 2 个输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高电平有效：

- 输出信号 OCx 与参考信号相同，只是其上升沿相对参考上升沿存在延迟。
- 输出信号 OCxN 与参考信号相反，并且其上升沿相对参考下降沿存在延迟。

如果延迟时间大于有效输出（OCx 或 OCxN）的宽度，则不会产生相应的脉冲。

下图所示为死区发生器的输出信号与参考信号 OCxREF 之间的关系。（在这些示例中，假定 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1）。

图 198. 带死区插入的互补输出。

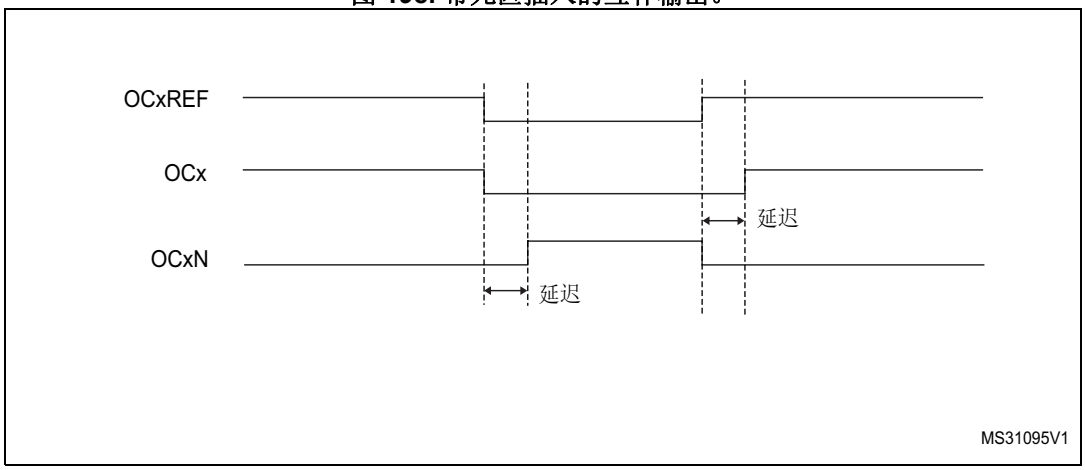


图 199. 延迟时间大于负脉冲宽度的死区波形。

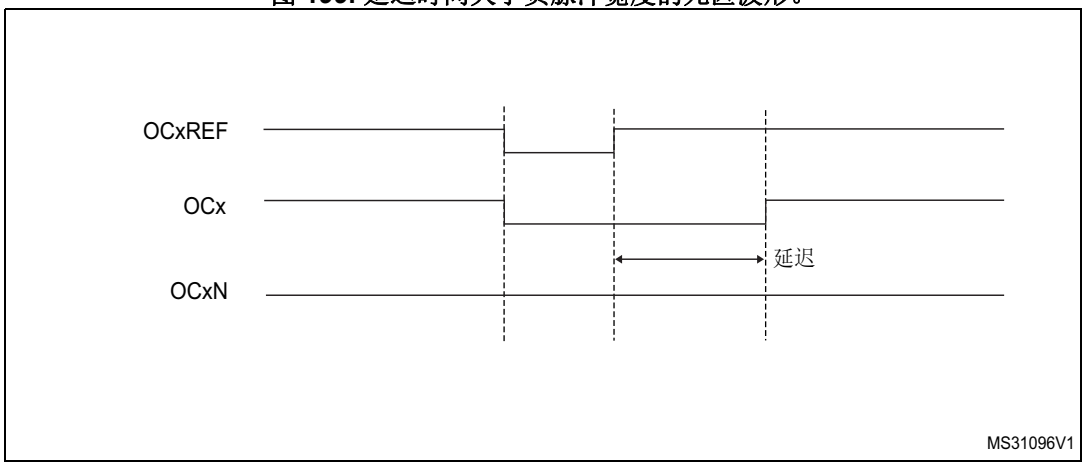
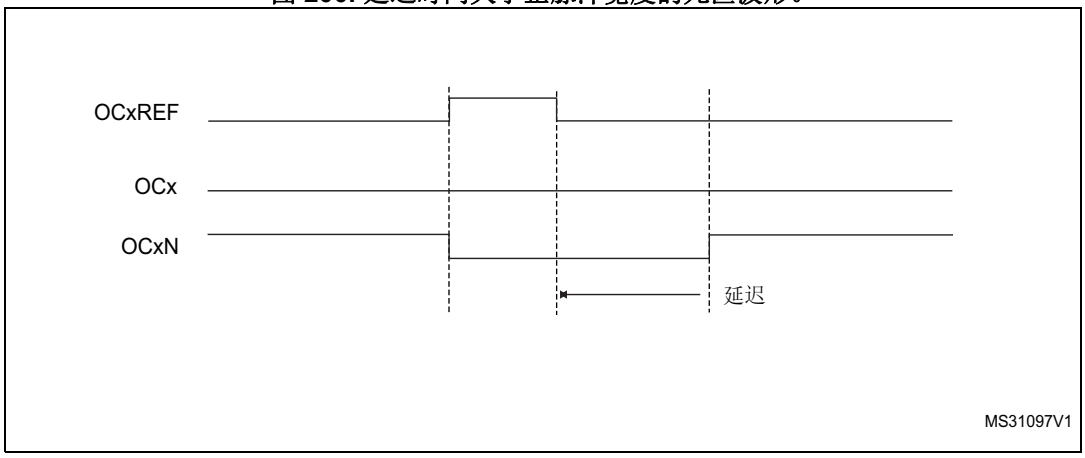


图 200. 延迟时间大于正脉冲宽度的死区波形。



死区延迟对于所有通道均相同，可通过 TIMx_BDTR 寄存器中的 DTG 位进行编程。有关延迟时间计算的信息，请参见*第 491 页的第 18.4.14 节：TIMx 刹车和死区寄存器 (TIMx_BDTR)（x = 16 到 17）*。

将 OCxREF 重定向到 OCx 或 OCxN

在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位，可将 OCxREF 重定向到 OCx 输出或 OCxN 输出。

通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。或者，使两个输出同时保持无效电平，或者两个输出同时处于有效电平，两者互补并且带死区。

注意： *如果仅使能 OCxN (CCxE=0, CCxNE=1)，两者不互补，一旦 OCxREF 为高电平，OCxN 即变为有效。例如，如果 CCxNP=0，则 OCxN=OCxRef。另一方面，如果同时使能 OCx 和 OCxN (CCxE=CCxNE=1)，OCx 在 OCxREF 为高电平时变为有效，而 OCxN 则与之互补，在 OCxREF 为低电平时变为有效。*

18.3.11 使用刹车功能

刹车功能的目的是保护由 TIM16/TIM17 定时器生成的 PWM 信号所驱动的电源开关。刹车输入通常被连接到功率级和三相逆变器的故障输出。激活时，刹车电路会关闭 PWM 输出，并将其强制为预定义的安全状态。

刹车通道收集系统级故障（时钟失效和奇偶校验错误等）和应用故障（来自输入引脚）。刹车电路可以在死区持续时间后将输出强制为预定义的电平（有效或无效）。

刹车期间的输出使能信号和输出电平取决于多个控制位：

- TIMx_BDTR 寄存器中的 MOE 位，允许通过软件使能 / 禁止输出，在发生刹车事件和刹车 2 事件时复位。
- TIMx_BDTR 寄存器中的 OSSI 位，定义定时器将输出控制在无效状态下，还是将输出控制释放给 GPIO 控制器（通常使其处于高阻态模式）
- TIMx_CR2 寄存器中的 OISx 和 OISxN 位，将输出设置为关断电平（有效或无效）。无论 OISx 和 OISxN 的值为何，均无法在给定时间将 OCx 和 OCxN 输出同时设置为有效电平。更多详细信息，请参见 *第 488 页的表 75：具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17)*。

退出复位状态后，刹车功能处于禁止状态，MOE 位处于低电平。通过设置 TIMx_BDTR 寄存器中的 BKE 位来使能刹车功能。刹车输入的极性可通过该寄存器中的 BKP 位来选择。BKE 和 BKP 位可同时修改。对 BKE 和 BKP 位执行写操作时，写操作会在 1 个 APB 时钟周期的延迟后生效。因此，执行写操作后，需要等待 1 个 APB 时钟周期，才能准确回读该位。

由于 MOE 下降沿可能是异步信号，因此在实际信号（作用于输出）与同步控制位（位于 TIMx_BDTR 寄存器中）之间插入了再同步电路，从而在异步信号与同步信号之间产生延迟。具体而言，如果在 MOE 处于低电平时将其置 1，则必须首先插入延迟（空指令），才能准确进行读取。这是因为写入的是异步信号，而读取的却是同步信号。

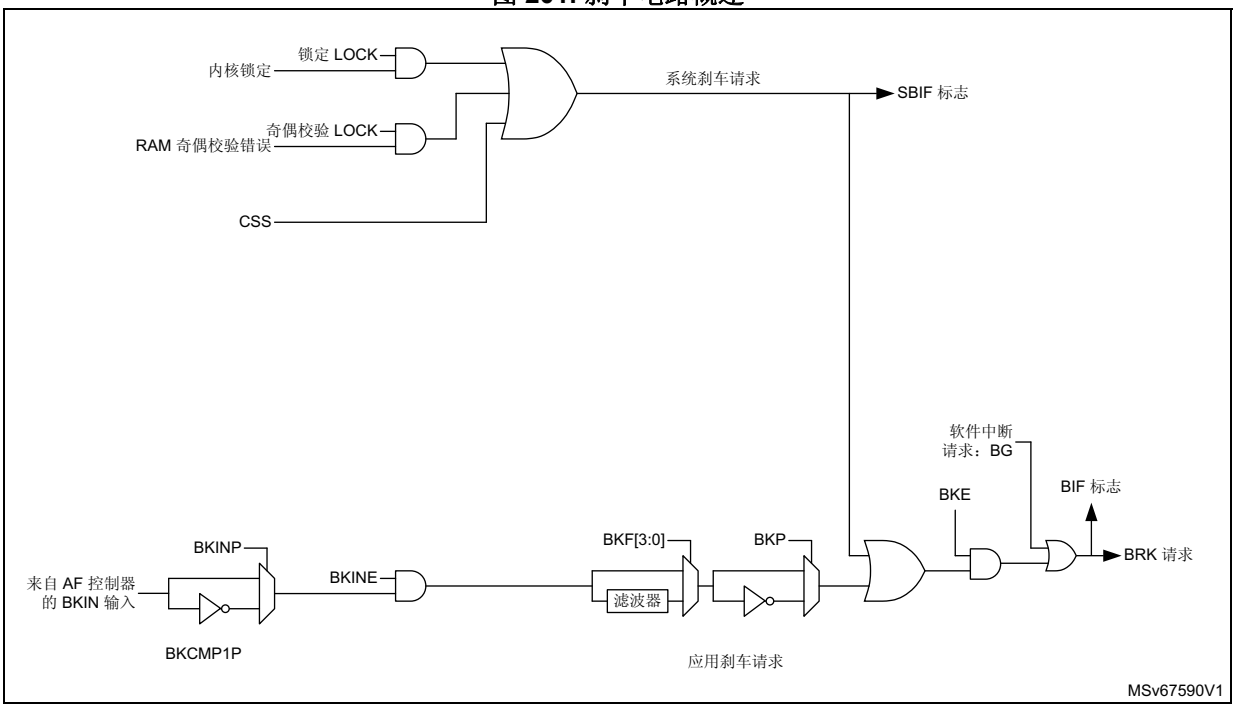
滤波器可编程（TIMx_BDTR 寄存器中的 BKF[3:0] 位），以滤除伪事件。

可以使用 TIMx_AF1 寄存器从多个源产生刹车，这些源可以单独使能并使用可编程边沿有效性。

刹车 (BRK) 通道的源为：

- 连接到 BKIN 引脚的外部源（由 GPIO 复用功能寄存器选定），具有极性选择和可选的数字滤波。
- 内部源：
  - 系统中断：
    - Cortex®-M0+ LOCKUP 输出
    - SRAM 奇偶校验错误信号
    - CSS 检测器产生时钟故障事件

图 201. 刹车电路概述



小心： 只有禁止可编程滤波器时才能保证异步（无时钟）操作。如果使能可编程滤波器，必须使用故障安全时钟模式（例如，使用 CSS）来保证能够处理刹车事件。

发生刹车（刹车输入上出现所选电平）时：

- MOE 位异步清零，使输出处于无效状态、空闲状态甚至释放控制权给 GPIO 控制器（通过 OSSI 位进行选择）。即使 MCU 振荡器关闭，该功能仍然有效。
- MOE=0 时，将以 TIMx_CR2 寄存器 OISx 位中设定的电平驱动每个输出通道。如果 OSSI=0，定时器将释放输出控制（由 GPIO 控制器接管），否则使能输出保持高电平。
- 使用互补输出时：
  - 输出首先置于复位状态或无效状态（取决于极性）。这是异步操作，因此即使没有为定时器提供时钟，该操作仍有效。
  - 如果定时器时钟仍存在，则将重新激活死区发生器，进而在死区后以 OISx 和 OISxN 位中编程的电平驱动输出。即使在这种情况下，也不能同时将 OCx 和 OCxN 驱动至其有效电平。请注意，MOE 进行再同步，因此死区的持续时间会比通常情况长一些（约 2 个 ck_tim 时钟周期）。
  - 如果 OSSI=0，定时器将释放使能输出（由强制高阻态的 GPIO 接管），否则使能输出将保持高电平或在 CCxE 或 CCxNE 位之一为高电平时立即变为高电平。
- 将刹车状态标志（TIMx_SR 寄存器中的 BIF 位）置 1。如果 TIMx_DIER 寄存器中的 BIE 位置 1，可产生中断。
- 如果 TIMx_BDTR 寄存器中的 AOE 位置 1，则 MOE 位会在发生下一更新事件 (UEV) 时自动再次置 1。这一特性有许多用处，比如，可用于实现调节器的功能。否则，MOE 将始终保持低电平，直到再次向该位写入 1。这种情况下，这一特性可用于确保安全。可以将刹车输入连接到功率驱动器的警报、温度传感器或任何安全元件。

注意： *当 AOE 位置 1 时，如果 CPU 将 MOE 复位，则输出处于空闲状态并根据 OSSI 值被强制设为无效电平或高阻态。*
如果 CPU 同时将 MOE 和 AOE 位复位，则输出处于禁止状态并由 TIMx_CR2 寄存器中 OISx 位设定的电平驱动。

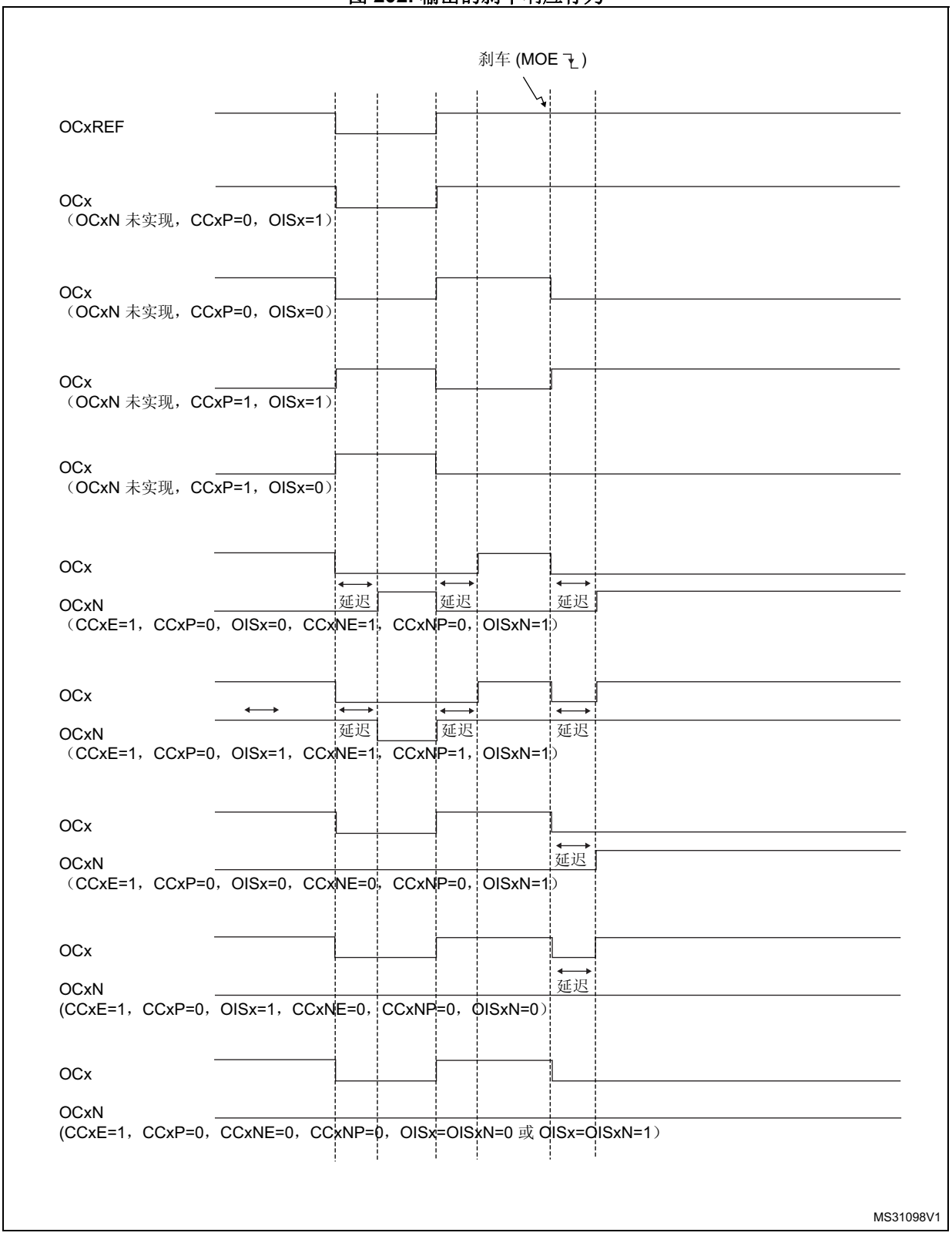
注意： *刹车输入为电平有效。因此，当刹车输入为有效电平时，不能将 MOE 位置 1（自动或通过软件）。同时，不能将状态标志 BIF 清零。*

刹车可由 BRK 输入生成，该输入具有可编程极性，其使能位 BKE 位于 TIMx_BDTR 寄存器中。

除刹车输入和输出管理外，刹车电路内部还实施了写保护，用以保护应用的安全。通过该功能，用户可冻结多个参数配置（死区持续时间、OCx/OCxN 极性和禁止时的状态、OCxM 配置、刹车使能和极性）。可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。请参见*第 491 页的第 18.4.14 节：TIMx 刹车和死区寄存器 (TIMx_BDTR)（x = 16 到 17）*。MCU 复位后只能对 LOCK 位执行一次写操作。

图 202 所示为输出对刹车响应行为的示例。

图 202. 输出的刹车响应行为



18.3.12 双向刹车输入

TIM16/TIM17 具有双向刹车 I/O，如*图 203* 所示。

它们可以实现以下用途：

- 将板级全局刹车信号用于向外部 MCU 或栅极驱动器发送故障信号，唯一的特定引脚做为输入输出双向状态脚。
- 在必须将多个内部和外部刹车输入合并时，将内部刹车源和多个外部开漏比较器输出“或”连接在一起，触发唯一刹车事件。

使用 TIMxBDTR 寄存器的 BKBID 位将刹车输入配置为双向模式。可以使用 TIMxBDTR 寄存器中的 LOCK 位，将 BKBID 编程位锁定在只读模式（处于锁定级别 1 或更高级别）。

双向模式需要将 I/O 配置为开漏模式且使极性低电平有效（使用 BKINP 和 BKP 位）。任何来自系统（例如 CSS）、片上外设或刹车输入的刹车请求都会强制将刹车输入置为低电平，以通知发生了故障事件。如果未正确设置极性位（高电平有效极性），则出于安全目的禁止双向模式。

软件刹车事件 (BG) 也会导致刹车 I/O 被强制为“0”，从而向外部组件指示定时器已进入刹车状态。但是仅在刹车使能时 (BKE = 1) 有效。当生成软件刹车事件且 BKE = 0 时，输出将被置于安全状态，并且刹车标志置 1，但对刹车 I/O 无影响。

安全解除机制可防止系统最终锁定（刹车输入上的低电平会触发刹车，进而将相同输入强制置为低电平）。

当 BKDSRM 位置 1 时，会释放刹车输出以清除故障信号，从而使系统能够重新启动。

在任何情况下都不能禁止刹车保护电路：

- 刹车输入路径始终有效：即使 BKDSRM 位置 1 且释放开漏控制，刹车事件也仍然有效。这样可以在发生刹车期间防止 PWM 输出重新启动。
- 使能输出（MOE 位置 1）后，BKDSRM 位不能解除刹车保护（请参见*表 74*）。

表 74. 刹车保护解除条件

MOE	BKDIR	BKDSRM	刹车保护状态
0	0	X	启动
0	1	0	启动
0	1	1	解除
1	X	X	启动

启动和重新启动刹车电路

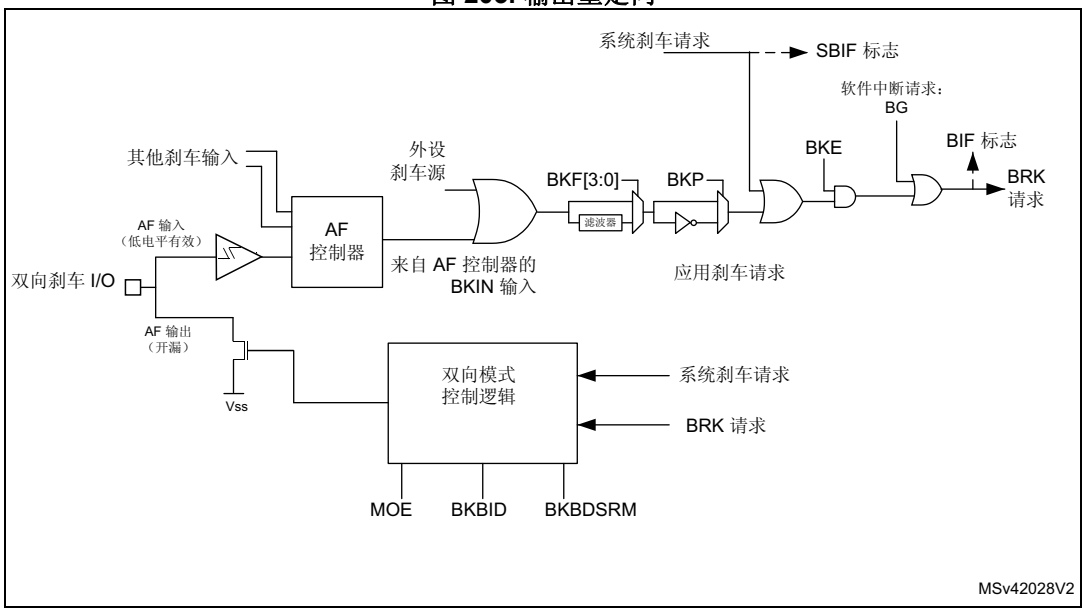
默认情况下（外设复位配置）会启动刹车电路（在输入或双向模式下）。

发生刹车事件后，必须按照以下步骤重新启动保护：

- 必须将 BKDSRM 位置 1，以释放输出控制
- 软件必须等待系统刹车条件消失（如果有），并清零 SBIF 状态标志（或在重新启动前由系统清零）
- 软件必须轮询 BKDSRM 位，直到该位由硬件清零（当应用刹车条件消失时）

此后，刹车电路即启动并激活，可以通过将 MOE 位置 1 来重新使能 PWM 输出。

图 203. 输出重定向



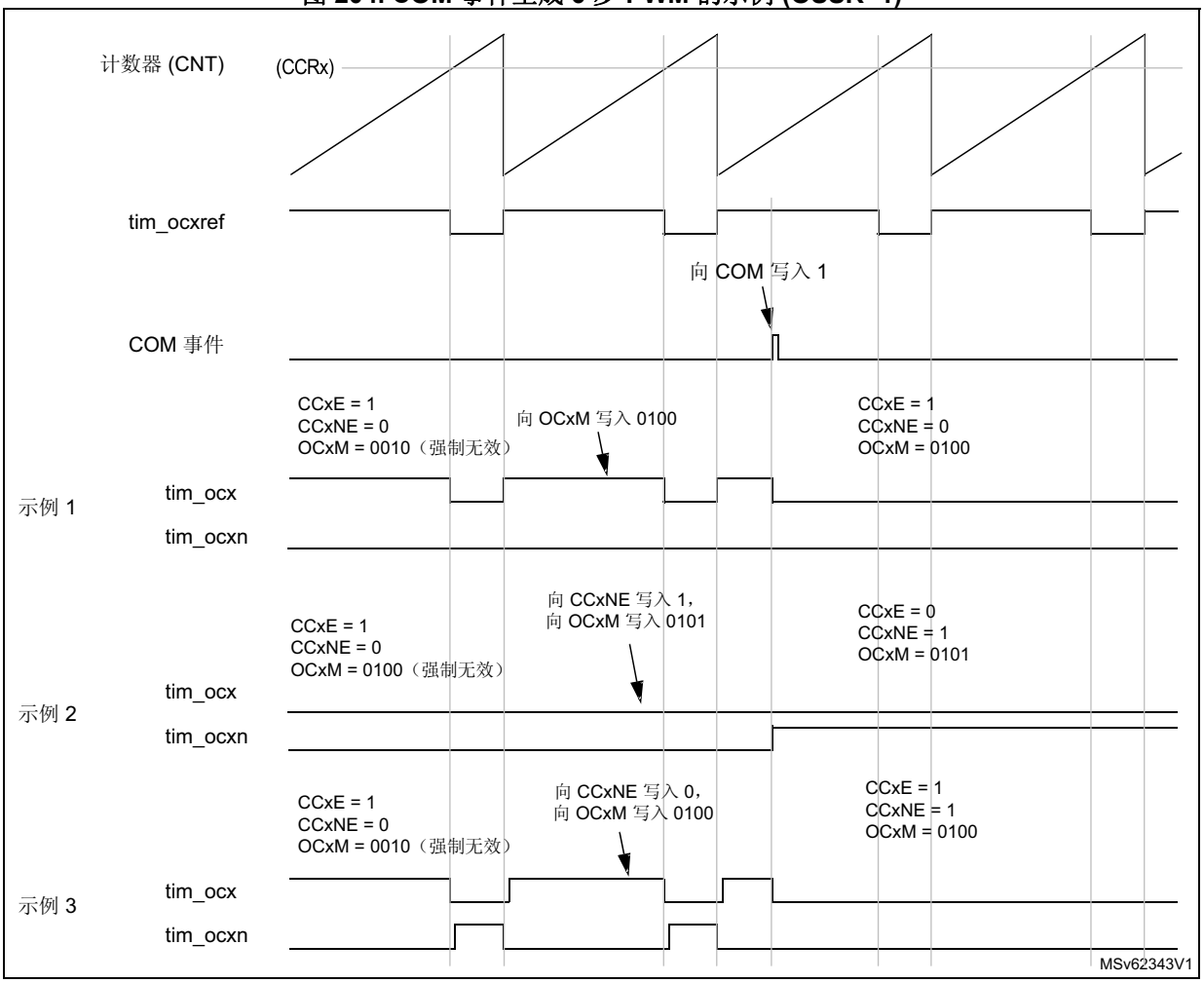
18.3.13 生成 6 步 PWM

当通道使用互补输出时，OCxM、CCxE 和 CCxNE 位上提供预装载位。发生 COM 换向事件时，这些预装载位将传输到影子位。因此，用户可以预先编程下一步骤的配置，并同时更改所有通道的配置。COM 可由软件通过将 TIMx_EGR 寄存器中的 COM 位置 1 而生成，也可以由硬件在 tim_trgi 上升沿生成。

发生 COM 事件时，某个标志位（TIMx_SR 寄存器中的 COMIF 位）将会置 1。这时，如果 TIMx_DIER 寄存器中的 COMIE 位置 1，将产生中断；如果 TIMx_DIER 寄存器中的 COMDE 位置 1，则将产生 DMA 请求。

图 204 以 3 种不同的编程配置为例，显示了发生 COM 事件时 tim_ocx 和 tim_ocxn 输出的行为。

图 204. COM 事件生成 6 步 PWM 的示例 (OSSR=1)



18.3.14 单脉冲模式

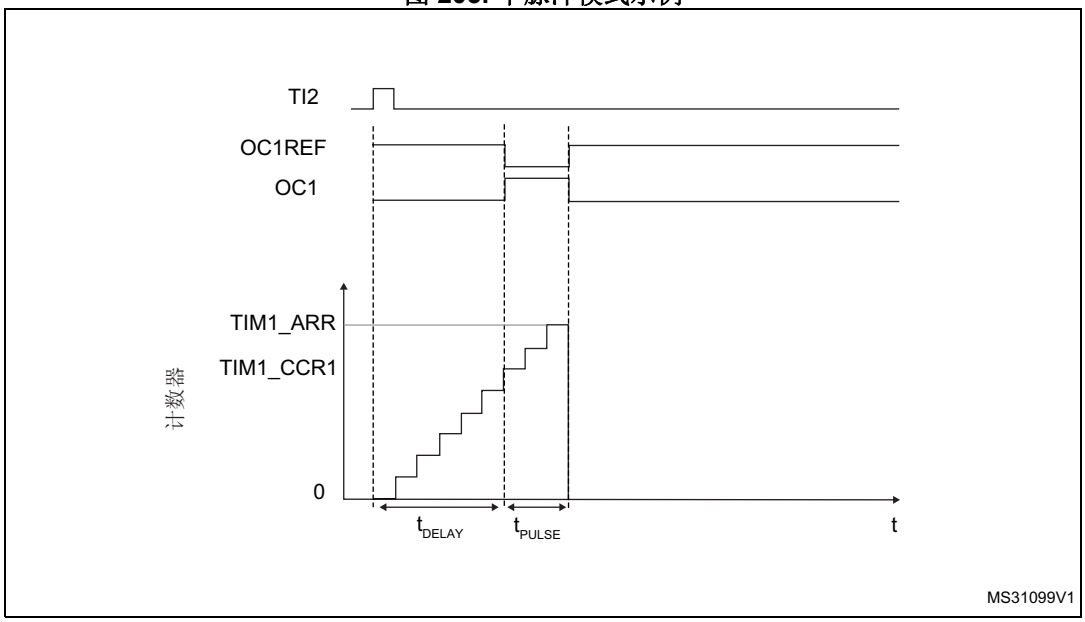
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。通过将 TIMx_CR1 寄存器中的 OPM 位置 1 选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- CNT < CCRx ≤ ARR（特别注意，0 < CCRx）

图 205. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2[x] 源（内部或外部）。
2. 在 TIMx_CCMR1 寄存器中写入 CC2S=“01”，以将 TI2FP2 映射到 TI2。
3. 在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
4. 在 TIMx_SMCR 寄存器中写入 TS=“00110”，以将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
5. 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），以使用 TI2FP2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，必须通过在 TIMx_CCMR1 寄存器中写入 OC1M=111，来使能 PWM 模式 2。可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

由于仅需要 1 个脉冲，因此必须向 TIMx_CR1 寄存器的 OPM 位写入 1，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。

特例：OCx 快速使能

在单脉冲模式下，TIx 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟（t_{DELAY} 最小值）。

如果要输出延迟时间最短的波形，可以将 TIMx_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 OCxRef（和 OCx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

18.3.15 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可按原子方式读取计数器值以及由 UIFCPY 标志发出的更新事件情况。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

UIF 和 UIFCPY 标志之间的置位没有延迟。

18.3.16 从模式 —— 复位 + 触发组合模式

在这种情况下，在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个定时器更新事件，并启动计数器。

该模式用于单脉冲模式。

18.3.17 DMA 分组传送模式

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的几个寄存器多次重新编程而无需软件开销，但也可以用于定期地一次性地批量读取定时器的多个寄存器的内容。

DMA 控制器访问目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（分组）。每次写入 TIMx_DMAR 寄存器都会重定向到定时器其中的一个寄存器。

TIMx_DCR 寄存器中的 DBL[4:0] 位设置 DMA 连续传送长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送，DBL 即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA[4:0] 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读 / 写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：

00000：TIMx_CR1，

00001：TIMx_CR2，

00010：TIMx_SMCR，

例如，定时器 DMA 连续传送功能用于在发生更新事件后将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   - DMA 通道外设地址为 DMAR 寄存器地址。
   - DMA 通道存储器地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
   - 要传输的数据量 = 3（参见下文注释）。
   - 禁止循环模式。
2. 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器：
   DBL = 3 次传输，DBA = 0xE。
3. 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx
5. 使能 DMA 通道

本例适用于每个 CCRx 寄存器只更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。

注意： *可以将空值写入保留的寄存器中。*

18.3.18 使用定时器输出作为其他定时器的触发 (TIM16/TIM17)

单通道定时器没有主模式。但是，可以使用 OC1 输出信号来触发其他定时器（包括本文档其他部分所述的定时器）。通过查看关于器件上任何 TIMx_SMCR 寄存器的“TIMx 内部触发连接”表，可以确定哪些定时器可以作为目标从定时器。

OC1 信号的脉冲宽度必须至少设为目标定时器时钟周期的 2 倍，以便确保从定时器能够检测到触发。

例如，如果目标定时器 CK_INT 时钟频率是源定时器的 1/4，则 OC1 的脉冲宽度必须至少是 8 个源定时器时钟周期。

18.3.19 调试模式

当微控制器进入调试模式（Cortex®-M0+ 内核停止）时，TIMx 计数器会根据 DBG 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。更多详细信息，请参见 *第 26.9.2 节：对定时器、看门狗和 I2C 的调试支持*。

为了安全起见，当计数器停止 (DBG_TIMx_STOP = 1) 时，输出被禁止（就像 MOE 位被复位一样）。可以将输出强制变为无效状态（OSSI 位 = 1），或者通过 GPIO 控制器（OSSI 位 = 0）来控制输出，以将其强制为高阻态。

18.4 TIM16/TIM17 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.2 节*。

外设寄存器可按半字（16 位）或字（32 位）访问。

18.4.1 TIMx 控制寄存器 1 (TIMx_CR1)（x = 16 到 17）

TIMx control register 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
Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
				rw		rw	rw	rw				rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)
该位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器所使用的采样时钟 (t_{DTS}) 之间的分频比。
00：$t_{DTS} = t_{CK\_INT}$
01：$t_{DTS} = 2 * t_{CK\_INT}$
10：$t_{DTS} = 4 * t_{CK\_INT}$
11：保留，不要设置成此值

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)
0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

位 6:4 保留，必须保持复位值。

位 3 **OPM**：单脉冲模式 (One pulse mode)
0：计数器在发生更新事件时不会停止计数
1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

位 2 **URS**：更新请求源 (Update request source)
该位由软件置 1 和清零，用以选择 UEV 事件源。
0：使能时，所有以下事件都会生成更新中断或 DMA 请求。此类事件包括：
– 计数器上溢 / 下溢
– 将 UG 位置 1
– 通过从模式控制器生成的更新事件
1：使能时，只有计数器上溢 / 下溢会生成更新中断或 DMA 请求。

位 1 **UDIS**：更新禁止 (Update disable)

该位由软件置 1 和清零，用以使能 / 禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢 / 下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

带缓冲的寄存器被加载为预加载数值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注意：只有事先通过软件将 CEN 位置 1，才可以使用外部时钟和门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

18.4.2 TIMx 控制寄存器 2 (TIMx_CR2)（x = 16 到 17）

TIMx control register 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.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC
						rw	rw					rw	rw		rw

位 15:10 保留，必须保持复位值。

位 9 **OIS1N**：输出空闲状态 1（OC1N 输出）(Output Idle state 1 (OC1N output))

0：当 MOE=0 时，经过死区时间后 OC1N=0

1：当 MOE=0 时，经过死区时间后 OC1N=1

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位即无法修改。

位 8 **OIS1**：输出空闲状态 1（OC1 输出）(Output Idle state 1 (OC1 output))

0：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0

1：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=1

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位即无法修改。

位 7:4 保留，必须保持复位值。

位 3 **CCDS**：捕获 / 比较 DMA 选择 (Capture/compare DMA selection)

0：发生 CCx 事件时发送 CCx DMA 请求

1：发生更新事件时发送 CCx DMA 请求

位 2 **CCUS**：捕获 / 比较控制更新选择 (Capture/compare control update selection)

0： 如果捕获 / 比较控制位 (CCPC=1) 进行预装载，仅通过将 COMG 位置 1 来对这些位进行更新。

1： 如果捕获 / 比较控制位 (CCPC=1) 进行预装载，可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。

注意：该位仅对具有互补输出的通道有效。

位 1 保留，必须保持复位值。

位 0 **CCPC**：捕获 / 比较预装载控制 (Capture/compare preloaded control)
0： CCxE、CCxNE 和 OCxM 位未进行预装载
1： CCxE、CCxNE 和 OCxM 位在写入后被预加载，只有当 COM 位置 1 时才进行更新。
注意：该位仅对具有互补输出的通道有效。

18.4.3 TIMx DMA/ 中断使能寄存器 (TIMx_DIER)（x = 16 到 17）

TIMx DMA/interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1DE	UDE	BIE	Res.	COMIE	Res.	Res.	Res.	CC1IE	UIE
						rw	rw	rw		rw				rw	rw

位 15:10 保留，必须保持复位值。

位 9 **CC1DE**：捕获 / 比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)
0：禁止 CC1 DMA 请求
1：使能 CC1 DMA 请求

位 8 **UDE**：更新 DMA 请求使能 (Update DMA request enable)
0：禁止更新 DMA 请求
1：使能更新 DMA 请求

位 7 **BIE**：刹车中断使能 (Break interrupt enable)
0：禁止刹车中断
1：使能刹车中断

位 6 保留，必须保持复位值。

位 5 **COMIE**：COM 中断使能 (COM interrupt enable)
0：禁止 COM 中断
1：使能 COM 中断

位 4:2 保留，必须保持复位值。

位 1 **CC1IE**：捕获 / 比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断
1：使能 CC1 中断

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断
1：使能更新中断

18.4.4 TIMx 状态寄存器 (TIMx_SR)（x = 16 到 17）

TIMx status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	Res.	COMIF	Res.	Res.	Res.	CC1IF	UIF
						rc_w0		rc_w0		rc_w0				rc_w0	rc_w0

位 15:10 保留，必须保持复位值。

位 9 **CC1OF**：捕获 / 比较 1 过捕获标志 (Capture/Compare 1 overcapture flag)

仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。

0：未检测到过捕获

1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8 保留，必须保持复位值。

位 7 **BIF**：刹车中断标志 (Break interrupt flag)

只要刹车输入变为有效状态，此标志便由硬件置 1。刹车输入无效后可通过软件对其清零。

0：未发生刹车事件

1：在刹车输入上检测到有效电平

位 6 保留，必须保持复位值。

位 5 **COMIF**：COM 中断标志 (COM interrupt flag)

此标志在发生一个 COM 事件时（一旦捕获 / 比较控制位——CCxE、CCxNE、OCxM——已更新）由硬件置 1。但需要通过软件清零。

0：未发生 COM 事件

1：COM 中断挂起

位 4:2 保留，必须保持复位值。

位 1 **CC1IF**：捕获 / 比较 1 中断标志 (Capture/Compare 1 interrupt flag)

该标志由硬件置 1，但需要通过软件清零（输入捕获或输出比较模式）或通过读取 TIMx_CCR1 寄存器清零（仅限输入捕获模式）。

0：未发生比较匹配 / 输入捕获

1：发生了比较匹配 / 输入捕获

如果通道 CC1 配置为输出：当计数器 TIMx_CNT 的值与 TIMx_CCR1 的值匹配时，此标志置 1。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。在中心对齐模式下，可通过 3 种方式将此标志置 1，完整说明请参见 TIMx_CR1 寄存器中的 CMS 位。

如果通道 CC1 配置为输入：当 TIMx_CCR1 寄存器中捕获到计数器值时（如果通过设置 TIMx_CCER 中的 CC1P 和 CC1NP 位定义为边沿有效，则改为在 IC1 上检测到边沿时），该位置 1。

位 0 **UIF**：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

- TIMx_CR1 寄存器中的 UDIS = 0，并且重复计数器值上溢时（重复计数器 = 0 时更新）。
- TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

18.4.5 TIMx 事件生成寄存器 (TIMx_EGR)（x = 16 到 17）

TIMx event generation register

偏移地址：0x14

复位值：0x0000

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

位 15:8 保留，必须保持复位值。

位 7 **BG**：刹车生成 (Break generation)

该位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作。

1：生成刹车事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 6 保留，必须保持复位值。

位 5 **COMG**：捕获 / 比较控制更新生成 (Capture/Compare control update generation)

该位可通过软件置 1，并由硬件自动清零。

0：无操作

1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位

注意：该位仅对具有互补输出的通道有效。

位 4:2 保留，必须保持复位值。

位 1 **CC1G**：捕获 / 比较 1 生成 (Capture/Compare 1 generation)

该位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作。

1：通道 1 上生成捕获 / 比较事件：

如果通道 CC1 配置为输出：

使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。

如果通道 CC1 配置为输入：

TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)

该位可通过软件置 1，并由硬件自动清零。

0：不执行任何操作。

1：重新初始化计数器并生成一个寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。

18.4.6 TIMx 捕获 / 比较模式寄存器 1 (TIMx_CCMR1)（x = 16 到 17）

TIMx capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)

该位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC[1:0]**：输入捕获 1 预分频器 (Input capture 1 prescaler)

该位域定义 CC1 输入 (IC1) 的预分频比。
只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。
00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S[1:0]**：捕获 / 比较 1 选择 (Capture/Compare 1 Selection)

该位域定义通道方向（输入 / 输出）以及所使用的输入。
00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
其他值：保留

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E =“0”），才可向 CC1S 位写入数据。

18.4.7 TIMx 捕获 / 比较模式寄存器 1 [复用] (TIMx_CCMR1)（x = 16 到 17）

TIMx capture/compare mode register 1 [alternate]

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
									rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 15:7 保留，必须保持复位值。

位 16，6:4 **OC1M[3:0]**：输出比较 1 模式 (Output Compare 1 mode)

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：冻结 —— 输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。当定时器用作软件时基时，可以使用该模式。在定时器操作期间使能冻结模式时，输出保持进入冻结状态前的状态（有效或无效）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获 / 比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获 / 比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转 ——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平 ——OC1REF 强制变为低电平。

0101：强制变为有效电平 ——OC1REF 强制变为高电平。

0110：PWM 模式 1—— 只要 TIMx_CNT < TIMx_CCR1，通道 1 便输出有效电平，否则输出无效电平。

0111：PWM 模式 2—— 只要 TIMx_CNT < TIMx_CCR1，通道 1 便输出无效电平，否则输出有效电平。

所有其他值：保留

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。

在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。

OC1M[3] 位不是连续的，而是在位 16 中。

位 3 **OC1PE**：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读 / 写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到影子寄存器中。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。

位 2 **OC1FE**：输出比较 1 快速使能 (Output Compare 1 fast enable)

该位可降低触发事件与定时器输出跳变之间的延迟。单脉冲模式下必须使用（TIMx_CR1 寄存器中的 OPM 位置 1），以便在启动触发后快速发送输出脉冲。

0： 即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1： 触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OC1FE 才会起作用。

位 1:0 **CC1S[1:0]**：捕获 / 比较 1 选择 (Capture/Compare 1 selection)

该位域定义通道方向（输入 / 输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

其他值：保留

注意：仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

18.4.8 TIMx 捕获 / 比较使能寄存器 (TIMx_CCER)（x = 16 到 17）

TIMx capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E
												rw	rw	rw	rw

位 15:4 保留，必须保持复位值。

位 3 **CC1NP**：捕获 / 比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)

CC1 通道配置为输出：

0：OC1N 高电平有效

1：OC1N 低电平有效

CC1 通道配置为输入：

该位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3 且 CC1S= “00”（通道配置为输出），该位立即变为不可写状态。

该位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NP 有效位才会从预装载位获取新值。

位 2 **CC1NE**：捕捉 / 比较 1 互补输出使能 (Capture/Compare 1 complementary output enable)

0： 关闭 ——OC1N 未激活。OC1N 电平与 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位存在函数关系。

1： 开启——基于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值，在相应输出引脚上输出 OC1N 信号。

位 1 **CC1P**：捕获 / 比较 1 输出极性 (Capture/Compare 1 output polarity)

0：OC1 高电平有效（输出模式）/ 边沿有效性选择（输入模式，见下文）

1：OC1 低电平有效（输出模式）/ 边沿有效性选择（输入模式，见下文）

CC1 通道配置为输入时，CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。

CC1NP=0，CC1P=0： 非反相 / 上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=0，CC1P=1： 反相 / 下降沿触发。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=1，CC1P=1： 非反相 / 上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。

CC1NP=1，CC1P=0： 该配置保留，不得使用。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，该位立即变为不可写状态。

该位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。

位 0 **CC1E**：捕捉 / 比较 1 输出使能 (Capture/Compare 1 output enable)

0：禁止捕获模式 /OC1 无效（见下文）

1：使能捕获模式 / 在相应输出引脚上输出 OC1 信号

当 CC1 通道配置为输出时，OC1 电平取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位，与 CC1E 位的状态无关。有关详细信息，请参见*表 75*。

表 75. 具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17)

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	X	0	0	禁止输出（不由定时器驱动：高阻态 OCx=0 OCxN=0、OCxN_EN=0	
		0	0	1	禁止输出（不由定时器驱动：高阻态 OCx=0	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	禁止输出（不由定时器驱动：高阻态 OCxN=0
		X	1	1	OCREF + 极性 + 死区	OCREF 互补项（对 OCREF 进行“非”运算）+ 极性 + 死区
		1	0	1	关闭状态（输出使能为无效状态） OCx=CCxP	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		1	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP、 OCx_EN=1	关闭状态（输出使能为无效状态） OCxN=CCxNP、OCxN_EN=1

表 75. 具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17)（续）

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
0	0	X	X	X	禁止输出（不由定时器驱动：高阻态）。	
	1		0	0		
			0	1	关闭状态（输出使能为无效状态） 异步：OCx=CCxP、OCxN=CCxNP 如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，从而假定 OISx 和 OISxN 在有效状态下与 OCX 和 OCxN 不对应	
			1	0		
			1	1		

1. 如果一个通道的两个输出均未使用（由 GPIO 控制器接管控制），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注意： *与互补通道 OCx 和 OCxN 相连的外部 I/O 引脚的状态取决于通道 OCx 和 OCxN 的状态以及 GPIO 控制和复用功能寄存器。*

18.4.9 TIMx 计数器 (TIMx_CNT)（x = 16 到 17）

TIMx counter

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIF CPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **UIFCPY**：UIF 副本 (UIF Copy)
该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。如果 TIMx_CR1 中的 UIFREMAP 位复位，则位 31 保留，读为 0。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

18.4.10 TIMx 预分频器 (TIMx_PSC)（x = 16 到 17）

TIMx prescaler

偏移地址：0x28

复位值：0x0000

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

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)
计数器时钟频率 (CK_CNT) 等于 f_{CK_PSC} / (PSC[15:0] + 1)。
PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到实际预分频器寄存器的值。

18.4.11 TIMx 自动重载寄存器 (TIMx_ARR)（x = 16 到 17）

TIMx auto-reload register

偏移地址：0x2C

复位值：0xFFFF

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

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的更多详细信息，请参见*第 454 页的第 18.3.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

18.4.12 TIMx 重复计数器寄存器 (TIMx_RCR)（x = 16 到 17）

TIMx repetition counter register

偏移地址：0x30

复位值：0x0000

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

位 15:8 保留，必须保持复位值。

位 7:0 **REP[7:0]**：重复计数器值 (Repetition counter value)

使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器向影子寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。

与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，向 TIMx_RCR 寄存器写入任何值都无影响。

这意味着在 PWM 模式下，每产生 (REP+1) 个 PWM 脉冲就产生更新事件。

18.4.13 TIMx 捕获 / 比较寄存器 1 (TIMx_CCR1)（x = 16 到 17）

TIMx capture/compare register 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[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR1[15:0]**：捕获 / 比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：

CCR1 是捕获 / 比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际的捕获 / 比较寄存器 1）。

实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出相应电平的值。

如果通道 CC1 配置为输入：

CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。

18.4.14 TIMx 刹车和死区寄存器 (TIMx_BDTR)（x = 16 到 17）

TIMx break and dead-time register

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.	Res.	Res.	BKBID	Res.	BK DSRM	Res.	Res.	Res.	Res.	Res.	Res.	BKF[3:0]			
			rw		rw							rw	rw	rw	rw

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

注意： *由于可以根据 LOCK 配置锁定位 BKBID、BKDSRM、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 位的写操作，因此必须在第一次对 TIMx_BDTR 寄存器执行写访问时对这些位进行配置。*

位 31:29 保留，必须保持复位值。

位 28 **BKBID**：双向刹车 (Break bidirectional)

0：刹车输入 BRK 为输入模式
1：刹车输入 BRK 为双向模式
在双向模式下（BKBID 位置 1），刹车输入配置为输入模式和开漏输出模式。任何激活的刹车事件都将使刹车输入上呈逻辑低电平，以向外部器件指示发生了内部刹车事件。

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 27 保留，必须保持复位值。

位 26 **BKDSRM**：刹车解除 (Break Disarm)

0：启动刹车输入 BRK
1：解除刹车输入 BRK
当没有刹车源被激活时，该位由硬件清零。
必须通过软件将 BKDSRM 位置 1 以释放双向输出控制（开漏输出处于高阻态），然后不断轮询该位，直到其由硬件复位，指示故障条件已消失。

注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 25:20 保留，必须保持复位值。

位 19:16 **BKF[3:0]**：刹车滤波器 (Break filter)

该位域可定义 BRK 输入的采样频率和适用于 BRK 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：

0000：无滤波器，BRK 异步工作
0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，该位即无法修改。

位 15 **MOE**：主输出使能 (Main output enable)

只要刹车输入变为有效状态，该位便由硬件异步清零。该位由软件置 1，也可根据 AOE 位状态自动置 1。该位仅对配置为输出的通道有效。

0：OC 和 OCN 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。
1：如果 OC 和 OCN 输出的相应使能位（TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位）均置 1，则使能 OC 和 OCN 输出。

有关详细信息，请参见 OC/OCN 使能说明（*第 487 页的第 18.4.8 节：TIMx 捕获 / 比较使能寄存器 (TIMx_CCER)（x = 16 到 17）*）。

位 14 **AOE**：自动输出使能 (Automatic output enable)

0：MOE 只能由软件置 1
1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1（如果刹车输入无效）

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 13 **BKP**：刹车极性 (Break polarity)

0：刹车输入 BRK 为低电平有效
1：刹车输入 BRK 为高电平有效

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。
对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 12 **BKE**：刹车使能 (Break enable)

0：禁止刹车输入（BRK 和 CCS 时钟故障事件）
1：使能刹车输入（BRK 和 CCS 时钟故障事件）

注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，该位即无法修改。
对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 11 **OSSR**：运行模式下的关闭状态选择 (Off-state selection for Run mode)

该位在 MOE = 1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。

有关详细信息，请参见 OC/OCN 使能说明（*第 487 页的第 18.4.8 节：TIMx 捕获 / 比较使能寄存器 (TIMx_CCER)（x = 16 到 17）*）。

0：处于无效状态时，禁止 OC/OCN 输出（输出控制器呈高阻态，相应输出由GPIO接管。）
1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平（输出仍由定时器控制）。

注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，该位即无法修改。

位 10 **OSSI**：空闲模式下的关闭状态选择 (Off-state selection for Idle mode)

该位在 MOE=0 时作用于配置为输出的通道。
有关详细信息，请参见 OC/OCN 使能说明（*第 487 页的第 18.4.8 节：TIMx 捕获 / 比较使能寄存器 (TIMx_CCER)（x = 16 到 17）*）。
0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号 =0）
1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便将 OC/OCN 输出首先强制为其空闲电平。然后设置 OC/OCN 使能输出信号 =1

注意：编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，该位即无法修改。

位 9:8 **LOCK[1:0]**：锁定配置 (Lock configuration)

这些位用于针对软件错误提供写保护。
00：关闭锁定 —— 不对任何位提供写保护
01：锁定级别 1，此时无法对 TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_BDTR 寄存器中的 BKE/BKP/AOE 位执行写操作。
10：锁定级别 2，此时无法对锁定级别 1 中适用的各位、CC 极性位（TIMx_CCER 寄存器中的 CCxP/CCxNP 位，只要通过 CCxS 位将相关通道配置为输出）以及 OSSR 和 OSSI 位执行写操作。
11：锁定级别 3，此时无法对锁定级别 2 中适用的各位、CC 控制位（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

注意：复位后只能对 LOCK 位执行一次写操作。对 TIMx_BDTR 寄存器执行写操作后其中的内容将冻结，直到下一次复位。

位 7:0 **DTG[7:0]**：配置死区发生器 (Dead-time generator setup)

该位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。
DTG[7:5] = 0xx => DT = DTG[7:0] x t_{dtg}，其中 $t_{dtg} = t_{DTS}$
DTG[7:5] = 10x => DT = (64 + DTG[5:0]) x t_{dtg}，其中 $t_{dtg} = 2 \times t_{DTS}$
DTG[7:5] = 110 => DT = (32 + DTG[4:0]) x t_{dtg}，其中 $t_{dtg} = 8 \times t_{DTS}$
DTG[7:5] = 111 => DT = (32 + DTG[4:0]) x t_{dtg}，其中 $t_{dtg} = 16 \times t_{DTS}$
示例：如果 t_{DTS} = 125 ns (8 MHz)，则可能的死区值为：
0 到 15875 ns（步长为 125 ns），
16 µs 到 31750 ns（步长为 250 ns），
32 µs 到 63 µs（步长为 1 µs），
64 µs 到 126 µs（步长为 2 µs）

注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，该位域即无法修改。

18.4.15 TIMx DMA 控制寄存器 (TIMx_DCR)（x = 16 到 17）

TIMx DMA control register

偏移地址：0x48

复位值：0x0000

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

位 15:13 保留，必须保持复位值。

位 12:8 **DBL[4:0]**：DMA 连续传送长度 (DMA burst length)

该 5 位域向量定义了 DMA 的传送长度（当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送），即一个 DMA 要连续传送的次数。可按半字或字节进行传送（请参见下面的示例）。
00000：1 次传输，
00001：2 次传输，
00010：3 次传输，
...
10001：18 次传输。

位 7:5 保留，必须保持复位值。

位 4:0 **DBA[4:0]**：DMA 基址 (DMA base address)

该 5 位域定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读 / 写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。
示例：
00000：TIMx_CR1，
00001：TIMx_CR2，
00010：TIMx_SMCR，
...

示例：考虑以下传输：DBL = 7 次传输，DBA = TIMx_CR1。这种情况下将向 / 从自 TIMx_CR1 地址开始的 7 个寄存器传输数据。

18.4.16 TIMx 全传输 DMA 地址 (TIMx_DMAR)（x = 16 到 17）

TIMx DMA address for full transfer

偏移地址：0x4C

复位值：0x0000

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

位 15:0 **DMAB[15:0]**：DMA 分组传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器

(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

18.4.17 TIM16 复用功能寄存器 1 (TIM16_AF1)

TIM16 alternate function register 1

偏移地址：0x60

复位值：0x0000 0001

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

位 31:10 保留，必须保持复位值。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
该位选择 BKIN 复用功能输入有效电平，必须与 BKP 极性位一起编程。
0：BKIN 输入为低电平有效
1：BKIN 输入为高电平有效
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 8:1 保留，必须保持复位值。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
该位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

18.4.18 TIM16 输入选择寄存器 (TIM16_TISEL)

TIM16 input selection register

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM16_CH1 输入
0001：LSI
0010：LSE
0011：保留
0100：MCO2
其他值：保留

18.4.19 TIM17 复用功能寄存器 1 (TIM17_AF1)

TIM17 alternate function register 1

偏移地址：0x60

复位值：0x0000 0001

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

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

位 31:10 保留，必须保持复位值。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
该位选择 BKIN 复用功能输入有效电平，必须与 BKP 极性位一起编程。
0：BKIN 输入为低电平有效
1：BKIN 输入为高电平有效
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

位 8:1 保留，必须保持复位值。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
该位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注意：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，该位即无法修改。

18.4.20 TIM17 输入选择寄存器 (TIM17_TISEL)

TIM17 input selection register

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM17_CH1 输入
0001：保留
0010：HSE/32
0011：MCO
0100：MCO2
其他值：保留

18.4.21 TIM16/TIM17 寄存器映射

TIM16/TIM17 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 76. TIM16/TIM17 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0				0	0	0	0
0x04	**TIMx_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC
	Reset value																							0	0					0	0		0
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1DE	UDE	BIE	Res.	COMIE	Res.	Res.	Res.	CC1IE	UIE
	Reset value																							0	0	0		0				0	0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	Res.	COMIF	Res.	Res.	Res.	CC1IF	UIF
	Reset value																							0		0		0				0	0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	Res.	COMG	Res.	Res.	Res.	CC1G	UG
	Reset value																									0		0				0	0
0x18	**TIMx_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																0										0	0	0	0	0	0	0
	TIMx_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1PSC [1:0]		CC1S [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0x20	**TIMx_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E
	Reset value																													0	0	0	0
0x24	**TIMx_CNT**	UIFCPY or Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIMx_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIMx_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表 76. TIM16/TIM17 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x30	**TIMx_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x34	**TIMx_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIMx_BDTR**	Res.	Res.	Res.	BKBID	Res.	BKDSRM	Res.	Res.	Res.	Res.	Res.	Res.	BKF[3:0]				MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DTG[7:0]							
	Reset value				0		0							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**TIMx_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIMx_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM16_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINE
	Reset value																							0									1
0x60	**TIM17_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKINE
	Reset value																							0									1
0x68	**TIM16_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																													0	0	0	0
0x68	**TIM17_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

19 红外接口 (IRTIM)

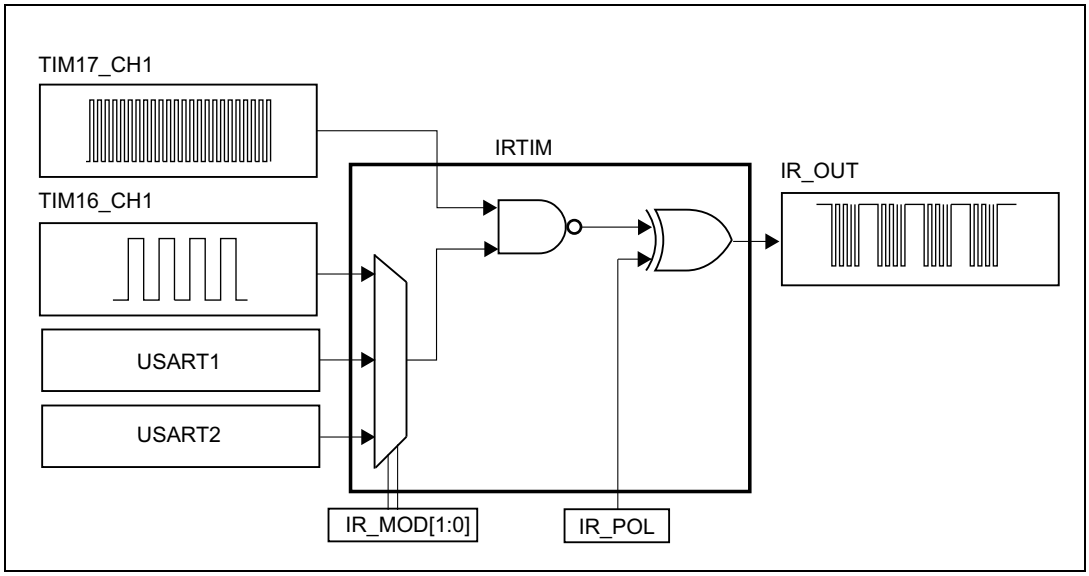
器件具有用于遥控的红外接口 (IRTIM)。此接口与红外 LED 一起用于实现遥控功能。

此接口内部连接到 USART1、 USART2、 TIM16 和 TIM17，如*图 206* 所示。

要生成红外遥控信号，必须使能 IR 接口并正确配置 TIM16 通道 1 (TIM16_OC1) 和 TIM17 通道 1 (TIM17_OC1)，以生成正确波形。

通过基本输入捕获模式可以轻松实现红外接收器。

图 206.IRTIM 与 TIM16 和 TIM17 的内部硬件连接



所有标准 IR 脉冲调制模式都可通过编程两个定时器输出比较通道获得。

TIM17 用于生成高频载波信号，而 TIM16、 USART1 或 USART2 生成调制包络，具体取决于 SYSCFG_CFGR1 寄存器中 IR_MOD[1:0] 位的设置。

IRTIM 输出信号的极性由 SYSCFG_CFGR1 寄存器中的 IR_POL 位控制，可以通过将此位置 1 来反相。

在 IR_OUT 引脚上输出红外功能。通过 GPIOx_AFRx 寄存器使能相关复用功能位来激活此功能。

高灌电流 LED 驱动能力（仅在 PB9 引脚和 PC14 引脚上可用）可以通过 SYSCFG_CFGR1 寄存器中的 I2C_PB9_FMP 位和/或 I2C_PC14_FMP 位激活，并用于吸收直接控制红外 LED 所需的高电流。

20 独立看门狗 (IWDG)

20.1 简介

此器件具有一个嵌入式看门狗外设，具有安全性高、定时准确及使用灵活的优点。此独立看门狗外设可检测并解决由软件错误导致的故障，并在计数器达到给定的超时值时触发系统复位。

独立看门狗 (IWDG) 由内置的低速时钟振荡器驱动，因此即便在主时钟发生故障时仍然保持工作状态。

IWDG 最适合应用于需要看门狗作为一个在主程序之外，能够完全独立工作，并且对时间精度要求较低的应用。有关窗口看门狗的详细信息，请参见*第 21 节：系统窗口看门狗 (WWDG)*。

20.2 IWDG 主要特性

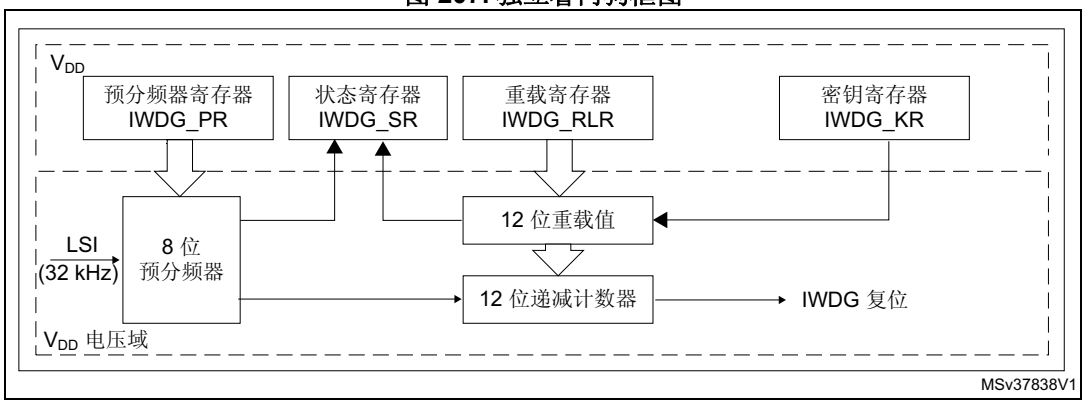
- 自由运行递减计数器
- 时钟由独立 RC 振荡器提供（可在待机和停止模式下运行）
- 复位条件
  - 当递减计数器值小于 0x000 时复位（如果看门狗已激活）
  - 在窗口之外重载递减计数器时复位（如果看门狗已激活）

20.3 IWDG 功能描述

20.3.1 IWDG 框图

图 207 给出了独立看门狗模块的功能框图。

图 207. 独立看门狗框图



1. 寄存器接口位于 V_{DD} 电压域。看门狗功能位于 V_{DD} 电压域，在停止模式和待机模式下仍能工作。

当通过对 *IWDG 键寄存器 (IWDG_KR)* 写入值 0x0000 CCCC 启动独立看门狗时，计数器开始从复位值 0xFFF 递减计数。当计数器计数到终值 (0x000) 时会产生一个复位信号（IWDG 复位）。

任何时候将键值 0x0000 AAAA 写到 *IWDG 键寄存器 (IWDG_KR)* 中，IWDG_RLR 的值就会被重载到计数器，从而避免产生看门狗复位。

一旦运行，IWDG 便无法停止。

20.3.2 窗口选项

通过在 *IWDG 窗口寄存器 (IWDG_WINR)* 中设置合适的窗口，IWDG 也可以用作窗口看门狗。

当计数器值大于 *IWDG 窗口寄存器 (IWDG_WINR)* 中存储的值时，如果执行重载操作，则会产生复位。

IWDG 窗口寄存器 (IWDG_WINR) 的默认值为 0x0000 0FFF，因此，如果不更新此默认值，将禁止窗口选项。

窗口值一经更改，便执行重载操作，以便将递减计数器复位为 *IWDG 重载寄存器 (IWDG_RLR)* 值，并方便计算周期数以生成下一次重载。

使能窗口选项时配置 IWDG

1. 通过在 *IWDG 键寄存器 (IWDG_KR)* 中写入 0x0000 CCCC 来使能 IWDG。
2. 通过在 *IWDG 键寄存器 (IWDG_KR)* 中写入 0x0000 5555 来使能寄存器访问。
3. 通过将 *IWDG 预分频器寄存器 (IWDG_PR)* 编程为 0~7 中的数值来配置 IWDG 预分频器。
4. 对 *IWDG 重载寄存器 (IWDG_RLR)* 进行写操作。
5. 等待寄存器更新 (IWDG_SR = 0x0000 0000)。
6. 对 *IWDG 窗口寄存器 (IWDG_WINR)* 进行写操作。这会自动刷新 *IWDG 重载寄存器 (IWDG_RLR)* 中的计数器值。

注意： *当 IWDG 状态寄存器 (IWDG_SR) 设置为 0x0000 0000 时，写入窗口值允许通过 RLR 刷新计数器值。*

禁止窗口选项时配置 IWDG

不使用窗口选项时，可按以下步骤配置 IWDG：

1. 通过在 *IWDG 键寄存器 (IWDG_KR)* 中写入 0x0000 CCCC 来使能 IWDG。
2. 通过在 *IWDG 键寄存器 (IWDG_KR)* 中写入 0x0000 5555 来使能寄存器访问。
3. 通过将 *IWDG 预分频器寄存器 (IWDG_PR)* 编程为 0~7 中的数值来配置预分频器。
4. 对 *IWDG 重载寄存器 (IWDG_RLR)* 进行写操作。
5. 等待寄存器更新 (IWDG_SR = 0x0000 0000)。
6. 刷新计数器值为 IWDG_RLR 的值 (IWDG_KR = 0x0000 AAAA)

20.3.3 硬件看门狗

如果通过器件选项位使能“硬件看门狗”功能，上电时将自动使能看门狗；如果在计数器计数结束前，若软件没有向 *IWDG 键寄存器 (IWDG_KR)* 写入相应的值，或者在窗口内部重载了递减计数器，则系统会产生复位。

20.3.4 寄存器访问保护

IWDG 预分频器寄存器 (IWDG_PR)、*IWDG 重载寄存器 (IWDG_RLR)* 和 *IWDG 窗口寄存器 (IWDG_WINR)* 寄存器具有写访问保护。若要对其进行修改，用户必须首先对 *IWDG 键寄存器 (IWDG_KR)* 写入键值 0x0000 5555。而写入其他值则会破坏该序列，从而使寄存器访问保护再次生效。这表示重载操作（即写入 0x0000 AAAA）也会启动写保护功能。

状态寄存器指示预分频值、递减计数器重载值或窗口值是否正在被更新。

20.3.5 调试模式

当器件进入调试模式时（内核停止），IWDG 计数器会根据 DBGMCU 冻结寄存器中相应位的配置选择继续正常工作或者停止工作。

20.4 IWDG 寄存器

有关寄存器说明中使用的缩写，请参见*第 33 页的第 1.2 节*。

外设寄存器可按半字（16 位）或字（32 位）访问。

20.4.1 IWDG 键寄存器 (IWDG_KR)

IWDG key register

偏移地址：0x00

复位值：0x0000 0000（通过待机模式复位）

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

位 31:16 保留，必须保持复位值。

位 15:0 **KEY[15:0]**：键值 (Key value)（只能写，读为 0x0000）

必须每隔一段时间通过软件对这些位写入键值 0xAAAA，否则当计数器计数到 0 时，看门狗会产生复位。

写入键值 0x5555 可使能对 IWDG_PR、IWDG_RLR 和 IWDG_WINR 寄存器的访问（请参见*第 20.3.4 节：寄存器访问保护*）

写入键值 0xCCCC 可启动看门狗（选中硬件看门狗选项的情况除外）

20.4.2 IWDG 预分频器寄存器 (IWDG_PR)

IWDG prescaler register

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

位 31:3 保留，必须保持复位值。

位 2:0 **PR[2:0]**：预分频器 (Prescaler divider)

这些位受写访问保护，请参见*第 20.3.4 节：寄存器访问保护*。通过软件设置这些位来选择计数器时钟的预分频因子。若要更改预分频器的分频系数，*IWDG 状态寄存器 (IWDG_SR)* 的 PVU 位必须为 0。

000：4 分频
001：8 分频
010：16 分频
011：32 分频
100：64 分频
101：128 分频
110：256 分频
111：256 分频

注意：读取该寄存器会返回 V_{DD} 电压域的预分频器值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的/有效的。因此，只有在 IWDG 状态寄存器 (IWDG_SR) 中的 PVU 位为 0 时，从寄存器读取的值才有效。

20.4.3 IWDG 重载寄存器 (IWDG_RLR)

IWDG reload register

偏移地址：0x08

复位值：0x0000 0FFF（待机模式时复位）

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

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

位 31:12 保留，必须保持复位值。

位 11:0 **RL[11:0]**：看门狗计数器重载值 (Watchdog counter reload value)

这些位受写访问保护，请参见*寄存器访问保护*。这个值由软件设置，每次对 *IWDG 键寄存器 (IWDG_KR)* 写入值 0xAAAA 时，这个值就会重装载到看门狗计数器中。之后，看门狗计数器便从该装载的值开始递减计数。超时周期由该值和时钟预分频器共同决定。有关超时信息，请参见数据手册。

若要更改重载值，*IWDG 状态寄存器 (IWDG_SR)* 中的 RVU 位必须为 0。

注意：读取该寄存器会返回 V_{DD} 电压域的重载值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的/有效的。因此，只有在 IWDG 状态寄存器 (IWDG_SR) 中的 RVU 位为 0 时，从寄存器读取的值才有效。

20.4.4 IWDG 状态寄存器 (IWDG_SR)

IWDG status register

偏移地址：0x0C

复位值：0x0000 0000（待机模式时不复位）

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

位 31:3 保留，必须保持复位值。

位 2 **WVU**：看门狗计数器窗口值更新 (Watchdog counter window value update)

该位由硬件置 1 以指示窗口值正在更新。当在 V_{DD} 电压域下完成重载值更新操作后（需要多达 5 个 LSI 周期），会通过硬件将该位复位。

窗口值只有在 WVU 位为 0 时才可更新。

位 1 **RVU**：看门狗计数器重载值更新 (Watchdog counter reload value update)

该位由硬件置 1 以指示重载值正在更新。当在 V_{DD} 电压域下完成重载值更新操作后（需要多达 5 个 LSI 周期），会通过硬件将该位复位。

重载值只有在 RVU 位为 0 时才可更新。

位 0 **PVU**：看门狗预分频器值更新 (Watchdog prescaler value update)

该位由硬件置 1 以指示预分频器值正在更新。当在 V_{DD} 电压域下完成预分频器值更新操作后（需要多达 5 个 LSI 周期），会通过硬件将该位复位。

预分频器值只有在 PVU 位为 0 时才可更新。

注意： *如果应用使用多个重载值、预分频器值或窗口值，则必须等到 RVU 位被复位后才能更改重载值，等到 PVU 位被复位后才能更改预分频器值，而且必须等到 WVU 位被复位后才能更改窗口值。但是，在更新预分频器和 / 或重载 / 窗口值之后，则无需等到 RVU、PVU 或 WVU 复位后再继续执行代码（进入低功耗模式时除外）。*

20.4.5 IWDG 窗口寄存器 (IWDG_WINR)

IWDG window register

偏移地址：0x10

复位值：0x0000 0FFF（待机模式时复位）

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

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

位 31:12 保留，必须保持复位值。

位 11:0 **WIN[11:0]**：看门狗计数器窗口值 (Watchdog counter window value)

这些位受写访问保护，请参见*第 20.3.4 节*，它们包含用于与递减计数器进行比较的窗口值上限。

为防止发生复位，当递减计数器的值低于窗口寄存器值且大于 0x0 时必须重载。

若要更改重载值，*IWDG 状态寄存器 (IWDG_SR)* 中的 WVU 位必须为 0。

注意：读取该寄存器会返回 V_{DD} 电压域的重载值。如果正在对该寄存器执行写操作，则读取的值可能无效。因此，只有在 IWDG 状态寄存器 (IWDG_SR) 中的 WVU 位为 0 时，从寄存器读取的值才有效。

20.4.6 IWDG 寄存器映射

下表提供了 IWDG 寄存器映射和复位值。

表 77. IWDG 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**IWDG_KR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**IWDG_PR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[2:0]		
	Reset value																														0	0	0
0x08	**IWDG_RLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RL[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x0C	**IWDG_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WVU	RVU	PVU
	Reset value																														0	0	0
0x10	**IWDG_WINR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WIN[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

21 系统窗口看门狗 (WWDG)

21.1 简介

系统窗口看门狗 (WWDG) 通常用来监测特定软件故障，这些软件故障来源于外部干扰或错误的逻辑条件造成的应用程序运行序列异常。

除非程序在 T6 位清零前刷新递减计数器的值，否则看门狗电路在达到预置的时间周期时，会产生一个 MCU 复位。如果在递减计数器达到窗口寄存器值之前刷新控制寄存器中的 7 位递减计数器值，看门狗电路也会产生一个 MCU 复位。这意味着必须在限定的时间窗口内刷新计数器。

WWDG 时钟由 APB 时钟经预分频后提供，通过可配置的时间窗口来检测应用程序非正常的过迟或过早的操作。

WWDG 最适合那些要求看门狗在精确计时窗口起作用的应用。

21.2 WWDG 主要特性

- 可编程的自由运行递减计数器
- 复位条件
  - 当递减计数器值小于 0x40 时复位（如果看门狗已激活）
  - 在窗口之外重载递减计数器时复位（如果看门狗已激活）（请参见*图 209*）
- 提前唤醒中断 (EWI)：当递减计数器等于 0x40 时触发（如果已使能且看门狗已激活）

21.3 WWDG 功能描述

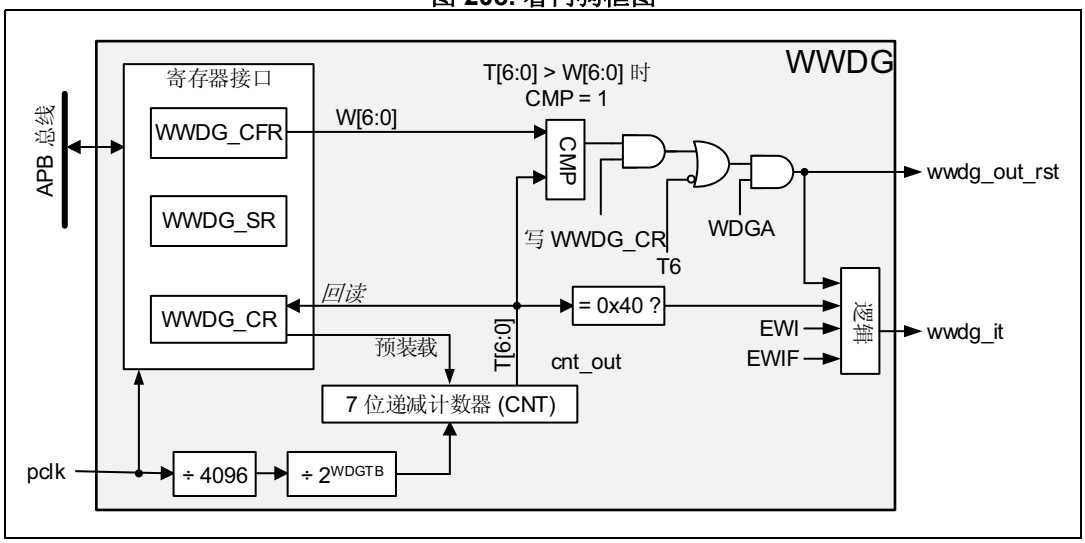
如果激活看门狗（WWDG_CR 寄存器中的 WDGA 位置 1），则当 7 位递减计数器（T[6:0] 位）从 0x40 递减到 0x3F（T6 已清零）时会引发复位。如果软件在重载计数器时，计数器的值大于窗口寄存器，则会产生复位。

应用程序在正常运行过程中必须定期地写入 WWDG_CR 寄存器以防止 MCU 发生复位。只有当计数器值低于窗口寄存器值且高于 0x3F 时，才能执行此操作。存储在 WWDG_CR 寄存器中的值必须介于 0xFF 和 0xC0 之间。

请参见*图 208* 了解 WWDG 框图。

21.3.1 WWDG 框图

图 208. 看门狗框图



21.3.2 使能看门狗

当用户选项 WWDG_SW 选择“软件窗口看门狗”时，看门狗在复位后总处于关闭状态。可通过设置 WWDG_CR 寄存器中的 WDGA 位来使能看门狗，之后除非执行复位操作，否则不能再次关闭。

当用户选项 WWDG_SW 选择“硬件窗口看门狗”时，看门狗在复位后始终使能，无法禁止。

21.3.3 控制递减计数器

递减计数器处于自由运行状态，即使禁止看门狗，递减计数器仍继续递减计数。当使能看门狗时，必须将 T6 位置 1，以防止立即复位。

T[5:0] 位包含了看门狗产生复位之前的计时数目；复位前的延时时间在一个最小值和一个最大值之间变化，这是因为写入 WWDG_CR 寄存器时，预分频器的状态是未知的（请参见 *图 209*）。*WWDG 配置寄存器 (WWDG_CFR)* 包含窗口的上限：为防止发生复位，当递减计数器的值低于或等于窗口寄存器值且大于 0x3F 时必须重载。*图 209* 介绍了窗口看门狗的工作过程。

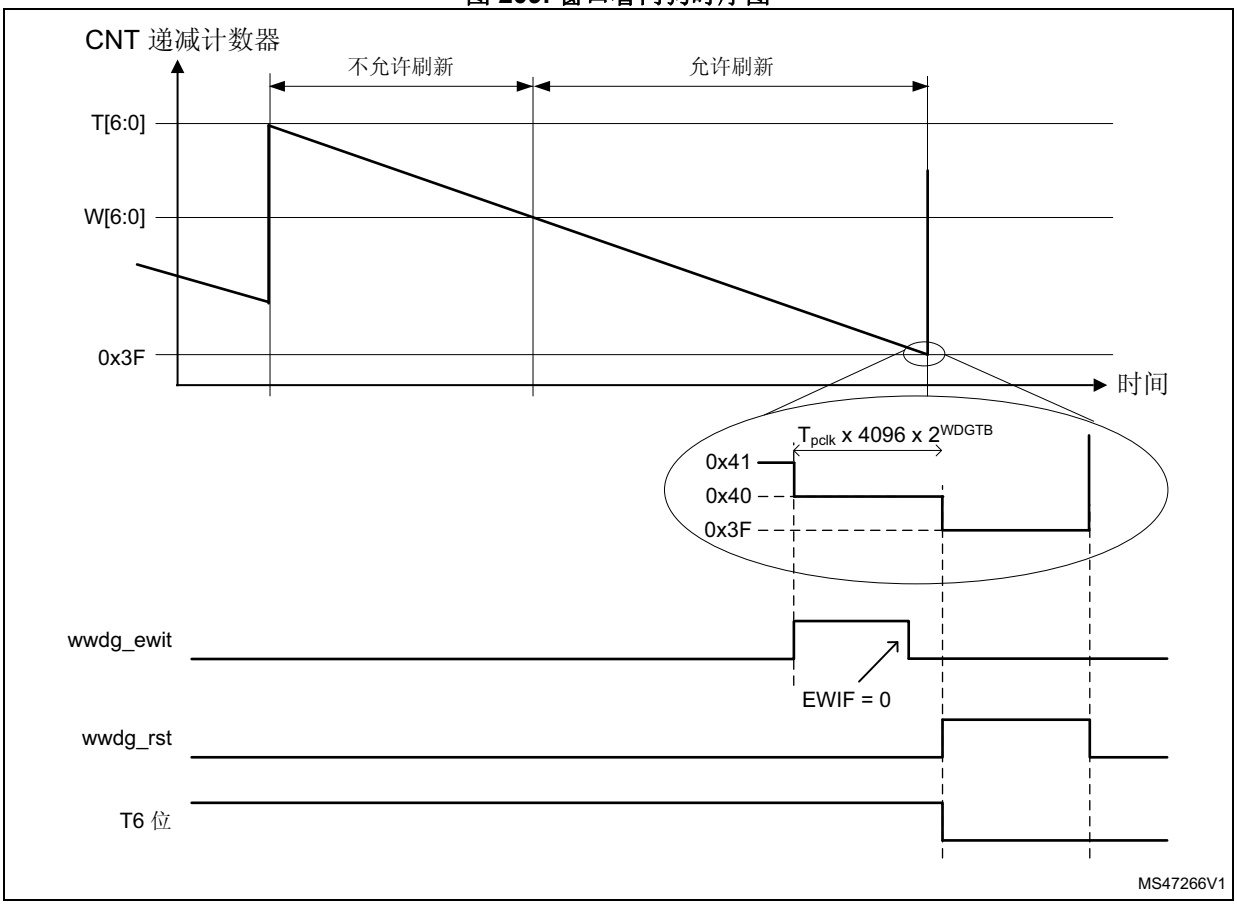
注意： *可使用 T6 位产生软件复位（将 WDGA 位置 1 并将 T6 位清零）。*

21.3.4 如何设置看门狗超时

使用 *图 209* 中的公式来计算 WWDG 超时。

警告： **写入 WWDG_CR 寄存器时，始终将 1 写入 T6 位，以避免生成立即复位。**

图 209. 窗口看门狗时序图



超时值的计算公式如下：

$$t_{WWDG} = t_{PCLK} \times 4096 \times 2^{WDGTB[1:0]} \times (T[5:0] + 1) \quad (ms)$$

其中：

t_{WWDG}：WWDG 超时

T_{PCLK}：APB 时钟周期，以 ms 为测量单位

4096：对应于内部分频器的值

例如，如果 APB 频率为 48 MHz，将 WDGTB[1:0] 设置为 3 并将 T[5:0] 设置为 63：

$$t_{WWDG} = (1/48000) \times 4096 \times 2^{3} \times (63 + 1) = 43.69ms$$

有关 t_{WWDG} 的最小值和最大值，请参见数据手册。

21.3.5 调试模式

当器件进入调试模式时（内核停止），WWDG 计数器会根据 DBG 模块中的配置位选择继续正常工作或者停止工作。有关详细信息，请参见*第 26 节：调试支持 (DBG)*。

21.4 WWDG 中断

如果在产生复位之前必须执行特定的安全操作或数据记录，则可使用提前唤醒中断 (EWI)。要使能提前唤醒中断，应用程序必须：

- 向 WWDG_SR 寄存器的 EWIF 位写入 0，以清除意外的挂起中断
- 向 WWDG_CFR 寄存器的 EWI 位写入 1，以使能中断

当递减计数器的值为 0x40 时，将生成看门狗中断。在复位器件之前，可以使用相应的中断服务程序 (ISR) 来触发特定操作（例如通信或数据记录）。

在某些应用中，可以使用 EWI 中断来管理软件系统检查和/或系统恢复/功能退化，而不会生成 WWDG 复位。在这种情况下，相应的 ISR 必须重载 WWDG 计数器以避免 WWDG 复位，然后再触发所需操作。

通过将 0 写入 WWDG_SR 寄存器中的 EWIF 位来清除看门狗中断。

注意： *当由于在更高优先级任务中有系统锁死等原因而无法使用看门狗中断时，最终会产生 WWDG 复位。*

21.5 WWDG 寄存器

有关寄存器说明中使用的缩写，请参见*第 33 页的第 1.2 节*。

外设寄存器可支持半字（16 位）或字（32 位）访问。

21.5.1 WWDG 控制寄存器 (WWDG_CR)

WWDG control register

偏移地址：0x000

复位值：0x0000 007F

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

位 31:8 保留，必须保持复位值。

位 7 **WDGA**：激活位 (Activation bit)
该位由软件置 1，只有复位后才由硬件清零。当 WDGA = 1 时，看门狗可产生复位。
0：禁止看门狗
1：使能看门狗

位 6:0 **T[6:0]**：7 位计数器（MSB 到 LSB）
这些位用来存储看门狗计数器的值，每隔 $(4096 \times 2^{WDGTB[1:0]})$ PCLK 个周期递减一次。当它从 0x40 递减到 0x3F（T6 清零）时会产生复位。

21.5.2 WWDG 配置寄存器 (WWDG_CFR)

WWDG configuration register

偏移地址：0x004

复位值：0x0000 007F

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	WDGTB[2:0]			Res.	EWI	Res.	Res.	W[6:0]						
		rw	rw	rw		rs			rw	rw	rw	rw	rw	rw	rw

位 31:14 保留，必须保持复位值。

位 13:11 **WDGTB[2:0]**：定时器时基 (Timer base)

可按如下方式修改预分频器的时基：

000：CK 计数器时钟 (PCLK div 4096) 分频器 1

001：CK 计数器时钟 (PCLK div 4096) 分频器 2

010：CK 计数器时钟 (PCLK div 4096) 分频器 4

011：CK 计数器时钟 (PCLK div 4096) 分频器 8

100：CK 计数器时钟 (PCLK div 4096) 分频器 16

101：CK 计数器时钟 (PCLK div 4096) 分频器 32

110：CK 计数器时钟 (PCLK div 4096) 分频器 64

111：CK 计数器时钟 (PCLK div 4096) 分频器 128

位 10 保留，必须保持复位值。

位 9 **EWI**：提前唤醒中断使能 (Early wake-up interrupt enable)

由软件置 1，只有复位后才由硬件清零。置 1 后，只要计数器值达到 0x40 就会产生中断。

位 8:7 保留，必须保持复位值。

位 6:0 **W[6:0]**：7 位窗口值 (7-bit window value)

这些位包含用于与递减计数器进行比较的窗口值。

21.5.3 WWDG 状态寄存器 (WWDG_SR)

WWDG status register

偏移地址：0x008

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF
															rc_w0

位 31:1 保留，必须保持复位值。

位 0 **EWIF**：提前唤醒中断标志 (Early wake-up interrupt flag)

当计数器值达到 0x40 时，该位由硬件置 1。它必须由软件通过写入 0 来清零。写入 1 无效。如果不使能中断，该位也会被置 1。

21.5.4 WWDG 寄存器映射

下表提供了 WWDG 寄存器映射和复位值。

表 78. WWDG 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	WWDG_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGA	T[6:0]						
	Reset value																									0	1	1	1	1	1	1	1
0x004	WWDG_CFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGTB [2:0]			Res.	EWI	Res.	Res.	W[6:0]						
	Reset value																			0	0	0		0			1	1	1	1	1	1	1
0x008	WWDG_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF
	Reset value																																0

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

22 实时时钟 (RTC)

22.1 简介

实时时钟 (RTC) 提供用于管理所有低功耗模式的自动唤醒单元。

实时时钟 (RTC) 是一个独立的 BCD 定时器/计数器。RTC 提供具有可编程闹钟中断功能的日历时钟/日历。

无论器件状态如何（运行模式、低功耗模式或处于复位状态），只要电源电压保持在工作范围内，RTC 便不会停止工作。

22.2 RTC 主要特性

RTC 支持以下特性（请参见*图 210：RTC 框图*）：

- 日历具有亚秒、秒、分、小时（12 小时制或 24 小时制）、星期几、日、月、年，格式为 BCD（二进码十进数）。
- 自动调整每月是 28、29（闰年）、30 还是 31 天。
- 一个可编程闹钟。
- 可运行时纠正 1 到 32767 个 RTC 时钟脉冲。这可用于与主时钟同步。
- 参考时钟检测：可使用更加精确的第二时钟源（50 或 60 Hz）来提高日历的精确度。
- 数字校准电路具有 0.95 ppm 的分辨率，以补偿石英晶振的不准确性。
- 时间戳特性可用于保存日历内容。此功能可由时间戳引脚上的事件触发。

RTC 时钟源可为：

- 32.768 kHz 外部晶振 (LSE)
- 外部谐振器或振荡器 (LSE)
- 内部低功耗 RC 振荡器（LSI，典型频率为 32 kHz）
- 高速外部时钟 (HSE)，由 RCC 中的预分频器进行分频

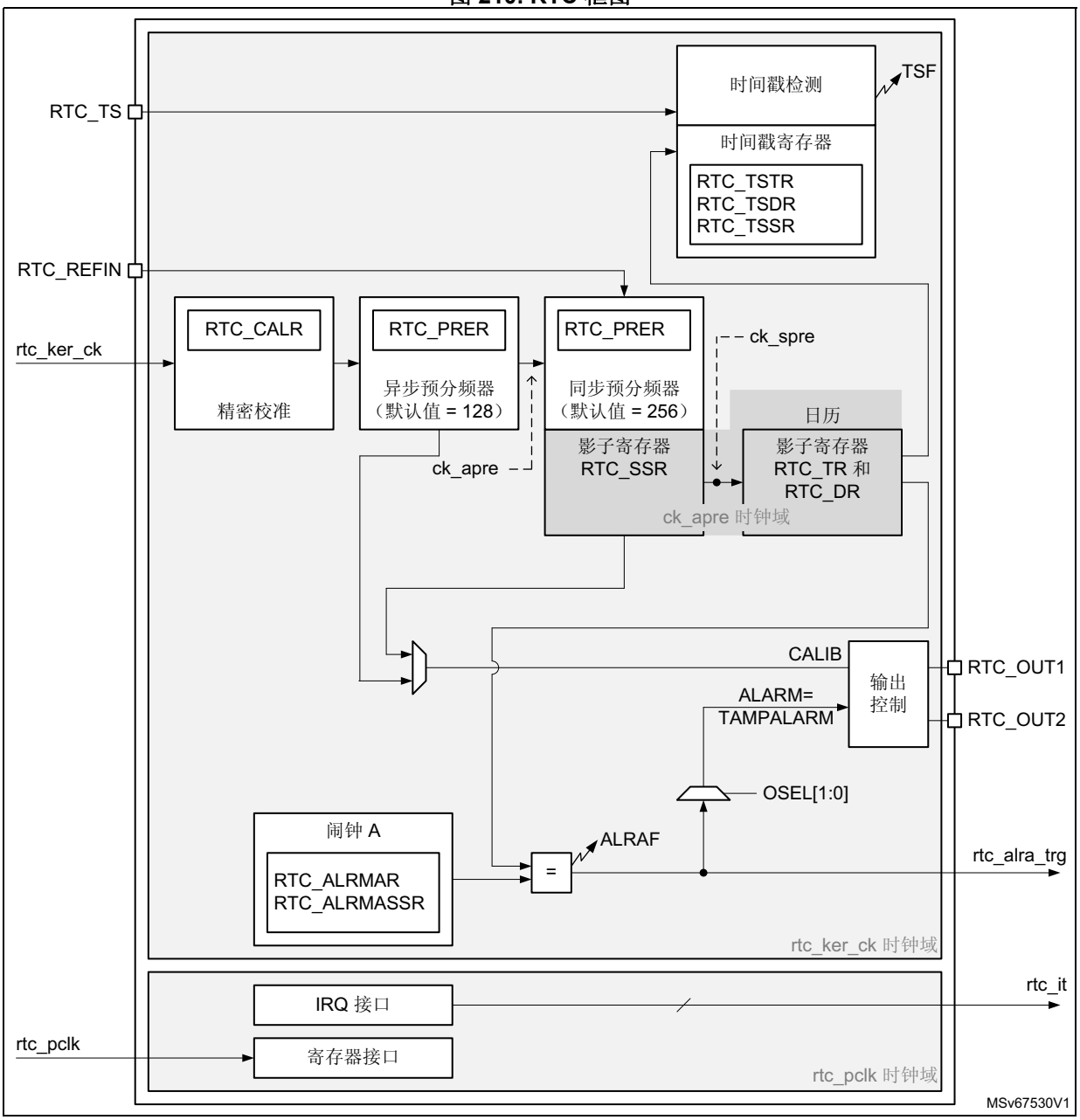
通过 LSE 或 LSI 提供时钟时，RTC 可以在除待机模式和关断模式之外的所有低功耗模式下正常工作。

所有 RTC 事件（闹钟、时间戳）都可以产生中断，将器件从低功耗模式唤醒。

22.3 RTC 功能描述

22.3.1 RTC 框图

图 210. RTC 框图



22.3.2 RTC 引脚和内部信号

表 79. RTC 输入/输出引脚

引脚名称	信号类型	说明
RTC_TS	输入	RTC 时间戳输入
RTC_REFIN	输入	RTC 50 Hz 或 60 Hz 参考时钟输入
RTC_OUT1	输出	RTC 输出 1
RTC_OUT2	输出	RTC 输出 2

- RTC_OUT1 和 RTC_OUT2 可选择以下两个输出之一：
  - CALIB：512 Hz 或 1 Hz 时钟输出（LSE 频率为 32.768 kHz）。可通过将 RTC_CR 寄存器中的 COE 位置 1 来使能此输出。
  - TAMPALRM：此输出为 ALARM 输出。

可通过配置 RTC_CR 寄存器中的 OSEL[1:0] 位使能 ALARM，可选择闹钟 A 输出。

表 80. RTC 内部输入/输出信号

内部信号名称	信号类型	说明
rtc_ker_ck	输入	RTC 内核时钟，在本文档中也称为 RTCCLK
rtc_pclk	输入	RTC APB 时钟
rtc_it	输出	RTC 中断（有关详细信息，请参见*第 22.5 节：RTC 中断*）
rtc_alra_trg	输出	RTC 闹钟 A 事件检测触发

RTC 内核时钟通常是 32.768 kHz 的 LSE，但也可以在 RCC 中选择其他时钟源（更多详细信息，请参见 RCC）。

触发输出可以用作其他外设的触发。

22.3.3 RTC 控制的 GPIO

RTC_OUT1 和 RTC_TS 映射到同一引脚。

此引脚输出机制遵循*表 81* 中所示的优先级顺序。

表 81. 引脚配置(1)

引脚功能		OSEL[1:0]（ALARM 输出使能）	COE（CALIB 输出使能）	OUT2EN	TAMPALRM_TYPE	TAMPALRM_PU	TSE（RTC_TS 输入使能）
TAMPALRM 输出推挽		01	无关	无关	0	0	无关
TAMPALRM 输出开漏(2)	无上下拉	01	无关	无关	1	0	无关
	内部上拉	01	无关	无关	1	1	无关
CALIB 输出 PP		00	1	0	无关	无关	无关
RTC_TS 输入浮空		00	0	无关	无关	无关	1
		00	1	1			
		无关	0				
唤醒引脚或标准 GPIO		00	0	无关	无关	无关	0
		00	1	1			
		无关	0				

1. OD：开漏；PP：推挽。
2. 在此配置下，GPIO 必须配置为输入。

此外，可借助 OUT2EN 位输出 RTC_OUT2。可以将不同的功能映射到 RTC_OUT1 或 RTC_OUT2 上，具体取决于 OSEL、COE 和 OUT2EN 的配置，如*表 82* 所示。

表 82. RTC_OUT 映射

OSEL[1:0] 位 ALARM 输出使能）	COE 位（CALIB 输出使能）	OUT2EN 位	RTC_OUT1	RTC_OUT2
00	0	0	-	-
00	1		CALIB	-
01 或 10 或 11	无关		TAMPALRM	-
00	0	1	-	-
00	1		-	CALIB
01 或 10 或 11	0		-	TAMPALRM
01 或 10 或 11	1		TAMPALRM	CALIB

22.3.4 时钟和预分频器

RTC 时钟源 (RTCCLK) 通过时钟控制器从 LSE 时钟、LSI 振荡器时钟以及 HSE 时钟三者中选择。有关 RTC 时钟源配置的更多信息，请参见*第 5 节：复位和时钟控制 (RCC)*。

可编程的预分频器阶段可生成 1 Hz 的时钟，用于更新日历。为最大程度地降低功耗，预分频器分为 2 个可编程的预分频器（参见*图 210：RTC 框图*）：

- 一个通过 RTC_PRER 寄存器的 PREDIV_A 位配置的 7 位异步预分频器。
- 一个通过 RTC_PRER 寄存器的 PREDIV_S 位配置的 15 位同步预分频器。

注意： *使用两个预分频器时，推荐将异步预分频器配置为较高的值，以最大程度降低功耗。*

要使用频率为 32.768 kHz 的 LSE 获得频率为 1 Hz 的内部时钟 (ck_spre)，需要将异步预分频系数设置为 128，并将同步预分频系数设置为 256。

分频系数的最小值为 1，最大值为 2^{22}。

这对应于约为 4 MHz 的最大输入频率。

f_{ck_apre} 可根据以下公式得出：

$$f_{CK\_APRE} = \frac{f_{RTCCLK}}{PREDIV\_A + 1}$$

ck_apre 时钟用于为二进制 RTC_SSR 亚秒递减计数器提供时钟。当该计数器计数到 0 时，会使用 PREDIV_S 的内容重载 RTC_SSR。

f_{ck_spre} 可根据以下公式得出：

$$f_{CK\_SPRE} = \frac{f_{RTCCLK}}{(PREDIV\_S + 1) \times (PREDIV\_A + 1)}$$

22.3.5 实时时钟和日历

RTC 日历时间和日期寄存器可通过与 PCLK（APB 时钟）同步的影子寄存器来访问。这些时间和日期寄存器也可以直接访问，无同步时间。

- RTC_SSR 对应于亚秒
- RTC_TR 对应于时间
- RTC_DR 对应于日期

每个 RTCCLK 周期，都将当前日历值复制到影子寄存器，并将 RTC_ICSR 寄存器的 RSF 位置 1（请参见*第 22.6.9 节：RTC 平移控制寄存器 (RTC_SHIFTR)*）。在停止和待机模式下不会执行复制操作。退出这两种模式时，影子寄存器会在最长 4 个 RTCCLK 周期后进行更新。

当应用读取日历寄存器时，它会访问影子寄存器的内容。也可以通过将 RTC_CR 寄存器的 BYPSHAD 控制位置 1 来直接访问日历寄存器。默认情况下，该位被清零，用户访问影子寄存器。

在 BYPSHAD=0 模式下读取 RTC_SSR、RTC_TR 或 RTC_DR 寄存器时，APB 时钟频率 (f_{APB}) 必须至少为 RTC 时钟频率 (f_{RTCCLK}) 的 7 倍。

影子寄存器通过系统复位来复位。

22.3.6 可编程闹钟

RTC 单元提供可编程闹钟：闹钟 A。

可通过 RTC_CR 寄存器中的 ALRAE 位来使能可编程闹钟功能。

如果日历亚秒、秒、分钟、小时、日期或日分别与闹钟寄存器 RTC_ALRMASSR 和 RTC_ALRMAR 中编程的值相匹配，则 ALRAF 标志会被置为 1。可通过 RTC_ALRMAR 寄存器的 MSKx 位以及 RTC_ALRMASSR 寄存器的 MASKSSx 位单独选择各日历字段。

可通过 RTC_CR 寄存器中的 ALRAIE 位来使能闹钟中断。

小心： 如果选择秒字段（RTC_ALRMAR 中的 MSK1 位复位），则 RTC_PRER 寄存器中设置的同步预分频器分频系数必须至少为 3，才能确保闹钟正确地运行。

闹钟 A（如果已通过 RTC_CR 寄存器中的位 OSEL[1:0] 使能）可连接到 TAMPALRM 输出。可通过 RTC_CR 寄存器的 POL 位配置 TAMPALRM 输出极性。

22.3.7 RTC 初始化和配置

RTC 寄存器访问

RTC 寄存器为 32 位寄存器。除了当 BYPSHAD=0 时对日历影子寄存器执行的读访问之外，APB 接口会在访问 RTC 寄存器时引入 2 个等待周期。

RTC 寄存器写保护

上电复位后，部分 RTC 寄存器受到写保护。

通过向写保护寄存器 (RTC_WPR) 写入一个键值来使能对受保护 RTC 寄存器的写操作。

要解锁 RTC 寄存器的写保护，需要执行以下步骤。

1. 将 0xCA 写入 RTC_WPR 寄存器。
2. 将 0x53 写入 RTC_WPR 寄存器。

写入一个错误的关键字会再次激活写保护。

保护机制不受系统复位影响。

日历初始化和配置

要编程包括时间格式和预分频器配置在内的初始时间和日期日历值，需按照以下顺序操作：

1. 将 RTC_ICSR 寄存器中的 INIT 位置 1 以进入初始化模式。在此模式下，日历计数器将停止工作并且其值可更新。
2. 轮询 RTC_ICSR 寄存器中的 INITF 位。当 INITF 置 1 时进入初始化阶段模式。大约需要 2 个 RTCCLK 时钟周期（由于时钟同步）。
3. 要为日历计数器生成 1 Hz 时钟，应编程 RTC_PRER 寄存器中的两个预分频系数。
4. 在影子寄存器（RTC_TR 和 RTC_DR）中加载初始时间和日期值，然后通过 RTC_CR 寄存器中的 FMT 位配置时间格式（12 或 24 小时制）。
5. 通过清零 INIT 位退出初始化模式。随后，自动加载实际日历计数器值，在 4 个 RTCCLK 时钟周期后重新开始计数。

当初始化序列完成之后，日历开始计数。

注意： *系统复位后，应用可读取 RTC_ICSR 寄存器中的 INITS 标志，以检查日历是否已初始化。如果该标志为 0，表明自系统复位以来，日历还尚未初始化过，其年份字段一直还保持着上电复位默认值 (0x00)。*

要在初始化之后读取日历，必须首先用软件检查 RTC_ICSR 寄存器的 RSF 标志是否置 1。

夏令时

可通过 RTC_CR 寄存器的 SUB1H、ADD1H 和 BKP 位管理夏令时。

利用 SUB1H 或 ADD1H，软件只需单次操作便可在日历中减去或增加一个小时，无需执行整个初始化步骤。

此外，软件还可以使用 BKP 位来记录是否曾经执行过此操作。

编程闹钟

要对可编程的闹钟进行编程或更新，必须执行类似的步骤。下面给出的步骤针对闹钟 A。

1. 将 RTC_CR 中的 ALRAE 位清零以禁止闹钟 A。
2. 编程闹钟 A 寄存器 (RTC_ALRMASSR/RTC_ALRMAR)。
3. 将 RTC_CR 寄存器中的 ALRAE 位置 1 以再次使能闹钟 A。

注意： *由于时钟同步的缘故，RTC_CR 寄存器的每次更改都将需要大约 2 个 RTCCLK 时钟周期来完成。*

22.3.8 读取日历

当 RTC_CR 寄存器中的 BYPSHAD 控制位清零时

要正确读取 RTC 日历寄存器（RTC_SSR、RTC_TR 和 RTC_DR），APB1 时钟频率 (f_{PCLK}) 必须等于或大于 RTC 时钟频率 (f_{RTCCLK}) 的七倍。这可以确保同步机制的安全性。

如果 APB1 时钟频率低于 RTC 时钟频率的七倍，则软件必须分两次读取日历时间寄存器和日期寄存器。这样，当两次读取的 RTC_TR 结果相同时，才能确保数据正确。否则必须执行第三次读访问。任何情况下，APB1 的时钟频率都不能低于 RTC 的时钟频率。

每次将日历寄存器中的值复制到 RTC_SSR、RTC_TR 和 RTC_DR 影子寄存器时，RTC_ICSR 寄存器中的 RSF 位都会置 1。每个 RTCCLK 周期执行一次复制。为确保这 3 个值一致，读取 RTC_SSR 或 RTC_TR 时会锁定高阶日历影子寄存器中的值，直到读取 RTC_DR。为避免软件对日历执行读访问的时间间隔小于 1 个 RTCCLK 周期：第一次读取日历之后必须通过软件将 RSF 清零，并且软件必须等待到 RSF 置 1 之后才可再次读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

从低功耗模式（停止模式或待机模式）唤醒之后，必须通过软件将 RSF 清零。之后，软件必须等待至 RSF 再次置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

RSF 位必须在唤醒之后而不是进入低功耗模式之前进行清零。

系统复位之后，软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。实际上，系统复位会将影子寄存器复位为其默认值。

初始化之后（请参见*第 519 页的日历初始化和配置*），软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

同步之后（请参见*第 22.3.10 节：RTC 同步*），软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

当 RTC_CR 寄存器中的 BYPSHAD 控制位置 1 时（旁路影子寄存器）

读取日历寄存器时会直接从日历计数器获取值，这样便无需等待至 RSF 位置 1。这对于从低功耗模式（停止模式或待机模式）退出后的情况特别有用，因为影子寄存器在这些模式下不更新。

当 BYPSHAD 位置 1 时，如果在对寄存器的两次读访问之间出现 RTCCLK 沿，则不同寄存器的结果彼此可能不一致。此外，如果在读操作期间出现 RTCCLK 沿，则可能导致其中一个寄存器的值不正确。软件必须分两次读取所有寄存器，然后将两次结果加以比较来确认数据是否一致和正确。此外，软件也可以只比较两次读取日历寄存器得到的结果的最低位。

注意： *当 BYPSHAD = 1 时，读取日历寄存器的指令需要一个额外的 APB 周期才能完成。*

22.3.9 复位 RTC

日历影子寄存器（RTC_SSR、RTC_TR 和 RTC_DR）以及 RTC 状态寄存器 (RTC_ICSR) 的某些位通过所有可用的系统复位源复位为各自的默认值。

而以下寄存器则通过上电复位来复位为各自的默认值并且不受系统复位的影响：RTC 当前日历寄存器、RTC 控制寄存器 (RTC_CR)、预分频器寄存器 (RTC_PRER)、RTC 校准寄存器 (RTC_CALR)、RTC 移位寄存器 (RTC_SHIFTR)、RTC 时间戳寄存器（RTC_TSSSR、RTC_TSTR 和 RTC_TSDR）和闹钟 A 寄存器 (RTC_ALRMASSR/RTC_ALRMAR)。

此外，当由 LSE 提供时钟时，如果复位源并非上电复位源（有关不受系统复位影响的 RTC 时钟源的详细信息，请参见 RCC），则 RTC 将在系统复位时保持运行状态。发生上电复位时，RTC 会停止工作，并且所有 RTC 寄存器都会设置为各自的复位值。

22.3.10 RTC 同步

RTC 可与高精度的远程时钟同步。在读取亚秒字段后（RTC_SSR 或 RTC_TSSSR），即可计算远程时钟的时间与 RTC 之间的精准偏差。之后，可使用 RTC_SHIFTR 对 RTC 的时钟进行零点几秒的“平移”，经过调整后可消除此偏差。

RTC_SSR 包含同步预分频器计数器的值。这样，便可计算分辨率低至 1 / (PREDIV_S + 1) 秒的 RTC 的准确时间。因此，可通过增大同步预分频器的值 (PREDIV_S[14:0]) 来提高分辨率。将 PREDIV_S 设置为 0x7FFF 时，可得到允许的最大分辨率（30.52 µs，时钟频率为 32768 Hz）。

但是，提高 PREDIV_S 意味着必须降低 PREDIV_A 才能将同步预分频器的输出维持在 1 Hz。这样，异步预分频器的输出频率会增大，RTC 的动态功耗也会相应增加。

可以使用 RTC 平移控制寄存器 (RTC_SHIFTR) 对 RTC 进行微调。可以用大小为 1 / (PREDIV_S + 1) 秒的分辨率对 RTC_SHIFTR 进行写操作，将时钟平移（延迟或提前）最长 1 秒。在这种平移操作中，会将 SUBFS[14:0] 值加到同步预分频器计数器 SS[15:0] 中：这将使时钟产生延迟。如果同时将 ADD1S 位置 1，则会增加一秒，与此同时减去的时间为零点几秒，因此将使时钟提前。

小心： 初始化平移操作前，用户必须检查确认 SS[15] = 0，以确保不会发生上溢。

对 RTC_SHIFTR 寄存器执行写操作以启动平移操作时，硬件会将 SHPF 标志置 1 以指示平移操作挂起。完成平移操作时，硬件会将该位清零。

小心： 该同步功能与参考时钟检测功能不兼容：当 REFCKON = 1 时，固件不能对 RTC_SHIFTR 执行写操作。

22.3.11 RTC 参考时钟检测

RTC 日历更新可与参考时钟 RTC_REFIN（通常为市电频率，50 Hz 或 60 Hz）同步。RTC_REFIN 参考时钟的精度应高于 32.768 kHz LSE 时钟。使能 RTC_REFIN 检测时（将 RTC_CR 的 REFCKON 位置 1），日历仍由 LSE 提供时钟，而 RTC_REFIN 用于补偿不准确的日历更新频率 (1 Hz)。

每个 1 Hz 时钟边沿都与最近的 RTC_REFIN 时钟边沿进行比较（如果在给定的时间窗口内发现一个边沿）。在大多数情况下，两个时钟边沿恰好对齐。当 1 Hz 时钟由于 LSE 时钟不精确而发生偏离时，RTC 会稍微偏移 1 Hz 时钟，以便后续的 1 Hz 时钟边沿能够对齐。利用这种机制，可使日历像参考时钟一样精确。

RTC 使用 32.768 kHz 石英产生的 256 Hz 时钟 (ck_apre) 检测是否存在参考时钟源。大约在日历每次更新时（每 1 秒钟），便会在时间窗口期间执行一次检测。检测到第一个参考时钟边沿时，该窗口等于 7 个 ck_apre 周期。随后的日历更新使用长度为 3 个 ck_apre 周期的较小窗口。

每次在窗口中检测到参考时钟时，都会强制输出 ck_spre 时钟的异步预分频器进行重载。当参考时钟与 1 Hz 时钟对齐时，此操作不起作用，因为预分频器会在同一时刻重载。当时钟不对齐时，重载操作会微调后续的 1 Hz 时钟边沿，使其与参考时钟对齐。

如果参考时钟停止（在 3 个 ck_apre 窗口内未出现参考时钟边沿），日历将仅根据 LSE 时钟进行连续更新。RTC 随后使用 ck_spre 边沿上居中的大检测窗口（7 个 ck_apre 周期）等待参考时钟。

使能 RTC_REFIN 检测后，必须将 PREDIV_A 和 PREDIV_S 设置为各自的默认值：

- PREDIV_A = 0x007F
- PREVID_S = 0x00FF

注意： *RTC_REFIN 时钟检测在待机模式下不可用。*

22.3.12 RTC 精密数字校准

RTC 频率可采用约 0.954 ppm 的分辨率进行数字校准，校准范围为 -487.1 ppm 到 +488.5 ppm。使用一系列微调（增加和/或减少单独的 RTCCLK 脉冲）进行频率校正。这些微调的分布非常均匀，因此 RTC 的校准效果相当好，即使在短时间内持续观察也是如此。

当输入频率为 32768 Hz 时，精密数字校准的校准周期约为 2^{20} 个 RTCCLK 脉冲或 32 秒。此周期由一个通过 RTCCLK 提供时钟信号的 20 位计数器 cal_cnt[19:0] 维持。

精密数字校准寄存器 (RTC_CALR) 可指定校准周期内要减少的 RTCCLK 时钟周期数：

- 将位 CALM[0] 置 1 时，校准周期内将只减少一个时钟。
- 将 CALM[1] 置 1 时，将减少两个时钟。
- 将 CALM[2] 置 1 时，将减少四个时钟。
- 依此类推，将 CALM[8] 置 1 时，将减少 256 个时钟。

注意： *CALM[8:0] (RTC_CALR) 可指定校准周期内要减少的 RTCCLK 脉冲数。将位 CALM[0] 置 1 时，校准周期内将只减少 1 个脉冲（当 cal_cnt[19:0] = 0x80000 时）；将 CALM[1] 置 1 时，将减少 2 个周期（当 cal_cnt = 0x40000 和 0xC0000 时）；将 CALM[2] 置 1 时，将减少 4 个周期（当 cal_cnt = 0x20000/0x60000/0xA0000/0xE0000 时）；依此类推，将 CALM[8] 置 1 时，将减少 256 个时钟（当 cal_cnt = 0xXX800 时）。*

使用适当分辨率时，CALM 可使 RTC 频率减少最多 487.1 ppm，而 CALP 可用于使频率增加 488.5 ppm。将 CALP 置 1，可每隔 2^{11} 个 RTCCLK 周期有效插入一个额外的 RTCCLK 脉冲，这意味着每个校准周期内可增加 512 个时钟。

将 CALM 和 CALP 配合使用时，可在校准周期内增加一个范围为 -511 到 +512 RTCCLK 周期的偏差，对应的校准范围为 -487.1 ppm 到 +488.5 ppm，分辨率约为 0.954 ppm。

若输入频率 (F_{RTCCLK}) 已知，可通过以下公式计算有效校准频率 (F_{CAL})：

$$F_{CAL} = F_{RTCCLK} \times [1 + (CALP \times 512 - CALM) / (2^{20} + CALM - CALP \times 512)]$$

PREDIV_A < 3 条件下的校准

当异步预分频器值（RTC_PRER 寄存器中的 PREDIV_A 位）小于 3 时，不能将 CALP 位置 1。如果 CALP 已置 1 并且 PREDIV_A 位的值小于 3，则会忽略 CALP，即假定 CALP 等于 0 而执行校准。

要在 PREDIV_A 小于 3 的条件下执行校准，应降低同步预分频器值 (PREDIV_S) 以便每秒内可加速 8 个 RTCCLK 时钟周期，这意味着每个校准周期可增加 256 个时钟周期。因此，仅使用 CALM 位，可在每个校准周期内有效增加 255 到 256 个时钟脉冲（对应的校准范围为 243.3 ppm 到 244.1 ppm）。

在标称 RTCCLK 频率 32768 Hz 下，当 PREDIV_A 等于 1 时（分频系数为 2），应将 PREDIV_S 设置为 16379 而不是 16383（少 4）。唯一相关的其他情况是，当 PREDIV_A 等于 0 时，应将 PREDIV_S 设置为 32759 而不是 32767（少 8）。

如果以这种方式减少 PREDIV_S，则采用以下公式计算校准输入时钟的有效频率：

$$F_{CAL} = F_{RTCCLK} \times [1 + (256 - CALM) / (2^{20} + CALM - 256)]$$

在这种情况下，如果 RTCCLK 恰好为 32768.00 Hz，则当 CALM[7:0] 等于 0x100 时（CALM 范围的中值），说明设置正确。

验证 RTC 校准

通过测量 RTCCLK 的精确频率，计算正确的 CALM 和 CALP 值以确保 RTC 精度。此外，还为应用提供了一个可选的 1 Hz 输出，用来测量和验证 RTC 精度。

如果在有限的间隔内测量 RTC 的精确频率，则会导致测量期间产生最多 2 个 RTCCLK 时钟周期的测量误差，具体取决于数字校准周期与测量周期的对齐方式。

但是，如果测量周期与校准周期的长度相同，则可以消除此测量误差。在这种情况下，观测到的唯一误差是由数字校准的分辨率导致的误差。

- 默认情况下，校准周期为 32 秒。

在此模式下，测量整个 32 秒内 1 Hz 输出的精度，可确保测量误差在 0.477 ppm 内（32 秒内为 0.5 个 RTCCLK 周期，受校准分辨率限制）。

- 可将 RTC_CALR 寄存器的 CALW16 位置 1，以强制 16 秒的校准周期。

此时，可在 16 秒内测量 RTC 精度，产生的最大误差为 0.954 ppm（16 秒内为 0.5 个 RTCCLK 周期）。但是，由于校准分辨率降低，长期的 RTC 精度也会降到 0.954 ppm：将 CALW16 置 1 时，CALM[0] 位将始终保持为 0。

- 可将 RTC_CALR 寄存器的 CALW8 位置 1，以强制 8 秒的校准周期。

此时，可在 8 秒内测量 RTC 精度，产生的最大误差为 1.907 ppm（8 秒内为 0.5 个 RTCCLK 周期）。长期的 RTC 精度也会降到 1.907 ppm：将 CALW8 置 1 时，CALM[1:0] 位将始终保持为 00。

动态重校准

当 RTC_ICSR/INITF = 0 时，可动态更新校准寄存器 (RTC_CALR)，具体步骤如下：

1. 轮询 RTC_ICSR/RECALPF（重新校准挂起标志）。
2. 如果该标志为 0，则可以根据需要向 RTC_CALR 写入新值。随后 RECALPF 位会被自动置为 1。
3. 新校准设置将在对 RTC_CALR 执行写操作之后的三个 ck_apre 周期内生效。

22.3.13 时间戳功能

将 RTC_CR 寄存器的 TSE 或 ITSE 位置 1 可使能时间戳。

将 TSE 置 1 时：

当在 RTC_TS 引脚上检测到时间戳事件时，日历会保存到时间戳寄存器（RTC_TSSSR、RTC_TSTR 和 RTC_TSDR）中。

发生时间戳事件时，RTC_SR 寄存器中的时间戳标志位 (TSF) 将置 1。

通过将 RTC_CR 寄存器中的 TSIE 位置 1，可在发生时间戳事件时生成中断。

如果在时间戳标志 (TSF) 已置 1 的条件下检测到新的时间戳事件，则时间戳上溢标志 (TSOVF) 将置 1，而时间戳寄存器（RTC_TSTR 和 RTC_TSDR）将保持上一事件的结果。

注意： *由于同步过程，TSF 将在时间戳事件后 2 个 ck_apre 周期置 1。*

将 TSOVF 置 1 时不存在延迟。这意味着，如果两个时间戳事件接连发生，则 TSOVF 可能为“1”而 TSF 仍为“0”。因此，建议只在检测到 TSF 为“1”后再轮询 TSOVF。

小心： 如果在 TSF 位清零后紧接着发生时间戳事件，则 TSF 和 TSOVF 位都将置 1。为防止在时间戳事件发生的同时屏蔽该事件，除非已将 TSF 位读取为 1，否则应用程序不得将 0 写入 TSF 位。

22.3.14 校准时钟输出

将 RTC_CR 寄存器中的 COE 位置 1 时，会在 CALIB 器件输出上提供一个参考时钟。

如果 RTC_CR 寄存器中的 COSEL 位复位且 PREDIV_A = 0x7F，则 CALIB 频率为 $f_{RTCCLK/64}$。这相当于 RTCCLK 频率为 32.768 kHz 时，512 Hz 的校准输出。CALIB 占空比是不规则的：下降沿上存在轻微抖动。因此推荐使用上升沿。

当 COSEL 置 1 且“PREDIV_S+1”为 256 的非零整数倍（即：PREDIV_S[7:0] = 0xFF）时，CALIB 频率为 $f_{RTCCLK}/(256 * (PREDIV\_A+1))$。这相当于 RTCCLK 频率为 32.768 kHz 时，1 Hz 的校准输出，其中预分频器为默认值（PREDIV_A = Ox7F、PREDIV_S = 0xFF）。

注意： *COSEL 清零时，CALIB 输出为异步预分频器的第 6 级输出。*

COSEL 置 1 时，CALIB 输出为同步预分频器的第 8 级输出。

22.3.15 闹钟输出

RTC_CR 寄存器中的 OSEL[1:0] 控制位用于激活闹钟输出 TAMPALRM，以及选择输出的功能。这些功能可反映 RTC_SR 寄存器中相应标志的内容。

TAMPALRM 输出的极性由 RTC_CR 中的 POL 控制位确定，这样当 POL 置 1 时会输出选定标志位的相反值。

TAMPALRM 输出

使用 RTC_CR 寄存器中的控制位 TAMPALRM_TYPE 可将 TAMPALRM 引脚配置为输出开漏或输出上拉。借助 RTC_CR 中的 TAMPALRM_PU，可以在输出模式下应用内部上拉。

注意： *使能* TAMPALRM *输出后，其优先级高于 RTC_OUT1 上的 CALIB。*

如果在 RTC 中将 TAMPALRM 配置为开漏，则必须将 RTC_OUT1 GPIO 配置为输入。

22.4 RTC 低功耗模式

表 83. 低功耗模式对 RTC 的作用

模式	说明
睡眠	无影响 RTC 中断可使器件退出睡眠模式。
停止	当 RTC 时钟源为 LSE 或 LSI 时，RTC 保持工作状态。RTC 中断可使器件退出停止模式。
待机	RTC 掉电，退出待机模式后必须重新初始化。
关断	RTC 掉电，退出关断模式后必须重新初始化。

下表汇总了所有模式下的 RTC 引脚和功能。

表 84. 各个模式下的 RTC 引脚功能

功能	待机和关断模式以外的所有低功耗模式下的功能正常	待机和关断模式下的功能正常
RTC_TS	是	否
RTC_REFIN	是	否
RTC_OUT1	是	否
RTC_OUT2	是	否

22.5 RTC 中断

中断通道在屏蔽中断状态寄存器中置 1。中断输出也会激活。

表 85. 中断请求

中断缩写	中断事件	事件标志(1)	使能控制位(2)	中断清除方法	退出睡眠模式	退出停止模式	退出待机模式和关断模式
RTC	闹钟 A	ALRAF	ALRAIE	向 CALRAF 中写入 1	是	是(3)	否
	时间戳	TSF	TSIE	向 CTSF 中写入 1	是	是(3)	否

1. 事件标志位于 RTC_SR 寄存器中。
2. 中断屏蔽标志（由事件标志和使能控制位的逻辑与运算得出）位于 RTC_MISR 寄存器中。
3. 仅当 RTC 时钟源为 LSE 或 LSI 时，才能从停止模式唤醒。

22.6 RTC 寄存器

有关寄存器说明中使用的缩写，请参见参考手册中的*第 33 页的第 1.2 节*。

外设寄存器可按字（32 位）进行访问。

22.6.1 RTC 时间寄存器 (RTC_TR)

RTC time register

RTC_TR 是日历时间影子寄存器。只能在初始化模式下对该寄存器执行写操作。请参见*第 519 页的日历初始化和配置*和*第 520 页的读取日历*。

此寄存器受写保护。*第 519 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x00

上电复位值：0x0000 0000

系统复位值：0x0000 0000（当 BYPSHAD = 0 时，当 BYPSHAD = 1 时不受影响）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]			
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)

位 15 保留，必须保持复位值。

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)

位 7 保留，必须保持复位值。

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)

22.6.2 RTC 日期寄存器 (RTC_DR)

RTC date register

RTC_DR 是日历日期影子寄存器。只能在初始化模式下对该寄存器执行写操作。请参见*第 519 页的日历初始化和配置*和*第 520 页的读取日历*。

此寄存器受写保护。*第 519 页的 RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x04

上电复位值：0x0000 2101

系统复位值：0x0000 2101（当 BYPSHAD = 0 时，当 BYPSHAD = 1 时不受影响）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]				YU[3:0]			
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:20 **YT[3:0]**：年份的十位（BCD 格式）(Year tens in BCD format)

位 19:16 **YU[3:0]**：年份的个位（BCD 格式）(Year units in BCD format)

位 15:13 **WDU[2:0]**：星期几的个位 (Week day units)

000：禁止
001：星期一
...
111：星期日

位 12 **MT**：月份的十位（BCD 格式）(Month tens in BCD format)

位 11:8 **MU[3:0]**：月份的个位（BCD 格式）(Month units in BCD format)

位 7:6 保留，必须保持复位值。

位 5:4 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 3:0 **DU[3:0]**：日期的个位（BCD 格式）(Date units in BCD format)

注意： *日历在达到最大值时会冻结，而不会翻转。*

22.6.3 RTC 亚秒寄存器 (RTC_SSR)

RTC sub second register

偏移地址：0x08

上电复位值：0x0000 0000

系统复位值：0x0000 0000（当 BYPSHAD = 0 时，当 BYPSHAD = 1 时不受影响）

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

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

位 31:16 保留，必须保持复位值。

位 15:0 **SS[15:0]**：亚秒值 (Sub second value)

SS[15:0] 是同步预分频器计数器的值。此亚秒值可根据以下公式得出：

亚秒值 = (PREDIV_S - SS) / (PREDIV_S + 1)

注意：仅当执行平移操作之后，SS 才能大于 PREDIV_S。在这种情况下，正确的时间/日期比 RTC_TR/RTC_DR 所指示的时间/日期慢一秒钟。

22.6.4 RTC 初始化控制和状态寄存器 (RTC_ICSR)

此寄存器受写保护。*第 519 页的 RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x0C

上电复位值：0x0000 0007

系统复位值：0bxxxx xxxx xxxx xxxx xxxx xxxx 000x xxxx（除了 INIT、INITF 和 RSF 位会清零之外，其他位均不受影响）

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INIT	INITF	RSF	INITS	SHPF	Res.	Res.	ALRAWF
								rw	r	rc_w0	r	r			r

位 31:17 保留，必须保持复位值。

位 16 **RECALPF**：重新校准挂起标志

当软件对 RTC_CALR 寄存器执行写操作时，RECALPF 状态标志将自动置 1，指示 RTC_CALR 寄存器已屏蔽。当采用新的校准设置时，该位恢复为 0。请参见*动态重校准*。

位 15:8 保留，必须保持复位值。

位 7 **INIT**：初始化模式

0：自由运行模式

1：初始化模式，用于编程时间和日期寄存器（RTC_TR 和 RTC_DR）以及预分频器寄存器 (RTC_PRER)。计数器停止计数，当 INIT 被复位后，计数器从新值开始计数。

位 6 **INITF**：初始化标志 (Initialization flag)

当该位置 1 时，RTC 处于初始化状态，此时可更新事件、日期和预分频器寄存器。

0：不允许更新日历寄存器

1：允许更新日历寄存器

位 5 **RSF**：寄存器同步标志 (Registers synchronization flag)

每次将日历寄存器的值复制到影子寄存器（RTC_SSR、RTC_TR 和 RTC_DR）时，都会由硬件将该位置 1。在初始化模式下、平移操作挂起时 (SHPF = 1) 或在旁路影子寄存器模式 (BYPSHAD = 1) 下，该位由硬件清零。该位还可由软件清零。

在初始化模式下，该位可由软件或硬件清零。

0：日历影子寄存器尚未同步

1：日历影子寄存器已同步

位 4 **INITS**：初始化状态标志 (Initialization status flag)

当日历年份字段不为 0 时（上电复位状态），由硬件将该位置 1。

0：日历尚未初始化

1：日历已经初始化

位 3 **SHPF**：平移操作挂起 (Shift operation pending)

只要通过对 RTC_SHIFTR 寄存器执行写操作来启动平移操作，此标志便由硬件置 1。执行完相应的平移操作后，此标志由硬件清零。对 SHPF 位执行写入操作不起作用。

0：没有平移操作挂起

1：某个平移操作挂起

位 2:1 保留，必须保持复位值。

位 0 **ALRAWF**：闹钟 A 写标志 (Alarm A write flag)

在 RTC_CR 寄存器中的 ALRAE 位置 0 后，当闹钟 A 的值可更改时，由硬件将该位置 1。该位在初始化模式下由硬件清零。

0：不允许更新闹钟 A

1：允许更新闹钟 A

22.6.5 RTC 预分频器寄存器 (RTC_PRER)

RTC prescaler register

只能在初始化模式下对该寄存器执行写操作。必须通过两次独立的写访问执行初始化。请参见*第 519 页的日历初始化和配置*。

此寄存器受写保护。*第 519 页的 RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x10

上电复位值：0x007F 00FF

系统复位：不受影响

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

位 31:23 保留，必须保持复位值。

位 22:16 **PREDIV_A[6:0]**：异步预分频系数 (Asynchronous prescaler factor)

下面是异步分频系数的公式：

ck_apre 频率 = RTCCLK 频率/(PREDIV_A+1)

位 15 保留，必须保持复位值。

位 14:0 **PREDIV_S[14:0]**：同步预分频系数 (Synchronous prescaler factor)

下面是同步分频系数的公式：

ck_spre 频率 = ck_apre 频率/(PREDIV_S+1)

22.6.6 RTC 控制寄存器 (RTC_CR)

RTC control register

此寄存器受写保护。第 519 页的 RTC 寄存器写保护中介绍了写访问的过程。

偏移地址：0x18

上电复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUT2EN	TAMPALRM_TYPE	TAMPALRM_PU	Res.	Res.	Res.	Res.	Res.	COE	OSEL[1:0]		POL	COSEL	BKP	SUB1H	ADD1H
rw	rw	rw						rw	rw	rw	rw	rw	rw	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	Res.	Res.	ALRAIE	TSE	Res.	Res.	ALRAE	Res.	FMT	BYPSHAD	REFCKON	TSEDGE	Res.	Res.	Res.
rw			rw	rw			rw		rw	rw	rw	rw			

位 31 **OUT2EN**：RTC_OUT2 输出使能 (RTC_OUT2 output enable)
将该位置 1 可将 RTC 输出重映射到 RTC_OUT2，具体如下：
OUT2EN = 0：RTC 输出 2 禁止
如果 OSEL ≠ 00 或 TAMPOE = 1：在 RTC_OUT1 上输出 TAMPALRM
如果 OSEL = 00、TAMPOE = 0 且 COE = 1：在 RTC_OUT1 上输出 CALIB
OUT2EN = 1：RTC 输出 2 使能
如果 (OSEL ≠ 00 或 TAMPOE = 1) 且 COE = 0：在 RTC_OUT2 上输出 TAMPALRM
如果 OSEL = 00、TAMPOE = 0 且 COE = 1：在 RTC_OUT2 上输出 CALIB
如果（OSEL ≠ 00 或 TAMPOE = 1）且 COE = 1：在 RTC_OUT2 上输出 CALIB，在 RTC_OUT1 上输出 TAMPALRM。

位 30 **TAMPALRM_TYPE**：TAMPALRM 输出类型 (TAMPALRM output type)
0：TAMPALRM 为推挽输出
1：TAMPALRM 为开漏输出

位 29 **TAMPALRM_PU**：TAMPALRM 上拉使能 (TAMPALRM pull-up enable)
0：不对 TAMPALRM 输出应用上拉
1：对 TAMPALRM 输出应用上拉

位 28:24 保留，必须保持复位值。

位 23 **COE**：校准输出使能 (Calibration output enable)
该位使能 CALIB 输出
0：禁止校准输出
1：使能校准输出

位 22:21 **OSEL[1:0]**：输出选择 (Output selection)
这些位用于选择要连接到 TAMPALRM 输出的标志。
00：禁止输出
01：使能闹钟 A 输出
10：保留
11：保留

位 20 **POL**：输出极性 (Output polarity)
该位用于配置 TAMPALRM 输出的极性。
0：当 ALRAF 置 1 时（取决于 OSEL[1:0]），该引脚为高电平
1：当 ALRAF 置 1 时（取决于 OSEL[1:0]），该引脚为低电平

位 19 **COSEL**：校准输出选择 (Calibration output selection)
当 COE = 1 时，该位可选择 CALIB 上输出的信号。
0：校准输出为 512 Hz
1：校准输出为 1 Hz
在 RTCCLK 为 32.768 kHz 且预分频器为其默认值（PREDIV_A = 127 且 PREDIV_S = 255）的条件下，这些频率有效。请参见*第 22.3.14 节：校准时钟输出*。

位 18 **BKP**：备份 (Backup)
用户可对该位执行写操作以记录是否已对夏令时进行更改。

位 17 **SUB1H**：减少 1 小时（冬季时间更改）(Subtract 1 hour (winter time change))
当该位在初始化模式以外的模式下置 1 时，如果当前小时不是 0，则日历时间将减少 1 小时。该位始终读为 0。
当前小时为 0 时，将该位置 1 没有任何作用。
0：无影响
1：将当前时间减少 1 小时。这可用于冬季时间更改。

位 16 **ADD1H**：增加 1 小时（夏季时间更改）(Add 1 hour (summer time change))
当该位在初始化模式以外的模式下置 1 时，日历时间将增加 1 小时。该位始终读为 0。
0：无影响
1：将当前时间增加 1 小时。这可用于夏季时间更改

位 15 **TSIE**：时间戳中断使能 (Timestamp interrupt enable)
0：时间戳中断禁止 (Timestamp interrupt disable)
1：时间戳中断使能 (Timestamp interrupt enable)

位 14:13 保留，必须保持复位值。

位 12 **ALRAIE**：闹钟 A 中断使能 (Alarm A interrupt enable)
0：禁止闹钟 A 中断
1：使能闹钟 A 中断

位 11 **TSE**：时间戳使能 (timestamp enable)
0：禁止时间戳
1：使能时间戳

位 10:9 保留，必须保持复位值。

位 8 **ALRAE**：闹钟 A 使能 (Alarm A enable)
0：禁止闹钟 A
1：使能闹钟 A

位 7 保留，必须保持复位值。

位 6 **FMT**：小时格式 (Hour format)
0：24 小时/天格式
1：AM/PM 小时格式

位 5 **BYPSHAD**：旁路影子寄存器 (Bypass the shadow registers)
0：日历值（从 RTC_SSR、RTC_TR 和 RTC_DR 读取时）取自影子寄存器，该影子寄存器每两个 RTCCLK 周期更新一次。
1：日历值（从 RTC_SSR、RTC_TR 和 RTC_DR 读取时）直接取自日历计数器。
注意：如果 APB1 时钟的频率低于 7 倍的 RTCCLK 频率，则必须将 BYPSHAD 置 1。

位 4 **REFCKON**：RTC_REFIN 参考时钟检测使能（50 Hz 或 60 Hz）(RTC_REFIN Reference clock detection enable (50 or 60 Hz))
0：禁止 RTC_REFIN 检测
1：使能 RTC_REFIN 检测
注意：PREDIV_S 必须为 0x00FF。

位 3 **TSEDGE**：时间戳事件有效边沿 (Timestamp event active edge)
0：RTC_TS 输入上升沿生成时间戳事件
1：RTC_TS 输入下降沿生成时间戳事件
TSEDGE 发生更改时，必须复位 TSE 以避免将 TSF 意外置 1。

位 2:0 保留，必须保持复位值。

注意：只能在初始化模式下 (RTC_ICSR/INITF = 1) 对该寄存器的位 6 和 4 执行写操作。
建议不要在日历小时递增时更改小时，因为这样做会屏蔽日历小时的增量。
ADD1H 和 SUB1H 的更改在下一秒生效。

22.6.7 RTC 写保护寄存器 (RTC_WPR)

RTC write protection register

偏移地址：0x24

复位值：0x0000 0000

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

位 31:8 保留，必须保持复位值。

位 7:0 **KEY[7:0]**：写保护关键字 (Write protection key)

可通过软件对该字节执行写操作。

读取该字节时，始终返回 0x00。

有关如何解锁 RTC 寄存器写保护的介绍，请参见*RTC 寄存器写保护*。

22.6.8 RTC 校准寄存器 (RTC_CALR)

RTC calibration register

此寄存器受写保护。*第 519 页的 RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x28

上电复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALP	CALW8	CALW16	Res.	Res.	Res.	Res.	CALM[8:0]								
rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **CALP**：将 RTC 的频率增加 488.5 ppm (Increase frequency of RTC by 488.5 ppm)

0：不增加 RTCCLK 脉冲。

1：每 2^{11} 个脉冲有效插入一个 RTCCLK 脉冲（将频率增加 488.5 ppm）。

此功能应与 CALM 结合使用，后者在高分辨率下会降低日历的频率。如果输入频率为 32768 Hz，则在 32 秒窗口中增加的 RTCCLK 脉冲数按如下公式计算：(512 × CALP) - CALM。

请参见*第 22.3.12 节：RTC 精密数字校准*。

位 14 **CALW8**：使用 8 秒校准周期 (Use an 8-second calibration cycle period)

当 CALW8 置 1 时，选择 8 秒校准周期。

注意：当 CALW8 = 1 时，CALM[1:0] 将始终保持为 00。请参见第 22.3.12 节：RTC 精密数字校准。

位 13 **CALW16**：使用 16 秒校准周期 (Use a 16-second calibration cycle period)

当 CALW16 置 1 时，选择 16 秒校准周期。如果 CALW8 = 1，则该位不可以置 1。

注意：当 CALW16= 1 时，CALM[0] 将始终保持为 0。请参见第 22.3.12 节：RTC 精密数字校准。

位 12:9 保留，必须保持复位值。

位 8:0 **CALM[8:0]**：负校准 (Calibration minus)

在 2^{20} 个 RTCCLK 脉冲内屏蔽 CALM 个脉冲（如果输入频率为 32768 Hz，则为 32 秒）来降低日历的频率。其分辨率为 0.9537 ppm。

要提高日历的频率，则应将此功能与 CALP 结合使用。请参见*第 22.3.12 节：第 522 页的 RTC 精密数字校准*。

22.6.9 RTC 平移控制寄存器 (RTC_SHIFTR)

RTC shift control register

此寄存器受写保护。*第 519 页的 RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x2C

上电复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w															

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

位 31 **ADD1S**：增加一秒钟 (Add one second)

0：无影响

1：对时钟 / 日历增加一秒钟

该位为只写位且始终读为 0。当平移操作挂起（RTC_ICSR 中的 SHPF = 1）时，对该位执行写操作无作用。

此函数应与 SUBFS 配合使用（请参见下文介绍），以便有效地向原子操作机制的时钟添加亚秒值。

位 30:15 保留，必须保持复位值。

位 14:0 **SUBFS[14:0]**：减少亚秒值 (Subtract a fraction of a second)

这些位为只写位且始终读为 0。当平移操作挂起（RTC_ICSR 中的 SHPF = 1）时，对该位执行写操作无作用。

写入 SUBFS 的值将加到同步预分频器计数器中。由于该计数器递减计数，此操作可有效地从时钟减去（延迟）以下时间：

延迟（秒）= SUBFS / (PREDIV_S + 1)

当 ADD1S 函数与 SUBFS 结合使用时，可有效地将亚秒值增加到时钟（提前时钟），使时钟提前以下时间：

提前（秒）= (1 - (SUBFS / (PREDIV_S + 1)))。

注意：对 SUBFS 执行写操作将使 RSF 清零。软件随后会等待至 RSF = 1 以确定影子寄存器已更新为平移后的时间。

22.6.10 RTC 时间戳时间寄存器 (RTC_TSTR)

RTC timestamp time register

仅当 RTC_SR 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

偏移地址：0x30

上电复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]			
									r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	r	r	r	r	r	r	r		r	r	r	r	r	r	r

位 31:23 保留，必须保持复位值。

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)。

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)。

位 15 保留，必须保持复位值。

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)。

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)。

位 7 保留，必须保持复位值。

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)。

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)。

22.6.11 RTC 时间戳日期寄存器 (RTC_TSDR)

RTC timestamp date register

仅当 RTC_SR 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

偏移地址：0x34

上电复位值：0x0000 0000

系统复位：不受影响

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
r	r	r	r	r	r	r	r			r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:13 **WDU[2:0]**：星期几的个位 (Week day units)

位 12 **MT**：月份的十位（BCD 格式）(Month tens in BCD format)

位 11:8 **MU[3:0]**：月份的个位（BCD 格式）(Month units in BCD format)

位 7:6 保留，必须保持复位值。

位 5:4 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 3:0 **DU[3:0]**：日期的个位（BCD 格式）(Date units in BCD format)

22.6.12 RTC 时间戳亚秒寄存器 (RTC_TSSSR)

RTC timestamp sub second register

仅当 RTC_SR 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

偏移地址：0x38

上电复位值：0x0000 0000

系统复位：不受影响

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

位 31:16 保留，必须保持复位值。

位 15:0 SS[15:0]：亚秒值 (Sub second value)

当发生时间戳事件时，SS[15:0] 是同步预分频器计数器的值。

22.6.13 RTC 闹钟 A 寄存器 (RTC_ALRMAR)

RTC alarm A register

仅当 RTC_ICSR 中的 ALRAWF 置 1 时或在初始化模式下，才可以对该寄存器执行写操作。

此寄存器受写保护。*第 519 页的 RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x40

上电复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WD SEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **MSK4**：闹钟 A 日期掩码 (Alarm A date mask)
0：如果日期/日匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，日期/日无关

位 30 **WDSEL**：星期几选择 (Week day selection)
0：DU[3:0] 代表日期的个位
1：DU[3:0] 代表星期几。DT[1:0] 为无关位。

位 29:28 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 27:24 **DU[3:0]**：日期个位或日（BCD 格式）(Date units or day in BCD format)

位 23 **MSK3**：闹钟 A 小时掩码 (Alarm A hours mask)
0：如果小时匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，小时无关

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)

位 15 **MSK2**：闹钟 A 分钟掩码 (Alarm A minutes mask)
0：如果分钟匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，分钟无关

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)

位 7 **MSK1**：闹钟 A 秒掩码 (Alarm A seconds mask)
0：如果秒匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，秒无关

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)。

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)。

22.6.14 RTC 闹钟 A 亚秒寄存器 (RTC_ALRMASSR)

RTC alarm A sub second register

仅当 RTC_ICSR 中的 ALRAWF 置 1 时或在初始化模式下，才可以对该寄存器执行写操作。

此寄存器受写保护。*第 519 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x44

上电复位值：0x0000 0000

系统复位：不受影响

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

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

位 31:28 保留，必须保持复位值。

位 27:24 **MASKSS[3:0]**：屏蔽从该位开始的最高有效位 (Mask the most-significant bits starting at this bit)
0：不对闹钟 A 的亚秒进行比较。当秒单元递增时设置闹钟（假定其余字段均匹配）。
1：在闹钟 A 比较中，SS[14:1] 为无关位。仅比较 SS[0]。
2：在闹钟 A 比较中，SS[14:2] 为无关位。仅比较 SS[1:0]。
3：在闹钟 A 比较中，SS[14:3] 为无关位。仅比较 SS[2:0]。
...
12：在闹钟 A 比较中，SS[14:12] 为无关位。比较 SS[11:0]。
13：在闹钟 A 比较中，SS[14:13] 为无关位。比较 SS[12:0]。
14：在闹钟 A 比较中，SS[14] 为无关位。比较 SS[13:0]。
15：所有 15 个 SS 位均进行比较，并且必须全部匹配才能激活闹钟。
同步计数器的溢出位（位 15）从不进行比较。仅当执行平移操作之后，该位才不为 0。
注意：同步计数器的溢出位（位 15）从不进行比较。仅当执行平移操作之后，该位才不为 0。

位 23:15 保留，必须保持复位值。

位 14:0 **SS[14:0]**：亚秒值 (Sub seconds value)
该值与同步预分频器计数器的内容进行比较以确定是否要激活闹钟 A。仅比较位 0 到 MASKSS-1。

22.6.15 RTC 状态寄存器 (RTC_SR)

RTC status register

偏移地址：0x50

上电复位值：0x0000 0000

系统复位：不受影响

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

位 31:5 保留，必须保持复位值。

位 4 **TSOVF**：时间戳溢出标志 (Timestamp overflow flag)
当在 TSF 已置 1 的情况下发生时间戳事件时，由硬件将此标志置 1。
建议仅在 TSF 位清零之后再检查并清零 TSOVF 位。否则，如果时间戳事件恰好在清零 TSF 位之前刚刚发生，则溢出事件可能会被漏掉。

位 3 **TSF**：时间戳标志 (Timestamp flag)
发生时间戳事件时，由硬件将此标志置 1。

位 2:1 保留，必须保持复位值。

位 0 **ALRAF**：闹钟 A 标志 (Alarm A flag)
当时间/日期寄存器（RTC_TR 和 RTC_DR）与闹钟 A 寄存器 (RTC_ALRMAR) 匹配时，由硬件将该标志置 1。

注意：　该寄存器的位在 RTC_SCR 寄存器中相应的清零位置 1 后的 2 个 APB 时钟周期之后清零。

22.6.16 RTC 屏蔽中断状态寄存器 (RTC_MISR)

RTC masked interrupt status register

偏移地址：0x54

上电复位值：0x0000 0000

系统复位：不受影响

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSOVMF	TSMF	Res.	Res.	ALRAMF
											r	r			r

位 31:5 保留，必须保持复位值。

位 4 **TSOVMF**：时间戳溢出屏蔽标志 (Timestamp overflow masked flag)
当在 TSMF 已置 1 的情况下发生时间戳中断时，由硬件将此标志置 1。
建议仅在 TSF 位清零之后再检查并清零 TSOVF 位。否则，如果时间戳事件恰好在清零 TSF 位之前刚刚发生，则溢出事件可能会被漏掉。

位 3 **TSMF**：时间戳屏蔽标志 (Timestamp masked flag)
发生时间戳中断时，由硬件将此标志置 1。

位 2:1 保留，必须保持复位值。

位 0 **ALRAMF**：闹钟 A 屏蔽标志 (Alarm A masked flag)
发生闹钟 A 中断时，由硬件将此标志置 1。

22.6.17 RTC 状态清零寄存器 (RTC_SCR)

RTC status clear register

偏移地址：0x5C

上电复位值：0x0000 0000

系统复位：不受影响

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTSOVF	CTSF	Res.	Res.	CALRAF
											w	w			w

位 31:5 保留，必须保持复位值。

位 4 **CTSOVF**：清零时间戳溢出标志 (Clear timestamp overflow flag)

向该位写入 1 可清零 RTC_SR 寄存器中的 TSOVF 位。

建议仅在 TSF 位清零之后再检查并清零 TSOVF 位。否则，如果时间戳事件恰好在清零 TSF 位之前刚刚发生，则溢出事件可能会被漏掉。

位 3 **CTSF**：清零时间戳标志 (Clear timestamp flag)

向该位写入 1 可清零 RTC_SR 寄存器中的 TSOVF 位。

位 2:1 保留，必须保持复位值。

位 0 **CALRAF**：清零闹钟 A 标志 (Clear alarm A flag)

向该位写入 1 可清零 RTC_SR 寄存器中的 ALRAF 位。

22.6.18 RTC 寄存器映射

表 86. RTC 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**RTC_TR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT [1:0]		HU[3:0]				Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	Reset value										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x04	**RTC_DR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]				YU[3:0]				WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT [1:0]		DU[3:0]			
	Reset value									0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1			0	0	0	0	0	1
0x08	**RTC_SSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**RTC_ICSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RECALPF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INIT	INITF	RSF	INITS	SHPF	Res.	Res.	ALRAWF
	Reset value																0									0	0	0	0	0			1
0x10	**RTC_PRER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV_A[6:0]							PREDIV_S[14:0]															
	Reset value										1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0x14	Reserved	Res.																															
0x18	**RTC_CR**	OUT2EN	TAMPALRM_TYPE	TAMPALRM_PU	Res.	Res.	Res.	Res.	Res.	COE	OSEL [1:0]		POL	COSEL	BKP	SUB1H	ADD1H	TSIE	Res.	Res.	ALRAIE	TSE	Res.	Res.	ALRAE	Res.	FMT	BYPSHAD	REFCKON	TSEDGE	Res.	Res.	Res.
	Reset value	0	0	0						0	0	0	0	0	0	0	0	0			0	0			0		0	0	0	0			
0x20	Reserved	Res.																															
0x24	**RTC_WPR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x28	**RTC_ CALR**	Res.	Res.	Res	Res	Res.	Res.	Res.	Res	Res	Res.	Res.	Res.	Res	Res.	Res.	Res.	CALP	CALW8	CALW16	Res.	Res	Res.	Res.	CALM[8:0]								
	Reset value																	0	0	0					0	0	0	0	0	0	0	0	0
0x2C	**RTC_SHIFTR**	ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBFS[14:0]														
	Reset value	0																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	**RTC_TSTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]				Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	Reset value										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x34	**RTC_TSDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDU[1:0]			MT	MU[3:0]				Res.	Res.	DT [1:0]		DU[3:0]			
	Reset value																	0	0	0	0	0	0	0	0			0	0	0	0	0	0
0x38	**RTC_TSSSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 86. RTC 寄存器映射和复位值（续）

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x40	**RTC_ALRMAR**	MSK4	WDSEL	DT [1:0]		DU[3:0]				MSK3	PM	HT [1:0]		HU[3:0]				MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**RTC_ ALRMASSR**	Res.	Res.	Res.	Res.	MASKSS [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]														
	Reset value					0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48 - 0x4C	Reserved	Res.																															
0x50	**RTC_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSOVF	TSF	Res.	Res.	ALRAF
	Reset value																												0	0			0
0x54	**RTC_MISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSOVMF	TSMF	Res.	Res.	ALRAMF
	Reset value																												0	0			0
0x58	Reserved	Res.																															
0x5C	**RTC_SCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTSOVF	CTSF	Res.	Res.	CALRAF
	Reset value																												0	0			0

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

23 内部集成电路 (I2C) 接口

23.1 简介

I^2C（内部集成电路）总线接口处理微控制器与串行 I^2C 总线间的通信。它提供多主模式功能，可以控制所有 I^2C 总线特定的序列、协议、仲裁和时序。它支持标准模式 (Sm)、快速模式 (Fm) 和超快速模式 (Fm+)。

它还与 SMBus（系统管理总线）和 PMBus®（电源管理总线）兼容。

可使用 DMA 来减轻 CPU 的工作量。

23.2 I2C 主要特性

- 兼容 I^2C 总线规范第 03 版：
  - 从模式和主模式
  - 多主模式功能
  - 标准速度模式（高达 100 kHz）
  - 快速模式（高达 400 kHz）
  - 超快速模式（高达 1 MHz）
  - 7 位和 10 位寻址模式
  - 多个 7 位从地址（2 个从设备地址寄存器，1 个具有可配置的掩码位段）
  - 所有 7 位地址应答模式
  - 广播呼叫
  - 总线上的数据建立和保持时间可软件配置
  - 方便易用的事件管理
  - 可选的时钟延长
  - 软件复位
- 带 DMA 功能的 1 字节缓冲
- 可编程模拟和数字噪声滤波器

还可额外提供以下特性，具体取决于产品实现（请参见*第 23.3 节*）：

- 兼容 SMBus 规范第 3.0 版：
  - 具有 ACK 控制的硬件 PEC（数据包错误校验）生成和验证
  - 命令和数据应答控制
  - 支持地址解析协议 (ARP)
  - 支持主机和从设备
  - SMBus 报警
  - 超时和空闲条件检测
- 兼容 PMBus 第 1.3 版标准
- 独立时钟：选择独立时钟源可使 I2C 通信速度不受 PCLK 时钟频率更改的影响
- 地址匹配时从停止模式唤醒

23.3 I2C 特性实现

器件包含一个 I²C 总线控制器，其特性在下表中列出

表 87. STM32C0x1 I2C 特性实现

I2C 特性(1)	I2C1
7 位寻址模式	X
10 位寻址模式	X
标准模式（高达 100 kb/s）	X
快速模式（高达 400 kb/s）	X
超快速模式，20 mA 输出驱动 I/O（高达 1 Mb/s）	X
独立时钟	X
从停止模式唤醒	X
SMBus/PMbus	X

1. X = 支持。

23.4 I2C 功能描述

除了接收和发送数据之外，此接口还可以从串行格式转换为并行格式，反之亦然。中断由软件使能或禁止。该接口通过数据引脚 (SDA) 和时钟引脚 (SCL) 连接到 I^2C 总线。它可以连接到标准速度（高达 100 kHz）、快速（高达 400 kHz）或超快速（高达 1 MHz） I^2C 总线。

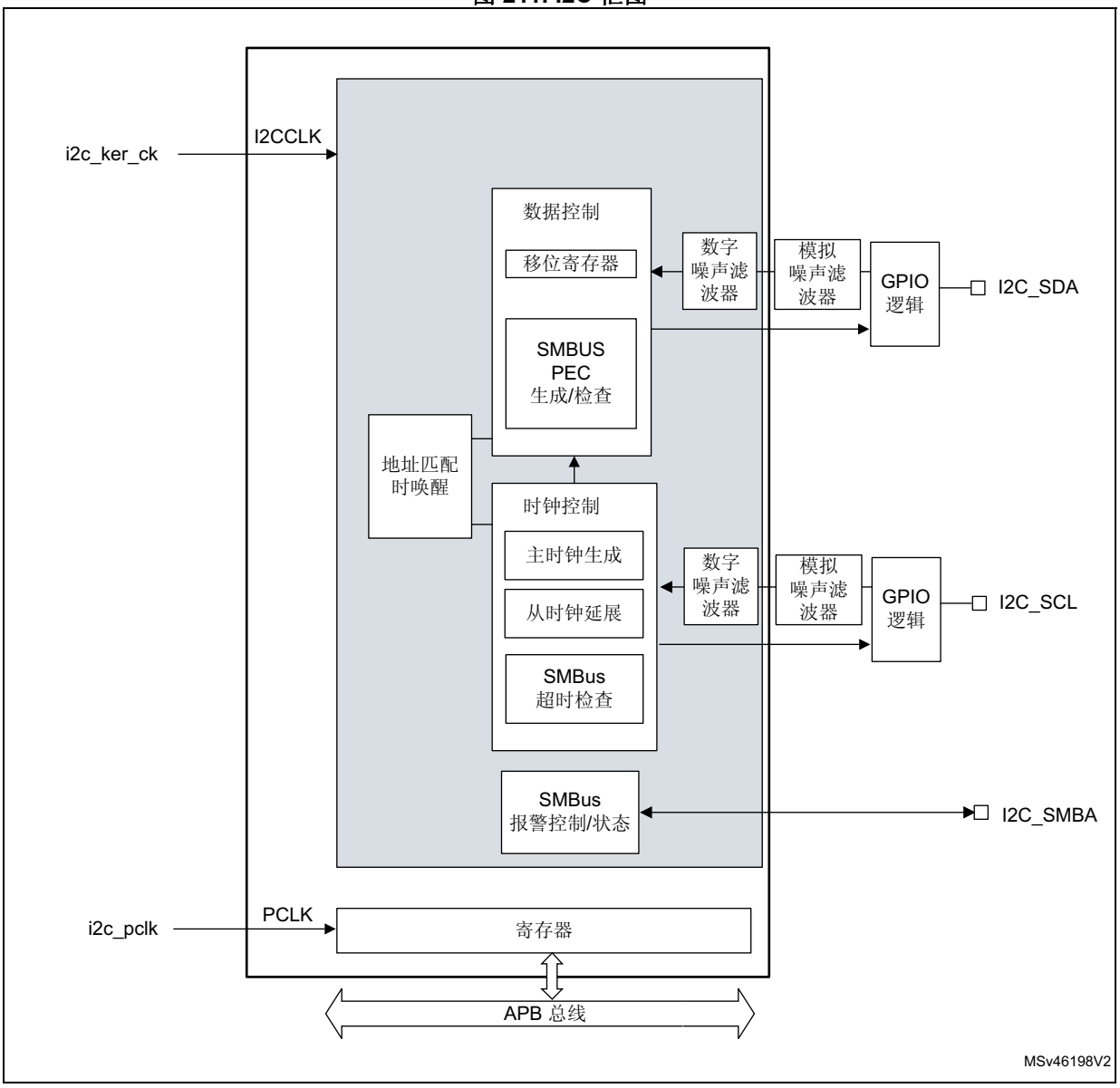
该接口也可通过数据 (SDA) 引脚和时钟 (SCL) 引脚连接到 SMBus。

如果支持 SMBus 功能：还可选用 SMBus 报警引脚 (SMBA)。

23.4.1 I2C 框图

I2C 接口如*图 211* 所示。

图 211. I2C 框图



I2C 的时钟由独立时钟源提供，这使得 I2C 能够独立于 PCLK 频率工作。

对于支持以 20 mA 输出电流驱动超快速模式操作的 I2C I/O，可以通过系统配置控制器 (SYSCFG) 中的控制位使能驱动功能。*请参见第 23.3 节：I2C 特性实现。*

23.4.2 I2C 引脚和内部信号

表 88. I2C 输入/输出引脚

引脚名称	信号类型	说明
I2C_SDA	双向	I2C 数据
I2C_SCL	双向	I2C 时钟
I2C_SMBA	双向	SMBus 报警

表 89. I2C 内部输入/输出信号

内部信号名称	信号类型	说明
i2c_ker_ck	输入	I2C 内核时钟，在本文档中也称为 I2CCLK
i2c_pclk	输入	I2C APB 时钟
i2c_it	输出	I2C 中断，有关中断源的完整列表，请参见*表 103*
i2c_rx_dma	输出	I2C 接收数据 DMA 请求 (I2C_RX)
i2c_tx_dma	输出	I2C 发送数据 DMA 请求 (I2C_TX)

23.4.3 I2C 时钟要求

I2C 内核的时钟由 I2CCLK 提供。

I2CCLK 周期 t_{I2CCLK} 必须遵循以下条件：

- $t_{I2CCLK} < (tLOW - tfilters) / 4$
- $t_{I2CCLK} < tHIGH$

其中：

t_{LOW}：SCL 低电平时间，tHIGH：SCL 高电平时间

$t_{filters}$：滤波器使能时，该值为模拟滤波器和数字滤波器引入的延时总和。

数字滤波器延时为 DNF x t_{I2CCLK}。

PCLK 时钟周期 t_{PCLK} 必须遵循以下条件：

- $t_{PCLK} < 4 / 3\ tSCL$（tSCL：SCL 周期）

小心： 当 I2C 内核的时钟由 PCLK 提供时，该时钟必须遵循 t_{I2CCLK} 的条件。

23.4.4 模式选择

该接口在工作时可选用以下四种模式之一：

- 从发送器
- 从接收器
- 主发送器
- 主接收器

默认情况下，它以从模式工作。接口在生成起始位后会自动由从模式切换为主模式，并在出现仲裁丢失或生成停止位时从主模式切换为从模式，从而实现多主模式功能。

通信流程

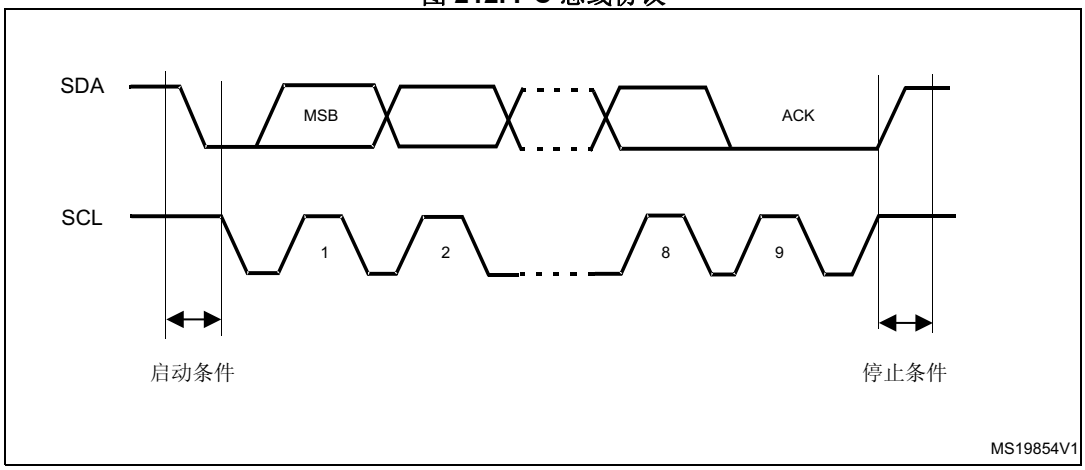
在主模式下，I2C 接口会启动数据传输并生成时钟信号。串行数据传输始终是在出现起始位时开始，在出现停止位时结束。起始位和停止位均在主模式下由软件生成。

在从模式下，该接口能够识别其自身地址（7 或 10 位）以及广播呼叫地址。广播呼叫地址检测可由软件使能或禁止。保留的 SMBus 地址也可由软件使能。

数据和地址均以 8 位字节传输，MSB 在前。起始位后紧随地址字节（7 位地址占据一个字节；10 位地址占据两个字节）。地址始终在主模式下发送。

在字节传输八个时钟周期后是第九个时钟脉冲，在此期间接收器必须向发送器发送一个应答位（请参见*图 212*）。

图 212. I²C 总线协议



应答位可由软件使能或禁止。I2C 接口地址可通过软件进行选择。

23.4.5 I2C 初始化

使能和禁止外设

必须先在时钟控制器中配置并使能 I2C 外设时钟，之后才能通过将 I2C_CR1 寄存器中的 PE 位置 1 来使能 I2C。

当禁止 I2C (PE = 0) 时，I²C 将执行软件复位。更多详细信息，请参见*第 23.4.6 节*。

噪声滤波器

通过将 I2C_CR1 寄存器中的 PE 位置 1 来使能 I2C 外设之前，如有必要，用户必须配置噪声滤波器。默认情况下，SDA 和 SCL 输入上集成了模拟噪声滤波器。该滤波器符合 I²C 规范，此规范要求在快速模式和超快速模式下对脉宽在 50ns 以下的脉冲都要抑制。用户可通过将 ANFOFF 位置 1 来禁止该模拟滤波器，和/或通过配置 I2C_CR1 寄存器中的 DNF[3:0] 位来选择数字滤波器。

使能数字滤波器时，SCL 或 SDA 线的电平只有在电平稳定时间超过 DNF x I2CCLK 个周期后才会发生内部变化。这样可抑制的尖峰脉宽在 1 到 15 个 I2CCLK 周期可编程。

表 90. 模拟滤波器与数字滤波器对比

-	模拟滤波器	数字滤波器
抑制的脉冲宽度	≥ 50 ns	长度可编程为 1 到 15 个 I2C 外设时钟
优点	停止模式中仍可用	– 长度可编程：额外的滤波能力与标准要求 – 稳定长度
缺点	随温度、电压和过程变化会发生变化	当使能数字滤波器后，无法在地址匹配时从停止模式唤醒

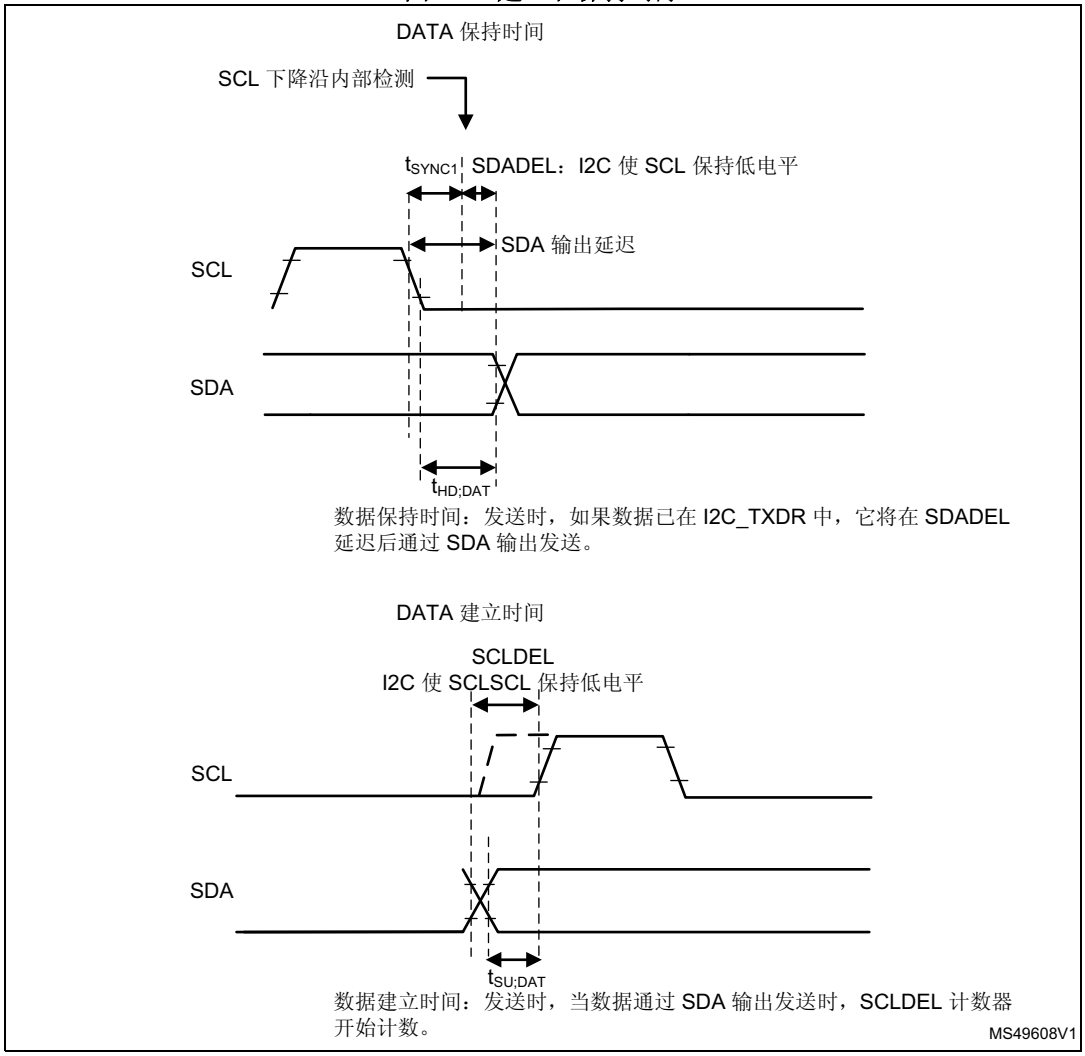
小心： 当 I2C 处于使能状态时，不允许更改滤波器配置。

I2C 时序

必须配置时序，以便保证主模式和从模式下使用正确的数据保持和建立时间。配置方法是编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLDEL[3:0] 和 SDADEL[3:0] 位。

STM32CubeMX 工具计算 I2C_TIMINGR 内容并在 I2C 配置窗口中进行显示。

图 213. 建立和保持时序



当内部检测到 SCL 下降沿时，会在发送 SDA 输出之前插入一段延时（t_{SDADEL}，会影响保持时间 $t_{HD;DAT}$）：t_{SDADEL} = SDADEL x t_{PRESC} + t_{I2CCLK}，其中 t_{PRESC} = (PRESC + 1) x t_{I2CCLK}。

SDA 总输出延时为：

t_{SYNC1} + {[SDADEL x (PRESC + 1) + 1] x t_{I2CCLK}}

t_{SYNC1} 持续时间取决于：

- SCL 下降斜率
- 模拟滤波器（使能时）引入的输入延时：$t_{AF(min)} < t_{AF} < t_{AF(max)}$
- 数字滤波器（使能时）引入的输入延时：t_{DNF} = DNF x t_{I2CCLK}
- SCL 与 I2CCLK 时钟建立同步而产生的延时（两到三个 I2CCLK 周期）

要桥接 SCL 下降沿的未定义区域，用户编程 SDADEL 时必须遵循以下条件：

{$t_{f\ (max)}$ + $t_{HD;DAT\ (min)}$ - $t_{AF(min)}$ - [(DNF + 3) x t_{I2CCLK}]} / {(PRESC + 1) x t_{I2CCLK}} ≤ SDADEL

SDADEL ≤ {$t_{HD;DAT\ (max)}$ - $t_{AF(max)}$ - [(DNF + 4) x t_{I2CCLK}]} / {(PRESC + 1) x t_{I2CCLK}}

注意： *只有使能模拟滤波器时，公式中才包含 $t_{AF(min)}$ / $t_{AF(max)}$。有关 t_{AF} 值的信息，请参见器件数据手册。*

标准模式、快速模式和超快速模式下的 $t_{HD;DAT}$ 最大值分别可达 3.45 µs、0.9 µs 和 0.45 µs。但必须小于 $t_{VD;DAT}$ 最大值（差值为跳变时间）。只有器件未延长 SCL 信号的低电平周期 (t_{LOW}) 时，才必须满足该最大值条件。如果时钟延长 SCL，数据必须在建立时间内保持有效，之后才能释放时钟。

SDA 上升沿通常为最坏情况。在这种情况下，上述公式变成如下形式：

SDADEL ≤ {$t_{VD;DAT\ (max)}$ - $t_{r\ (max)}$ - $t_{AF\ (max)}$ - [(DNF + 4) x t_{I2CCLK}]} / {(PRESC + 1) x t_{I2CCLK}}。

注意： *NOSTRETCH=0 时会违反该条件，这是因为器件会根据 SCLDEL 值来延长 SCL 低电平时间，以保证建立时间。*

有关 t_f、t_r、$t_{HD;DAT}$ 和 $t_{VD;DAT}$ 标准值的信息，请参见*表 91*。

- 在 t_{SDADEL} 后，或在因数据未写入 I2C_TXDR 寄存器而导致从器件必须延长时钟时发送 SDA 输出后，SCL 线会在建立时间内保持低电平。该建立时间为 t_{SCLDEL} = (SCLDEL+ 1) x t_{PRESC}，其中
  t_{PRESC} = (PRESC+ 1) x t_{I2CCLK}。t_{SCLDEL} 会影响建立时间 $t_{SU;DAT}$。

要桥接 SDA 跳变（上升沿通常为最坏情况）的未定义区域，编程 SCLDEL 时必须遵循以下条件：

{[$t_{r\ (max)}$ + $t_{SU;DAT\ (min)}$] / [(PRESC+ 1)] x t_{I2CCLK}]} - 1 ≤ SCLDEL

有关 t_r 和 $t_{SU;DAT}$ 标准值的信息，请参见*表 91*。

将使用的 SDA 和 SCL 跳变时间值就是应用中的值。使用最大值而非标准值会增加 SDADEL 和 SCLDEL 计算的约束条件，但能够确保任意应用的特性。

注意： *在发送和接收模式下，对于每个时钟脉冲，检测到 SCL 下降沿后，I2C 主器件或从器件会至少在 [(SDADEL+SCLDEL+1) x (PRESC+1) + 1] x t_{I2CCLK} 期间内延长 SCL 低电平时间。在发送模式下，如果 SDADEL 计数器计数结束后数据还未写入 I2C_TXDR，则 I2C 会继续延长 SCL 低电平时间，直到写入下一个数据。随后，会将新数据 MSB 发送到 SDA 输出，SCLDEL 计数器将开始计数，同时会继续延长 SCL 低电平时间以确保提供充足的数据建立时间。*

如果从模式下 NOSTRETCH = 1，则 SCL 不会延长。因此，编程 SDADEL 时还必须确保提供充足的建立时间。

表 91. I²C-SMBus 规范数据建立和保持时间

符号	参数	标准模式 (Sm)		快速模式 (Fm)		超快速模式 (Fm+)		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
$t_{HD;DAT}$	数据保持时间	0	-	0	-	0	-	0.3	-	µs
$t_{VD;DAT}$	数据有效时间	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	数据建立时间	250	-	100	-	50	-	250	-	ns
t_r	SDA 和 SCL 信号的上升时间	-	1000	-	300	-	120	-	1000	
t_f	SDA 和 SCL 信号的下降时间	-	300	-	300	-	120	-	300	

此外，在主模式下，必须通过编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLH[7:0] 和 SCLL[7:0] 位域来配置 SCL 时钟的高电平和低电平。

- 当内部检测到 SCL 下降沿时，会在释放 SCL 输出之前插入一段延时。
  该延时为 $t_{SCLL} = (SCLL + 1) \times t_{PRESC}$，其中 $t_{PRESC} = (PRESC + 1) \times t_{I2CCLK}$。
  t_{SCLL} 会影响 SCL 低电平时间 t_{LOW}。
- 当内部检测到 SCL 上升沿时，会在将 SCL 输出强制为低电平之前插入一段延时。该延时为 $t_{SCLH} = (SCLH + 1) \times t_{PRESC}$，其中
  $t_{PRESC} = (PRESC+ 1) \times t_{I2CCLK}$. t_{SCLH} 会影响 SCL 高电平时间 t_{HIGH}。

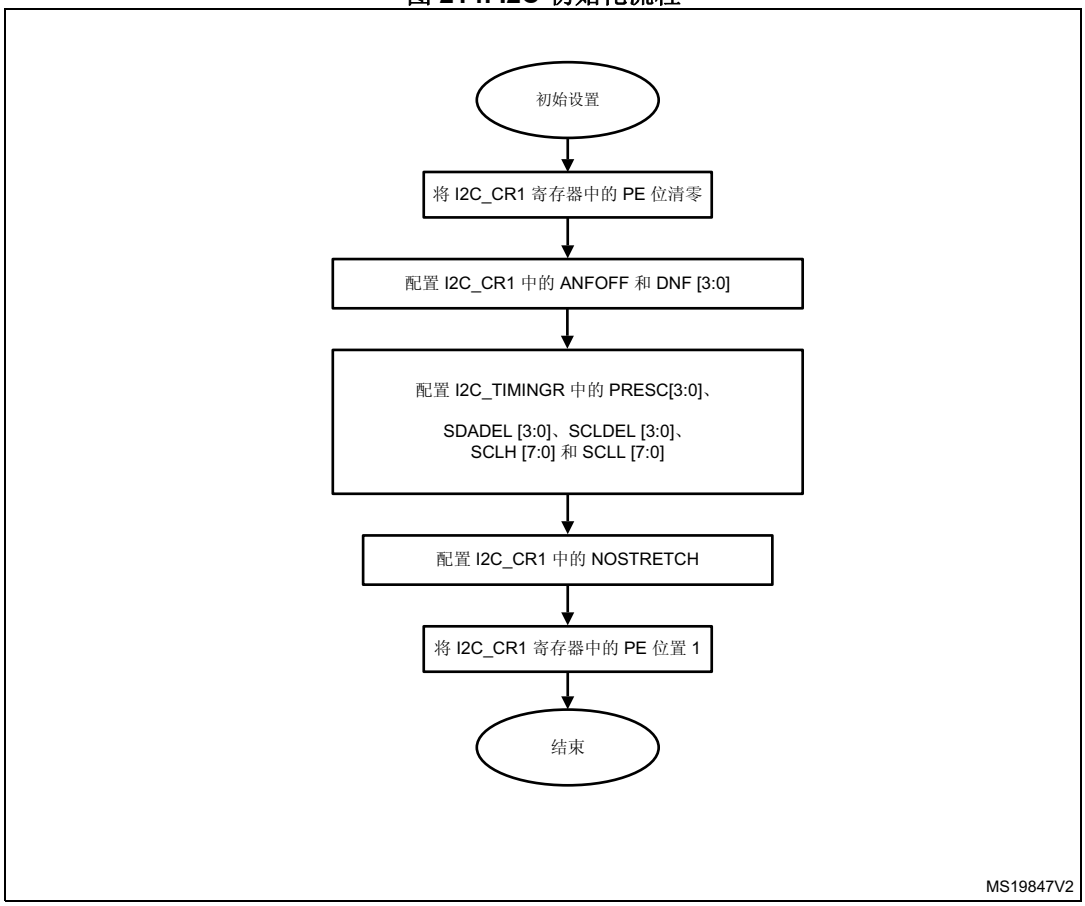
更多详细信息，请参见 *I2C 主模式初始化*。

小心： 使能 I2C 后，不允许更改时序配置。

此外，还必须在使能外设之前配置 I2C 从 NOSTRETCH 模式。更多详细信息，请参见 *I2C 从模式初始化*。

小心： 使能 I2C 后，不允许更改 NOSTRETCH 配置。

图 214. I2C 初始化流程



23.4.6 软件复位

可通过将 I2C_CR1 寄存器中的 PE 位清零来执行软件复位。在这种情况下，I2C 线 SCL 和 SDA 被释放。内部状态机复位，通信控制位和状态位恢复为其复位值。配置寄存器不受影响。

下面列出了受影响的寄存器位：

1. I2C_CR2 寄存器：START、STOP 和 NACK
2. I2C_ISR 寄存器：BUSY、TXE、TXIS、RXNE、ADDR、NACKF、TCR、TC、STOPF、BERR、ARLO 和 OVR

支持 SMBus 功能时还会影响到以下寄存器位：

1. I2C_CR2 寄存器：PECBYTE
2. I2C_ISR 寄存器：PECERR、TIMEOUT 和 ALERT

必须使 PE 保持低电平持续至少三个 APB 时钟周期，才能成功执行软件复位。使用以下软件写序列可确保这一点：

1. 写入 PE = 0
2. 检查 PE = 0
3. 写入 PE = 1

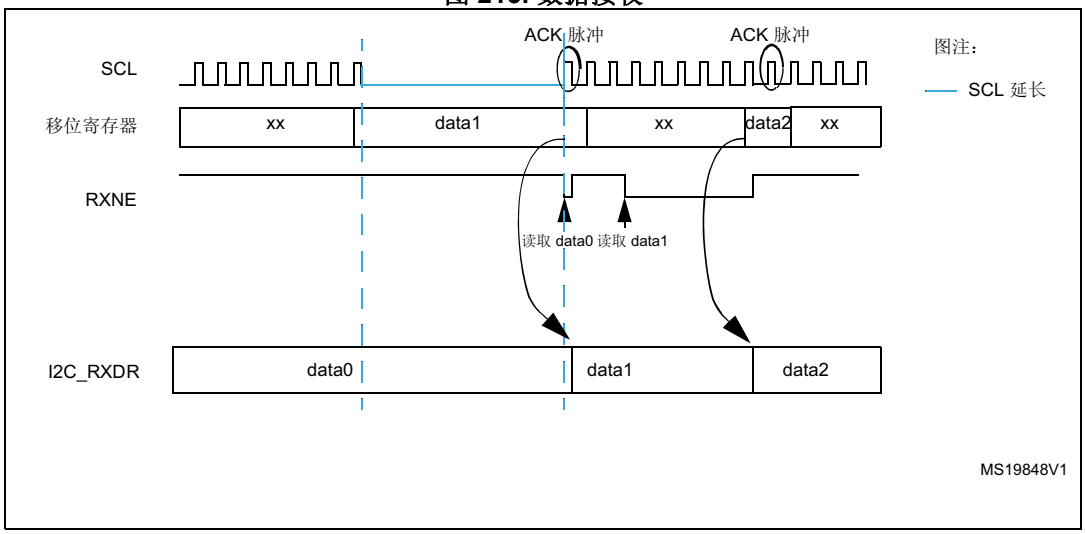
23.4.7 数据传输

数据传输由发送和接收数据寄存器以及移位寄存器来管理。

接收

SDA 输入填充移位寄存器。在第八个 SCL 脉冲后（接收到完整的数据字节时），如果 I2C_RXDR 寄存器为空 (RXNE=0)，则移位寄存器的内容会复制到其中。如果 RXNE = 1（意味着尚未读取上一次接收到的数据字节），则将延长 SCL 线的低电平时间，直到读取了 I2C_RXDR 为止。在第八个和第九个 SCL 脉冲之间（应答脉冲之前）插入一段延长的时间。

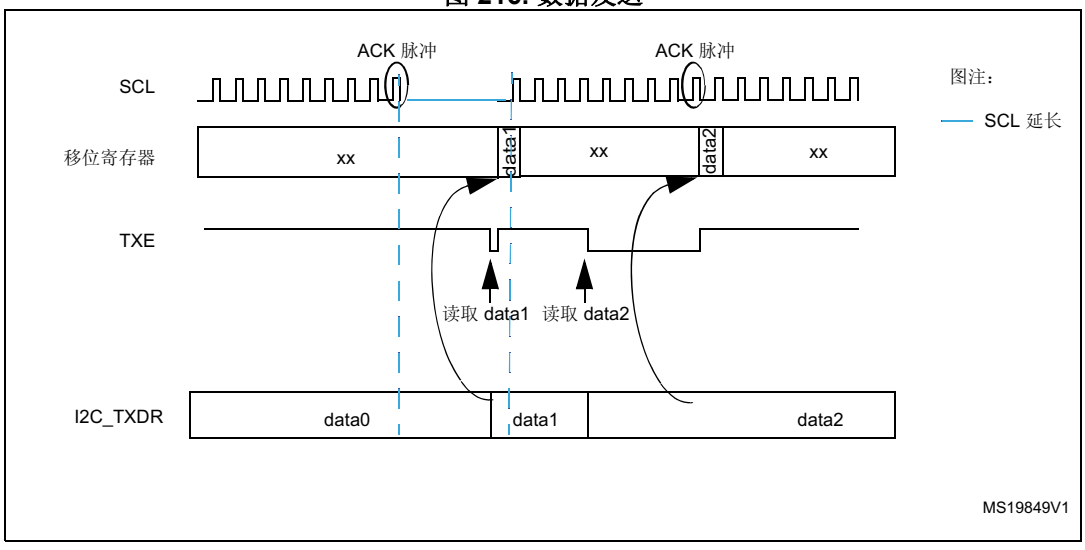
图 215. 数据接收



发送

如果 I2C_TXDR 寄存器不为空 (TXE = 0)，则其内容会在第九个 SCL 脉冲（应答脉冲）后复制到移位寄存器中。然后移位寄存器的内容会移出到 SDA 线上。如果 TXE = 1（意味着 I2C_TXDR 内尚未写入任何数据），则将延长 SCL 线的低电平时间，直到写入了 I2C_TXDR 为止。在第九个 SCL 脉冲后进行延长。

图 216. 数据发送



硬件传输管理

I2C 在硬件中内置了字节计数器，以便在下列各种模式下管理字节传输和结束通信：

- 主模式下生成 NACK、STOP 和 ReSTART
- 从接收器模式下控制 ACK 是否发出
- SMBus 模式下生成/校验 PEC

字节计数器通常在主模式下使用。在从模式下，字节计数器默认为禁止状态，但可以通过软件来使能，方法是将 I2C_CR1 寄存器中的 SBC（从字节控制）位置 1。

待传输的字节数在 I2C_CR2 寄存器的 NBYTES[7:0] 位域中进行编程。如果待传输的字节数 (NBYTES) 大于 255，或者接收方希望控制是否对接收到的数据字节进行应答，则必须选择重载模式，方法是将 I2C_CR2 寄存器的 RELOAD 位置 1。在该模式下，完成 NBYTES 中所编程字节数的数据传输之后，TCR 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TCR 标志置 1，SCL 便会延长。向 NBYTES 写入一个非零值时，TCR 由软件清零。

在 NBYTES 中设置最后一次传输的字节数前，必须将 RELOAD 位清零。

当主模式下 RELOAD = 0 时，可在以下两种模式下使用计数器：

- **自动结束模式**（I2C_CR2 寄存器中的 AUTOEND = “1”）。在该模式下，一旦完成 NBYTES[7:0] 位域中所编程字节数的数据传输，主器件便会自动发送停止位。
- **软件结束模式**（I2C_CR2 寄存器中的 AUTOEND = “0”）在该模式下，一旦完成 NBYTES[7:0] 位域中所编程字节数的数据传输，TC 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TC 标志置 1，SCL 信号便会延长，需要软件介入操作。当软件将 I2C_CR2 寄存器中的起始位或停止位置 1 时，TC 标志将被清零。当主器件要发送重复起始位时，必须使用该模式。

小心： 当 RELOAD 位置 1 时，AUTOEND 位将不起作用。

表 92. I2C 配置

功能	SBC 位	RELOAD 位	AUTOEND 位
主 Tx/Rx NBYTES + STOP	x	0	1
主 Tx/Rx + NBYTES + RESTART	x	0	0
从 Tx/Rx，接收的所有字节都要回复应答	0	x	x
具有 ACK 控制的从 Rx	1	1	x

23.4.8 I2C 从模式

I2C 从模式初始化

要在从模式下工作，用户必须至少使能一个从地址。可使用 I2C_OAR1 和 I2C_OAR2 这两个寄存器来编程自身从地址 OA1 和 OA2。

- OA1 既可配置为 7 位寻址模式（默认），也可通过将 I2C_OAR1 寄存器的 OA1MODE 位置 1 配置为 10 位寻址模式。

  通过将 I2C_OAR1 寄存器中的 OA1EN 位置 1 来使能 OA1。
- 如果需要额外的从地址，可配置第二个从地址 OA2。将 I2C_OAR2 寄存器的 OA2MSK[2:0] 位置 1 最多可屏蔽 7 个 OA2 LSB。因此，当 OA2MSK 配置为 1 到 6 时，将分别只有 OA2[7:2]、OA2[7:3]、OA2[7:4]、OA2[7:5]、OA2[7:6] 或 OA2[7] 与接收到的地址作比较。只要 OA2MSK 不等于 0，OA2 的地址比较器便会排除 I2C 保留地址（0000 XXX 和 1111 XXX），这些地址将不会得到应答。如果 OA2MSK = 7，接收到的所有 7 位地址（保留地址除外）均得到应答。OA2 始终为 7 位地址。

如果这些保留地址在 I2C_OAR1 或 I2C_OAR2 寄存器中进行了编程并且 OA2MSK = 0，则它们可以在通过特定使能位使能后得到应答。

通过将 I2C_OAR2 寄存器中的 OA2EN 位置 1 来使能 OA2。

- 通过将 I2C_CR1 寄存器中的 GCEN 位置 1 来使能广播呼叫地址。

当通过 I2C 的其中一个使能地址来寻址到该 I2C 设备时，ADDR 中断状态标志将置 1，并且 ADDRIE 位置 1 时将生成中断。

默认情况下，从器件使用其时钟延长功能（即必要时延长 SCL 信号的低电平时间）来为软件操作的执行提供时机。如果主器件不支持时钟延长，则必须对 I2C 进行如下配置：将 I2C_CR1 寄存器中 NOSTRETCH 位置 1。

接收到 ADDR 中断后，如果使能多个地址，则用户必须读取 I2C_ISR 寄存器中的 ADDCODE[6:0] 位，以确定是哪个地址匹配。还必须检查 DIR 标志，以获悉传输方向。

带时钟延长的从模式 (NOSTRETCH = 0)

在默认模式下，I2C 从器件会在以下情况下延长 SCL 时钟：

- ADDR 标志置 1 时：接收到的地址与其中一个使能的从地址匹配。通过软件将 ADDRCF 位置 1 以清零 ADDR 标志时，将释放该时钟延展。
- 发送时，前一次数据传输已完成但 I2C_TXDR 寄存器中未写入任何新数据，或者 ADDR 标志清零 (TXE = 1) 时未写入第一个数据字节。在 I2C_TXDR 寄存器中写入数据时，将释放该时钟延展。
- 接收时，尚未读取 I2C_RXDR 寄存器但新的数据接收已完成。读取 I2C_RXDR 时，将释放该时钟延展。
- 当从器件字节控制模式和重载模式（SBC = 1 且 RELOAD = 1）下 TCR = 1 时，这意味着最后一个数据字节已完成传输。通过向 NBYTES[7:0] 位域写入一个非零值以将 TCR 清零时，将释放该时钟延展。
- 检测到 SCL 下降沿后，I2C 会在 $[(SDADEL + SCLDEL + 1) \times (PRESC+ 1) + 1] \times t_{I2CCLK}$ 期间内延长 SCL 低电平时间。

不带时钟延长的从模式 (NOSTRETCH = 1)

当 I2C_CR1 寄存器中的 NOSTRETCH = 1 时，I2C 从器件不会延长 SCL 信号。

- ADDR 标志置 1 时，不会延长 SCL 时钟。
- 发送时，必须在与发送数据对应的第一个 SCL 脉冲出现之前，向 I2C_TXDR 寄存器写入数据。否则，会发生下溢，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。当第一次数据发送开始而 STOPF 位仍置 1（尚未清零）时，OVR 标志也将置 1。因此，如果写入下一次传输要发送的第一个数据后才清零上一次传输的 STOPF 标志，则应提供 OVR 状态，甚至对于待发送的第一个数据也是如此。
- 接收时，必须在下一个数据字节的第九个 SCL 脉冲（ACK 脉冲）出现之前，从 I2C_RXDR 寄存器读取数据。否则，会发生上溢，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

从器件字节控制模式

要在从接收模式下实现字节 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从器件字节控制模式。这样符合 SMBus 标准。

要在从接收模式下实现字节 ACK 控制，必须选择重载模式 (RELOAD = 1)。要控制每个字节，必须在 ADDR 中断子程序中将 NBYTES 初始化为 0x1，并在每接收一个字节后将 NBYTE 重载为 0x1。接收到字节后，TCR 位将置 1，从而延长 SCL 信号的第八个和第九个脉冲之间的低电平时间。用户可以从 I2C_RXDR 寄存器中读取数据，然后通过配置 I2C_CR2 寄存器中的 ACK 位来决定是否应答。通过将 NBYTES 编程为非零值来释放 SCL 延长：发送应答或不应答信号，然后可继续接收下一个字节。

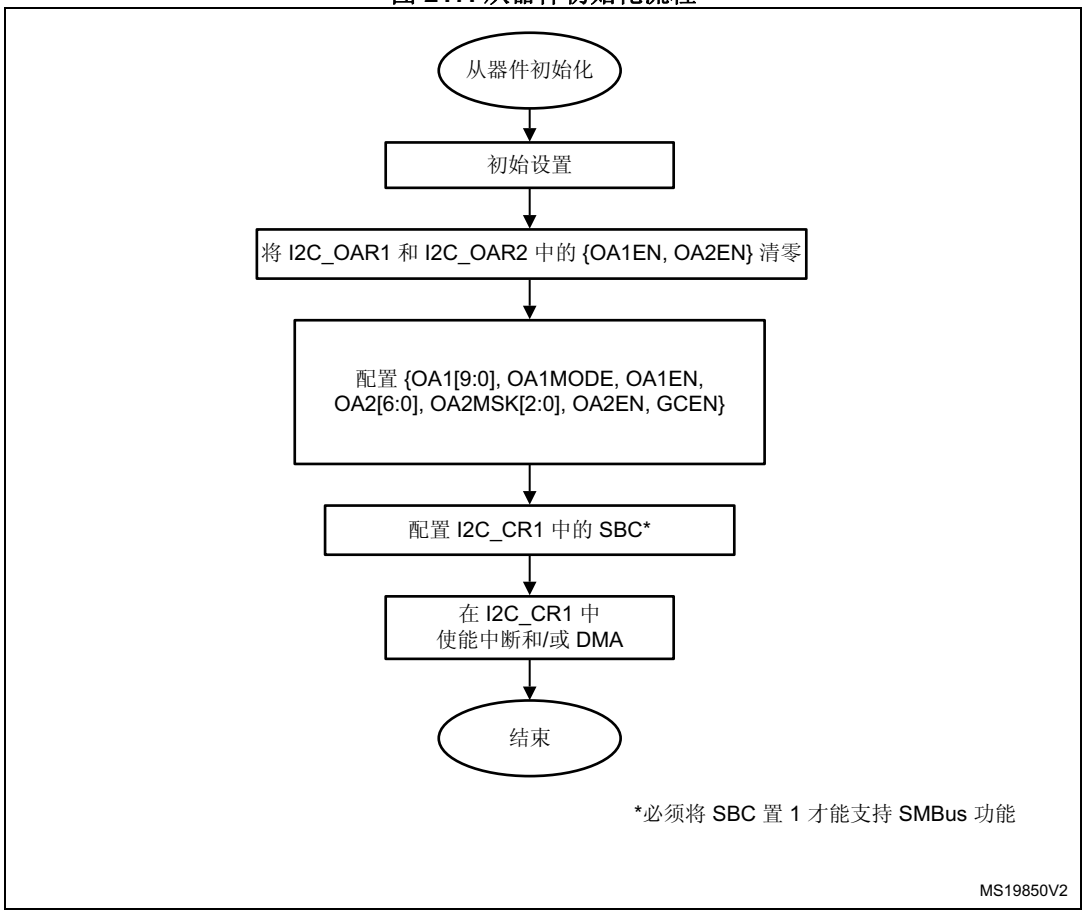
NBYTES 可加载大于 0x1 的值，在这种情况下，接收流在 NBYTES 个数据接收期间是连续的。

注意： *SBC 位只能在 I2C 被禁止时、从器件不被寻址时或 ADDR = 1 时配置。*

ADDR = 1 或 TCR = 1 时，可以更改 RELOAD 位的值。

小心： 从器件字节控制模式与 NOSTRETCH 模式不兼容。不允许在 NOSTRETCH = 1 时将 SBC 位置 1。

图 217. 从器件初始化流程



从发送器

当 I2C_TXDR 寄存器为空时，将生成发送中断状态 (TXIS)。如果 I2C_CR1 寄存器中的 TXIE 位置 1，将生成中断。

I2C_TXDR 寄存器中写入待发送的下一个数据字节时，TXIS 位将被清零。

接收到 NACK 时，I2C_ISR 寄存器中的 NACKF 位将置 1，如果 I2C_CR1 寄存器中的 NACKIE 位置 1，还将生成中断。从器件自动释放 SCL 和 SDA 线，以使主器件执行停止或重复起始位的发送。收到 NACK 时，TXIS 位不会置 1。

当接收到停止位且 I2C_CR1 寄存器中的 STOPIE 位置 1 时，I2C_ISR 寄存器中的 STOPF 标志将置 1 并且会生成中断。在大多数应用中，SBC 位通常编程为“0”。在这种情况下，如果接收到从地址 (ADDR=1) 时 TXE = 0，用户可以选择发送 I2C_TXDR 寄存器的内容作为第一个数据字节，也可以选择通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器以编程新的数据字节。

在从器件字节控制模式 (SBC = 1) 下，必须在地址匹配中断子程序 (ADDR = 1) 中向 NBYTES 写入待发送数据的个数。在这种情况下，传输期间 TXIS 事件的数量对应于 NBYTES 中编程的值。

小心： 如果 NOSTRETCH = 1，当 ADDR 标志置 1 时不会延长 SCL 时钟，因此用户无法在 ADDR 子程序中刷新 I2C_TXDR 寄存器的内容，从而编程第一个数据字节。必须在 I2C_TXDR 寄存器中预编程待发送的第一个数据字节：

- 该数据可以是前一个传输消息的最后一个 TXIS 事件中写入的数据。
- 如果该数据字节不是待发送的数据字节，可通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器，从而编程新的数据字节。必须仅在执行完这些操作后再清零 STOPF 位，以确保在地址应答之后第一次数据传输之前执行这些操作。

  如果第一次数据传输开始时 STOPF 仍置 1，则将生成下溢错误（OVR 标志置 1）。

  如果需要 TXIS 事件（发送中断或发送 DMA 请求），用户必须将 TXE 位和 TXIS 位均置 1，以便生成 TXIS 事件。

图 218. I2C 从发送器的传输序列流程 (NOSTRETCH = 0)

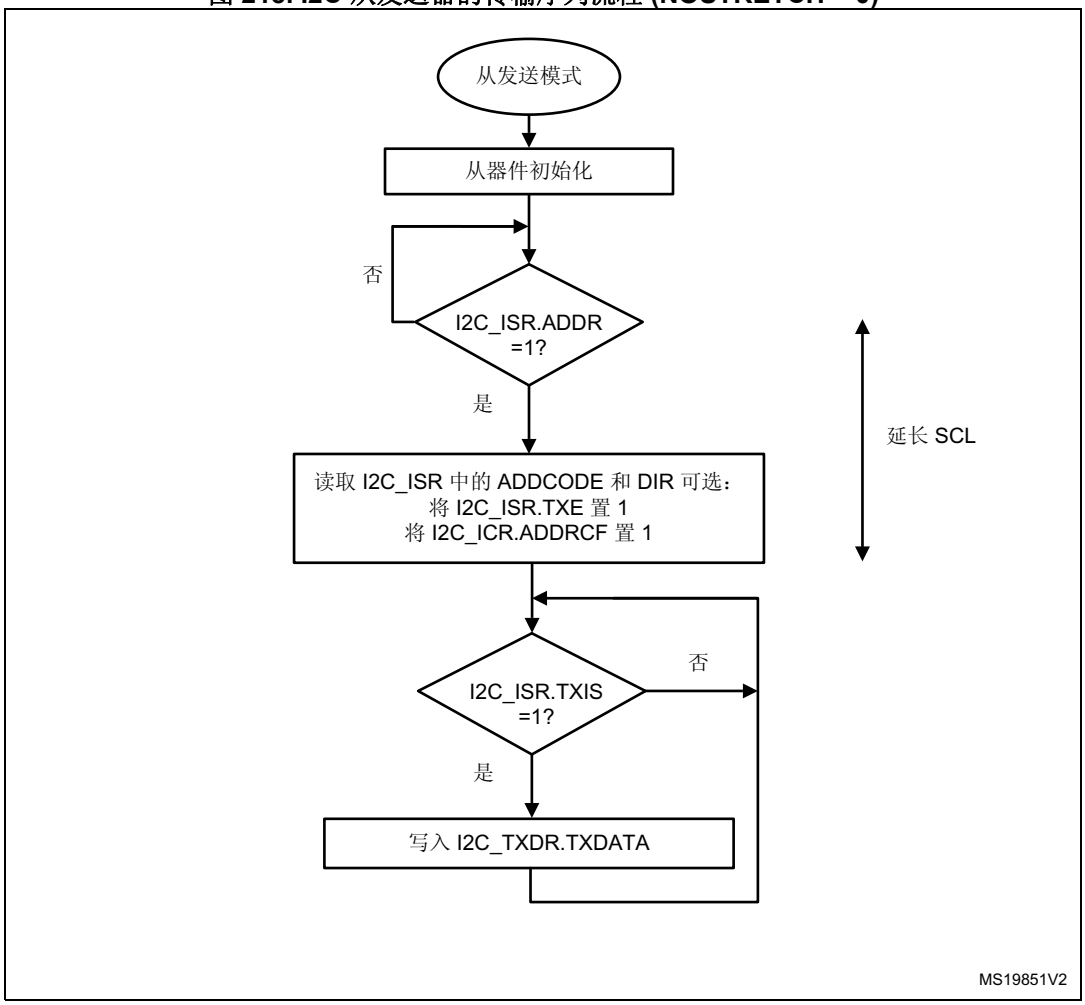


图 219. I2C 从发送器的传输序列流程 (NOSTRETCH = 1)

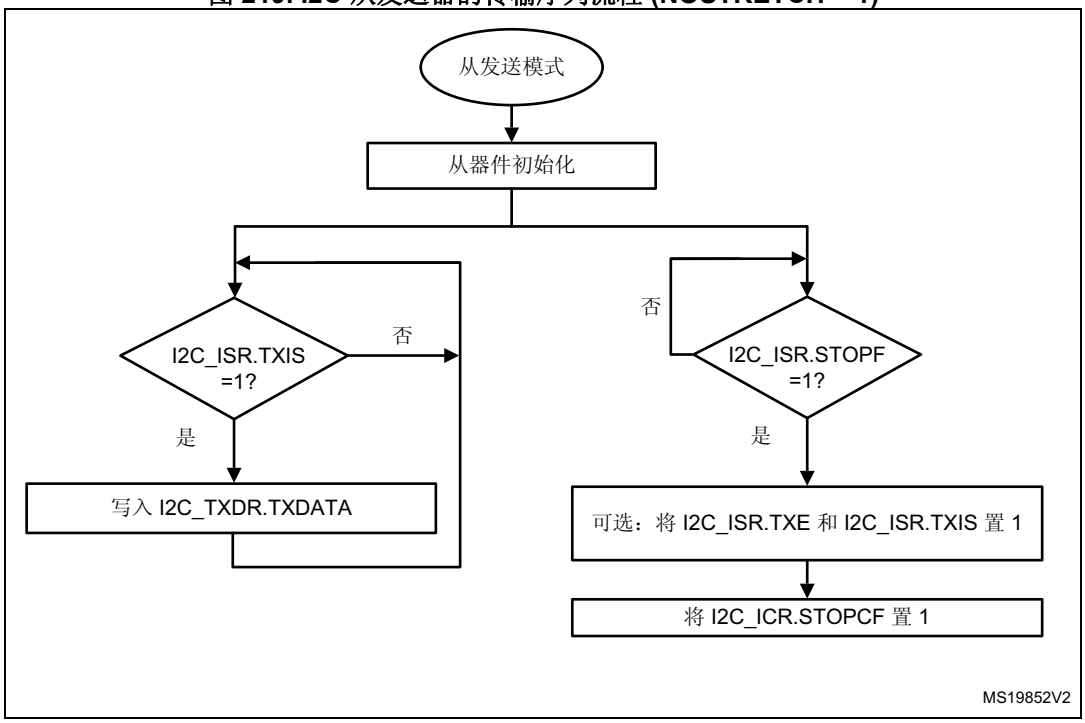
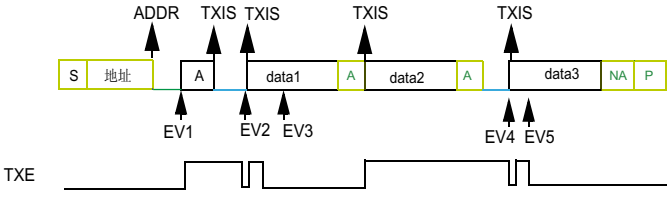


图 220. I2C 从发送器的传输总线图（仅限强制性事件）

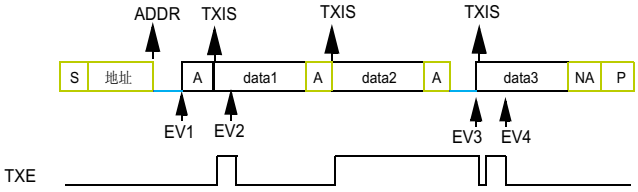
示例：I2C 从发送器 3 个字节，刷新第 1 个数据，NOSTRETCH=0：



图注：
发送
接收
SCL 延长

EV1：ADDR ISR：检查 ADDCODE 和 DIR，将 TXE 置 1，将 ADDRCF 置 1
EV2：TXIS ISR：写入 data1
EV3：TXIS ISR：写入 data2
EV4：TXIS ISR：写入 data3
EV5：TXIS ISR：写入 data4（不发送）

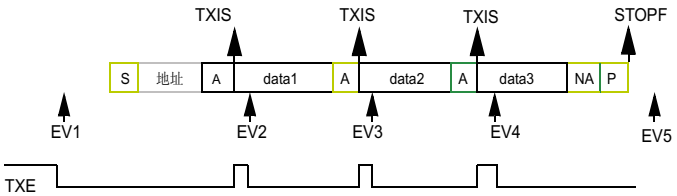
示例：I2C 从发送器 3 个字节，不刷新第 1 个数据，NOSTRETCH=0：



图注：
发送
接收
SCL 延长

EV1：ADDR ISR：检查 ADDCODE 和 DIR，将 ADDRCF 置 1
EV2：TXIS ISR：写入 data2
EV3：TXIS ISR：写入 data3
EV4：TXIS ISR：写入 data4（不发送）

示例：I2C 从发送器 3 个字节，NOSTRETCH=1：



图注：
发送
接收
SCL 延长

EV1：写入 data1
EV2：TXIS ISR：写入 data2
EV3：TXIS ISR：写入 data3
EV4：TXIS ISR：写入 data4（不发送）
EV5：STOPF ISR：（可选：将 TXE 和 TXIS 置 1），将 STOPCF 置 1

MS19853V2

从接收器

当 I2C_RXDR 满时，I2C_ISR 中的 RXNE 将置 1，如果 I2C_CR1 中的 RXIE 置 1，还将生成中断。读取 I2C_RXDR 时，将清零 RXNE。

接收到停止条件且 I2C_CR1 寄存器中的 STOPIE 置 1 时，I2C_ISR 中的 STOPF 将置 1 并且会生成中断。

图 221. 从接收器的传输序列流程 (NOSTRETCH = 0)

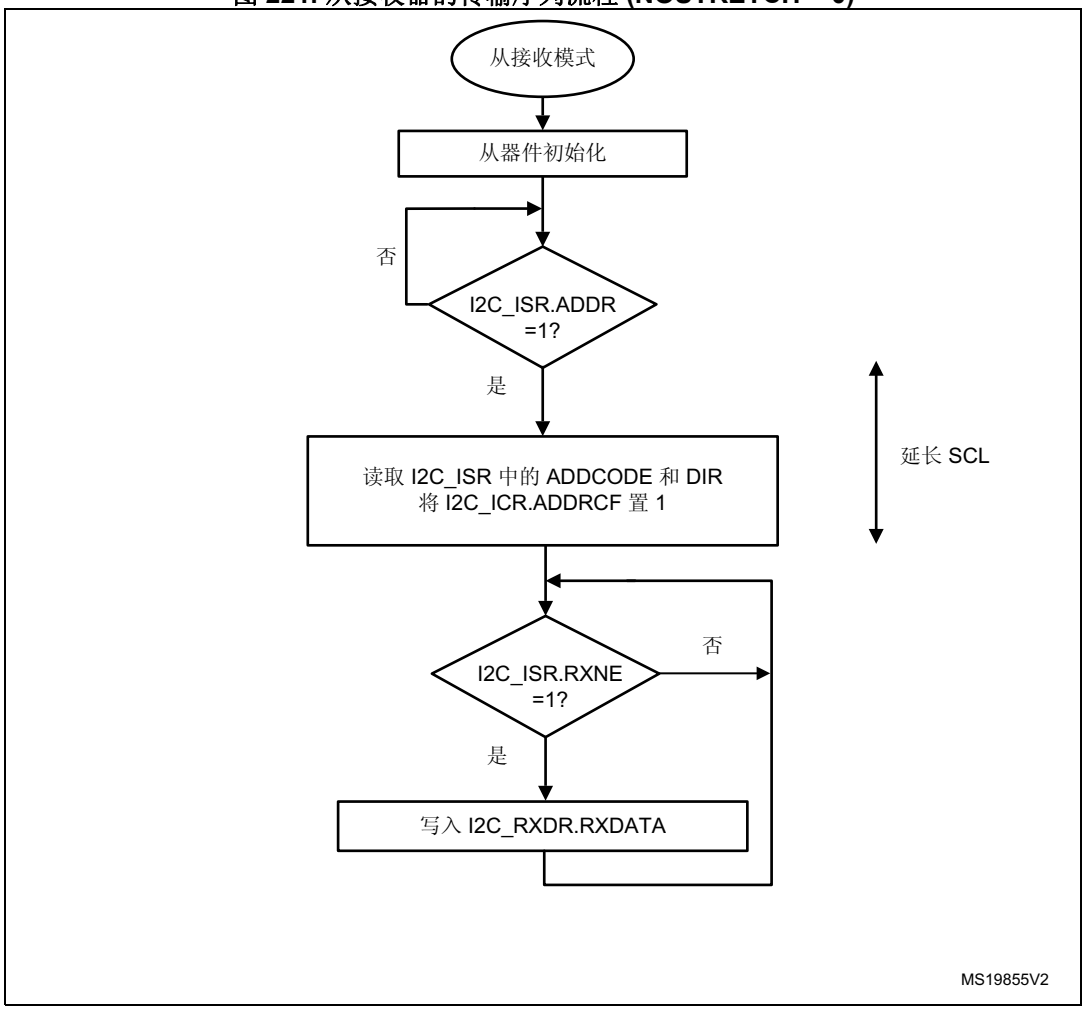


图 222. 从接收器的传输序列流程 (NOSTRETCH = 1)

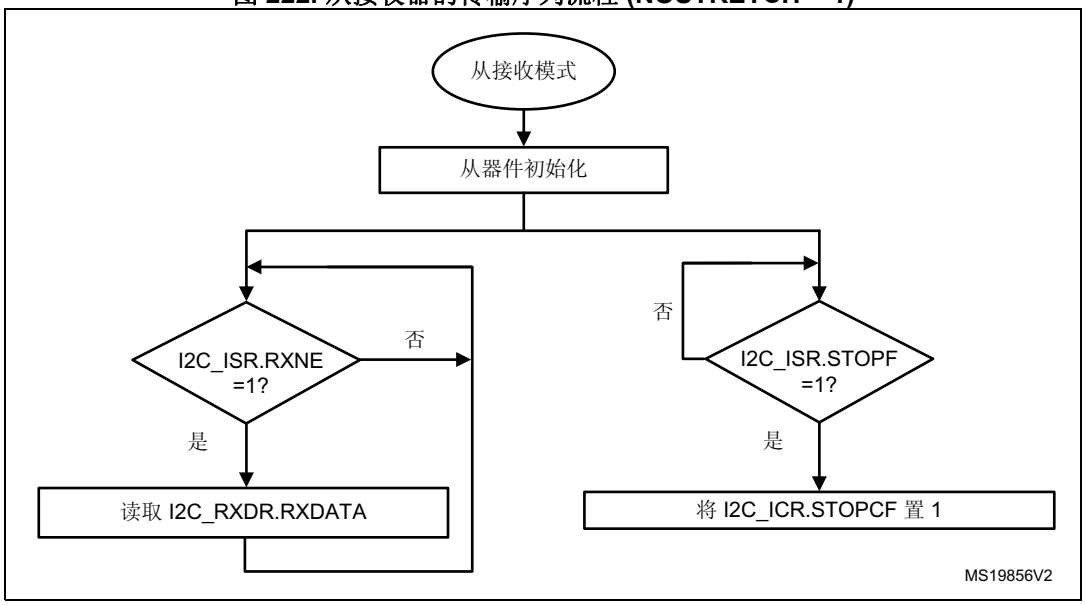
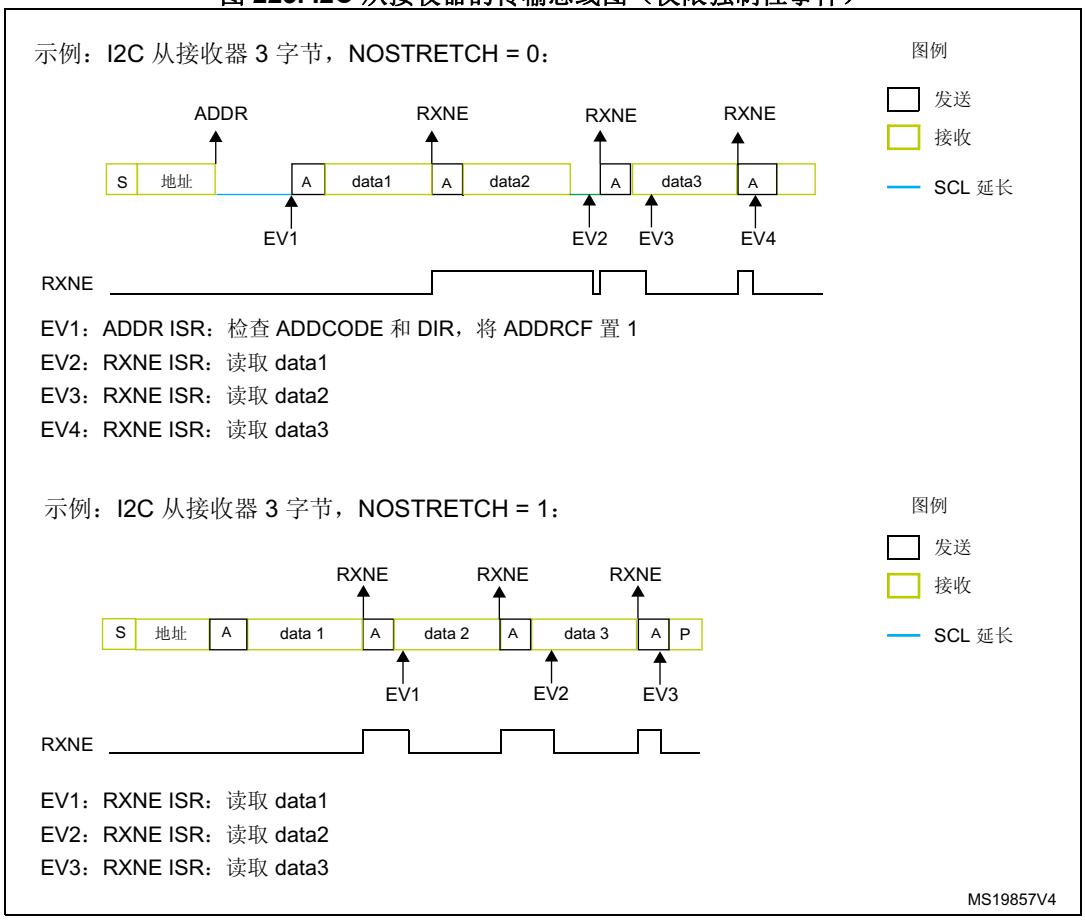


图 223. I2C 从接收器的传输总线图（仅限强制性事件）



23.4.9 I2C 主模式

I2C 主模式初始化

使能外设前，必须通过设置 I2C_TIMINGR 寄存器中的 SCLH 和 SCLL 位来配置 I2C 主时钟。

STM32CubeMX 工具计算 I2C_TIMINGR 内容并在 I2C 配置窗口中进行显示。

为了支持多主环境和从时钟延长，I2C 实现了时钟同步机制。

为了实现时钟同步，需执行以下操作：

- 使用 SCLL 计数器从 SCL 低电平内部检测开始对时钟的低电平进行计数。
- 使用 SCLH 计数器从 SCL 高电平内部检测开始对时钟的高电平进行计数。

I2C 经过 t_{SYNC1} 延时后检测其自身的 SCL 低电平，该延时取决于 SCL 下降沿、SCL 输入噪声滤波器（模拟 + 数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLL 计数器达到 I2C_TIMINGR 寄存器的 SCLL[7:0] 位中编程的值，I2C 便会将 SCL 释放为高电平。

I2C 经过 t_{SYNC2} 延时后检测其自身的 SCL 高电平，该延时取决于 SCL 上升沿、SCL 输入噪声滤波器（模拟 + 数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLH 计数器达到 I2C_TIMINGR 寄存器的 SCLH[7:0] 位中编程的值，I2C 便会使 SCL 变为低电平。

因此，主时钟周期为：

$$t_{SCL} = t_{SYNC1} + t_{SYNC2} + \{[(SCLH+ 1) + (SCLL+ 1)] \times (PRESC+ 1) \times t_{I2CCLK}\}$$

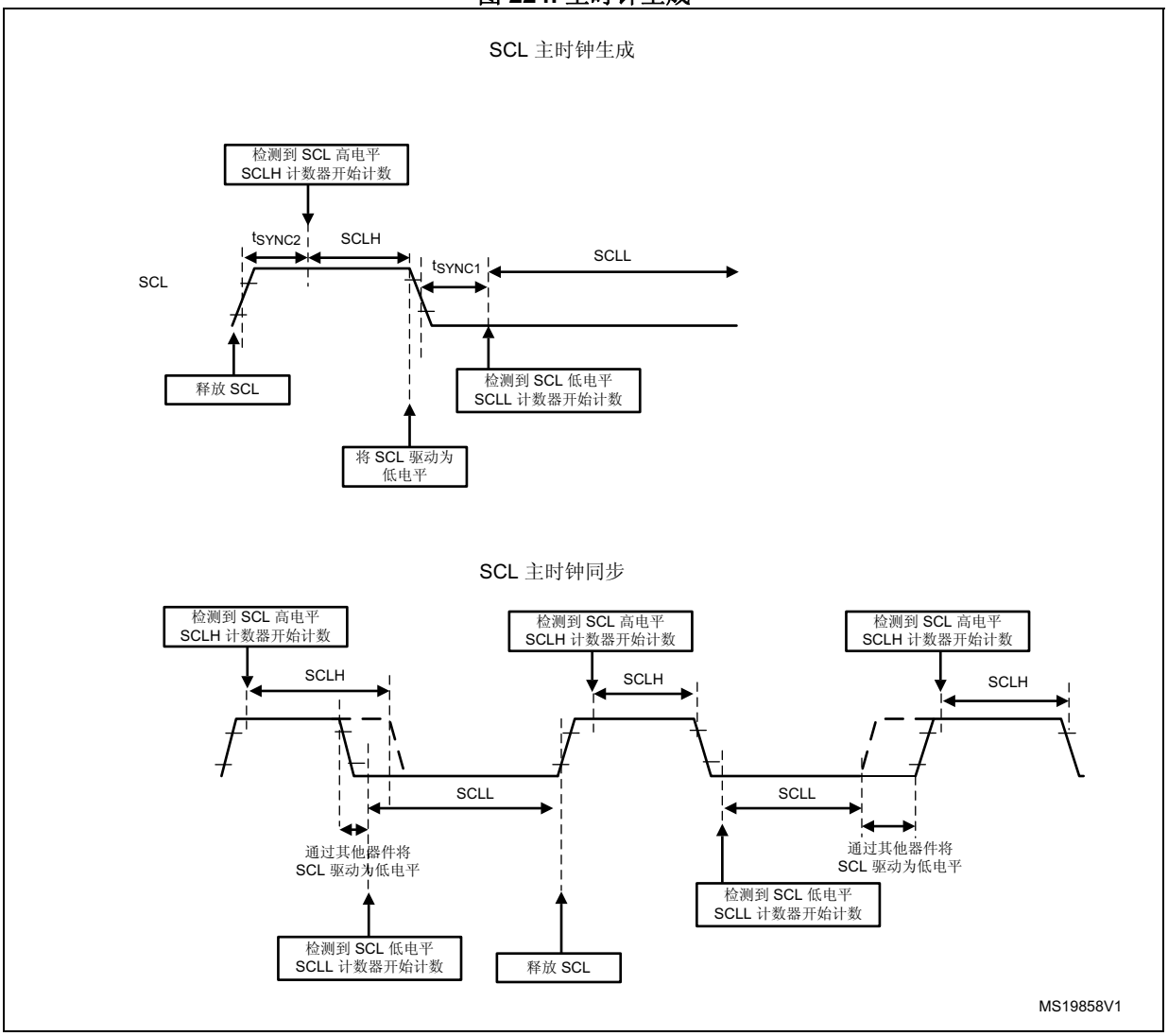
t_{SYNC1} 的持续时间取决于以下参数：

- SCL 下降斜率
- 模拟滤波器（使能时）引入的输入延时。
- 数字滤波器（使能时）引入的输入延时：DNF x t_{I2CCLK}
- SCL 与 I2CCLK 时钟建立同步而产生的延时（两到三个 I2CCLK 周期）

t_{SYNC2} 的持续时间取决于以下参数：

- SCL 上升斜率
- 模拟滤波器（使能时）引入的输入延时。
- 数字滤波器（使能时）引入的输入延时：DNF x t_{I2CCLK}
- SCL 与 I2CCLK 时钟建立同步而产生的延时（两到三个 I2CCLK 周期）

图 224. 主时钟生成


SCL 主时钟生成
检测到 SCL 高电平
SCLH 计数器开始计数
tSYNC2
SCLH
tSYNC1
SCLL
SCL
释放 SCL
检测到 SCL 低电平
SCLL 计数器开始计数
将 SCL 驱动为
低电平
SCL 主时钟同步
检测到 SCL 高电平
SCLH 计数器开始计数
SCLH
SCLL
通过其他器件将
SCL 驱动为低电平
检测到 SCL 低电平
SCLL 计数器开始计数
释放 SCL
MS19858V1

小心： 为了符合 I^2C 或 SMBus 规范，主时钟必须遵循下表中给出的时序。

表 93. I^2C-SMBus 规范时钟时序

符号	参数	标准模式 (Sm)		快速模式 (Fm)		超快速模式 (Fm+)		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
f_{SCL}	SCL 时钟频率	-	100	-	400	-	1000	-	100	kHz
$t_{HD:STA}$	（重复）起始条件的保持时间	4.0	-	0.6	-	0.26	-	4.0	-	μs
$t_{SU:STA}$	重复起始条件的建立时间	4.7	-	0.6	-	0.26	-	4.7	-	
$t_{SU:STO}$	停止条件的建立时间	4.0	-	0.6	-	0.26	-	4.0	-	
t_{BUF}	停止条件和起始条件之间的总线空闲时间	4.7	-	1.3	-	0.5	-	4.7	-	
t_{LOW}	SCL 时钟的低电平周期	4.7	-	1.3	-	0.5	-	4.7	-	
t_{HIGH}	SCL 时钟的高电平周期	4.0	-	0.6	-	0.26	-	4.0	50	
t_r	SDA 和 SCL 信号的上升时间	-	1000	-	300	-	120	-	1000	ns
t_f	SDA 和 SCL 信号的下降时间	-	300	-	300	-	120	-	300	

注意： *SCLL 还用于生成 t_{BUF} 和 $t_{SU:STA}$ 时序，SCLH 还用于生成 $t_{HD:STA}$ 和 $t_{SU:STO}$ 时序。*

有关 I2C_TIMINGR 设置与 I2CCLK 频率的示例，请参见*第 23.4.10 节*。

主模式通信初始化（地址阶段）

要发起通信，用户必须在 I2C_CR2 寄存器中为寻址的从器件编程以下参数：

- 寻址模式（7 位或 10 位）：ADD10
- 待发送的从地址：SADD[9:0]
- 传输方向：RD_WRN
- 读取 10 位地址时：HEAD10R 位。必须对 HEAD10R 进行相应配置，以指示传输方向变化时必须发送完整的地址序列，还是只发送地址头。
- 待传输的字节数：NBYTES[7:0]。如果字节数等于或大于 255，则初始化时必须将 NBYTES[7:0] 填充为 0xFF。

然后，用户必须将 I2C_CR2 寄存器中的 START 位置 1。START 位置 1 时，不允许更改上述所有位。

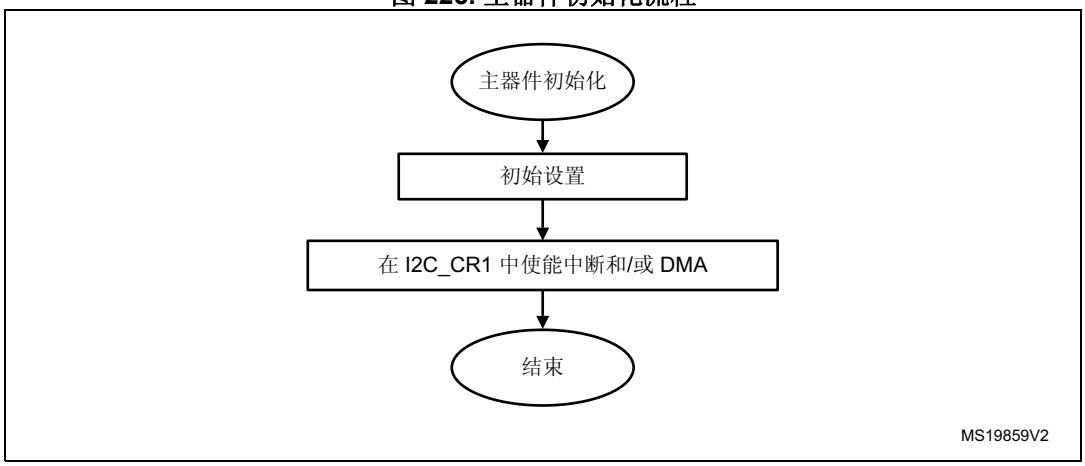
之后，当主器件检测到总线空闲 (BUSY = 0) 时，它会在经过 t_{BUF} 的延时后自动发送起始位，随后发出从器件地址。

仲裁丢失时，主器件将自动切换回从模式，如果作为从器件被寻址，还可对其自身地址进行应答。

注意： *无论接收到的应答值为何，只要已在总线上发送从地址，START 位便会由硬件复位。如果仲裁丢失，START 位也会由硬件复位。*
在 10 位寻址模式下，如果从器件不对从地址的前 7 位进行应答，则主器件将自动重新启动从地址发送，直至接收到 ACK。在这种情况下，如果从从器件接收到 NACK，则必须将 ADDRCF 置 1，以停止发送从地址。
如果当 START 位置 1 时，I2C 作为从器件 (ADDR = 1) 被寻址，则 I2C 将切换为从模式，START 位将清零。

注意： *该步骤同样适用于重复起始位。在这种情况下，BUSY = 1。*

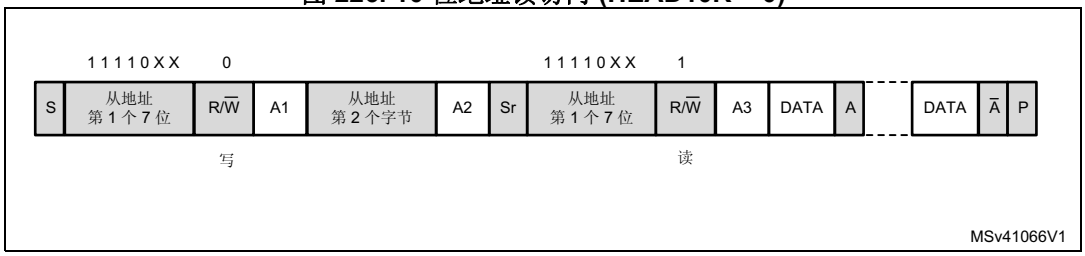
图 225. 主器件初始化流程



主机接收器寻址 10 位地址从器件的初始化过程

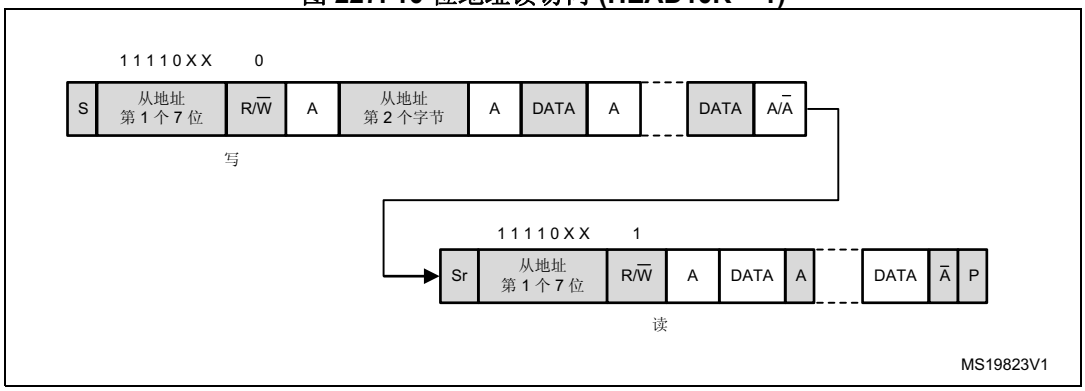
- 如果从地址采用 10 位格式，用户可选择将 I2C_CR2 寄存器中的 HEAD10R 位清零来发送完整的读序列。在这种情况下，主器件会在 START 位置 1 后自动发送以下完整序列：（重复）起始位 + 带写方向的从器件 10 位地址头字节 + 从器件地址第二个字节 + 重复起始位 + 带读方向的从器件 10 位地址头字节

图 226. 10 位地址读访问 (HEAD10R = 0)



- 如果主器件对 10 位地址从器件进行寻址、向该从器件发送数据、然后再从该从器件读取数据，则必须首先完成主器件发送过程。然后，重复起始位置 1，10 位从地址配置为 HEAD10R = 1。在这种情况下，主器件发送以下序列：重复起始位 + 带读方向的从器件 10 位地址头字节。

图 227. 10 位地址读访问 (HEAD10R = 1)



主发送器

写传输时，在发送完每个字节（即第九个 SCL 脉冲（接收到 ACK 时））后，TXIS 标志将置 1。

如果 I2C_CR1 寄存器中的 TXIE 位置 1，TXIS 事件将生成中断。当 I2C_TXDR 寄存器中写入待发送的下一个数据字节时，该标志将被清零。

传输期间的 TXIS 事件的数量对应于 NBYTES[7:0] 中编程的值。如果待发送的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES 数据传输完成时，TCR 标志将置 1，并且 SCL 线的低电平将被延展，直到 NBYTES[7:0] 被写入非零值。

收到 NACK 时，TXIS 标志不会置 1。

- 当 RELOAD = 0 且 NBYTES 数据传输完成时：
  - 在自动结束模式 (AUTOEND = 1) 下，将自动发送停止位。
  - 在软件结束模式 (AUTOEND = 0) 下，TC 标志将置 1 且 SCL 线的低电平将被延长，以便执行以下软件操作：

    可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从地址和待传输字节数来请求发送重复起始位。将 START 位置 1 会将 TC 标志清零，并在总线上发送起始位。

    可通过将 I2C_CR2 寄存器中的 STOP 位置 1 来请求停止位。将 STOP 位置 1 会将 TC 标志清零，并在总线上发送停止位。
- 如果接收到 NACK：TXIS 标志不会置 1，并且接收到 NACK 后会自动发送停止位。I2C_ISR 寄存器中的 NACKF 标志置 1，如果 NACKIE 位置 1，还将生成中断。

图 228. I2C 主发送器的传输序列流程（N ≤ 255 字节）

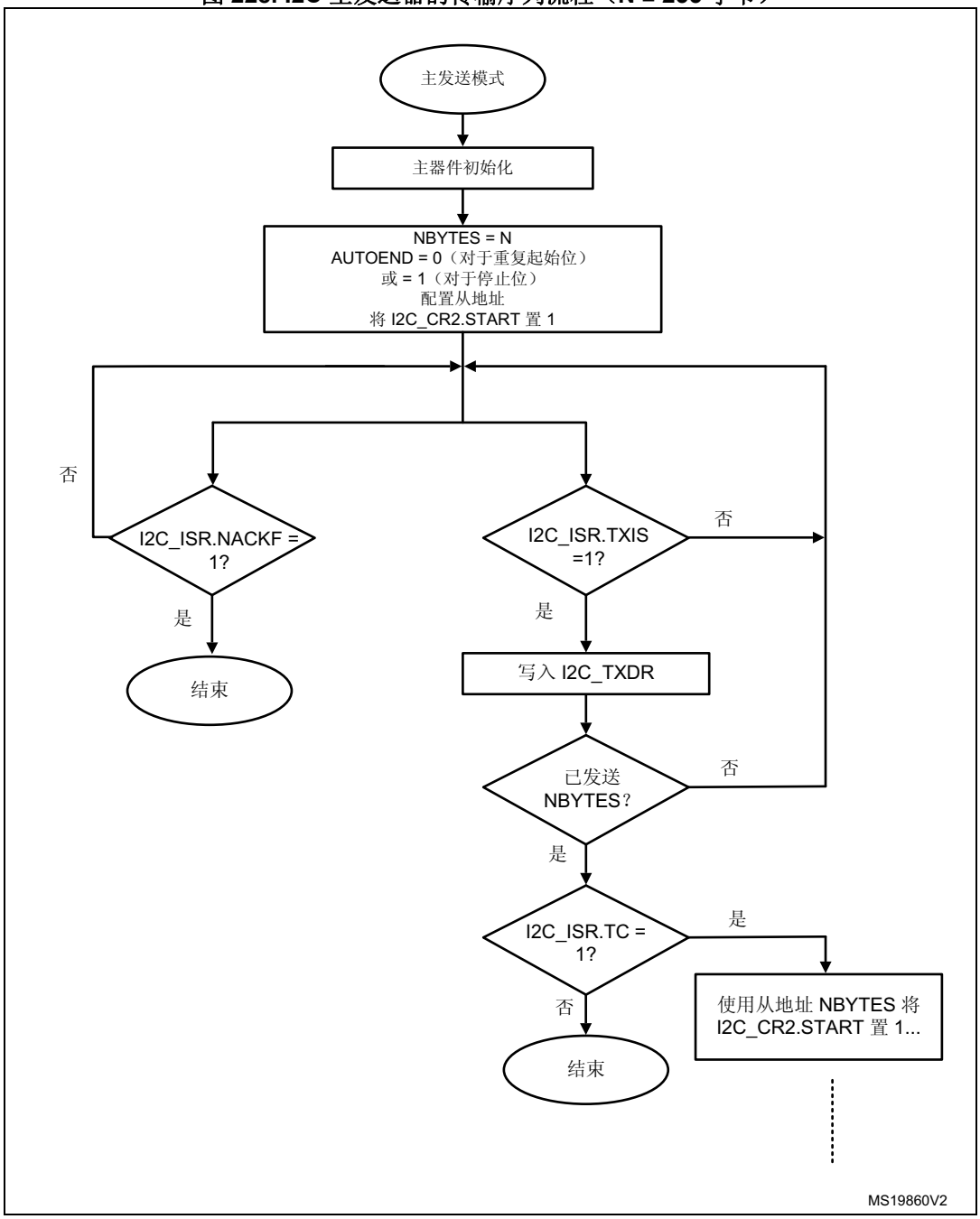


图 229. I2C 主发送器的传输序列流程（N > 255 字节）

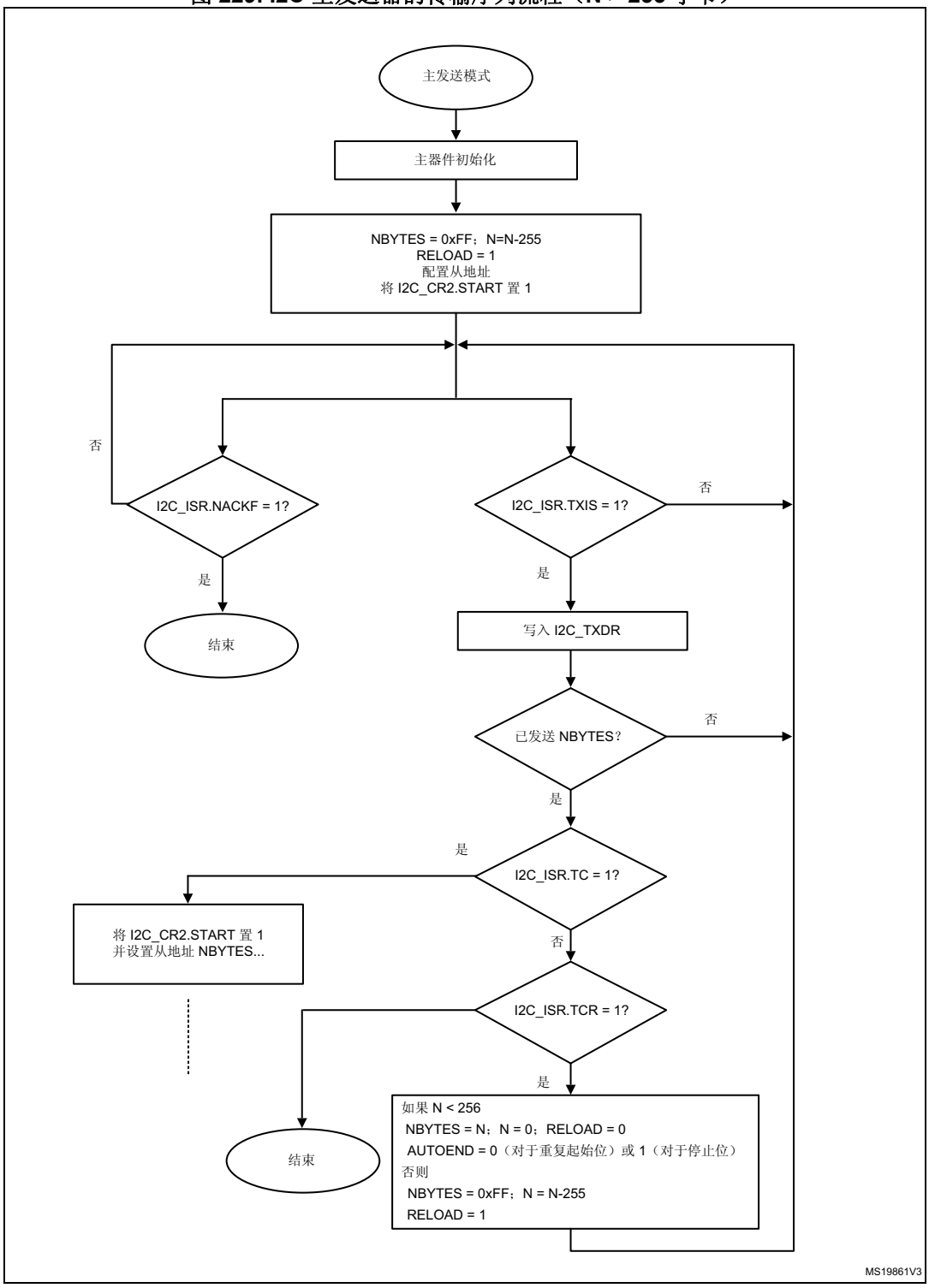
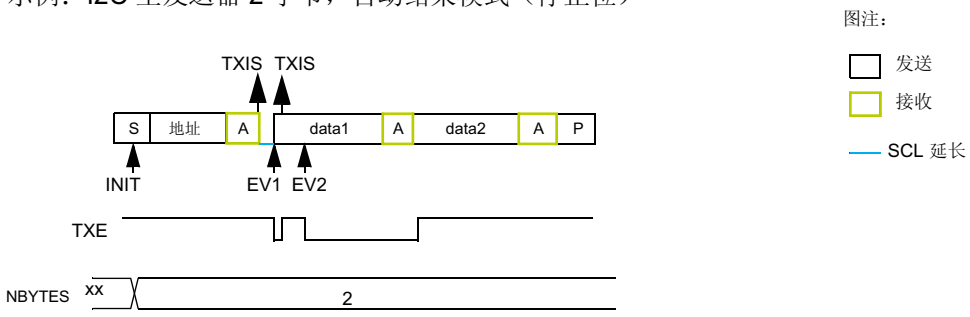


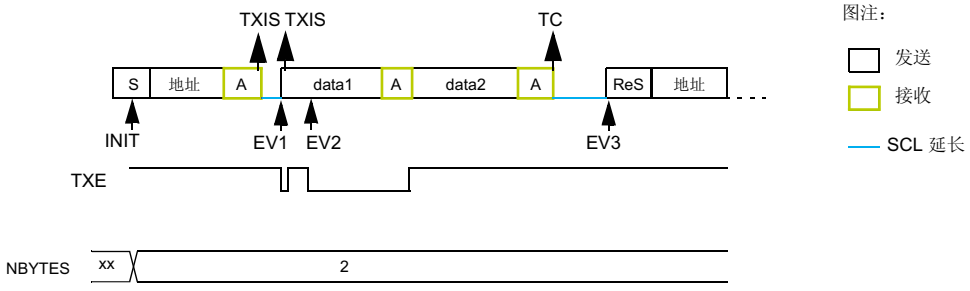
图 230. I2C 主发送器的传输总线图
（仅限强制性事件）

示例：I2C 主发送器 2 字节，自动结束模式（停止位）



INIT：编程从地址，编程 NBYTES = 2，AUTOEND=1，将 START 置 1
EV1：TXIS ISR：写入 data1
EV2：TXIS ISR：写入 data2

示例：I2C 主发送器 2 字节，软件结束模式（重复起始位）



INIT：编程从地址，编程 NBYTES = 2，AUTOEND=0，将 START 置 1
EV1：TXIS ISR：写入 data1
EV2：TXIS ISR：写入 data2
EV3：TC ISR：编程从地址，编程 NBYTES = N，将 START 置 1

MS19862V2

主接收器

读传输时，在接收到每个字节（即第八个 SCL 脉冲）后，RXNE 标志将置 1。如果 I2C_CR1 寄存器中的 RXIE 位置 1，RXNE 事件将生成中断。读取 I2C_RXDR 时，将清零该标志。

如果待接收的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES[7:0] 数据传输完成时，TCR 标志将置 1，并且 SCL 线的低电平将被延展，直到 NBYTES[7:0] 被写入非零值。

- 当 RELOAD = 0 且 NBYTES[7:0] 数据传输完成时：
  - 在自动结束模式 (AUTOEND = 1) 下，接收到最后一个字节后，将自动发送 NACK 和停止位。
  - 在软件结束模式 (AUTOEND = 0) 下，接收到最后一个字节后，将自动发送 NACK，TC 标志将置 1 且 SCL 线的低电平将被延长，以便执行以下软件操作：

    可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从地址和待传输字节数来请求发送重复起始位。将 START 位置 1 会将 TC 标志清零，并在总线上发送起始位，后跟从地址。

    可通过将 I2C_CR2 寄存器中的 STOP 位置 1 来请求停止位。将 STOP 位置 1 会将 TC 标志清零，并在总线上发送停止位。

图 231. I2C 主接收器的传输序列流程（N ≤ 255 字节）

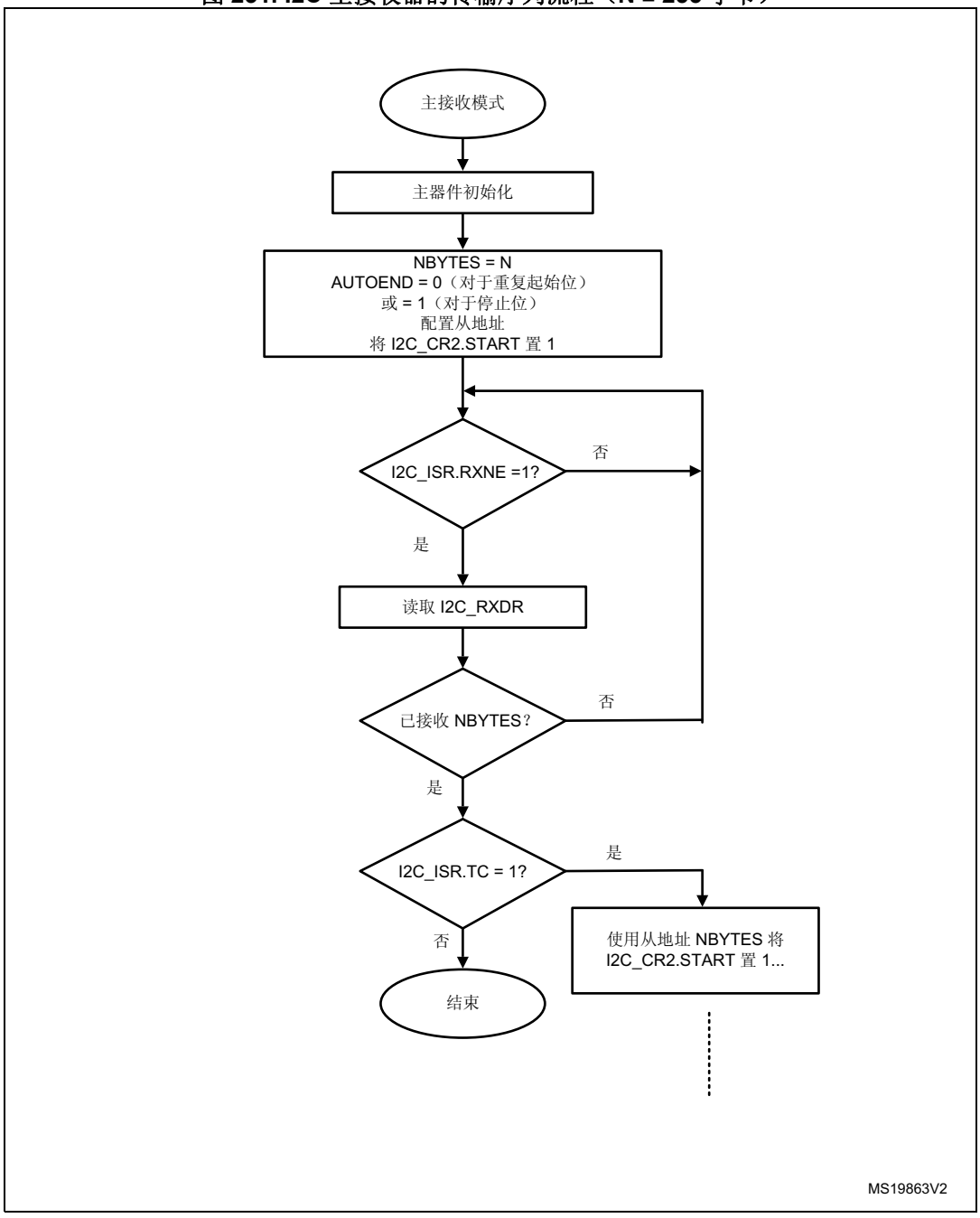


图 232. I2C 主接收器的传输序列流程（N > 255 字节）

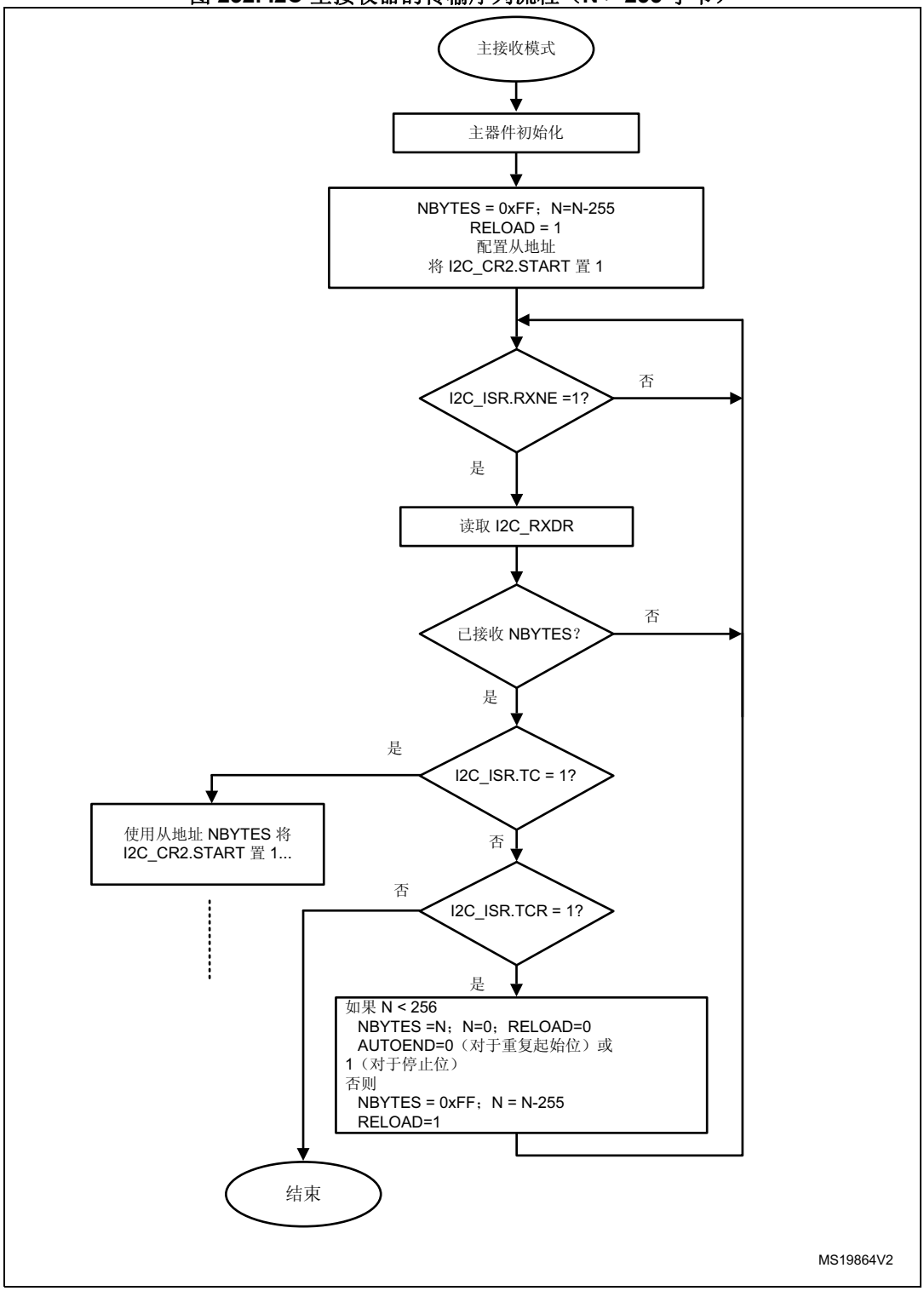
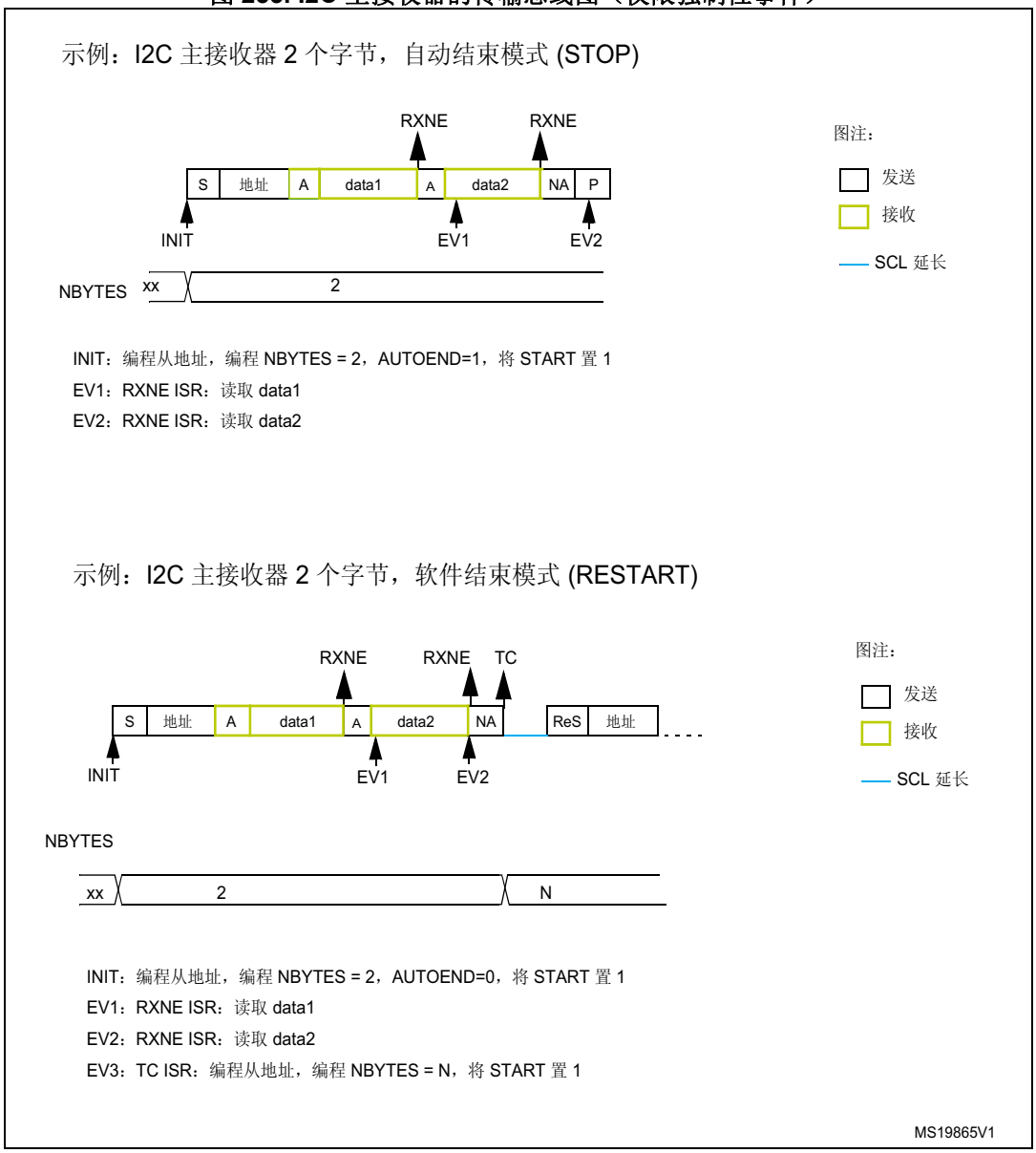


图 233. I2C 主接收器的传输总线图（仅限强制性事件）



23.4.10 I2C_TIMINGR 寄存器配置示例

下文各表提供了相应示例，以介绍如何编程 I2C_TIMINGR 才能获得符合 I²C 规范的时序。要获取更准确的配置值，应使用 STM32CubeMX 工具（I2C 配置窗口）。

表 94. f_{I2CCLK} = 8 MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	500 kHz
PRESC	0x1	0x1	0x0	0x0
SCLL	0xC7	0x13	0x9	0x6
t_{SCLL}	200 x 250 ns = 50 µs	20 x 250 ns = 5.0 µs	10 x 125 ns = 1250 ns	7 x 125 ns = 875 ns
SCLH	0xC3	0xF	0x3	0x3
t_{SCLH}	196 x 250 ns = 49 µs	16 x 250 ns = 4.0µs	4 x 125 ns = 500 ns	4 x 125 ns = 500 ns
t_{SCL}(1)	约 100 µs(2)	约 10 µs(2)	约 2500 ns(3)	约 2000 ns(4)
SDADEL	0x2	0x2	0x1	0x0
t_{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	1 x 125 ns = 125 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	2 x 125 ns = 250 ns

1. 由于 SCL 内部检测存在延时，t_{SCL} 大于 t_{SCLL} + t_{SCLH}。为 t_{SCL} 提供的值仅用于举例说明。
2. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 500 ns。t_{SYNC1} +t_{SYNC2} = 1000 ns 时的示例。
3. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 500 ns。t_{SYNC1} + t_{SYNC2} = 750 ns 时的示例。
4. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 500 ns。t_{SYNC1} + t_{SYNC2} = 655 ns 时的示例。

表 95. f_{I2CCLK} = 16 MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	0x3	0x3	0x1	0x0
SCLL	0xC7	0x13	0x9	0x4
t_{SCLL}	200 x 250 ns = 50 µs	20 x 250 ns = 5.0 µs	10 x 125 ns = 1250 ns	5 x 62.5 ns = 312.5 ns
SCLH	0xC3	0xF	0x3	0x2
t_{SCLH}	196 x 250 ns = 49 µs	16 x 250 ns = 4.0 µs	4 x 125 ns = 500 ns	3 x 62.5 ns = 187.5 ns
t_{SCL}(1)	约 100 µs(2)	约 10 µs(2)	约 2500 ns(3)	约 1000 ns(4)
SDADEL	0x2	0x2	0x2	0x0
t_{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	2 x 125 ns = 250 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x2
t_{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	3 x 62.5 ns = 187.5 ns

1. 由于 SCL 内部检测存在延时，t_{SCL} 大于 t_{SCLL} + t_{SCLH}。为 t_{SCL} 提供的值仅用于举例说明。
2. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 250 ns。t_{SYNC1} +t_{SYNC2} = 1000 ns 时的示例。
3. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 250 ns。t_{SYNC1} +t_{SYNC2} = 750 ns 时的示例。
4. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 250 ns。t_{SYNC1} +t_{SYNC2} = 500 ns 时的示例。

表 96. f_{I2CCLK} = 48 MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	0xB	0xB	0x5	0x5
SCLL	0xC7	0x13	0x9	0x3
t_{SCLL}	200 x 250 ns = 50 µs	20 x 250 ns = 5.0 µs	10 x 125 ns = 1250 ns	4 x 125 ns = 500 ns
SCLH	0xC3	0xF	0x3	0x1
t_{SCLH}	196 x 250 ns = 49 µs	16 x 250 ns = 4.0 µs	4 x 125 ns = 500 ns	2 x 125 ns = 250 ns
t_{SCL}(1)	约 100 µs(2)	约 10 µs(2)	约 2500 ns(3)	约 875 ns(4)
SDADEL	0x2	0x2	0x3	0x0
t_{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	3 x 125 ns = 375 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	2 x 125 ns = 250 ns

1. 由于 SCL 内部检测存在延时，t_{SCL} 大于 t_{SCLL} + t_{SCLH}。为 t_{SCL} 提供的值仅用于举例说明。
2. t_{SYNC1} + t_{SYNC2} 最小值为 4x t_{I2CCLK} = 83.3 ns。t_{SYNC1} + t_{SYNC2} = 1000 ns 时的示例
3. t_{SYNC1} + t_{SYNC2} 最小值为 4x t_{I2CCLK} = 83.3 ns。t_{SYNC1} + t_{SYNC2} = 750 ns 时的示例
4. t_{SYNC1} + t_{SYNC2} 最小值为 4x t_{I2CCLK} = 83.3 ns。t_{SYNC1} + t_{SYNC2} = 250 ns 时的示例

23.4.11 SMBus 特性

仅当支持 SMBus 功能（请参见*第 23.3 节*）时，才涉及本节内容。

简介

系统管理总线 (SMBus) 是一个双线制接口，各器件可通过它在彼此之间或者与系统的其余部分进行通信。它以 I^2C 的工作原理为基础。SMBus 可针对系统和电源管理相关的任务提供控制总线。

该外设与 SMBus 规范 (http://smbus.org) 兼容。

系统管理总线规范涉及三类器件。

- 从器件，用于接收或响应命令。
- 主器件，用于发出命令、生成时钟和中止传输。
- 主机，专用的主器件，可提供连接系统 CPU 的主接口。主机必须具有主 - 从设备功能，并且必须支持 SMBus 主机通知协议。系统中只允许存在一个主机。

该外设可配置为主器件或从器件，也可配置为主机。

总线协议

任何给定器件都有十一种可用命令协议。器件既可以在这十一种协议中任选其一，也可以使用全部十一种协议进行通信。这十一种协议分别为快速命令、发送字节、接收字节、写入字节、写入字、读取字节、读取字、过程调用、块读取、块写入以及块写入-块读取过程调用。这些协议必须通过用户软件实现。

有关这些协议的详细信息，请参见 SMBus 规范 (http://smbus.org)。

地址解析协议 (ARP)

通过为各个从器件动态分配一个新的唯一地址可解决 SMBus 从地址冲突的问题。为了提供一种机制来针对地址分配隔离各个器件，各器件必须具有唯一的器件标识符 (UDID)。该 128 位数字由软件实现。

该外设支持地址解析协议 (ARP)。通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 器件默认地址 (0b1100 001)。ARP 命令必须通过用户软件实现。

此外，还将在从模式下执行仲裁以支持 ARP。

有关 SMBus 地址解析协议的详细信息，请参见 SMBus 规范 (http://smbus.org)。

接收的命令和数据应答控制

SMBus 接收器必须能够对接收到的每个命令或数据进行 NACK 应答。要在从模式下实现 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。更多详细信息，请参见*从器件字节控制模式*。

主机通知协议

该外设通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来支持主机通知协议。在这种情况下，主机将应答 SMBus 主机地址 (0b0001 000)。

使用该协议时，器件用作主器件，而主机用作从器件。

SMBus 报警

器件支持 SMBus ALERT 可选信号。只具备从功能的器件可通过 SMBALERT# 引脚向主机发出信号，指示它想要通信。主机会处理该中断并通过报警响应地址 (0b0001 100) 同时访问所有 SMBALERT# 器件。只有那些将 SMBALERT# 拉到低电平的器件会应答报警响应地址。

如果配置为从器件 (SMBHEN = 0)，则通过将 I2C_CR1 寄存器中的 ALERTEN 位置 1 来将 SMBA 引脚拉为低电平。这同时还会使能报警响应地址。

如果配置为主机 (SMBHEN = 1)，则当 SMBA 引脚上检测到下降沿且 ALERTEN = 1 时，I2C_ISR 寄存器中的 ALERT 标志置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，将生成中断。当 ALERTEN = 0 时，即使外部 SMBA 引脚为低电平，ALERT 线也将被视为高电平。

如果无需 SMBus ALERT 引脚，则当 ALERTEN = 0 时，SMBA 引脚可用作标准 GPIO。

数据包错误校验

SMBus 规范中引入了数据包错误校验机制来提高可靠性和通信稳定性。数据包错误校验的实施方式是在每次消息传输结束时附加数据包错误代码 (PEC)。PEC 的计算方式是对所有消息字节（包括地址和读/写位）使用 CRC-8 多项式 $C(x) = x_8 + x^2 + x + 1$。

外设内置了硬件 PEC 计算器，可在接收到的字节与硬件计算的 PEC 不匹配时自动发送否定应答信号。

超时

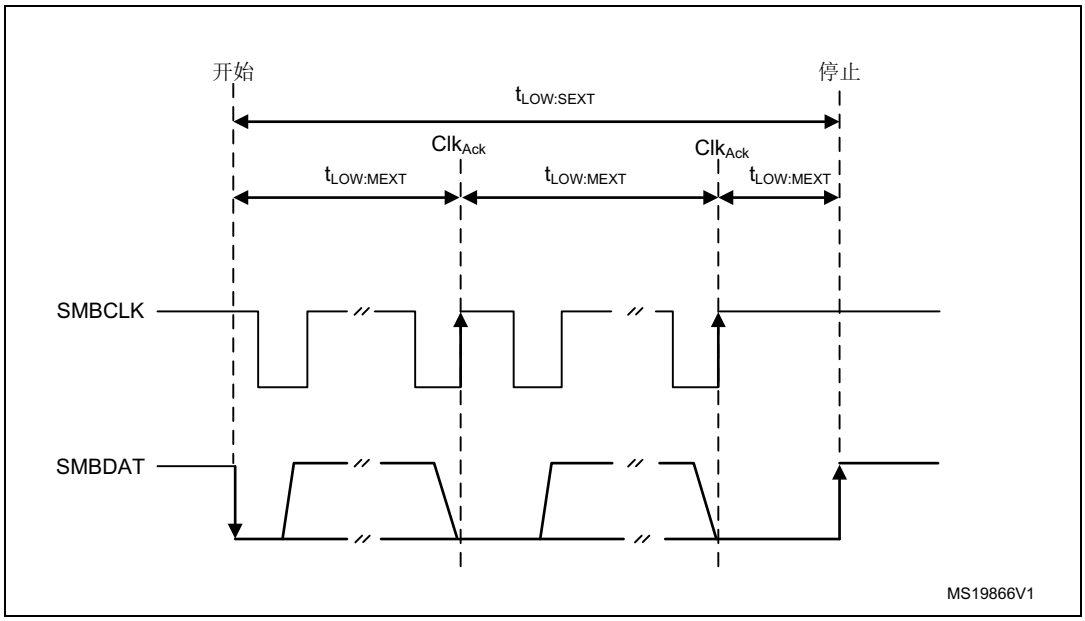
该外设内置了硬件定时器，以便符合 SMBus 规范中定义的三个超时。

表 97. SMBus 超时规范

符号	参数	限值		单位
		最小值	最大值	
$t_{TIMEOUT}$	检测时钟低电平超时	25	35	ms
$t_{LOW:SEXT}$(1)	累积时钟低电平延长时间（从器件）	-	25	
$t_{LOW:MEXT}$(2)	累积时钟低电平延长时间（主器件）	-	10	

1. $t_{LOW:SEXT}$ 是一段累积时间，即给定从器件在一条消息的最初起始到停止期间时钟信号可延长的时间。其他从器件或主器件也可能延长时钟，进而导致时钟低电平总延长时间超过 $t_{LOW:SEXT}$。因此，测量该参数时该全速主器件只寻址一个从器件。
2. $t_{LOW:MEXT}$ 是一段累积时间，即主器件在消息的每个字节（定义为 START 到 ACK、ACK 到 ACK 或 ACK 到 STOP）内时钟信号可延长的时间。从器件或其他主器件也可能延长时钟，进而导致时钟低电平总时间超过 $t_{LOW:MEXT}$（针对给定字节）。因此，测量该参数时该全速主器件只寻址一个从器件。

图 234. $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的超时间隔



总线空闲检测

如果主器件检测到时钟和数据信号的高电平时间已达 t_{IDLE}（超过 $t_{HIGH,MAX}$），则认为总线空闲（请参见*表 91*）。

该时序参数已考虑如下情况：主器件已动态添加至总线，但可能尚未检测到 SMBCLK 或 SMBDAT 线上的状态转换。在这种情况下，主器件必须等待足够长的时间，以确定当前未进行传输。外设支持硬件总线空闲检测。

23.4.12 SMBus 初始化

仅当支持 SMBus 功能（请参见*第 23.3 节*）时，才涉及本节内容。

除了 I2C 初始化之外，还必须进行一些其他的特定初始化，以便执行 SMBus 通信：

接收的命令和数据应答控制（从模式）

SMBus 接收器必须能够对接收到的每个命令或数据进行 NACK 应答。要在从模式下实现 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。更多详细信息，请参见*从器件字节控制模式*。

特定地址（从模式）

必要时必须使能特定的 SMBus 地址。更多详细信息，请参见*总线空闲检测*。

- 通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 器件默认地址 (0b1100 001)。
- 通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来使能 SMBus 主机地址 (0b0001 000)。
- 通过将 I2C_CR1 寄存器中的 ALERTEN 位置 1 来使能报警响应地址 (0b0001100)。

数据包错误校验

通过将 I2C_CR1 寄存器中的 PECEN 位置 1 来使能 PEC 的计算。然后，借助硬件字节计数器来管理 PEC 传输：I2C_CR2 寄存器中的 NBYTES[7:0]。使能 I2C 之前，必须配置 PECEN 位。

PEC 传输由硬件字节计数器来管理，因此在从模式下连接 SMBus 时必须将 SBC 位置 1。当 PECBYTE 位置 1 且 RELOAD 位清零时，传输完 NBYTES - 1 字节的数据后会传输 PEC。如果 RELOAD 置 1，PECBYTE 将不起作用。

小心： 使能 I2C 时，不允许更改 PECEN 配置。

表 98. 带 PEC 的 SMBus 配置

模式	SBC 位	RELOAD 位	AUTOEND 位	PECBYTE 位
主 Tx/Rx NBYTES + PEC+ STOP	x	0	1	1
主 Tx/Rx NBYTES + PEC + ReSTART	x	0	0	1
从 Tx/Rx + PEC	1	0	x	1

超时检测

将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 和 TEXTEN 位置 1 来使能超时检测。定时器必须按如下方式编程：即在 SMBus 规范规定的时间最大值之前检测出超时情况。

- $t_{TIMEOUT}$ 检查

  要使能 $t_{TIMEOUT}$ 检查，必须将 12 位 TIMEOUTA[11:0] 位编程为定时器重载值，以检查 $t_{TIMEOUT}$ 参数。必须将 TIDLE 位配置为“0”，以检测 SCL 低电平超时。

  然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。

  如果 SCL 的低电平持续时间超过 (TIMEOUTA + 1) x 2048 x t_{I2CCLK}，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。

  请参见*表 99*。

小心： TIMEOUTEN 位置 1 时，不允许更改 TIMEOUTA[11:0] 位和 TIDLE 位的配置。

- $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 检查
  必须根据外设配置为主器件还是从器件来配置 TIMEOUTB 定时器，以便为从器件校验 $t_{LOW:SEXT}$，为主器件校验 $t_{LOW:MEXT}$。由于标准只规定了最大值，用户可以为这两个参数选择相同的值。
  然后，通过将 I2C_TIMEOUTR 寄存器中的 TEXTEN 位置 1 来使能定时器。
  如果 SMBus 外设延展 SCL 的累积时间超过 (TIMEOUTB + 1) x 2048 x t_{I2CCLK}，并且达到*总线空闲检测*一节给出的超时间隔，则 I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。
  请参见*表 100*。

小心： TEXTEN 位置 1 时，不允许更改 TIMEOUTB 配置。

总线空闲检测

要使能 t_{IDLE} 检查，必须将 12 位 TIMEOUTA[11:0] 位域编程为定时器重载值，以获取 t_{IDLE} 参数。必须将 TIDLE 位配置为“1”，以检测 SCL 和 SDA 高电平超时。

然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。

如果 SCL 和 SDA 线的高电平持续时间超过 (TIMEOUTA+1) x 4 x t_{I2CCLK}，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。

请参见*表 101*。

小心： TIMEOUTEN 置 1 时，不允许更改 TIMEOUTA 和 TIDLE 配置。

23.4.13 SMBus：I2C_TIMEOUTR 寄存器配置示例

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 23.3 节*。

- 将 $t_{TIMEOUT}$ 的最大持续时间配置为 25 ms：

表 99. TIMEOUTA 设置示例（最大 $t_{TIMEOUT}$ = 25 ms）

f_{I2CCLK}	TIMEOUTA[11:0] 位	TIDLE 位	TIMEOUTEN 位	$t_{TIMEOUT}$
8 MHz	0x61	0	1	98 x 2048 x 125 ns = 25 ms
16 MHz	0xC3	0	1	196 x 2048 x 62.5 ns = 25 ms
48 MHz	0x249	0	1	586 x 2048 x 20.08 ns = 25 ms

- 将 $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的最大持续时间配置为 8 ms：

表 100. TIMEOUTB 设置示例

f_{I2CCLK}	TIMEOUTB[11:0] 位	TEXTEN 位	$t_{LOW:EXT}$
8 MHz	0x1F	1	32 x 2048 x 125 ns = 8 ms
16 MHz	0x3F	1	64 x 2048 x 62.5 ns = 8 ms
48 MHz	0xBB	1	188 x 2048 x 20.08 ns = 8 ms

- 将 t_{IDLE} 的最大持续时间配置为 50 µs

表 101. TIMEOUTA 设置示例（最大 t_{IDLE} = 50 µs）

f_{I2CCLK}	TIMEOUTA[11:0] 位	TIDLE 位	TIMEOUTEN 位	t_{TIDLE}
8 MHz	0x63	1	1	100 x 4 x 125 ns = 50 µs
16 MHz	0xC7	1	1	200 x 4 x 62.5 ns = 50 µs
48 MHz	0x257	1	1	600 x 4 x 20.08 ns = 50 µs

23.4.14 SMBus 从模式

仅当支持 SMBus 功能（请参见*第 23.3 节*）时，才涉及本节内容。

除了 I2C 从模式传输管理（请参见*第 23.4.8 节*）之外，还提供了一些额外的软件流程来支持 SMBus。

SMBus 从发送器

在 SMBus 模式下作为从发送器时，必须将 SBC 编程为“1”，以使能在完成已编程数据字节数的传输后进行 PEC 传输。当 PECBYTE 位置 1 时，NBYTES[7:0] 中编程的字节数包含 PEC 传输。在这种情况下，总 TXIS 中断数为 NBYTES - 1，如果主器件在完成 NBYTES - 1 字节的数据传输后请求传输额外的字节，则将自动发送 I2C_PECR 寄存器的内容。

小心： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 235. SMBus 从发送器的传输序列流程（N 字节 + PEC）

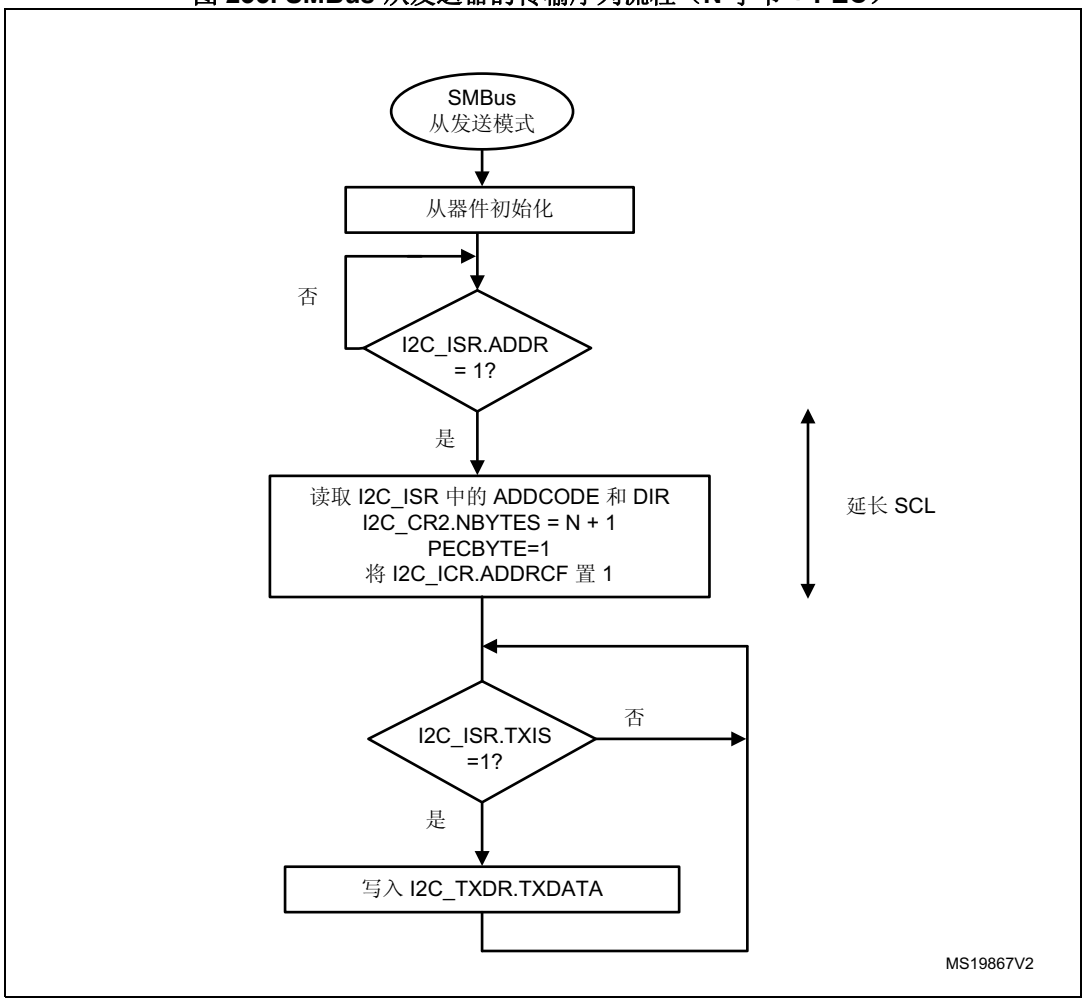
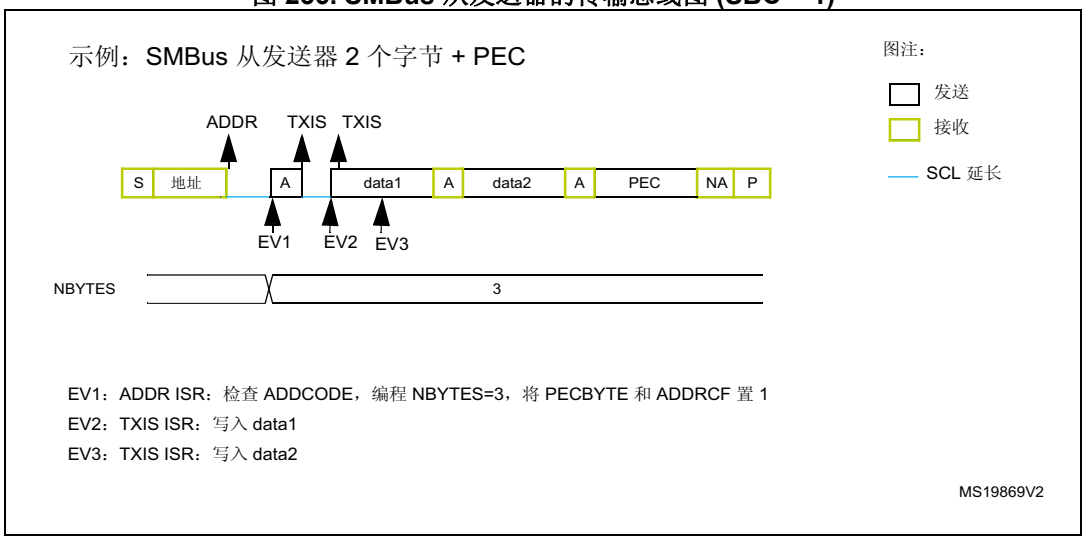


图 236. SMBus 从发送器的传输总线图 (SBC = 1)



SMBus 从接收器

在 SMBus 模式下使用 I2C 时，必须将 SBC 编程为“1”，以便在完成已编程数据字节数的传输后进行 PEC 校验。要对每个字节进行 ACK 控制，必须选择重载模式 (RELOAD = 1)。更多详细信息，请参见*从器件字节控制模式*。

要校验 PEC 字节，必须将 RELOAD 位清零并将 PECBYTE 位置 1。在这种情况下，当接收到 NBYTES - 1 字节的数据后，接收的下一个字节将与内部 I2C_PECR 寄存器的内容作比较。如果比较不匹配，则将自动生成 NACK 信号；如果比较匹配，则将自动生成 ACK 信号，而与 ACK 位的值无关。PEC 字节一经接收，便会像任何其他数据一样复制到 I2C_RXDR 寄存器中，并且 RXNE 标志将置 1。

当 PEC 不匹配时，PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

如果无需 ACK 软件控制，用户可编程 PECBYTE = 1，在同一写操作下，将 NBYTES 编程为连续接收的字节数。接收到 NBYTES - 1 字节的数据后，会将接收的下一个字节视为 PEC 进行校验。

小心： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 237. SMBus 从接收器的传输序列流程（N 字节 + PEC）

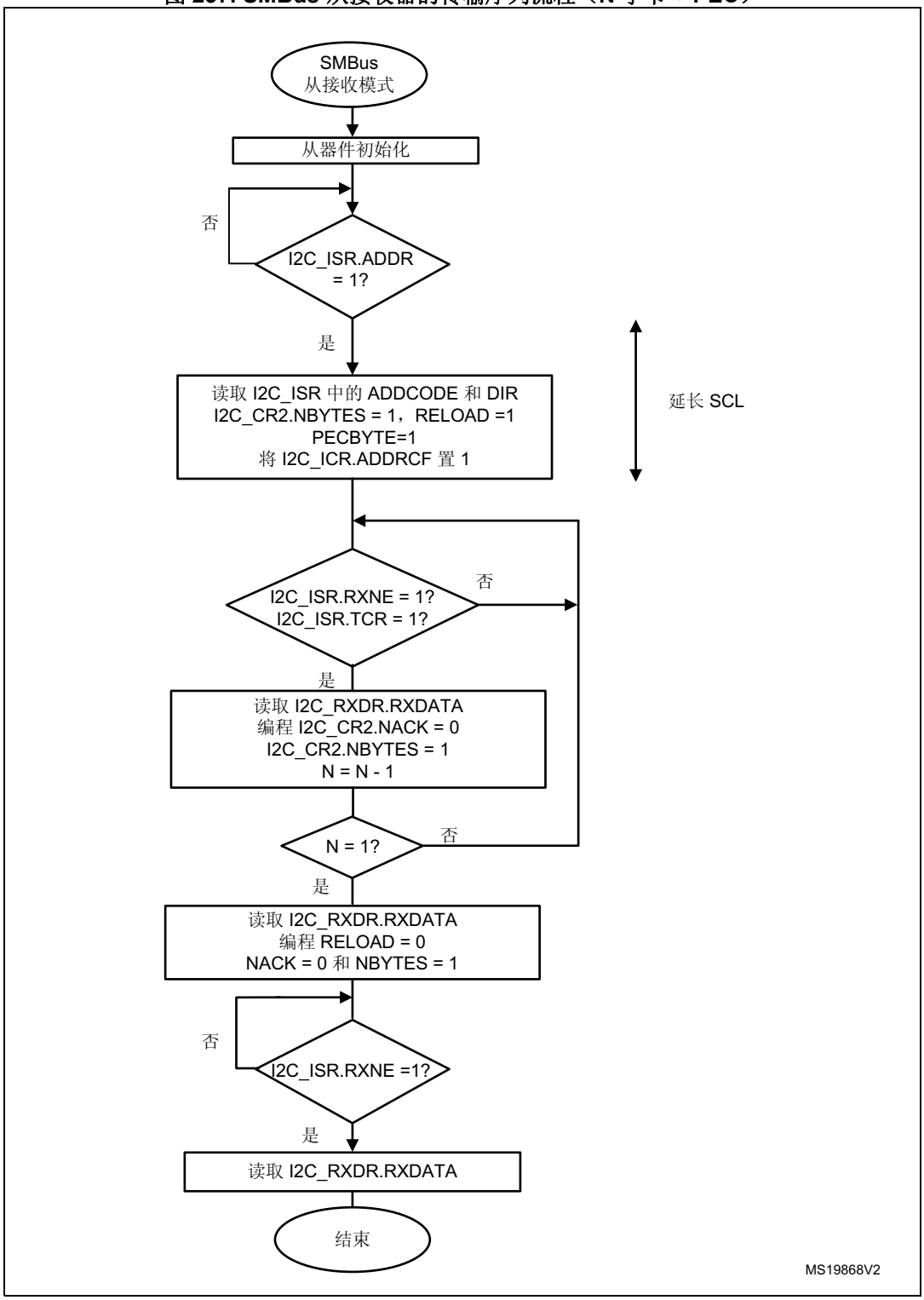
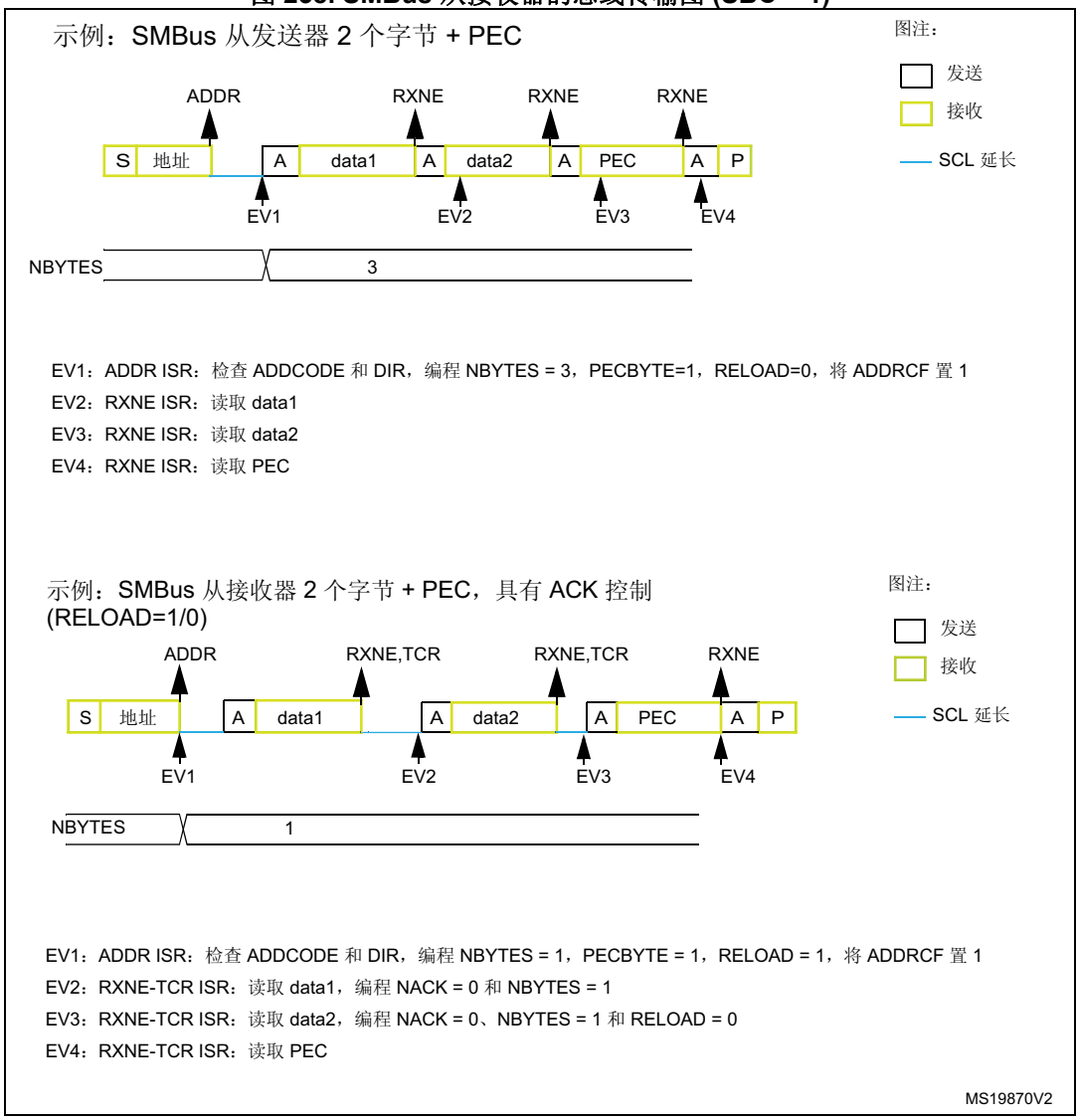


图 238. SMBus 从接收器的总线传输图 (SBC = 1)



仅当支持 SMBus 功能（请参见*第 23.3 节*）时，才涉及本节内容。

除了 I2C 主模式传输管理（请参见*第 23.4.9 节*）之外，还提供了一些额外的软件流程来支持 SMBus。

SMBus 主发送器

当 SMBus 主器件想要发送 PEC 时，必须在 START 位置 1 前，将 PECBYTE 位置 1 并在 NBYTES[7:0] 位域中设置字节数。在这种情况下，总 TXIS 中断数为 NBYTES - 1。因此，如果 PECBYTE 位在 NBYTES = 0x1 时置 1，则将自动发送 I2C_PECR 寄存器的内容。

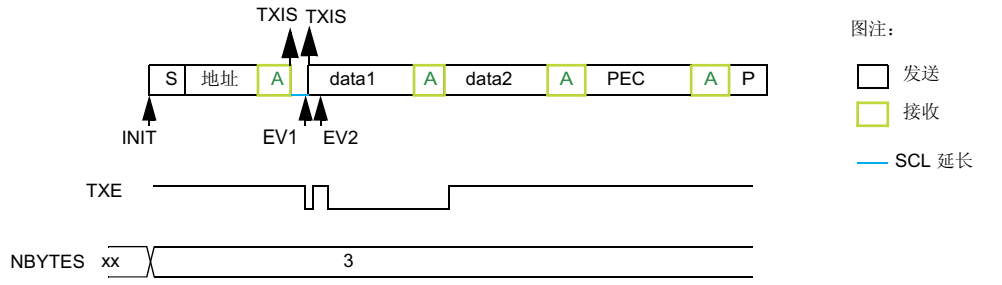
如果 SMBus 主器件想要在 PEC 后发送停止位，则必须选择自动结束模式 (AUTOEND = 1)。在这种情况下，传输 PEC 后将自动发送停止位。

如果 SMBus 主器件想要在 PEC 后发送重复起始位，则必须选择软件模式 (AUTOEND = 0)。在这种情况下，发送 NBYTES - 1 字节的数据后，将发送 I2C_PECR 寄存器的内容，TC 标志将在传输完 PEC 之后置 1，SCL 线的低电平时间将延长。必须在 TC 中断子程序中设置重复起始位。

小心： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

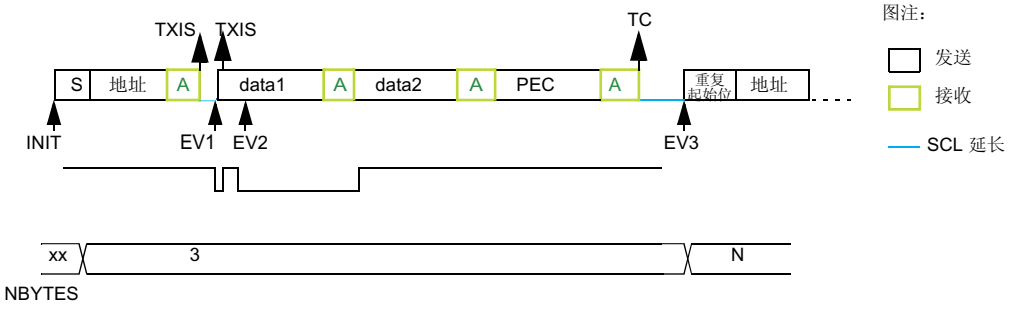
图 239. SMBus 主发送器的总线传输图

示例：SMBus 主发送器 2 个字节 + PEC，自动结束模式（停止位）



INIT：编程从地址，编程 NBYTES = 3，AUTOEND=1，将 PECBYTE 置 1，将 START 置 1
EV1：TXIS ISR：写入 data1
EV2：TXIS ISR：写入 data2

示例：SMBus 主发送器 2 个字节 + PEC，软件结束模式（重复起始位）



INIT：编程从地址，编程 NBYTES = 3，AUTOEND=0，将 PECBYTE 置 1，将 START 置 1
EV1：TXIS ISR：写入 data1
EV2：TXIS ISR：写入 data2
EV3：TC ISR：编程从地址，编程 NBYTES = N，将 START 置 1

MS19871V2

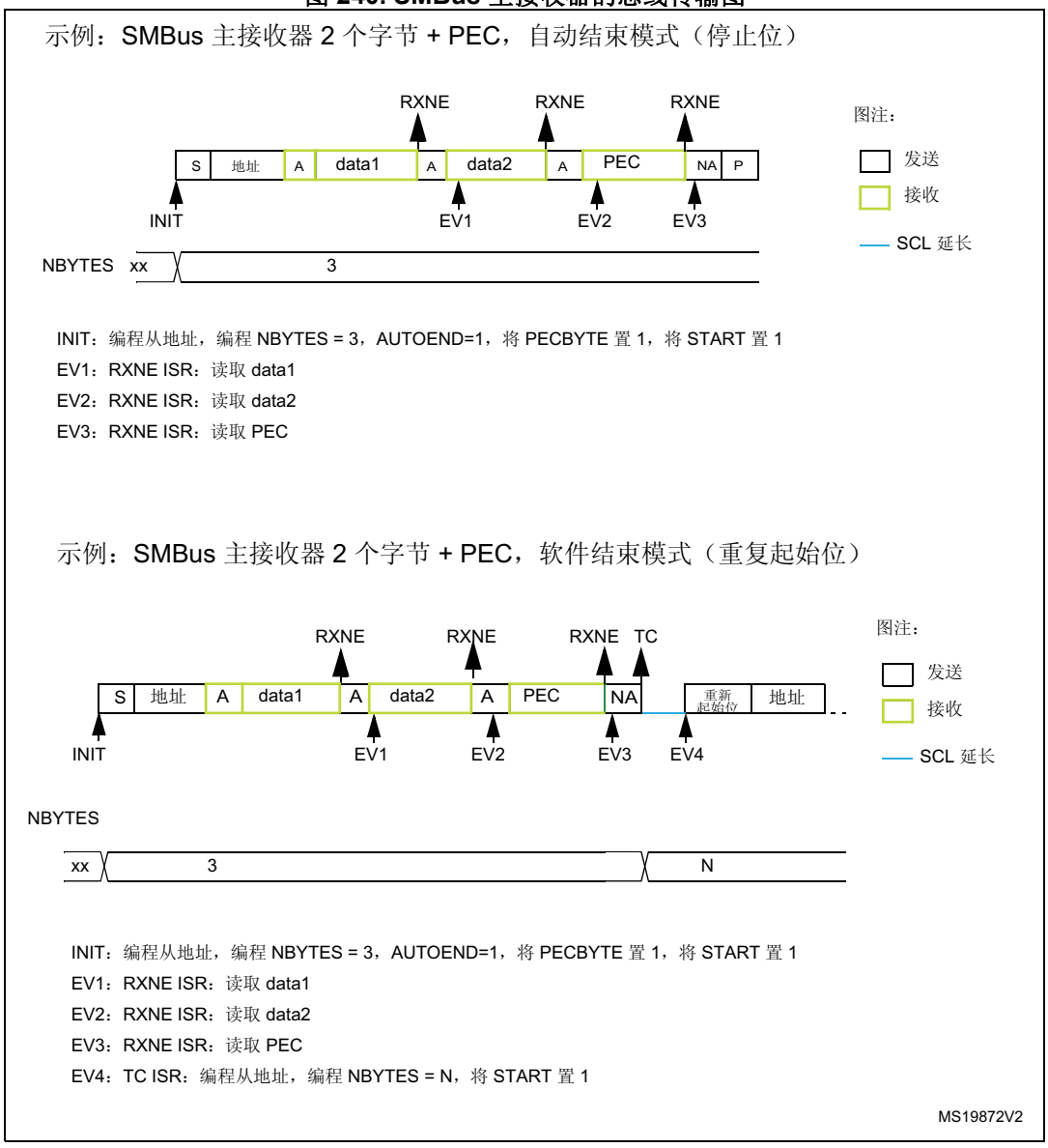
SMBus 主接收器

当 SMBus 主器件想要接收 PEC，并在传输结束后接收 STOP 时，可选择自动结束模式 (AUTOEND = 1)。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从地址。在这种情况下，当接收到 NBYTES - 1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。PEC 字节（其后跟有停止位）将得到 NACK 响应。

当 SMBus 主接收器想要接收 PEC 字节，并且在传输结束后接收重复起始位时，必须选择软件模式 (AUTOEND = 0)。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从地址。在这种情况下，当接收到 NBYTES - 1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。接收到 PEC 字节后，TC 标志将置 1，SCL 线的低电平时间将延长。可以在 TC 中断子程序中设置重复起始位。

小心： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 240. SMBus 主接收器的总线传输图



23.4.15 地址匹配时从停止模式唤醒

仅当支持从停止模式唤醒功能（请参见*第 23.3 节*）时，才涉及本节内容。

被寻址时，I2C 能够将 MCU 从停止模式唤醒（APB 时钟关断）。支持所有寻址模式。

将 I2C_CR1 寄存器中的 WUPEN 位置 1，可以使能从停止模式唤醒功能。对于 I2CCLK，必须选择 HSI48 振荡器作为时钟源，以便从停止模式唤醒。

在停止模式期间，HSI48 关闭。当检测到 START 时，I2C 接口将 HSI48 接通，并延长 SCL 使其处于低电平直到唤醒 HSI48。

HSI48 随后用来接收地址。

地址匹配的情况下，MCU 唤醒时间内，I2C 延长 SCL 的低电平持续时间。通过软件清零 ADDR 标志后，此延长操作被释放，传输正常进行。

如果地址不匹配，HSI48 再次关断，MCU 不被唤醒。

注意： *如果 I2C 时钟是系统时钟，或者 WUPEN = 0，则接收到 START 后，*HSI48 *不会接通。*

只有 ADDR 中断能够唤醒 MCU。因此，当 I2C 以主器件身份或在 ADDR 标志置 1 后以被寻址从器件身份执行传输时，不要进入停止模式。通过在 ADDR 中断程序中清零 SLEEPDEEP 位，然后仅在 STOPF 标志置 1 后再将其置 1，可对此进行管理。

小心： 数字滤波器与从停止模式唤醒功能不兼容。如果 DNF 位不等于 0，则将 WUPEN 位置 1 将不起任何作用。

小心： 只有当 I2C 时钟源为 HSI48 振荡器时，该功能才可用。

小心： 必须使能时钟延长 (NOSTRETCH = 0) 才能确保从停止模式唤醒功能正常工作。

小心： 如果禁止从停止模式唤醒 (WUPEN = 0)，则在进入停止模式前必须禁止 I2C 外设 (PE = 0)。

23.4.16 错误条件

以下错误是可能导致通信失败的条件。

总线错误 (BERR)

当检测到起始位或停止位但不位于第九个 SCL 时钟脉冲之后时，会检测到总线错误。当 SDA 边沿出现且 SCL 为高电平时，会检测到起始或停止位。

只有当 I2C 在传输过程中用作主器件或被寻址为从器件时（即未处于从模式下的地址阶段），才会将总线错误标志置 1。

在从模式下检测到错位的起始位或重复起始位时，I2C 会像接收到正确的起始位一样进入地址识别状态。

检测到总线错误时，I2C_ISR 寄存器中的 BERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

仲裁丢失 (ARLO)

当 SDA 线上发送高电平但在 SCL 上升沿却采样到低电平时，会检测到仲裁丢失。

- 在主模式下，将在地址阶段、数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，SDA 线和 SCL 线被释放，起始控制位由硬件清零，主器件自动切换为从模式。
- 在从模式下，将在数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，传输停止，SCL 和 SDA 线被释放。

检测到仲裁丢失时，I2C_ISR 寄存器中的 ARLO 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

上溢/下溢错误 (OVR)

当满足 NOSTRETCH = 1 和以下条件时，将在从模式下检测到上溢或下溢错误：

- 在接收过程中接收到一个新的字节，但 RXDR 寄存器的值还未被读取。接收的新字节丢失，自动发送 NACK 来响应新字节。
- 在发送过程中：
  - 当 STOPF = 1 且必须发送第一个数据字节时。TXE = 0 时发送 I2C_TXDR 寄存器的内容，否则发送 0xFF。
  - 必须发送一个新字节但尚未向 I2C_TXDR 寄存器写入数据时，将发送 0xFF。

检测到上溢或下溢错误时，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

数据包错误校验错误 (PECERR)

仅当支持 SMBus 功能（请参见*第 23.3 节*）时，才涉及本节内容。

当接收到的 PEC 字节与 I2C_PECR 寄存器的内容不匹配时，将检测到 PEC 错误。接收到错误的 PEC 后，将自动发送 NACK。

检测到 PEC 错误时，I2C_ISR 寄存器中的 PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

超时错误 (TIMEOUT)

仅当支持 SMBus 功能（请参见*第 23.3 节*）时，才涉及本节内容。

满足以下任何条件均会出现超时错误：

- TIDLE = 0 且 SCL 的低电平持续时间达到 TIMEOUTA[11:0] 位中定义的时间：这用于检测 SMBus 超时。
- TIDLE = 1 且 SDA 和 SCL 的高电平持续时间达到 TIMEOUTA[11:0] 位中定义的时间：这用于检测总线空闲情况。
- 主器件的累积时钟低电平延长时间达到了 TIMEOUTB[11:0] 位中定义的时间（SMBus $t_{LOW:MEXT}$ 参数）。
- 从器件的累积时钟低电平延长时间达到了 TIMEOUTB[11:0] 位中定义的时间（SMBus $t_{LOW:SEXT}$ 参数）。

当在主模式下检测到超时时，将自动发送停止位。

当在从模式下检测到超时时，将自动释放 SDA 和 SCL 线。

检测到超时错误时，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

报警 (ALERT)

仅当支持 SMBus 功能（请参见*第 23.3 节*）时，才涉及本节内容。

当 I2C 接口配置为主机 (SMBHEN = 1)、使能了报警引脚检测 (ALERTEN = 1) 并且在 SMBA 引脚上检测到下降沿时，ALERT 标志将置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，将生成中断。

23.4.17 DMA 请求

使用 DMA 进行发送

将 I2C_CR1 寄存器中的 TXDMAEN 位置 1 可以使能 DMA（直接存储器访问）进行发送。当 TXIS 位置 1 时，数据将从由 DMA 外设配置的 SRAM 区（请参见*第 169 页的第 9 节：直接存储器访问控制器 (DMA)*）装载进 I2C_TXDR 寄存器。

只有数据字节采用 DMA 进行传输。

- 在主模式下：初始化、从地址、方向、字节数和起始位均由软件编程（发送的从地址无法通过 DMA 传输）。当所有数据均通过 DMA 传输时，必须在起始位置 1 之前初始化 DMA。传输结束由 NBYTES 计数器来管理。请参见*主发送器*。
- 在从模式下：
  - 当 NOSTRETCH = 0 时，如果所有数据均通过 DMA 传输，则必须在地址匹配事件之前（或清零 ADDR 之前在 ADDR 中断子程序中）初始化 DMA。
  - 当 NOSTRETCH = 1 时，必须在地址匹配事件之前初始化 DMA。
- 支持 SMBus 时：PEC 传输由 NBYTES 计数器管理。请参见*SMBus 从发送器*和*SMBus 主发送器*。

注意： *如果使用 DMA 进行发送，则无需使能 TXIE 位。*

使用 DMA 进行接收

将 I2C_CR1 寄存器中的 RXDMAEN 位置 1 可以使能 DMA（直接存储器访问）进行接收。当 RXNE 位置 1 时，数据将从 I2C_RXDR 寄存器装载进由 DMA 外设配置的 SRAM 区（请参见*第 169 页的第 9 节：直接存储器访问控制器 (DMA)*）。只有数据字节（包括 PEC）采用 DMA 进行传输。

- 在主模式下：初始化、从地址、方向、字节数和起始位均由软件编程。当所有数据均通过 DMA 传输时，必须在起始位置 1 之前初始化 DMA。传输结束由 NBYTES 计数器来管理。
- 在从模式下，当 NOSTRETCH = 0 时，如果所有数据均通过 DMA 传输，则必须在地址匹配事件之前（或清零 ADDR 标志之前在 ADDR 中断子程序中）初始化 DMA。
- 如果支持 SMBus（请参见*第 23.3 节*）：PEC 传输由 NBYTES 计数器管理。请参见*SMBus 从接收器*和*SMBus 主接收器*。

注意： *如果使用 DMA 进行接收，则无需使能 RXIE 位。*

23.4.18 调试模式

当微控制器进入调试模式时（内核停止），SMBus 超时定时器会根据 DBG 模块中的 DBG_I2Cx_SMBUS_TIMEOUT 配置位选择继续正常工作还是停止工作。

23.5 I2C 低功耗模式

表 102. 低功耗模式对 I2C 的影响

模式	说明
睡眠	无影响。I2C 中断可使器件退出睡眠模式。
停止(1)	I2C 模块的寄存器内容仍被保持。 – WUPEN = 1 并且 I2C 的时钟由内部振荡器 (HSI48) 提供：地址识别功能正常。I2C 地址匹配条件会导致器件退出停止模式。 – WUPEN = 0：必须在进入停止模式之前禁止 I2C。
待机	I2C 外设掉电，退出待机模式后必须重新初始化。

1. 有关每个实例支持的停止模式的信息，请参见*第 23.3 节*。如果不支持从特定停止模式唤醒，则必须在进入此停止模式之前禁止实例。

23.6 I2C 中断

下表列出了 I2C 中断请求。

表 103. I2C 中断请求

中断缩写		中断事件	事件标志	使能控制位	中断清除方法	退出睡眠模式	退出 停止 模式	退出 待机 模式
I2C	I2C_EV	接收缓冲区非空	RXNE	RXIE	读取 I2C_RXDR 寄存器	是	否	否
		发送缓冲区中断状态	TXIS	TXIE	写入 I2C_TXDR 寄存器			
		停止位检测中断标志	STOPF	STOPIE	写入 STOPCF = 1			
		传输完成等待重载	TCR	TCIE	写入 I2C_CR2 (NBYTES[7:0] ≠ 0)			
		传输完成	TC		写入 START = 1 或 STOP = 1			
		地址匹配	ADDR	ADDRIE	写入 ADDRCF = 1		是(1)	
		接收到 NACK 应答	NACKF	NACKIE	写入 NACKCF = 1		否	
	I2C_ER	总线错误	BERR	ERRIE	写入 BERRCF = 1	是	否	否
		仲裁丢失	ARLO		写入 ARLOCF = 1			
		上溢/下溢	OVR		写入 OVRCF = 1			
		PEC 错误	PECERR		写入 PECERRCF = 1			
		超时/t_{LOW} 错误	TIMEOUT		写入 TIMEOUTCF = 1			
		SMBus 报警	ALERT		写入 ALERTCF = 1			

1. 仅当 I2C 实例支持从停止模式唤醒功能时，ADDR 匹配事件才能将器件从停止模式唤醒。请参见*第 23.3 节*。

23.7 I2C 寄存器

有关寄存器说明中使用的缩写，请参见*第 33 页的第 1.2 节*。

外设寄存器按字（32 位）进行访问。

23.7.1 I2C 控制寄存器 1 (I2C_CR1)

I2C control register 1

偏移地址：0x00

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	WUPEN	NOSTRETCH	SBC
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDMAEN	TXDMAEN	Res.	ANFOFF	DNF[3:0]				ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE
rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23 **PECEN**：PEC 使能 (PEC enable)
0：禁止 PEC 计算
1：使能 PEC 计算
注意：如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 23.3 节。

位 22 **ALERTEN**：SMBus 报警使能 (SMBus alert enable)
0：在主机模式 (SMBHEN = 1) 下，不支持 SMBus 报警引脚 (SMBA)。在设备模式 (SMBHEN = 0) 下，会释放 SMBA 引脚并禁止报警响应地址头（0001100x 后跟 NACK）。
1：在主机模式 (SMBHEN = 1) 下，支持 SMBus 报警引脚。在设备模式 (SMBHEN = 0) 下，会将 SMBA 引脚驱动为低电平并使能报警响应地址头（0001100x 后跟 ACK）。
注意：当 ALERTEN = 0 时，SMBA 引脚可用作标准 GPIO。
如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 23.3 节。

位 21 **SMBDEN**：SMBus 器件默认地址使能 (SMBus device default address enable)
0：禁止器件默认地址。不对地址 0b1100001x 应答。
1：使能器件默认地址。对地址 0b1100001x 应答。
注意：如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 23.3 节。

位 20 **SMBHEN**：SMBus 主机地址使能 (SMBus host address enable)
0：禁止主机地址。不对地址 0b0001000x 应答。
1：使能主机地址。对地址 0b0001000x 应答。
注意：如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 23.3 节。

位 19 **GCEN**：广播呼叫使能 (General call enable)
0：禁止广播呼叫。不对地址 0b00000000 应答。
1：使能广播呼叫。对地址 0b00000000 应答。

位 18 **WUPEN**：从停止模式唤醒使能 (Wake-up from Stop mode enable)

0：禁止从停止模式唤醒。

1：使能从停止模式唤醒。

注意：如果不支持从停止模式唤醒功能，该位保留并由硬件强制清零。请参见第 23.3 节。

注意：只有当 DNF = “0000”时，WUPEN 才能置 1

位 17 **NOSTRETCH**：时钟延长禁止 (Clock stretching disable)

该位用于在从模式下禁止时钟延长。它在主模式下必须保持清零。

0：使能时钟延长

1：禁止时钟延长

注意：该位只能在 I2C 禁止时 (PE = 0) 编程。

位 16 **SBC**：从器件字节控制 (Slave byte control)

该位用于在从设备模式下使能硬件字节控制。

0：禁止从器件字节控制

1：使能从器件字节控制

位 15 **RXDMAEN**：DMA 接收请求使能 (DMA reception requests enable)

0：禁止 DMA 接收请求

1：使能 DMA 接收请求

位 14 **TXDMAEN**：DMA 发送请求使能 (DMA transmission requests enable)

0：禁止 DMA 发送请求

1：使能 DMA 发送请求

位 13 保留，必须保持复位值。

位 12 **ANFOFF**：模拟噪声滤波器关闭 (Analog noise filter OFF)

0：使能模拟噪声滤波器

1：禁止模拟噪声滤波器

注意：该位只能在 I2C 禁止时 (PE = 0) 编程。

位 11:8 **DNF[3:0]**：数字噪声滤波器 (Digital noise filter)

这些位用于配置 SDA 和 SCL 输入端的数字噪声滤波器。数字滤波器可滤除脉宽 DNF[3:0] * t_{I2CCLK} 以下的尖峰

0000：禁止数字滤波器

0001：使能数字滤波器，可滤除的噪声尖峰脉宽可达 1 t_{I2CCLK}

...

1111：使能数字滤波器，可滤除的噪声尖峰脉宽可达 15 t_{I2CCLK}

注意：如果模拟滤波器也已使能，数字滤波将叠加在模拟滤波之上。

该滤波器只能在 I2C 禁止时 (PE = 0) 编程。

位 7 **ERRIE**：错误中断使能 (Error interrupt enable)

0：禁止错误检测中断

1：使能错误检测中断

注意：以下任一错误均会生成中断：

仲裁丢失 (ARLO)

总线错误检测 (BERR)

上溢/下溢 (OVR)

超时检测 (TIMEOUT)

PEC 错误检测 (PECERR)

报警引脚事件检测 (ALERT)

位 6 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)
0：禁止传输完成中断
1：使能传输完成中断
注意：以下任一事件均会生成中断：
传输完成 (TC)
传输完成等待重载 (TCR)

位 5 **STOPIE**：停止位检测中断使能 (Stop detection Interrupt enable)
0：禁止停止位检测 (STOPF) 中断
1：使能停止位检测 (STOPF) 中断

位 4 **NACKIE**：接收到否定应答中断使能 (Not acknowledge received Interrupt enable)
0：禁止接收到否定应答 (NACKF) 中断
1：使能接收到否定应答 (NACKF) 中断

位 3 **ADDRIE**：地址匹配中断使能（仅从模式） (Address match Interrupt enable (slave only))
0：禁止地址匹配 (ADDR) 中断
1：使能地址匹配 (ADDR) 中断

位 2 **RXIE**：RX 中断使能 (RX Interrupt enable)
0：禁止接收 (RXNE) 中断
1：使能接收 (RXNE) 中断

位 1 **TXIE**：TX 中断使能 (TX Interrupt enable)
0：禁止发送 (TXIS) 中断
1：使能发送 (TXIS) 中断

位 0 **PE**：使能外设
0：禁止外设
1：使能外设
注意：当 PE = 0 时，将释放 I2C SCL 线和 SDA 线。内部状态机和状态位均恢复为复位值。清零时，PE 必须保持低电平状态至少三个 APB 时钟周期。

23.7.2 I2C 控制寄存器 2 (I2C_CR2)

I2C control register 2

偏移地址：0x04

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	PEC BYTE	AUTO END	RE LOAD	NBYTES[7:0]							
					rs	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD[9:0]									
rs	rs	rs	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:27 保留，必须保持复位值。

位 26 **PECBYTE**：数据包错误校验字节 (Packet error checking byte)

该位由软件置 1，并可在 PEC 传输完成时、接收到停止位或匹配地址时、或者 PE=0 时由硬件清零。

0：不传输 PEC。

1：请求 PEC 发送/接收。

注意：向该位写入“0”不起作用。

当 RELOAD 置 1 时，该位不起作用。

当 SBC = 0 时，该位在从模式下不起作用。

如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 23.3 节。

位 25 **AUTOEND**：自动结束模式（主模式）(Automatic end mode (master mode))

该位由软件置 1 和清零。

0：软件结束模式：当 NBYTES 数据传输完成时，TC 标志将置 1，SCL 的低电平时间将延长直到相应软件操作结束。

1：自动结束模式：当 NBYTES 数据传输完成时，将自动发送停止位。

注意：在从模式下，或当 RELOAD 位置 1 时，该位不起作用。

位 24 **RELOAD**：NBYTES 重载模式 (NBYTES reload mode)

该位由软件置 1 和清零。

0：传输 NBYTES 数据（后跟停止位或重复起始位）之后即完成传输。

1：传输 NBYTES 数据之后未完成传输（将重载 NBYTES）。当 NBYTES 数据传输完成时，TCR 标志将置 1，SCL 的低电平时间将延长直到相应软件操作结束。

位 23:16 **NBYTES[7:0]**：字节数

在此设置待发送/接收的字节数。在从模式下，当 SBC = 0 时，该位域为无关位域。

注意：START 位置 1 时，不允许更改这些位。

位 15 **NACK**：NACK 生成（从模式）(NACK generation (slave mode))

该位由软件置 1，并可在发送 NACK 时、接收到停止位或匹配地址时、或者 PE = 0 时由硬件清零。

0：在当前接收的字节后发送 ACK。

1：在当前接收的字节后发送 NACK。

注意：向该位写入“0”不起作用。

该位仅在从模式下使用：在主接收器模式下，无论 NACK 位的值为何，最后一个字节（后跟停止位或重复起始位）后都将自动生成 NACK。

当从接收器 NOSTRETCH 模式下发生上溢时，无论 NACK 位的值为何，都将自动生成 NACK。

使能硬件 PEC 校验时 (PECBYTE = 1)，PEC 应答值与 NACK 值无关。

位 14 **STOP**：停止位生成（主模式）(Stop generation (master mode))

该位由软件置 1，并可在检测到停止位时或 PE = 0 时由硬件清零。

在主模式下：

0：不生成停止位。

1：在当前字节传输完成后生成停止位。

注意：向该位写入“0”不起作用。

位 13 **START**：生成起始位

该位由软件置 1，并可在发送起始位（后跟地址序列）之后、发生仲裁丢失时、从模式下发生地址匹配时、出现超时错误时、或者 PE = 0 时由硬件清零。

0：不生成起始位。

1：生成重复起始/起始位：

如果 I2C 已处于主模式下且 AUTOEND = 0，则将该位置 1 会在 NBYTES 传输结束后且 RELOAD = 0 的情况下生成重复起始位。

否则，将该位置 1 会在总线释放后立即生成起始位。

注意：向该位写入“0”不起作用。

即使总线繁忙或 I2C 处于从模式，也可将 START 位置 1。

当 RELOAD 置 1 时，该位不起作用。

位 12 **HEAD10R**：读方向传输时，只发送 10 位地址的前 7 位地址头字节（主接收器模式）(10-bit address header only read direction (master receiver mode))

0：主器件发送完整的 10 位从地址读序列：起始位 + 带写方向的 2 字节 10 位地址 + 重复起始位 + 带读方向的 10 位地址的前 7 位。

1：主器件只发送 10 位地址的前 7 位，后跟读方向。

注意：START 位置 1 时，不允许更改该位。

位 11 **ADD10**：10 位寻址模式（主模式）(10-bit addressing mode (master mode))

0：主器件工作在 7 位寻址模式下

1：主器件工作在 10 位寻址模式下

注意：START 位置 1 时，不允许更改该位。

位 10 **RD_WRN**：传输方向（主模式）(Transfer direction (master mode))

0：主器件请求写传输。

1：主器件请求读传输。

注意：START 位置 1 时，不允许更改该位。

位 9:0 **SADD[9:0]**：从地址（主模式）(Slave address (master mode))

在 7 位寻址模式 (ADD10 = 0) 下：

SADD[7:1] 必须写入待发送的 7 位从地址。SADD[9]、SADD[8] 和 SADD[0] 位为无关位。

在 10 位寻址模式 (ADD10 = 1) 下：

SADD[9:0] 必须写入待发送的 10 位从地址。

注意：START 位置 1 时，不允许更改这些位。

23.7.3 I2C 自身地址 1 寄存器 (I2C_OAR1)

I2C own address 1 register

偏移地址：0x08

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA1EN	Res.	Res.	Res.	Res.	OA1 MODE	OA1[9:0]									
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **OA1EN**：设备自身地址 1 使能 (Own address 1 enable)
0：禁止设备自身地址 1。不对接收的从地址 OA1 应答。
1：使能设备自身地址 1。对接收的从地址 OA1 应答。

位 14:11 保留，必须保持复位值。

位 10 **OA1MODE**：设备自身地址 1 10 位模式 (Own address 1 10-bit mode)
0：设备自身地址 1 为 7 位地址。
1：设备自身地址 1 为 10 位地址。
注意：仅可在 OA1EN = 0 时写入该位。

位 9:0 **OA1[9:0]**：接口自身从地址 (Interface own slave address)
7 位寻址模式：OA1[7:1] 包含 7 位自身从地址。OA1[9]、OA1[8] 和 OA1[0] 位为无关位。
10 位寻址模式：OA1[9:0] 包含 10 位自身从地址。
注意：仅可在 OA1EN = 0 时写入这些位。

23.7.4 I2C 自身地址 2 寄存器 (I2C_OAR2)

I2C own address 2 register

偏移地址：0x0C

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA2EN	Res.	Res.	Res.	Res.	OA2MSK[2:0]			OA2[7:1]							Res.
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位 31:16 保留，必须保持复位值。

位 15 **OA2EN**：设备自身地址 2 使能 (Own address 2 enable)

0：禁止设备自身地址 2。不对接收的从地址 OA2 应答。

1：使能设备自身地址 2。对接收的从地址 OA2 应答。

位 14:11 保留，必须保持复位值。

位 10:8 **OA2MSK[2:0]**：设备自身地址 2 屏蔽位 (Own Address 2 masks)

000：无屏蔽

001：OA2[1] 被屏蔽，为无关位。仅比较 OA2[7:2]。

010：OA2[2:1] 被屏蔽，为无关位。仅比较 OA2[7:3]。

011：OA2[3:1] 被屏蔽，为无关位。仅比较 OA2[7:4]。

100：OA2[4:1] 被屏蔽，为无关位。仅比较 OA2[7:5]。

101：OA2[5:1] 被屏蔽，为无关位。仅比较 OA2[7:6]。

110：OA2[6:1] 被屏蔽，为无关位。仅比较 OA2[7]。

111：OA2[7:1] 被屏蔽，为无关位。不进行比较，对接收到的全部 7 位地址（保留位除外）应答。

注意：仅可在 OA2EN = 0 时写入这些位。

只要 OA2MSK 不等于 0，即使比较匹配，也不会对保留的 I2C 地址（0b0000xxx 和 0b1111xxx）应答。

位 7:1 **OA2[7:1]**：接口地址

7 位寻址模式：7 位地址

注意：仅可在 OA2EN = 0 时写入这些位。

位 0 保留，必须保持复位值。

23.7.5 I2C 时序寄存器 (I2C_TIMINGR)

I2C timing register

偏移地址：0x10

复位值：0x0000 0000

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL[3:0]				SDADEL[3:0]			
rw	rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH[7:0]								SCLL[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 **PRESC[3:0]**：时序预分频因子 (Timing prescaler)

该位域用于对 I2CCLK 进行预分频，以生成用于数据建立和保持计数器（请参见 *I2C 时序*）以及 SCL 高电平和低电平计数器（请参见 *I2C 主模式初始化*）的时钟周期 t_{PRESC}。

t_{PRESC} = (PRESC + 1) x t_{I2CCLK}

位 27:24 保留，必须保持复位值。

位 23:20 **SCLDEL[3:0]**：数据建立时间

该位域用于在 SDA 边沿和 SCL 上升沿之间生成延时 t_{SCLDEL}。在主模式和从模式下，如果 NOSTRETCH = 0，则 SCL 线的低电平时间将在 t_{SCLDEL} 期间延长。

t_{SCLDEL} = (SCLDEL + 1) x t_{PRESC}

注意：t_{SCLDEL} 用于生成 $t_{SU:DAT}$ 时序。

位 19:16 **SDADEL[3:0]**：数据保持时间

该位域用于在 SCL 下降沿和 SDA 边沿之间生成延时 t_{SDADEL}。在主模式和从模式下，如果 NOSTRETCH = 0，则 SCL 线的低电平时间将在 t_{SDADEL} 期间延长。

t_{SDADEL}= SDADEL x t_{PRESC}

注意：SDADEL 用于生成 $t_{HD:DAT}$ 时序。

位 15:8 **SCLH[7:0]**：SCL 高电平周期（主模式）(SCL high period (master mode))

在主模式下，该位域用于生成 SCL 高电平周期。

t_{SCLH} = (SCLH + 1) x t_{PRESC}

*注意：SCLH 还用于生成 $t_{SU:STO}$ 和 t*HD:STA *时序。*

位 7:0 **SCLL[7:0]**：SCL 低电平周期（主模式）(SCL low period (master mode))

在主模式下，该位域用于生成 SCL 低电平周期。

t_{SCLL} = (SCLL + 1) x t_{PRESC}

注意：SCLL 还用于生成 t_{BUF} 和 $t_{SU:STA}$ 时序。

注意：　该寄存器必须在 I2C 禁止时 (PE = 0) 进行配置。

注意：　STM32CubeMX 工具计算 I2C_TIMINGR 内容并在 I2C 配置窗口中进行显示。

23.7.6 I2C 超时寄存器 (I2C_TIMEOUTR)

I2C timeout register

偏移地址：0x14

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEXTEN	Res.	Res.	Res.	TIMEOUTB[11:0]											
rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMOUTEN	Res.	Res.	TIDLE	TIMEOUTA[11:0]											
rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **TEXTEN**：时钟信号延展超时使能 (Extended clock timeout enable)

0：禁止时钟信号延展超时检测

1：使能时钟信号延展超时检测当 I2C 接口执行 SCL 延展的累积时间超过 $t_{LOW:EXT}$ 时，将检测到超时错误 (TIMEOUT = 1)。

位 30:28 保留，必须保持复位值。

位 27:16 **TIMEOUTB[11:0]**：总线超时 B (Bus timeout B)

该位域用于配置累积时钟延展超时：

在主模式下，将检测主器件的累积时钟低电平延展时间 ($t_{LOW:MEXT}$)

在从模式下，将检测从器件的累积时钟低电平延展时间 ($t_{LOW:SEXT}$)

$t_{LOW:EXT}$ = (TIMEOUTB + TIDLE = 01) x 2048 x t_{I2CCLK}

注意：仅可在 TEXTEN = 0 时写入这些位。

位 15 **TIMOUTEN**：时钟超时使能 (Clock timeout enable)

0：禁止 SCL 超时检测

1：使能 SCL 超时检测：当 SCL 的低电平时间超过 $t_{TIMEOUT}$ (TIDLE = 0)，或 SCL 的高电平时间超过 t_{IDLE} (TIDLE = 1) 时，将检测到超时错误 (TIMEOUT = 1)。

位 14:13 保留，必须保持复位值。

位 12 **TIDLE**：空闲时钟超时检测 (Idle clock timeout detection)

0：TIMEOUTA 用于检测 SCL 低电平超时

1：TIMEOUTA 用于检测 SCL 和 SDA 高电平超时（总线空闲条件）

注意：仅可在 TIMOUTEN = 0 时写入该位。

位 11:0 **TIMEOUTA[11:0]**：总线超时 A (Bus timeout A)

该位域用于配置：

SCL 低电平超时条件 $t_{TIMEOUT}$（当 TIDLE = 0 时）

$t_{TIMEOUT}$= (TIMEOUTA + 1) x 2048 x t_{I2CCLK}

总线空闲条件，即 SCL 和 SDA 高电平（当 TIDLE = 1 时）

t_{IDLE}= (TIMEOUTA + 1) x 4 x t_{I2CCLK}

注意：仅可在 TIMOUTEN = 0 时写入这些位。

注意：如果不支持 SMBus 功能，该寄存器保留，并由硬件强制为“0x00000000”。请参见第 23.3 节。

23.7.7 I2C 中断和状态寄存器 (I2C_ISR)

I2C interrupt and status register

偏移地址：0x18

复位值：0x0000 0001

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR
								r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res.	ALERT	TIME OUT	PEC ERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
r		r	r	r	r	r	r	r	r	r	r	r	r	rs	rs

位 31:24 保留，必须保持复位值。

位 23:17 **ADDCODE[6:0]**：地址匹配代码（从模式）(Address match code (Slave mode))

发生地址匹配事件时 (ADDR = 1)，这些位更新为接收到的地址。

在 10 位地址的情况下，ADDCODE 提供 10 位地址的头字节，后跟地址的两个 MSB。

位 16 **DIR**：传输方向（从模式）(Transfer direction (slave mode))

该标志在发生地址匹配事件时 (ADDR = 1) 更新。

0：写传输，从器件进入接收器模式。

1：读传输，从器件进入发送器模式。

位 15 **BUSY**：总线忙碌 (Bus busy)

该标志用于指示总线上正在进行通信。当检测到起始位时，该位由硬件置 1。当检测到停止位或 PE = 0 时，该位由硬件清零。

位 14 保留，必须保持复位值。

位 13 **ALERT**：SMBus 报警

当 SMBHEN = 1（SMBus 主机配置）、ALERTEN = 1 且在 SMBA 引脚上检测到 SMBALERT 事件（下降沿）时，该标志由硬件置 1。该位由软件清零，方法是将 ALERTCF 位置 1。

注意：当 PE = 0 时，该位由硬件清零。

如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 23.3 节。

位 12 **TIMEOUT**：超时或 t_{LOW} 检测标志 (Timeout or t_{LOW} detection flag)

发生超时或延长时钟超时时，该标志由硬件置 1。该位由软件清零，方法是将 TIMEOUTCF 位置 1。

注意：当 PE = 0 时，该位由硬件清零。

如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 23.3 节。

位 11 **PECERR**：接收期间 PEC 错误 (PEC Error in reception)

当接收到的 PEC 与 PEC 寄存器的内容不匹配时，该标志由硬件置 1。接收到错误的 PEC 后，将自动发送 NACK。该标志由软件清零，方法是将 PECCF 位置 1。

注意：当 PE = 0 时，该位由硬件清零。

如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 23.3 节。

位 10 **OVR**：上溢/下溢（从模式）(Overrun/Underrun (slave mode))

在从模式下且 NOSTRETCH = 1 时，如果发生上溢/下溢错误，该标志由硬件置 1。该标志由软件清零，方法是将 OVRCF 位置 1。

注意：当 PE = 0 时，该位由硬件清零。

位 9 **ARLO**：仲裁丢失

发生仲裁丢失时，该标志由硬件置 1。该标志由软件清零，方法是将 ARLOCF 位置 1。

注意：当 PE = 0 时，该位由硬件清零。

位 8 **BERR**：总线错误 (Bus error)

当检测到错位的起始位或停止位，而外设也参与传输时，该标志由硬件置 1。在从模式下的地址阶段，该标志不会置 1。该标志由软件清零，方法是将 *BERRCF 位置 1*。

注意：当 PE = 0 时，该位由硬件清零。

位 7 **TCR**：传输完成等待重载

当 RELOAD = 1 且 NBYTES 数据传输完成时，该标志由硬件置 1。当 NBYTES 写入一个非零值时，该标志由软件清零。

注意：当 PE = 0 时，该位由硬件清零。

该标志单独用于主模式，SBC 位置 1 时单独用于从模式。

位 6 **TC**：传输完成（主模式）(Transfer Complete (master mode))

当 RELOAD = 0、AUTOEND = 0 且 NBYTES 数据传输完成时，该标志由硬件置 1。当 START 位或 STOP 位置 1 时，该标志由软件清零。

注意：当 PE = 0 时，该位由硬件清零。

位 5 **STOPF**：停止位检测标志 (Stop detection flag)

当在总线上检测到停止位，且外设也参与本次传输时，该标志由硬件置 1：

– 外设作为主器件，该位置 1 的前提是外设已经发出停止位。

– 外设作为从器件，该位置位的前提条件是此次传输的寻址对象就是该外设。

该标志由软件清零，方法是将 STOPCF 位置 1。

注意：当 PE = 0 时，该位由硬件清零。

位 4 **NACKF**：接收到否定应答标志(Not Acknowledge received flag)

传输完字节后接收到 NACK 时，该标志由硬件置 1。该标志由软件清零，方法是将 NACKCF 位置 1。

注意：当 PE = 0 时，该位由硬件清零。

位 3 **ADDR**：地址匹配（从模式）(Address matched (slave mode))

接收到的地址与使能的从设备地址之一匹配时，该位由硬件置 1。该位由软件清零，方法是将 *ADDRCF 位置 1*。

注意：当 PE = 0 时，该位由硬件清零。

位 2 **RXNE**：接收数据寄存器不为空（接收器） (Receive data register not empty (receivers))

当接收到的数据已复制到 I2C_RXDR 寄存器且准备就绪可供读取时，该位由硬件置 1。读取 I2C_RXDR 时，将清零该位。

注意：当 PE = 0 时，该位由硬件清零。

位 1 **TXIS**：发送中断状态（发送器）(Transmit interrupt status (transmitters))

当 I2C_TXDR 寄存器为空时，该位由硬件置 1，待发送的数据必须写入 I2C_TXDR 寄存器。下一个待发送的数据写入 I2C_TXDR 寄存器时，该位被清零。

该位只能在 NOSTRETCH = 1 时由软件写入“1”，以生成 TXIS 事件（TXIE = 1 时为中断，TXDMAEN = 1 时为 DMA 请求）。

注意：当 PE = 0 时，该位由硬件清零。

位 0 **TXE**：发送数据寄存器为空（发送器）(Transmit data register empty (transmitters))

当 I2C_TXDR 寄存器为空时，该位由硬件置 1。下一个待发送的数据写入 I2C_TXDR 寄存器时，该位被清零。

该位可由软件写入“1”，以刷新发送数据寄存器 I2C_TXDR。

注意：当 PE = 0 时，该位由硬件置 1。

23.7.8 I2C 中断清零寄存器 (I2C_ICR)

I2C interrupt clear register

偏移地址：0x1C

复位值：0x0000 0000

访问：无等待周期

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ALERTCF	TIMOUTCF	PECCF	OVRCF	ARLOCF	BERRCF	Res.	Res.	STOPCF	NACKCF	ADDRCF	Res.	Res.	Res.
		w	w	w	w	w	w			w	w	w			

位 31:14 保留，必须保持复位值。

位 13 **ALERTCF**：报警标志清零 (Alert flag clear)
将 1 写入该位时，I2C_ISR 寄存器中的 ALERT 标志将清零。
注意：如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 23.3 节。

位 12 **TIMOUTCF**：超时检测标志清零 (Timeout detection flag clear)
将 1 写入该位时，I2C_ISR 寄存器中的 TIMEOUT 标志将清零。
注意：如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 23.3 节。

位 11 **PECCF**：PEC 错误标志清零 (PEC Error flag clear)
将 1 写入该位时，I2C_ISR 寄存器中的 PECERR 标志将清零。
注意：如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 23.3 节。

位 10 **OVRCF**：上溢/下溢标志清零 (Overrun/Underrun flag clear)
将 1 写入该位时，I2C_ISR 寄存器中的 OVR 标志将清零。

位 9 **ARLOCF**：仲裁丢失标志清零 (Arbitration lost flag clear)
将 1 写入该位时，I2C_ISR 寄存器中的 ARLO 标志将清零。

位 8 **BERRCF**：总线错误标志清零 (Bus error flag clear)
将 1 写入该位时，I2C_ISR 寄存器中的 BERRF 标志将清零。

位 7:6 保留，必须保持复位值。

位 5 **STOPCF**：停止位检测标志清零 (STOP detection flag clear)
将 1 写入该位时，I2C_ISR 寄存器中的 STOPF 标志将清零。

位 4 **NACKCF**：否定应答标志清零 (Not Acknowledge flag clear)
向该位写入 1 时，I2C_ISR 寄存器中的 NACKF 标志将清零。

位 3 **ADDRCF**：地址匹配标志清零 (Address Matched flag clear)
将 1 写入该位时，I2C_ISR 寄存器中的 ADDR 标志将清零。将 1 写入该位时，I2C_CR2 寄存器中的 START 位也将清零。

位 2:0 保留，必须保持复位值。

23.7.9 I2C PEC 寄存器 (I2C_PECR)

I2C PEC register

偏移地址：0x20

复位值：0x0000 0000

访问：无等待周期

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **PEC[7:0]**：数据包错误校验寄存器 (Packet error checking register)
当 PECEN=1 时，该位域包含内部 PEC。
当 PE = 0 时，PEC 由硬件清零。

注意： *如果不支持 SMBus 功能，该寄存器保留，并由硬件强制为“0x00000000”。请参见第 23.3 节。*

23.7.10 I2C 接收数据寄存器 (I2C_RXDR)

I2C receive data register

偏移地址：0x24

复位值：0x0000 0000

访问：无等待周期

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **RXDATA[7:0]**：8 位接收数据 (8-bit receive data)
从 I^2C 总线接收的数据字节。

23.7.11 I2C 发送数据寄存器 (I2C_TXDR)

I2C transmit data register

偏移地址：0x28

复位值：0x0000 0000

访问：无等待周期

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **TXDATA[7:0]**：8 位发送数据 (8-bit transmit data)
待发送到 I^2C 总线的数据字节
注意：仅可在 TXE = 1 时写入这些位。

23.7.12 I2C 寄存器映射

下表提供了 I2C 寄存器映射和复位值。

表 104. I2C 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0	I2C_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	WUPEN	NOSTRETCH	SBC	RXDMAEN	TXDMAEN	Res.	ANFOFF	DNF[3:0]				ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE
	Reset value									0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
0x4	I2C_CR2	Res.	Res.	Res.	Res.	Res.	PECBYTE	AUTOEND	RELOAD	NBYTES[7:0]								NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD[9:0]									
	Reset value						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8	I2C_OAR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA1EN	Res.	Res.	Res.	Res.	OA1MODE	OA1[9:0]									
	Reset value																	0					0	0	0	0	0	0	0	0	0	0	0
0xC	I2C_OAR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA2EN	Res.	Res.	Res.	Res.	OA2MSK [2:0]			OA2[7:1]							Res.
	Reset value																	0					0	0	0	0	0	0	0	0	0	0	
0x10	I2C_ TIMINGR	PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL [3:0]				SDADEL [3:0]				SCLH[7:0]								SCLL[7:0]							
	Reset value	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 104. I2C 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x14	I2C_ TIMEOUTR	TEXTEN	Res.	Res.	Res.	TIMEOUTB[11:0]												TIMOUTEN	Res.	Res.	TIDLE	TIMEOUTA[11:0]											
	Reset value	0				0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	I2C_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR	BUSY	Res.	ALERT	TIMEOUT	PECERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
	Reset value									0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x1C	I2C_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALERTCF	TIMOUTCF	PECCF	OVRCF	ARLOCF	BERRCF	Res.	Res.	STOPCF	NACKCF	ADDRCF	Res.	Res.	Res.
	Reset value																			0	0	0	0	0	0			0	0	0			
0x20	I2C_PECR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x24	I2C_RXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x28	I2C_TXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

24 通用同步/异步收发器 (USART)

本节介绍通用同步/异步收发器 (USART)。

24.1 USART 简介

USART 能够灵活地与外部设备进行全双工数据交换，满足外部设备对工业标准 NRZ 异步串行数据格式的要求。USART 通过小数波特率发生器实现了多种波特率。

USART 不仅支持同步单向通信和半双工单线通信，以及 LIN（局域互连网络）、智能卡协议、IrDA（红外线数据协会）SIR ENDEC 规范和调制解调器操作 (CTS/RTS)，还支持多处理器通信。

通过配置多个缓冲区使用 DMA（直接存储器访问）可实现高速数据通信。

24.2 USART 主要特性

- 全双工异步通信
- NRZ 标准格式（标记/空格）
- 可配置为 16 倍过采样或 8 倍过采样，从而在速度容差与时钟容差之间取得最佳平衡
- 波特率发生器系统
- 两个用于收发数据的内部 FIFO
  每个 FIFO 均可由软件使能/禁止，并且均带有一个状态标志。
- 通用可编程收发波特率
- 双时钟域，带有独立于 PCLK 的外设专用内核时钟
- 自动波特率检测
- 数据字长度可编程（7 位、8 位或 9 位）
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 停止位可配置（支持 1 个或 2 个停止位）
- 用于同步通信的同步主/从模式和时钟输出/输入
- SPI 从发送下溢错误标志
- 单线半双工通信
- 使用 DMA 实现连续通信
- 使用中央 DMA 在预留的 SRAM 缓冲区中收/发字节
- 发射器和接收器有单独的使能位
- 发送和接收的单独信号极性控制
- Tx/Rx 引脚可配置互换
- 调制解调器和 RS-485 收发器的硬件流控制
- 通信控制/错误检测标志
- 奇偶校验控制：
  – 发送奇偶校验位
  – 检查接收的数据字节的奇偶性
- 具有标志的中断源
- 多处理器通信：从静默模式唤醒（通过空闲线检测或地址标记检测）
- 从停止模式唤醒

24.3 USART 扩展特性

- LIN 主模式同步中断发送功能和 LIN 从模式中断检测功能
  - 对 USART 进行 LIN 硬件配置时可生成 13 位停止符号和检测 10/11 位停止符号
- IrDA SIR 编解码器在正常模式下支持 3/16 位持续时间
- 智能卡模式
  - 对于 ISO/IEC 7816-3 标准中定义的智能卡，支持 T = 0 和 T = 1 异步协议
  - 智能卡工作模式下，支持 0.5 和 1.5 个停止位
- 支持 ModBus 通信
  - 超时功能
  - CR/LF 字符识别

24.4 USART 实现

下表介绍了 USART 实现。其中还包含 LPUART 作为参照。

表 105. STM32C0x1 特性

USART 实例	STM32C0x1
USART1	FULL
USART2	BASIC

表 106. USART 特性

USART 模式/特性(1)	全面的功能集	基本的功能集
调制解调器的硬件流控制	X	X
使用 DMA 进行连续通信	X	X
多处理器通信	X	X
同步模式（主/从）	X	X
智能卡模式	X	-
单线半双工通信	X	X
IrDA SIR ENDEC 模块	X	-
LIN 模式	X	-
双时钟域和从低功耗模式唤醒	X	-
接收器超时中断	X	-
Modbus 通信	X	-
自动波特率检测	X	-
驱动器使能	X	X
USART 数据长度	7 位、8 位和 9 位	
Tx/Rx FIFO	X	-
Tx/Rx FIFO 大小	8	-
预分频器	X	-

1. X = 支持。

24.5 USART 功能描述

24.5.1 USART 框图

图 241. USART 框图

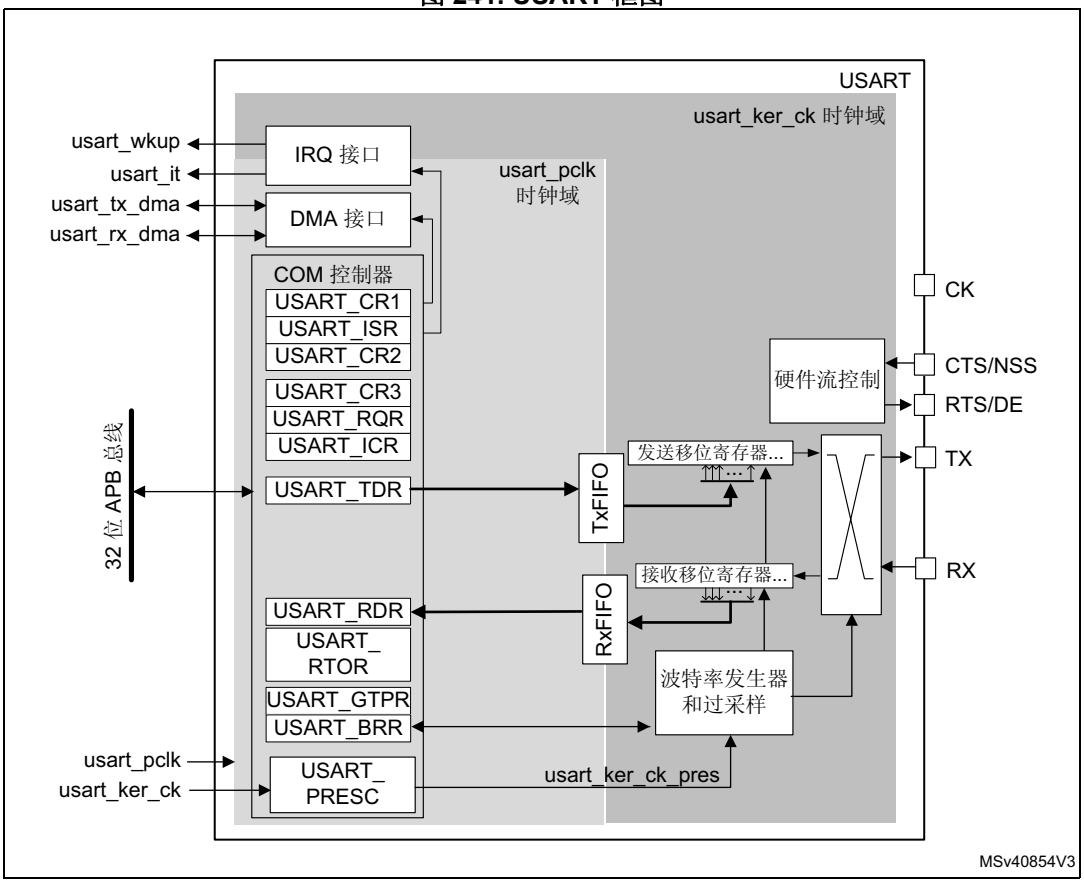


图 241 中的简化框图显示的是两个完全独立的时钟域：

- **usart_pclk** 时钟域

  usart_pclk 时钟信号为外设总线接口提供时钟。需要访问 USART 寄存器时，该信号必须有效。

- **usart_ker_ck** 内核时钟域。

  usart_ker_ck 是 USART 时钟源。它独立于 **usart_pclk** ，由 RCC 提供。因此，即使 **usart_ker_ck** 时钟停止，也可以对 USART 寄存器执行读/写操作。

  禁用双时钟域功能时，**usart_ker_ck** 时钟与 **usart_pclk** 时钟相同。

usart_pclk 和 **usart_ker_ck** 之间无任何约束：**usart_ker_ck** 可快于/可慢于 **usart_pclk**。唯一的限制是软件以足够快的速度管理通信的能力。

USART 工作在 SPI 从器件模式下时，会使用源自外部 CK 信号（由外部主 SPI 器件提供）的串行接口时钟来处理数据流。**usart_ker_ck** 的时钟频率必须至少 3 倍于 CK 上的输入时钟。

24.5.2 USART 信号

USART 双向通信

USART 双向通信需要至少两个引脚：接收数据输入引脚 (RX) 和发送数据输出引脚 (TX)：

- **RX**（接收数据输入引脚）

  RX 为串行数据输入引脚。采用过采样技术进行数据恢复，即区分有效输入数据和噪声。

- **TX**（发送数据输出引脚）

  如果关闭发送器，该输出引脚模式由其 I/O 端口配置决定。如果使能了发送器但没有需要发送的数据，则 TX 引脚处于高电平。在单线和智能卡模式下，该 I/O 用于发送和接收数据。

RS232 硬件流控制模式

在 RS232 硬件流控制模式下需要以下引脚：

- **CTS**（清除以发送）

  如果驱动为高电平，则该信号用于在当前传输结束时阻止数据发送。

- **RTS**（请求以发送）

  如果为低电平，则该信号用于指示 USART 已准备好接收数据。

RS485 硬件控制模式

在 RS485 硬件控制模式下需要以下引脚：

- **DE**（驱动器使能）

  该信号用于激活外部收发器的发送模式。

注意： *DE 和 RTS 共用同一个引脚。*

同步主/从模式和智能卡模式

在同步主/从模式和智能卡模式下需要以下引脚：

- CK

  该引脚在同步主模式和智能卡模式下用作时钟输出。

  它在同步从模式下用作时钟输入。

  在同步主模式下，该引脚用于输出发送器数据时钟，以便按照 SPI 主器件模式进行同步发送（起始位和结束位上无时钟脉冲，可通过软件向最后一个数据位发送时钟脉冲）。同时，RX 引脚可同步接收数据。该机制可用于控制带移位寄存器的外设（如 LCD 驱动器）。时钟相位和极性可通过软件编程。

  在智能卡模式下，CK 输出向智能卡提供时钟。

- NSS

  该引脚在同步从模式下用作从器件选择输入。

注意： *NSS 和 CTS 共用同一个引脚。*

24.5.3 USART 字符说明

可通过对 USART_CR1 寄存器中的 M 位（M0：位 12，M1：位 28）进行编程来将字长设置为 7 位、8 位或 9 位（请参见*图 242*）。

- 7 位字符长度：M[1:0] = “10”
- 8 位字符长度：M[1:0] = “00”
- 9 位字符长度：M[1:0] = “01”

注意： *在 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。*

在默认情况下，信号（TX 或 RX）在起始位工作期间处于低电平状态。在停止位工作期间处于高电平状态。

通过极性配置控制，可以单独针对每个信号对这些值取反。

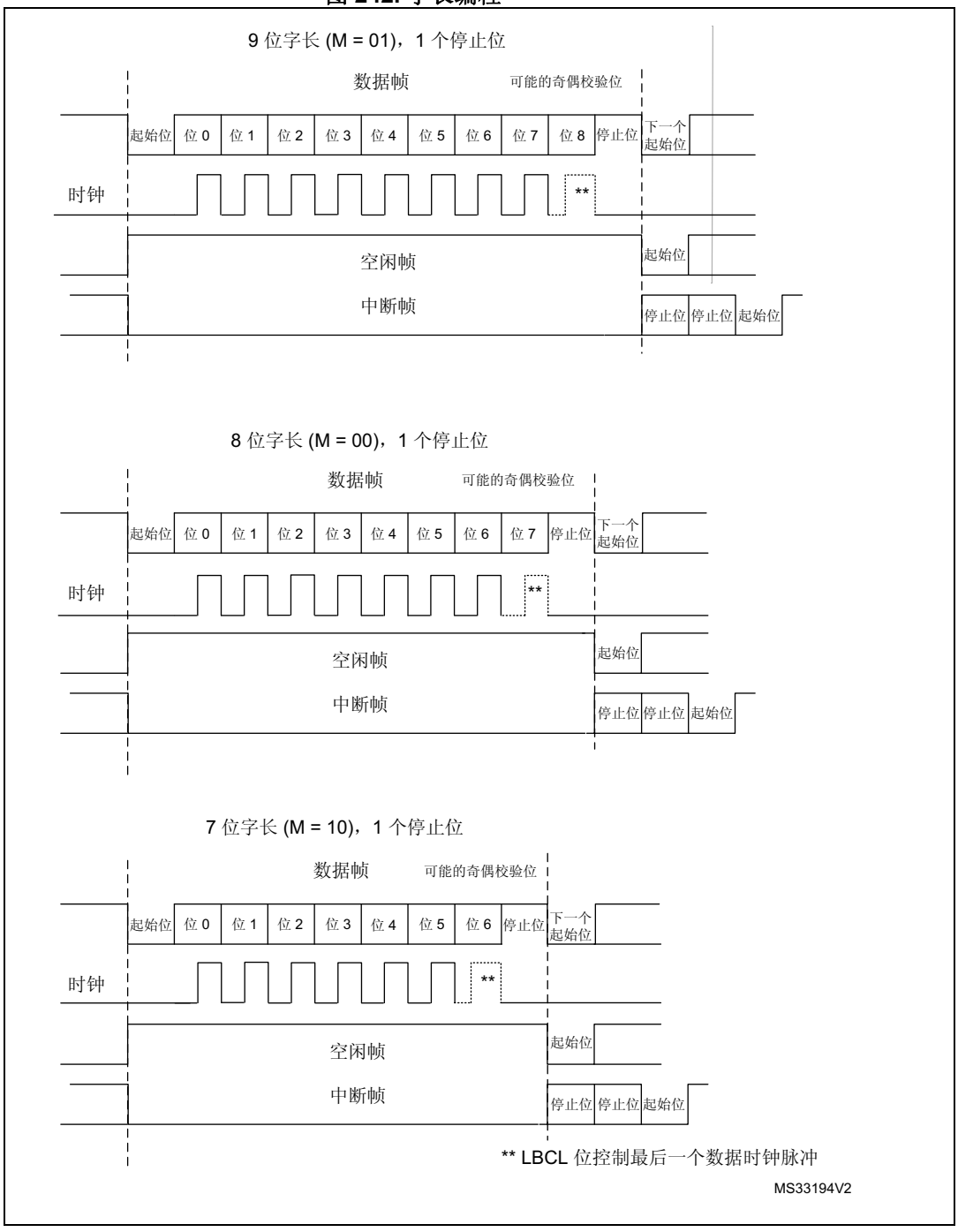
空闲字符可理解为整个帧周期内电平均为“1”（停止位的电平也是“1”）。

中断字符可理解为在一个帧周期内接收到的电平均为“0”。发送器在中断帧的末尾插入 2 个停止位。

发送和接收操作由通用波特率发生器驱动。当发送器和接收器的使能位置 1 时，将分别生成发送时钟和接收时钟。

下面给出了各个块的详细说明。

图 242. 字长编程


9 位字长 (M = 01)，1 个停止位
数据帧
可能的奇偶校验位
起始位
位 0
位 1
位 2
位 3
位 4
位 5
位 6
位 7
位 8
停止位
下一个起始位
时钟
**
空闲帧
起始位
中断帧
停止位
停止位
起始位
8 位字长 (M = 00)，1 个停止位
数据帧
可能的奇偶校验位
起始位
位 0
位 1
位 2
位 3
位 4
位 5
位 6
位 7
停止位
下一个起始位
时钟
**
空闲帧
起始位
中断帧
停止位
停止位
起始位
7 位字长 (M = 10)，1 个停止位
数据帧
可能的奇偶校验位
起始位
位 0
位 1
位 2
位 3
位 4
位 5
位 6
停止位
下一个起始位
时钟
**
空闲帧
起始位
中断帧
停止位
停止位
起始位
** LBCL 位控制最后一个数据时钟脉冲
MS33194V2

24.5.4 USART FIFO 和阈值

USART 可工作在 FIFO 模式下。

USART 具有一个发送 FIFO (TXFIFO) 和一个接收 FIFO (RXFIFO)。可通过将 USART_CR1 寄存器中的 FIFOEN（位 29）置 1 使能 FIFO 模式。仅 UART、SPI 和智能卡模式下支持该模式。

最大数据字长度为 9 位，因此 TXFIFO 为 9 位宽。不过，RXFIFO 的默认宽度为 12 位。这是因为接收器不仅在 FIFO 中存储数据，而且还存储与每个字符相关的错误标志（奇偶校验错误、噪声错误和帧错误标志）。

注意： *接收的数据与相应的标志一起存储在 RXFIFO 中，但读取 RDR 时仅读取数据。*

状态标志位于 USART_ISR 寄存器中。

可以配置触发 Tx 和 RX 中断的 TXFIFO 和 RXFIFO 阈值。这些阈值通过 USART_CR3 控制寄存器中的 RXFTCFG 和 TXFTCFG 位域进行编程。

在这种情况下：

- 当 RXFIFO 中接收的数据量达到 RXFTCFG 位域中编程的阈值时，USART_ISR 寄存器中的 RXFT 标志置 1 并生成相应中断（如果使能）。

  这意味着 RXFIFO 被填充，直到 RXFIFO 中的数据量等于编程的阈值。

  已接收到 RXFTCFG 数据：USART_RDR 中有 1 个数据，RXFIFO 中有 (RXFTCFG - 1) 个数据。例如，如果将 RXFTCFG 编程为“101”，则在接收到对应于 FIFO 大小的数据量（RXFIFO 中有（FIFO 大小 - 1）个数据，USART_RDR 中有 1 个数据）时，RXFT 标志将置 1。因此，下一个接收到的数据不会将上溢标志置 1。
- 当 TXFIFO 中空单元数量达到 TXFTCFG 位域中编程的阈值时，USART_ISR 寄存器中的 TXFT 标志置 1 并生成相应中断（如果使能）。

  这意味着 TXFIFO 被清空，直到 TXFIFO 中的空存储单元量等于编程的阈值。

24.5.5 USART 发送器

发送器可发送 7 位、8 位或 9 位的数据字，具体取决于 M 位的状态。要激活发送器功能，必须将发送使能位 (TE) 置 1。发送移位寄存器中的数据在 TX 引脚输出，相应的时钟脉冲在 CK 引脚输出。

字符发送

USART 发送期间，首先通过 TX 引脚移出数据的最低有效位（默认配置）。在该模式下，USART_TDR 寄存器的缓冲区 (TDR) 位于内部总线和发送移位寄存器之间。

使能 FIFO 模式时，写入到发送数据寄存器 (USART_TDR) 中的数据会在 TXFIFO 中排队。

每个字符前面都有一个起始位，其对应于一个位周期的逻辑低电平。字符由可配置数量的停止位终止。

停止位的数量可配置为 0.5、1、1.5 或 2。

注意： *向 USART_TDR 中写入要发送的数据前，TE 位必须先置 1。*

数据发送期间不应复位 TE 位。发送期间复位 TE 位会冻结波特率计数器，进而损坏 TX 引脚上的数据。当前发送的数据随即丢失。

使能 TE 位时，将发送空闲帧。

可配置的停止位

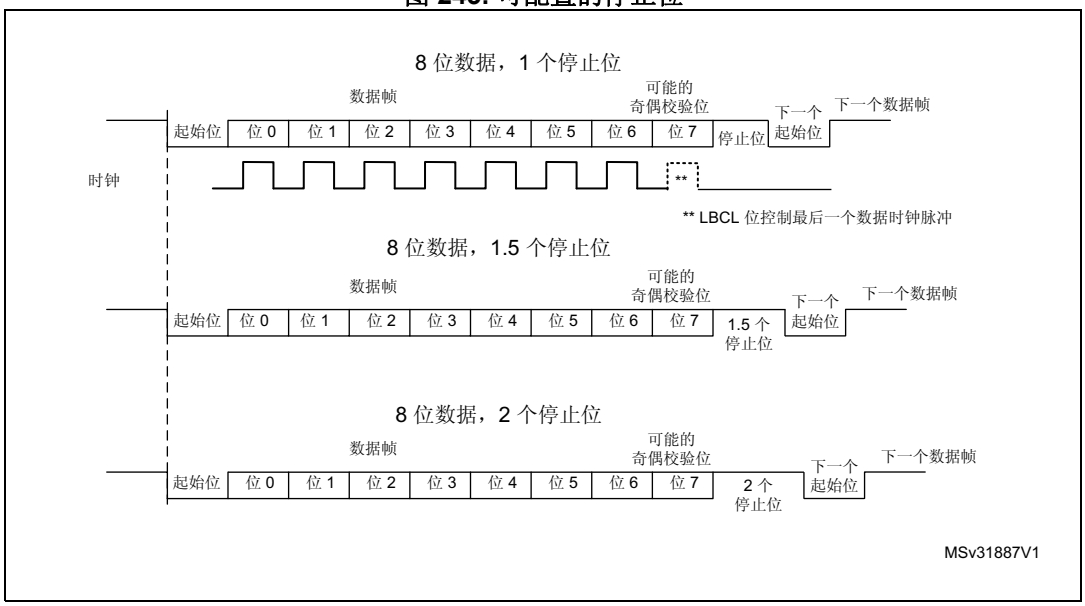
可以在 USART_CR2 的位 13 和位 12 中编程将随各个字符发送的停止位的数量。

- ***1 个停止位***：这是停止位数量的默认值。
- ***2 个停止位***：正常 USART 模式、单线模式和调制解调器模式支持该值。
- ***1.5 个停止位***：用于智能卡模式。

空闲帧发送包括停止位。

中断传输是 10 个低电平位（M[1:0] = “00”时）、11 个低电平位（M[1:0] = “01”时）或 9 个低电平位（M[1:0] = “10”时），然后是 2 个停止位（请参见*图 243*）。无法传送长中断（中断长度大于 9/10/11 个低电平位）。

图 243. 可配置的停止位



字符发送步骤

要发送字符，需遵循以下步骤：

1. 对 USART_CR1 中的 M 位进行编程以定义字长。
2. 使用 USART_BRR 寄存器选择所需波特率。
3. 对 USART_CR2 中的停止位数量进行编程。
4. 通过向 USART_CR1 寄存器中的 UE 位写入 1 使能 USART。
5. 如果必须进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAT)。按照*第 24.5.19 节：使用 USART 和 DMA 进行连续通信*中的说明配置 DMA 寄存器。
6. 将 USART_CR1 中的 TE 位置 1 以便在首次发送时发送一个空闲帧。
7. 在 USART_TDR 寄存器中写入要发送的数据。为每个要在单缓冲区模式下发送的数据重复这一步骤。
   - 禁止 FIFO 模式时，向 USART_TDR 写入数据会将 TXE 标志清零。
   - 使能 FIFO 模式时，向 USART_TDR 写入数据会为 TXFIFO 中增添一个数据。当 TXFNF 标志置 1 时，对 USART_TDR 寄存器的写操作可以执行。该标志会保持置 1，直到 TXFIFO 已满。

8. 将最后一个数据写入 USART_TDR 寄存器后，等待 TC = 1。
   – 禁止 FIFO 模式时，这表示最后一个帧的发送已完成。
   – 使能 FIFO 模式时，这表示 TXFIFO 和移位寄存器均为空。

   当 USART 被禁止或进入暂停模式时，需要执行此检查来避免损坏最后一次发送。

单字节通信

- 禁止 FIFO 模式时

  对发送数据寄存器进行写操作始终会清零 TXE 位。TXE 标志由硬件置 1。它表示：
  – 数据已从 USART_TDR 寄存器移到移位寄存器中且数据发送已开始；
  – USART_TDR 寄存器为空；
  – USART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。

  TXEIE 位置 1 时该标志位会生成中断。

  发送时，写 USART_TDR 寄存器，会将数据存储在 TDR 缓冲区。该数据随后会在当前发送结束时复制到移位寄存器中。

  未发送时，写 USART_TDR 寄存器，会将数据写入移位寄存器，数据发送开始时，TXE 位置 1。
- 使能 FIFO 模式时，TXFNF（TXFIFO 未满）标志由硬件置 1，以指示：
  – TXFIFO 未满；
  – USART_TDR 寄存器为空；
  – USART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。发送时，写 USART_TDR 寄存器，会将数据存储在 TXFIFO。该数据随后会在当前发送结束时从 TXFIFO 复制到移位寄存器中。

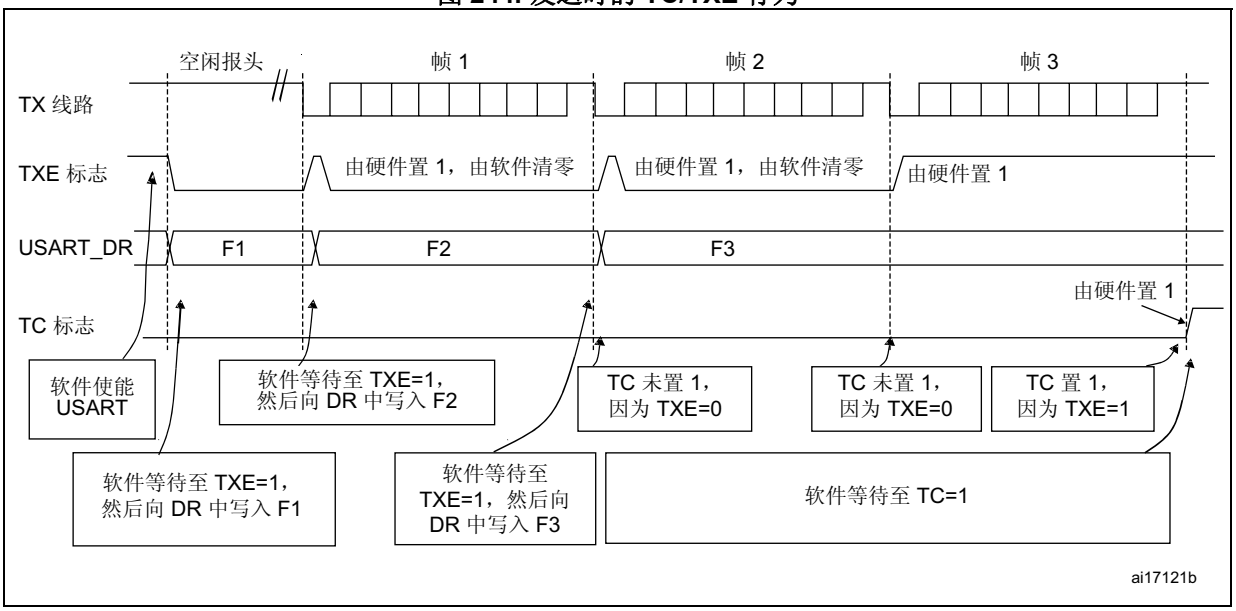
  TXFIFO 未满时，即使在对 USART_TDR 寄存器执行完写操作后，TXFNF 标志也保持为“1”。该标志在 TXFIFO 已满时清零。TXFNFIE 位置 1 时该标志位会生成中断。

  或者，当达到 TXFIFO 阈值时，会生成中断并将数据写入 FIFO。在这种情况下，CPU 可写入由编程的触发阈值定义的数据块。

  如果帧已发送（停止位后）且 TXE 标志（FIFO 模式下的 TXFE）置 1，TC 标志将变为高电平。如果 USART_CR1 寄存器中的 TCIE 位置 1，将生成中断。

向 USART_TDR 寄存器中写入最后一个数据后，必须等待至 TC 置 1，之后才可禁止 USART 或使器件进入低功耗模式（请参见*图 244：发送时的 TC/TXE 行为*）。

图 244. 发送时的 TC/TXE 行为



注意： *使能 FIFO 管理时，TXFNF 标志将用于数据发送。*

中断字符

将 SBKRQ 位置 1 将发送一个中断字符。中断帧的长度取决于 M 位（请参见*图 242*）。

如果将“1”写入 SBKRQ 位，则当前字符发送完成后，将在 TX 线路上发送一个中断字符。通过写操作将 SBKF 位置 1 并在中断字符发送完成时（发送中断字符后的停止位期间），该位由硬件复位。USART 在中断帧末尾的两位持续时间内插入一个逻辑“1”信号 (STOP)，以确保识别下个帧的起始位。

如果 SBKRQ 位置 1，则在当前发送结束时，会发送一个中断字符。

使能 FIFO 模式时，即使 TXFIFO 已满，发送中断字符的优先级也仍高于发送数据的优先级。

空闲字符

将 TE 位置 1 会驱动 USART 在第一个数据帧之前发送一个空闲帧。

24.5.6 USART 接收器

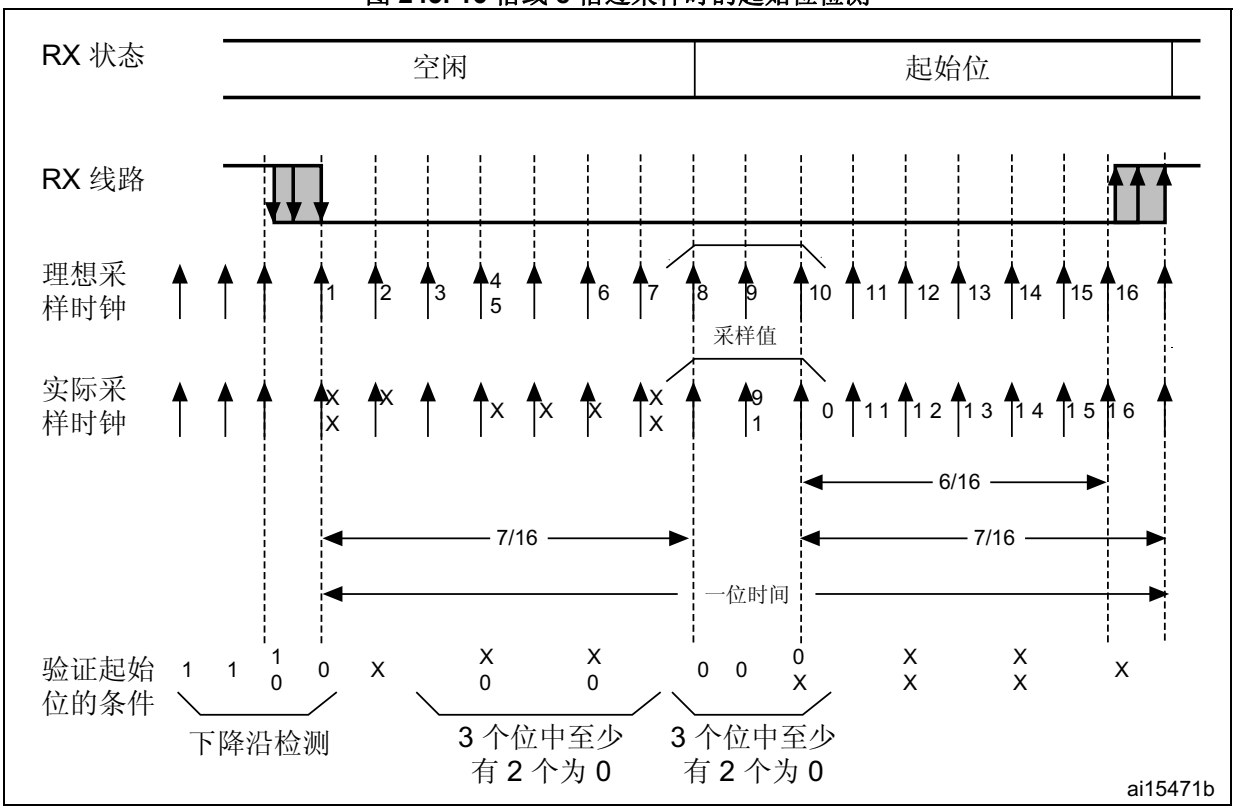
USART 可接收 7 位、8 位或 9 位的数据字，具体取决于 USART_CR1 寄存器中的 M 位。

起始位检测

16 倍或 8 倍过采样时，起始位检测序列相同。

在 USART 中，识别出特定序列的采样时会检测到起始位。该序列为：1 1 1 0 X 0 X 0X 0X 0 X 0X 0。

图 245. 16 倍或 8 倍过采样时的起始位检测



注意： *如果序列不完整，起始位检测将中止，接收器将返回空闲状态（无标志位置 1）等待下降沿。*

如果 3 个采样位均为“0”（针对第 3 位、第 5 位和第 7 位进行首次采样时检测到这 3 位均为“0”；针对第 8 位、第 9 位和第 10 位进行第二次采样时仍检测到这 3 位均为“0”），可确认起始位（RXNE 标志置 1 且 RXNEIE = 1 时生成中断；如果使能 FIFO 模式，则 RXFNE 标志置 1 且 RXFNEIE = 1 时生成中断）。

满足以下条件时，可验证起始位但 NE 噪声标志置 1：

a) 对于两次采样，3 个采样位中有 2 位为“0”（针对第 3 位、第 5 位和第 7 位进行采样并且针对第 8 位、第 9 位和第 10 位采样）

或

b) 对于其中一次采样（针对第 3 位、第 5 位和第 7 位进行采样或针对第 8 位、第 9 位和第 10 位进行采样），3 个采样位中有 2 个为“0”。

如果上述条件均不满足，则启动检测中止，接收器返回空闲状态（无标志置 1）。

字符接收

USART 接收期间，首先通过 RX 引脚移出数据的最低有效位（默认配置）。

字符接收步骤

要接收字符，需遵循以下步骤：

1. 对 USART_CR1 中的 M 位进行编程以定义字长。
2. 使用波特率寄存器 USART_BRR 选择所需波特率
3. 对 USART_CR2 中的停止位数量进行编程。
4. 通过向 USART_CR1 寄存器中的 UE 位写入“1”使能 USART。
5. 如果将进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAR)。按照*第 24.5.19 节：使用 USART 和 DMA 进行连续通信*中的说明配置 DMA 寄存器。
6. 将 RE 位 USART_CR1 置 1。这一操作将使能接收器开始搜索起始位。

接收到字符时：

- 如果已禁止 FIFO 模式，则 RXNE 位置 1，这表明移位寄存器的内容已传送到 RDR。也就是说，已接收到并可读取数据（以及其相应的错误标志）。
- 如果已使能 FIFO 模式，则 RXFNE 位置 1，这表示 RXFIFO 非空。读取 USART_RDR 会返回输入到 RXFIFO 中的最早数据。接收到数据时，数据以及相应的错误位将一起被存储在 RXFIFO 中。
- 如果 RXNEIE（使能 FIFO 模式时为 RXFNEIE）位置 1，则会生成中断。
- 如果接收期间已检测到帧错误、噪声错误、奇偶校验错误或上溢错误，错误标志会置 1。
- 在多缓冲区通信模式下：
  - 如果禁止 FIFO 模式，则 RXNE 标志会在每次接收到字节后置 1。该标志在 DMA 读取接收数据寄存器时被清零。
  - 如果使能 FIFO 模式，则 RXFNE 标志在 RXFIFO 非空时置 1。每次收到 DMA 请求后，都会从 RXFIFO 取回数据。当 RXFIFO 非空时（即，当存在要从 RXFIFO 中读取的数据时），会触发 DMA 请求。
- 在单缓冲区模式下：
  - 如果禁止 FIFO 模式，则通过软件对 USART_RDR 寄存器进行读操作来将 RXNE 标志清零。也可以通过将 USART_RQR 寄存器中的 RXFRQ 位编程为“1”来清零 RXNE 标志。RXNE 标志必须在结束接收下一个字符前清零，以避免发生上溢错误。
  - 如果使能 FIFO 模式，则 RXFNE 在 RXFIFO 非空时置 1。每次对 USART_RDR 执行完读操作后，都会从 RXFIFO 中取回数据。RXFIFO 为空时，RXFNE 标志将清零。也可以通过将 USART_RQR 寄存器中的 RXFRQ 位编程为“1”来清零 RXFNE 标志。RXFIFO 已满时，必须在结束接收下一个字符前读取 RXFIFO 中的第一个条目，以避免发生上溢错误。当 RXFNEIE 位置 1 时，RXFNE 标志会生成中断。或者，当达到 RXFIFO 阈值时，会生成中断并从 RXFIFO 中读取数据。在这种情况下，CPU 可读取由编程的阈值定义的数据块。

中断字符

接收到中断字符时，USART 将会按照帧错误对其进行处理。

空闲字符

检测到空闲帧时，除了在 IDLEIE 位置 1 时会生成中断外，处理步骤与接收到数据字符的情况相同。

上溢错误

- 禁止 FIFO 模式

  如果在 RXNE 未复位时接收到字符，则会发生上溢错误。

  RXNE 位清零前，数据无法从移位寄存器传送到 RDR 寄存器。每接收到一个字节后，RXNE 标志都将置 1。

  当 RXNE 标志位是 1 时，如果在接收到下一个数据或尚未处理上一个 DMA 请求，则会发生上溢错误。发生上溢错误时：

  – ORE 位置 1。
  – RDR 中的内容不会丢失。可通过读取 USART_RDR 寄存器获得之前的数据。
  – 移位寄存器会被覆盖。之后，上溢期间接收到的任何数据都将丢失。
  – 如果 RXNEIE 或 EIE 位置 1，则会生成中断。

- 使能 FIFO 模式

  移位寄存器准备好传送并且接收 FIFO 已满时，会发生上溢错误。

  在 RXFIFO 中出现一个空闲位置之前，数据无法从移位寄存器传送到 USART_RDR 寄存器。当 RXFIFO 非空时，RXFNE 标志置 1。

  如果 RXFIFO 已满且移位寄存器已准备好传送，则会发生上溢错误。发生上溢错误时：

  – ORE 位置 1。
  – RXFIFO 中的第一个条目不会丢失。通过读取 USART_RDR 寄存器可获得此条目。
  – 移位寄存器会被覆盖。之后，上溢期间接收到的任何数据都将丢失。
  – 如果 RXFNEIE 或 EIE 位置 1，则会生成中断。

通过将 USART_ICR 寄存器中的 ORECF 位置 1 来复位 ORE 位。

注意： *ORE 位置 1 时表示至少 1 个数据丢失。*

禁止 FIFO 模式时，有以下两种可能

- *如果 RXNE = 1，则最后一个有效数据存储于接收寄存器 (RDR) 中并且可进行读取，*
- *如果 RXNE = 0，则最后一个有效数据已被读取，因此 RDR 寄存器中没有要读取的数据。接收到新（丢失）数据的同时已读取 RDR 寄存器中的最后一个有效数据时，会发生该情况。*

选择时钟源和合适的过采样方法

时钟源的选择通过时钟控制系统完成（请参见 *复位和时钟控制 (RCC)* 部分）。在使能 USART 之前，必须通过 UE 位选择时钟源。

必须遵循以下两个条件选择时钟源：

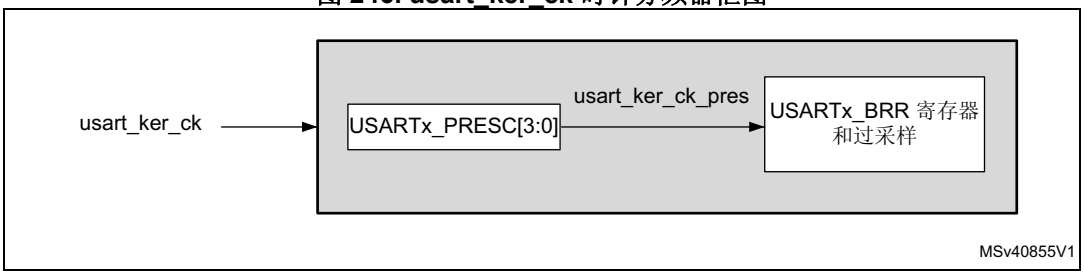
- 可在低功耗模式下使用 USART
- 通信速度。

时钟源频率为 usart_ker_ck。

如果支持双时钟域和从低功耗模式唤醒功能，则 usart_ker_ck 时钟源可在 RCC 中进行配置（请参见 *复位和时钟控制 (RCC)* 部分）。否则，usart_ker_ck 时钟与 usart_pclk 相同。

usart_ker_ck 时钟可根据可编程系数（在 USART_PRESC 寄存器中定义）进行分频。

图 246. usart_ker_ck 时钟分频器框图



某些 usart_ker_ck 时钟源允许 USART 在 MCU 处于低功耗模式时接收数据。需要时，USART 可基于所接收的数据和选择的唤醒模式来唤醒 MCU，以通过用软件读取 USART_RDR 寄存器的方式或通过 DMA 的方式传输已接收数据。

对于其他时钟源，系统必须激活才能使能 USART 通信。

时钟源还决定通信速度范围（尤其是最大通信速度）。

接收器采用不同的用户可配置过采样技术（除了同步模式下），可以从噪声中提取有效数据。这可在最大通信速度与抗噪声/时钟误差性能之间实现最佳平衡。

可通过编程 USART_CR1 寄存器中的 OVER8 位来选择采样方法，且采样时钟可以是波特率时钟的 16 倍或 8 倍（请参见*图 247* 和*图 248*）。

根据应用：

- 选择 8 倍过采样 (OVER8 = 1) 以获得更高的速度（高达 usart_ker_ck_pres/8）。这种情况下接收器对时钟偏差的最大容差将会降低（请参见*第 624 页的第 24.5.8 节：USART 接收器对时钟偏差的容差*）
- 选择 16 倍过采样 (OVER8 = 0) 以增加接收器对时钟偏差的容差。在这种情况下，最大速度被限制为最大 usart_ker_ck_pres/16（其中，usart_ker_ck_pres 为 USART 输入时钟通过预分频器进行分频得到的值）。

可通过编程 USART_CR3 寄存器中的 ONEBIT 位选择用于评估逻辑电平的方法。有两种选择可供使用：

- 在已接收位的中心进行三次采样，从而进行多数表决。这种情况下，如果用于多数表决的 3 次采样结果不相等，NE 位置 1。
- 在已接收位的中心进行单次采样

  根据应用：

  – 在噪声环境下工作时，请选择三次采样的多数表决法 (ONEBIT = 0)；在检测到噪声时请拒绝数据（请参见*图 107*），因为这表示采样过程中产生了干扰。

  – 线路无噪声时请选择单次采样法 (ONEBIT = 1) 以增加接收器对时钟偏差的容差（请参见*第 624 页的第 24.5.8 节：USART 接收器对时钟偏差的容差*）。这种情况下 NE 位始终不会置 1。

帧中检测到噪声时：

- 在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）的上升沿时 NE 位置 1。
- 无效数据从移位寄存器传送到 USART_RDR 寄存器。
- 单字节通信时无中断产生。然而，在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）生成中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时会发出中断。

通过将 USART_ICR 寄存器中的 NECF 位置 1 复位 NE 位。

注意： *SPI 模式不支持噪声错误。*

智能卡、IrDA 和 LIN 模式下不可采用 8 倍过采样。在这些模式下，OVER8 位由硬件强制清零。

图 247. 16 倍过采样时的数据采样

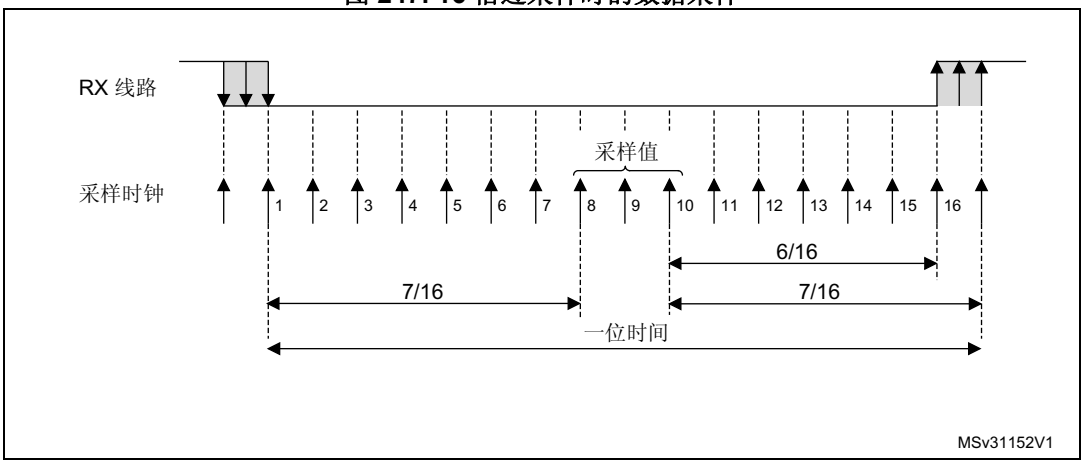


图 248. 8 倍过采样时的数据采样

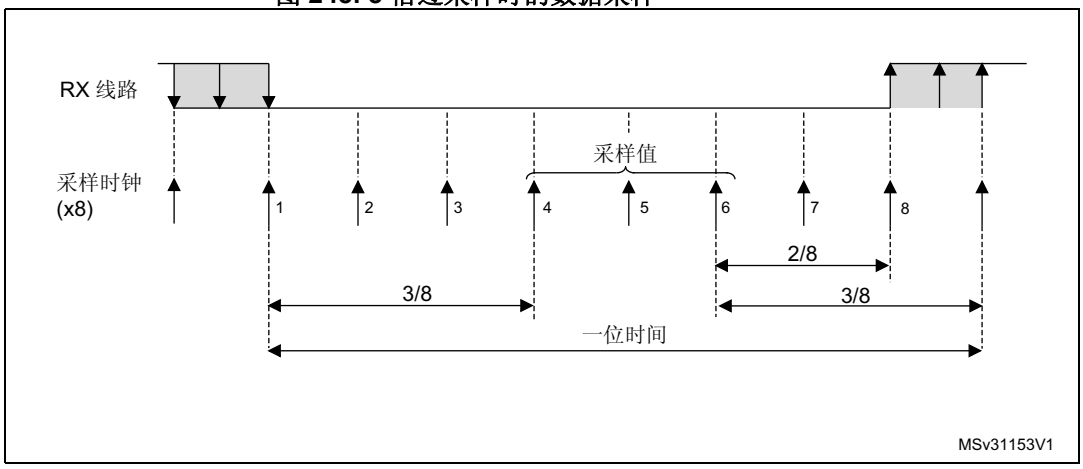


表 107. 通过采样数据进行噪声检测

采样值	NE 状态	接收的位值
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

帧错误

如果接收数据时未在预期时间内识别出停止位，从而出现同步失效或过度的噪声，则会检测到帧错误。

检测到帧错误时：

- FE 位由硬件置 1
- 无效数据从移位寄存器传送到 USART_RDR 寄存器（使能 FIFO 模式时为 RXFIFO）。
- 单字节通信时无中断产生。然而，在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）生成中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时会发出中断。

通过将“1”写入 USART_ICR 寄存器中的 FECF 位来复位 FE 位。

注意： *SPI 模式不支持帧错误。*

接收期间可配置的停止位

可通过 USART_CR 的控制位配置要接收的停止位的数量：可以是 1 或 2 个（正常模式下），也可以是 0.5 或 1.5 个（智能卡模式下）。

- **0.5 个停止位（在智能卡模式下接收时）**：不会对 0.5 个停止位进行采样。结果，选择 0.5 个停止位时，无法检测到帧错误和中断帧。
- **1 个停止位**：将在第 8、第 9 和第 10 次采样时对 1 个停止位进行采样。
- 1.5 个停止位（在智能卡模式下）

  在智能卡模式下发送时，设备必须检查数据是否正确发送。因此，必须使能接收器块（USART_CR1 中的 RE = 1）并检查停止位，以测试智能卡是否已检测到奇偶校验错误。

  发生奇偶校验错误时，智能卡会在采样时将数据信号强制为低电平（即 NACK 信号），该信号被标记为帧错误。之后，FE 标志在 1.5 个停止位的末尾由 RXNE 标志（使能 FIFO 模式时为 RXFNE）置 1。在第 16、第 17 和第 18 次采样时对 1.5 个停止位进行采样（停止位采样开始后维持 1 个波特时钟周期）。1.5 个停止位可分为 2 个部分：0.5 个波特时钟周期（未发生任何动作），然后是 1 个正常的停止位周期（一半时间处进行采样）（更多详细信息，请参见*第 636 页的第 24.5.16 节：USART 接收器超时*）。
- 2 个停止位

  采样 2 个停止位时在第 8、第 9 和第 10 次采样时对第一个停止位进行采样。

  如果在第一个停止位期间检测到帧错误，则帧错误标志会置 1。

  发生帧错误时不检测第 2 个停止位。RXNE 标志（使能 FIFO 模式时为 RXFNE）将在第一个停止位结束时置 1。

24.5.7 USART 波特率生成

接收器和发送器（Rx 和 Tx）的波特率均设置为 USART_BRR 寄存器中编程的值。

公式 1：适用于标准 USART（包括 SPI 模式）的波特率（OVER8 = “0”或“1”）

在 16 倍过采样的情况下，波特率通过以下公式得出：

$$\text{Tx/Rx 波特率} = \frac{\text{usart\_ker\_ckpres}}{\text{USARTDIV}}$$

在 8 倍过采样的情况下，波特率通过以下公式得出：

$$\text{Tx/Rx 波特率} = \frac{2 \times \text{usart\_ker\_ckpres}}{\text{USARTDIV}}$$

公式 2：智能卡、LIN 和 IrDA 模式下的波特率 (OVER8 = 0)

波特率可根据以下公式得出：

$$\text{Tx/Rx 波特率} = \frac{\text{usart\_ker\_ckpres}}{\text{USARTDIV}}$$

USARTDIV 是一个存放在 USART_BRR 寄存器中的无符号定点数。

- 当 OVER8 = 0 时，BRR = USARTDIV。
- 当 OVER8 = 1 时
  - BRR[2:0] = USARTDIV[3:0] 右移 1 位。
  - BRR[3] 必须保持清零。
  - BRR[15:4] = USARTDIV[15:4]

注意： *对 USART_BRR 执行写操作后，波特率计数器更新为波特率寄存器中的新值。因此，波特率寄存器的值不应在通信时发生更改。*

16 倍和 8 倍过采样时，USARTDIV 必须大于或等于 16。

如何从 USART_BRR 寄存器中获取 USARTDIV

示例 1

要通过 usart_ker_ck_pres = 8 MHz 获得 9600 波特：

- 16 倍过采样时：
  USARTDIV = 8 000 000/9600
  BRR = USARTDIV = 0d833 = 0x0341
- 8 倍过采样时：
  USARTDIV = 2 * 8 000 000/9600
  USARTDIV = 1666,66 (0d1667 = 0x683)
  BRR[3:0] = 0x3 >> 1 = 0x1
  BRR = 0x681

示例 2

要通过 usart_ker_ck_pres = 48 MHz 获得 921.6 K 波特：

- 16 倍过采样时：
  USARTDIV = 48 000 000/921 600
  BRR = USARTDIV = 0d52 = 0x34
- 8 倍过采样时：
  USARTDIV = 2 * 48 000 000/921 600
  USARTDIV = 104 (0d104 = 0x68)
  BRR[3:0] = USARTDIV[3:0] >> 1 = 0x8 >> 1 = 0x4
  BRR = 0x64

24.5.8 USART 接收器对时钟偏差的容差

仅当总时钟系统偏差小于 USART 接收器的容差时，USART 异步接收器才能正常工作。

影响总偏差的因素包括：

- DTRA：发送器误差引起的偏差（其中还包括发送器本地振荡器的偏差）
- DQUANT：接收器的波特率量化引起的误差
- DREC：接收器本地振荡器的偏差
- DTCL：传输线路引起的偏差（通常是由于收发器所引起，它可能会在低电平到高电平转换时序与高电平到低电平转换时序之间引入不对称）

$$\text{DTRA} + \text{DQUANT} + \text{DREC} + \text{DTCL} + \text{DWU} < \text{USART 接收器 容差}$$

其中

DWU 是使用从低功耗模式唤醒时因采样点偏差而产生的误差。

M[1:0] = 01 时：

$$\text{DWU} = \frac{t_{WUUSART}}{11 \times \text{Tbit}}$$

M[1:0] = 00 时：

$$\text{DWU} = \frac{t_{WUUSART}}{10 \times \text{Tbit}}$$

M[1:0] = 10 时：

$$\text{DWU} = \frac{t_{WUUSART}}{9 \times \text{Tbit}}$$

$t_{WUUSART}$ 是检测到起始位下降沿与时钟（由外设请求）就绪、达到外设且调压器就绪之间的时间。

USART 接收器在*表 108* 和*表 109* 中指定的最大容许偏差下可正确接收数据，具体取决于以下设置：

- 由 USART_CR1 寄存器中的 M 位定义的 9 位、10 位或 11 位字符长度
- 由 USART_CR1 寄存器中的 OVER8 位定义的 8 倍或 16 倍过采样
- USART_BRR 寄存器的 BRR[3:0] 位等于或不等于 0000。
- 使用 1 位或 3 位对数据进行采样，取决于 USART_CR3 寄存器中 ONEBIT 位的值

表 108. BRR [3:0] = 0000 时的 USART 接收器容差

M 位	OVER8 位 = 0		OVER8 位 = 1	
	ONEBIT = 0	ONEBIT = 1	ONEBIT = 0	ONEBIT = 1
00	3.75%	4.375%	2.50%	3.75%
01	3.41%	3.97%	2.27%	3.41%
10	4.16%	4.86%	2.77%	4.16%

表 109. BRR[3:0] 不等于 0000 时的 USART 接收器容差

M 位	OVER8 位 = 0		OVER8 位 = 1	
	ONEBIT = 0	ONEBIT = 1	ONEBIT = 0	ONEBIT = 1
00	3.33%	3.88%	2%	3%
01	3.03%	3.53%	1.82%	2.73%
10	3.7%	4.31%	2.22%	3.33%

注意： *当接收的帧恰好包含 10 个（M 位 = 00）、11 个（M 位 = 01）或 9 个（M 位 = 10）位时间的空闲帧时，表 108 和表 109 中指定的数据可能与特例中的数据略微不同。*

24.5.9 USART 自动波特率检测

USART 可根据接收一个字符检测并自动设置 USART_BRR 寄存器的值。自动波特率检测在以下两种情况下非常有用：

- 事先不知道系统的通信速度。
- 系统正在使用精确度相对较低的时钟源且该机制允许在不测量时钟偏差的情况下获得正确的波特率。

时钟源频率必须与预期通信速度兼容。

- 16 倍过采样时，波特率范围为 usart_ker_ck_pres/65535 到 usart_ker_ck_pres/16。
- 8 倍过采样时，波特率范围为 usart_ker_ck_pres/65535 到 usart_ker_ck_pres/8。

在激活自动波特率检测之前，必须通过 USART_CR2 寄存器中的 ABRMOD[1:0] 位域选择自动波特率检测模式。根据不同的字符模式，存在四种检测模式。在这些自动波特率模式下，波特率在同步接收数据期间被多次测量，每次测量的结果都与前一次进行比较。

这些模式如下：

- **模式 0**：以“1”位开头的任意字符。
  这种情况下，USART 会测量起始位的持续时间（下降沿到上升沿）。
- **模式 1**：以 10xx 位模式开头的任何字符。
  这种情况下，USART 会测量起始位和第一个数据位的持续时间。测量在下降沿到下降沿期间完成，可在信号斜率较小时确保较高的精度。
- **模式 2**：0x7F 字符帧（可以是 LSB 在前模式下的 0x7F 字符，也可以是 MSB 在前模式下的 0xFE 字符）。
  这种情况下，先在起始位结束时更新波特率 (BR)，然后在位 6 结束时更新波特率（根据从下降沿到下降沿执行的测量：BR6）。以 BR 对位 0 到位 6 进行采样，而以 BR6 对字符的其他位进行采样。

- **模式 3**：0x55字符帧。

  这种情况下，先在起始位结束时更新波特率 (BR)，然后在位 0 结束时更新波特率（根据从下降沿到下降沿执行的测量：BR0），最后在位 6 结束时更新波特率 (BR6)。以 BR 对位 0 进行采样，以 BR0 对位 1 到位 6 进行采样，以 BR6 对字符的其他位进行采样。同时，对 RX 线路的各个中间转换执行其他检查。如果 RX 上的转换与接收器（基于根据位 0 计算的波特率的接收器）未充分同步，则生成错误。

激活自动波特率检测之前，必须先通过向 USART_BRR 寄存器写入非零的波特率值来初始化该寄存器。

通过将 USART_CR2 寄存器中的 ABREN 位置 1 来激活自动波特率检测。之后 USART 会等待 RX 线路上的第一个字符。通过将 USART_ISR 寄存器中的 ABRF 标志置 1 来指示自动波特率操作完成。如果线路干扰较大，则无法保证正确的波特率检测。这种情况下，BRR 值可能会损坏，ABRE 错误标志位会置 1。如果通信速度不在自动波特率检测范围（位持续时间不在 16 个和 65536 个时钟周期（16 倍过采样时）之间，也不在 8 个和 65536 个时钟周期（8 倍过采样时）之间）内，也会出现这种情况。

稍后，可通过复位 ABRF 标志（通过写入“0”）重新启动自动波特率检测。

禁止 FIFO 管理且发生自动波特率错误时，ABRE 标志通过 RXNE 和 FE 位置 1。

使能 FIFO 管理且发生自动波特率错误时，ABRE 标志通过 RXFNE 和 FE 位置 1。

如果使能 FIFO 模式，则应使用第一个 RXFIFO 位置上的数据进行自动波特率检测。因此，在启动自动波特率检测之前，请通过检查 USART_ISR 寄存器的 RXFNE 标志来确保 RXFIFO 为空。

注意： *如果在自动波特率操作期间禁止 USART (UE = 0)，则可能损坏 BRR 值。*

24.5.10 USART 多处理器通信

可以执行 USART 多处理器通信（多个 USART 连接在一个网络中）。例如，其中一个 USART 可以是主 USART，其 TX 输出与其他 USART 的 RX 输入相连；而其他 USART 为从 USART，其各自的 TX 输出在逻辑上通过与运算连在一起，并与主 USART 的 RX 输入相连。

在多处理器配置中，理想情况下通常只有预期的消息接收方主动接收完整的消息内容，从而减少由所有未被寻址的接收器造成的冗余 USART 服务开销。

可通过静默功能将未被寻址的器件置于静默模式下。为了使用静默模式功能，必须将 USART_CR1 寄存器中的 MME 位置 1。

注意： *使能 FIFO 管理且 MME 已置 1 时，不得清零 MME 位再快速将其置 1（在两个 usart_ker_ck 周期内），否则静默模式可能保持有效。*

使能静默模式时：

- 不得将接收状态位置 1。
- 禁止任何接收中断。
- USART_ISR 寄存器中的 RWU 位置“1”。在某些情况下，RWU 可以由硬件或软件通过 USART_RQR 寄存器中的 MMRQ 位自动控制。

根据 USART_CR1 寄存器中 WAKE 位的设置，USART 可使用以下两种方法进入或退出静音模式：

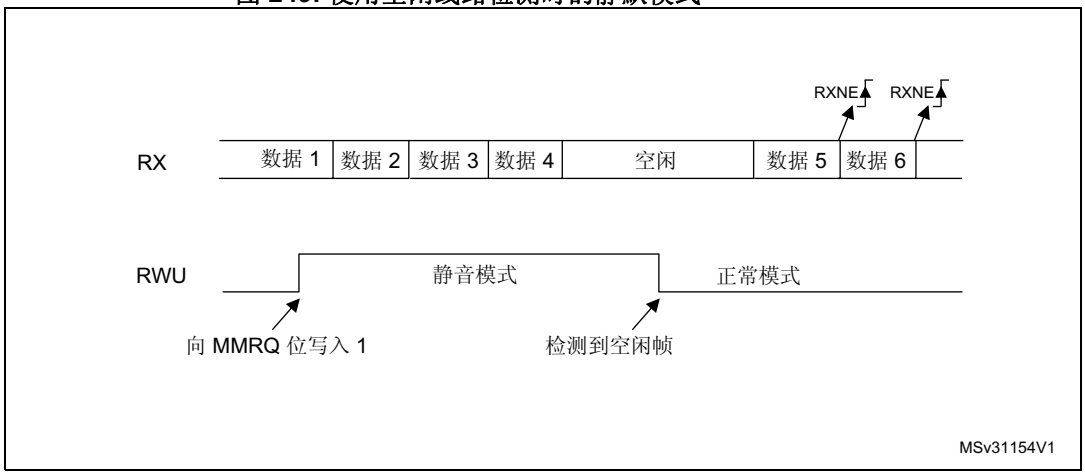
- 如果 WAKE 位被复位，则进行空闲线路检测，
- 如果 WAKE 位置 1，则进行地址标记检测。

空闲线路检测 (WAKE = 0)

向 MMRQ 位写入“1”且 RWU 位自动置 1 时，USART 进入静默模式。

检测到空闲帧时，USART 将唤醒。此时 RWU 位会由硬件清零，但 USART_ISR 寄存器中的 IDLE 位不会置 1。*图 249* 中给出了使用空闲线路检测时静默模式行为的示例。

图 249. 使用空闲线路检测时的静默模式



注意： *如果在 IDLE 字符已经过去时将 MMRQ 位置 1，则不会进入静默模式（RWU 未置 1）。*

如果在线路处于空闲状态时激活 USART，在一个 IDLE 帧持续时间后（不只在接收一个字符帧后）会检测到空闲状态。

4 位/7 位地址标记检测 (WAKE = 1)

在此模式下，如果字节的 MSB 为 1，则将这些字节识别为地址，否则将其识别为数据。在地址字节中，目标接收器的地址位于 4 个或 7 个 LSB 中。7 位或 4 位地址检测通过 ADDM7 位来选择。接收器会将此 4 位/7 位字与其地址进行比较，该接收器的地址在 USART_CR2 寄存器的 ADD 位中进行设置。

注意： *在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。*

当接收到与其编程地址不匹配的地址字符时，USART 会进入静默模式。此时，RWU 位将由硬件置 1。USART 进入静默模式后，RXNE 标志不会针对此地址字节置 1，也不会发出中断或 DMA 请求。使能 FIFO 管理时，软件应确保在进入静默模式之前 RXFIFO 中至少有一个空位置。

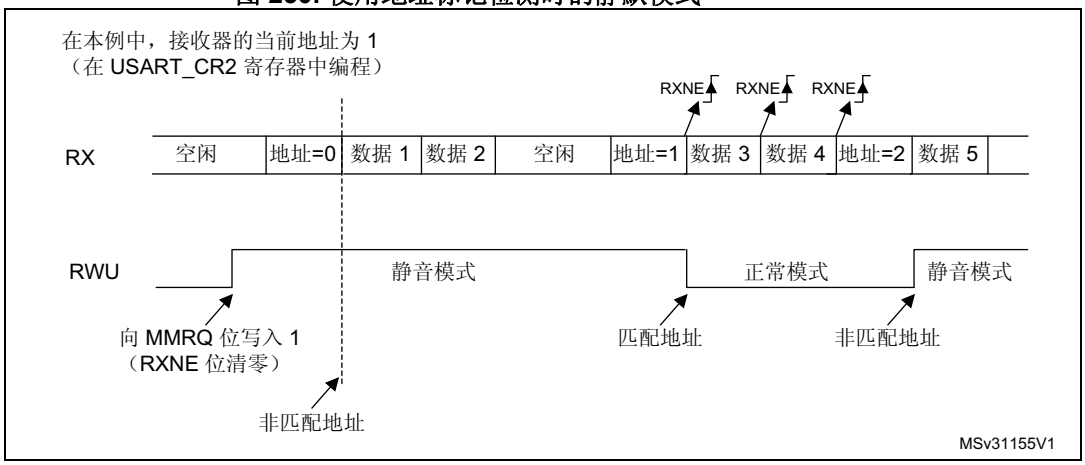
当向 MMRQ 位写入 1 时，USART 也会进入静默模式。这种情况下，RWU 位也自动置 1。

当接收到与编程地址匹配的地址字符时，USART 会退出静默模式。然后 RWU 位被清零，可以开始正常接收后续字节。由于 RWU 位已清零，RXNE/RXFNE 位会针对地址字符置 1。

注意： *使能 FIFO 管理时，如果在接收器对数据的最后一位进行采样时 MMRQ 置 1，则可在有效进入静默模式之前接收该数据*

图 250 中给出了使用地址标记检测时静默模式行为的示例。

图 250. 使用地址标记检测时的静默模式



24.5.11 USART Modbus 通信

USART 为 Modbus/RTU 和 Modbus/ASCII 协议的实现提供基本支持。Modbus/RTU 是一个半双工块传输协议。该协议的控制部分（地址识别、块完整性控制和命令解析）必须用软件实现。

USART 为块结束检测提供基本支持，无需软件开销或其他资源。

Modbus/RTU

在此模式下，一个块的结束通过超过 2 个字符时间的“静默”（空闲线路）来识别。此功能通过可编程的超时功能实现。

超时功能和中断必须分别通过 USART_CR2 寄存器中的 RTOEN 位和 USART_CR1 寄存器中的 RTOIE 位激活。与 2 个字符时间（例如 22 个位时间）的超时相对应的值必须在 RTO 寄存器中编程。如果在此期间接收线路空闲，则在接收到最后一个停止位后，将生成中断，同时通知软件当前块接收已完成。

Modbus/ASCII

在此模式下，块结束通过特定 (CR/LF) 字符序列识别。USART 通过字符匹配功能管理此机制。

通过在 ADD[7:0] 位域中编程 LF ASCII 码以及激活字符匹配中断 (CMIE = 1)，软件可在接收到 LF 时获得通知并检查 DMA 缓存区中的 CR/LF。

24.5.12 USART 极性控制

将 USART_CR1 寄存器中的 PCE 位置 1，可以使能奇偶校验控制（发送时生成奇偶校验位，接收时进行奇偶校验检查）。根据 M 位定义的帧长度，*表 110* 中列出了可能的 USART 帧格式。

表 110. USART 帧格式

M 位	PCE 位	USART 帧(1)
00	0	| SB | 8 位数据 | STB |
00	1	| SB | 7 位数据 | PB | STB |
01	0	| SB | 9 位数据 | STB |
01	1	| SB | 8 位数据 PB | STB |
10	0	| SB | 7 位数据 | STB |
10	1	| SB | 6 位数据 | PB | STB |

1. 图注：SB：起始位，STB：停止位，PB：奇偶校验位。在数据寄存器中，PB 始终位于 MSB 位置（第 8 位或第 7 位，具体取决于 M 位的值）。

偶校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为偶数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据 = 00110101 且 4 个位置 1，则在选择偶校验（USART_CR1 寄存器中的 PS 位 = 0）时，奇偶校验位等于 0。

奇校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为奇数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据 = 00110101 且 4 个位置 1，则在选择奇校验（USART_CR1 寄存器中的 PS 位 = 1）时，奇偶校验位等于 1。

接收时进行奇偶校验检查

如果奇偶校验检查失败，则 USART_ISR 寄存器中的 PE 标志置 1；如果 USART_CR1 寄存器中 PEIE 位置 1，则会生成中断。通过软件将 1 写入 USART_ICR 寄存器中的 PECF 位来清零 PE 标志。

发送时的奇偶校验生成

如果 USART_CR1 中的 PCE 位置 1，则在数据寄存器中所写入数据的 MSB 位会进行传送，但是会由奇偶校验位进行更改（如果选择偶校验（PS = 0），则“1”的数量为偶数；如果选择奇校验 (PS=1)，则“1”的数量为奇数）。

24.5.13 USART LIN（局域互连网络）模式

仅在支持 LIN 模式时才与本节相关。请参见*第 608 页的第 24.4 节：USART 实现*。

通过将 USART_CR2 寄存器中的 LINEN 位置 1 来选择 LIN 模式。在 LIN 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 CLKEN 位，
- USART_CR3 寄存器中的 STOP[1:0]、SCEN、HDSEL 和 IREN 位。

LIN 发送

与正常的 USART 发送相比，在 LIN 主器件中发送时必须采用*第 24.5.4 节*中介绍的步骤，同时还具有以下区别：

- M 位清零以配置 8 位字长度。
- LINEN 位置 1 以进入 LIN 模式。此时，将 SBKRQ 位置 1 会发送 13 个“0”位作为中断字符。然后会发送值为“1”的两个位以进行下一启动检测。

LIN 接收

使能 LIN 模式后，将激活中断检测电路。该检测完全独立于正常的 USART 接收器。在空闲状态或某个帧期间，只要发生中断即可检测出来。

接收器（USART_CR1 寄存器中 RE = 1）使能后，电路便开始监测 RX 输入上的启始信号。检测起始位的方法与搜索中断字符或数据的方法相同。检测到起始位后，电路会对接下来的位进行采样，方法与数据采样相同（第 8、第 9 和第 10 次采样）。如果 10 个（USART_CR2 寄存器中 LBDL = 0 时）或 11 个（USART_CR2 寄存器中 LBDL = 1 时）连续位均检测为“0”，且其后跟随分隔符，则 USART_ISR 寄存器中的 LBDF 标志将会置 1。如果 LBDIE 位 = 1，则会生成中断。在验证中断前，会对分隔符进行检查，因为它表示 RX 线路已恢复到高电平。

如果在第 10 或第 11 次采样前已采样到“1”信号，则中断检测电路会取消当前检测，并重新搜索起始位。

如果禁止 LIN 模式 (LINEN = 0)，接收器会作为正常的 USART 继续工作，不会再进行中断检测。

如果使能 LIN 模式 (LINEN = 1)，只要发生帧错误（例如，在“0”处检测到停止位，这种情况可能出现在任何中断帧中），接收器即会停止，直到中断检测电路接收到“1”（中断字不完整时）或接收到分隔符（检测到中断时）为止。

*第 631 页的图 251：LIN 模式下的中断检测（11 位中断长度——LBDL 位置 1）*中显示了中断检测器状态机和中断标志的行为。

*第 632 页的图 252：LIN 模式下的中断检测与帧错误检测*中列出了中断帧的示例。

图 251. LIN 模式下的中断检测（11 位中断长度——LBDL 位置 1）

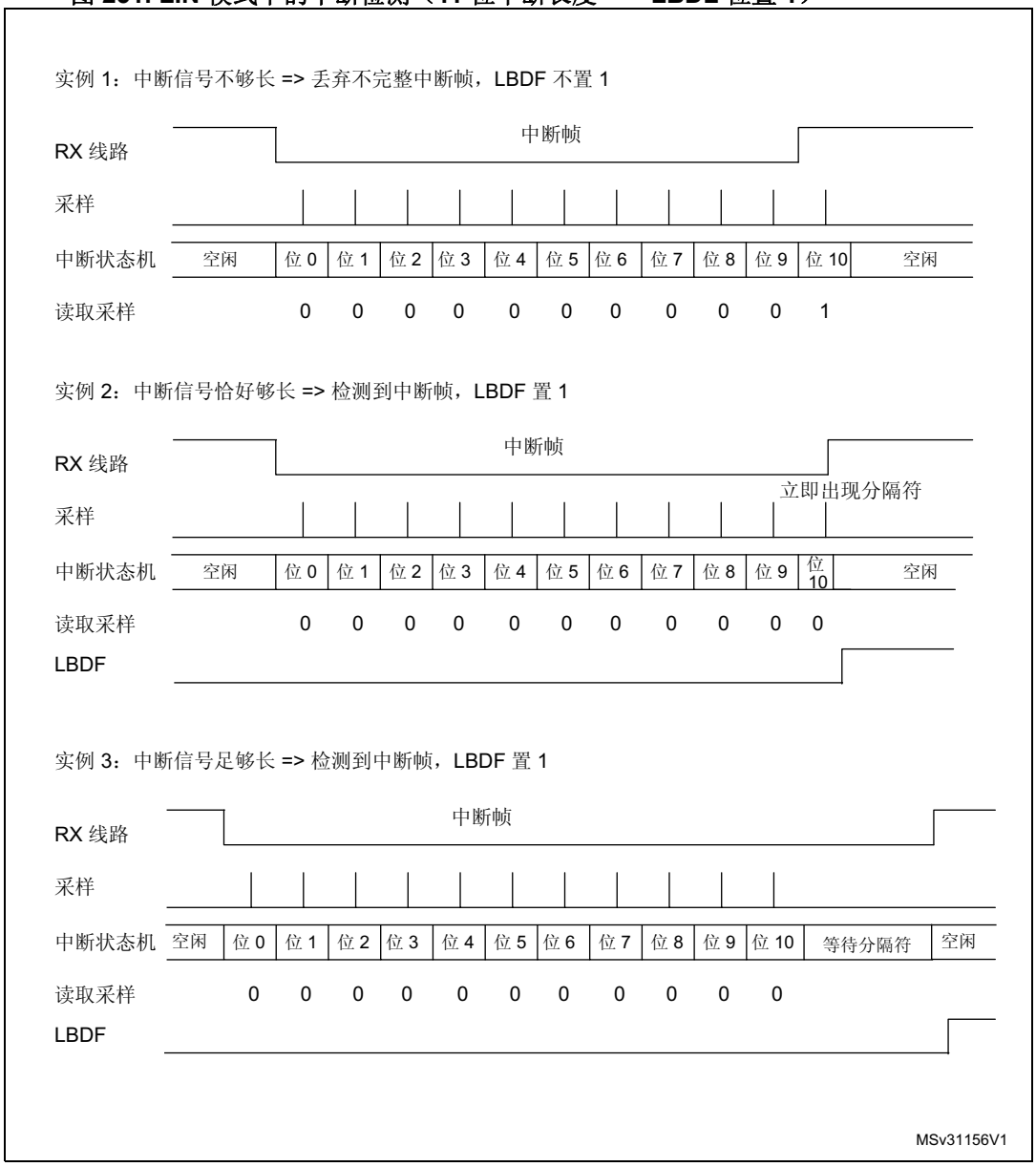
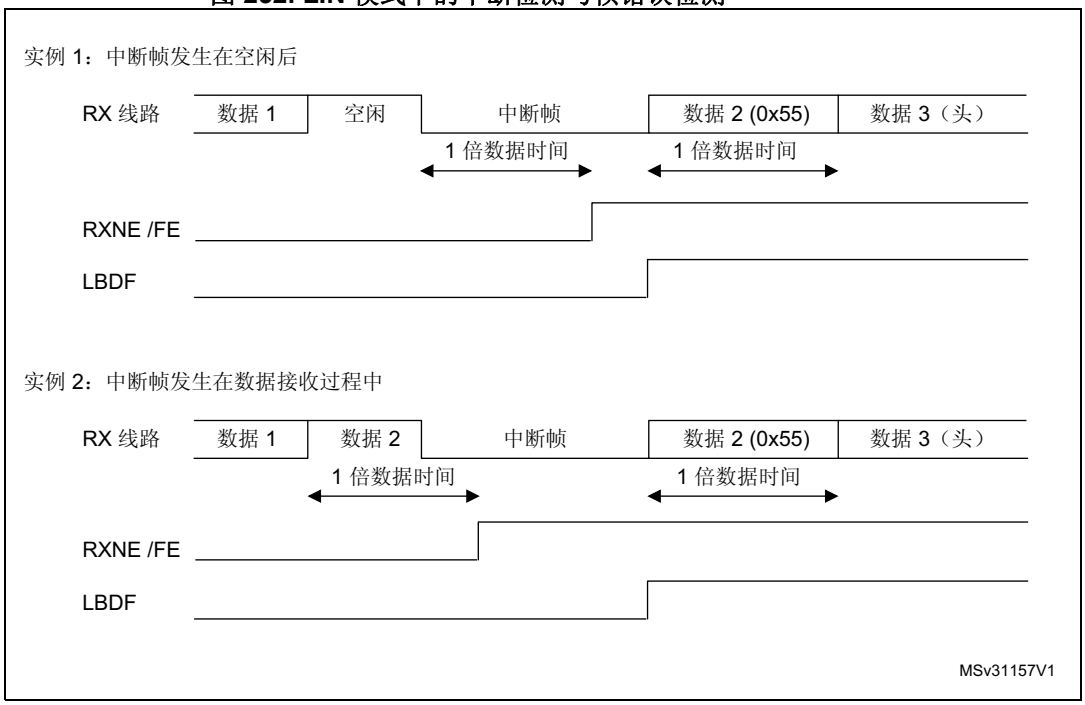


图 252. LIN 模式下的中断检测与帧错误检测



24.5.14 USART 同步模式

主模式

通过将 USART_CR2 寄存器中的 CLKEN 位编程为“1”来选择同步主模式。在同步模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 位，
- USART_CR3 寄存器中的 SCEN、HDSEL 和 IREN 位。

在此模式下，USART 可用于在主模式下控制双向同步串行通信。CK 引脚是 USART 发送器时钟的输出。在起始位或停止位期间，不会向 CK 引脚发送时钟脉冲。在最后一个有效数据位（地址标记）期间，是否生成时钟脉冲取决于 USART_CR2 寄存器中 LBCL 位的状态。USART_CR2 寄存器中的 CPOL 位用于选择时钟极性；USART_CR2 寄存器中的 CPHA 位用于选择外部时钟的相位（请参见*图 253*、*图 254* 和*图 255*）。

在空闲状态、报头模式和发送中断期间，外部 CK 时钟处于未激活状态。

在同步主模式下，USART 发送器的工作方式与异步模式下完全相同。但是，由于 CK 与 TX 同步（根据 CPOL 和 CPHA），因此 TX 上的数据是同步的。

在同步主模式下，USART 接收器的工作方式与异步模式下不同。如果 RE 置 1，则数据在 CK 上采样（上升或下降沿，具体取决于 CPOL 和 CPHA），而不会进行任何过采样。此时必须确保给定建立时间和保持时间（取决于波特率：1/16 位时间）符合要求。

注意： *在主模式下，CK 引脚可与 TX 引脚结合使用。因此，仅当使能发送器 (TE = 1) 且正在发送数据时（USART_TDR 数据寄存器已写入），才会提供时钟。这意味着，没有发送数据的情况下无法接收同步数据。*

图 253. USART 同步主发送示例

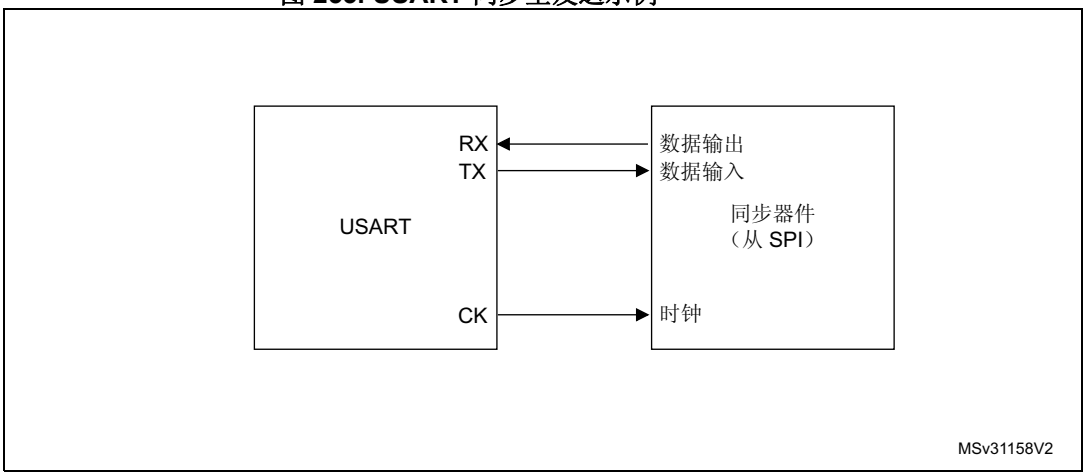


图 254. USART 在同步主模式下的数据时钟时序图（M 位 = 00）

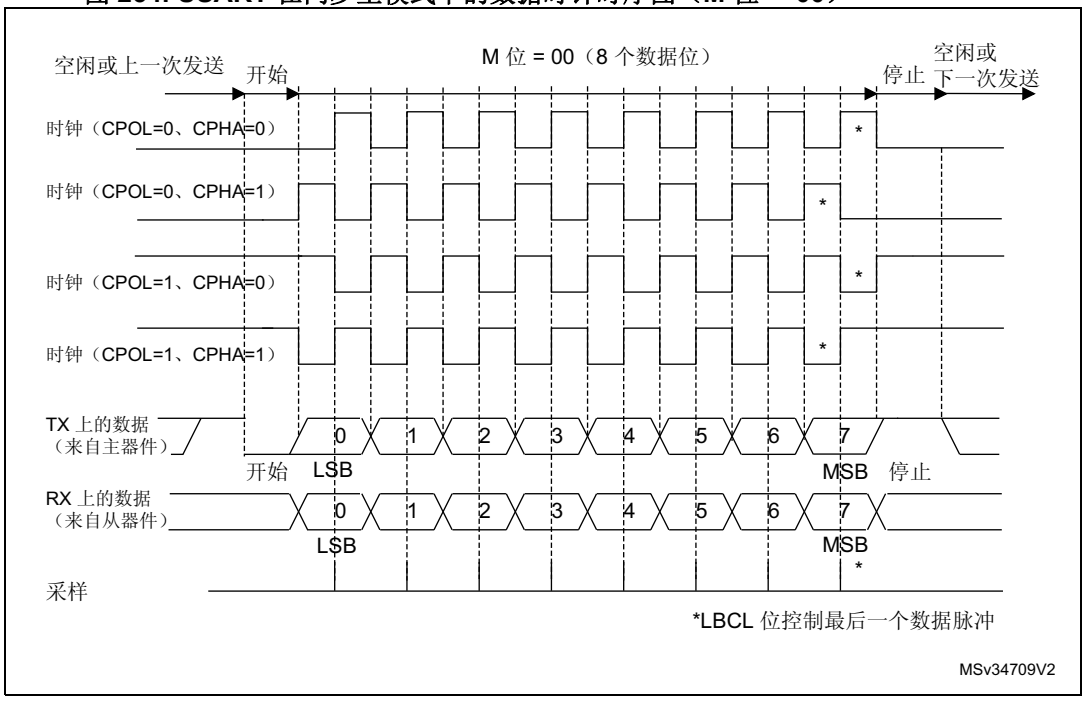
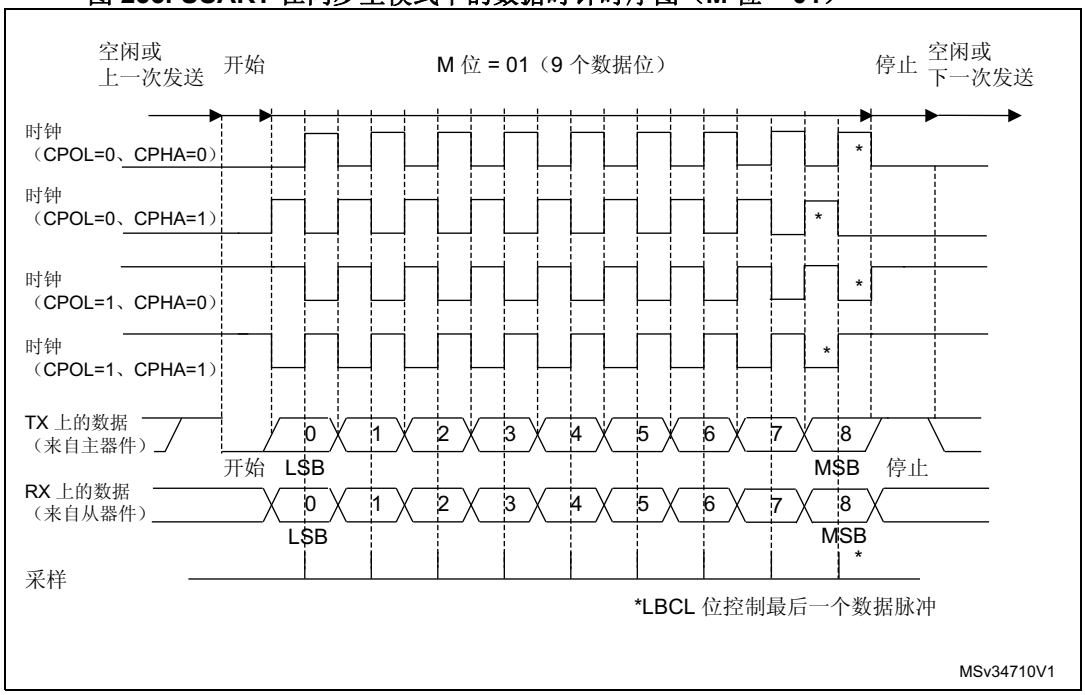


图 255. USART 在同步主模式下的数据时钟时序图（M 位 = 01）



从模式

通过将 USART_CR2 寄存器中的 SLVEN 位编程为“1”来选择同步从模式。在同步从模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 和 CLKEN 位，
- USART_CR3 寄存器中的 SCEN、HDSEL 和 IREN 位。

在此模式下，USART 可用于在从模式下控制双向同步串行通信。在从模式下，CK 引脚是 USART 的输入。

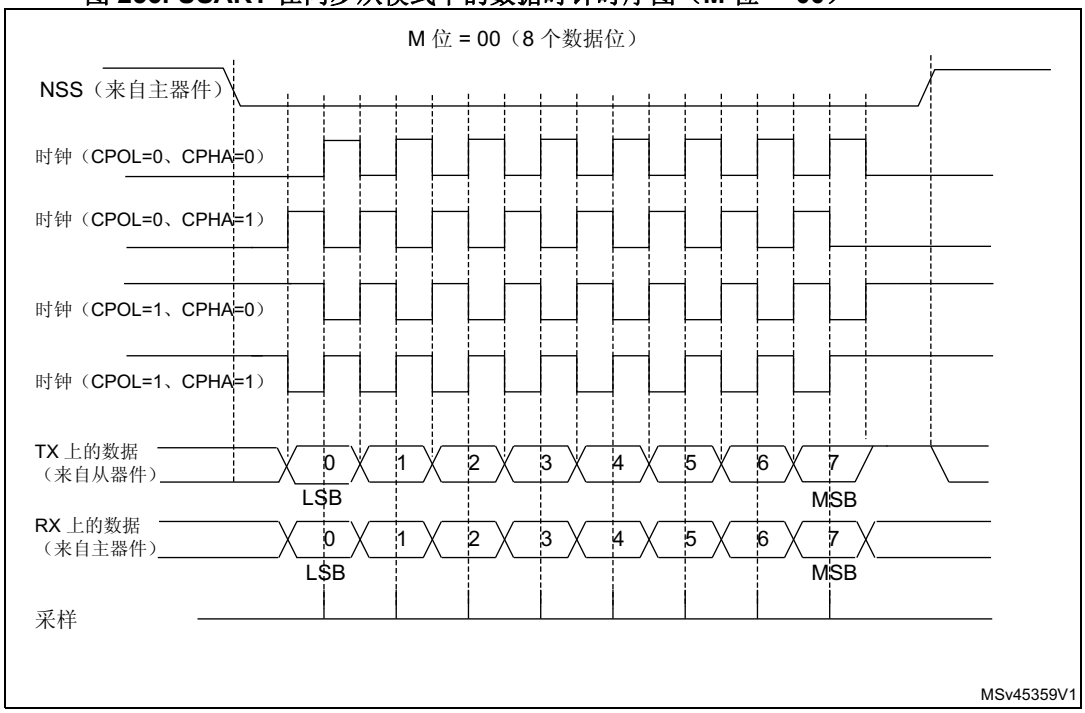
注意： *当外设用于 SPI 从器件模式时，外设时钟源 (*usart_ker_ck_pres*) 的频率必须是 CK 输入频率的 3 倍以上。*

USART_CR2 寄存器中的 CPOL 位和 CPHA 位分别用于选择时钟极性和外部时钟的相位（请参见*图 256*）。

从发送模式支持下溢错误标志。如果在软件尚未将任何值加载到 USART_TDR 之前出现第一个数据发送时钟脉冲，则此标志将置 1。

从器件支持硬件和软件 NSS 管理。

图 256. USART 在同步从模式下的数据时钟时序图（M 位 = 00）



从器件选择 (NSS) 引脚管理

可通过 USART_CR2 寄存器中的 DIS_NSS 位设置硬件或软件从器件选择管理：

- 软件 NSS 管理 (DIS_NSS = 1)
  始终选择 SPI 从器件，忽略 NSS 输入引脚。
  外部 NSS 引脚空闲，可供其他应用使用。
- 硬件 NSS 管理 (DIS_NSS = 0)
  SPI 从器件选择取决于 NSS 输入引脚。NSS 为低电平时选择从器件，NSS 为高电平时取消选择从器件。

注意： *在 USART 未使能时 (UE = 0)，必须选择 LBCL（仅用于 SPI 主器件模式）、CPOL 和 CPHA 位以确保时钟脉冲正常工作。*

在 SPI 从器件模式下，必须在启动主器件通信之前（或时钟稳定时在相邻帧之间）使能 USART。否则，如果 USART 从器件在主器件处于某个帧中间时使能，则它会与主器件失去同步。在通信时钟的第一个边沿到来之前或者正在进行的通信结束之前，从器件的数据寄存器就需要准备就绪，否则 SPI 从器件会发送零。

SPI 从器件下溢错误

发生下溢错误时，USART_ISR 寄存器中的 UDR 标志会置 1，SPI 从器件继续发送最后一个数据，直到下溢错误标志由软件清零。

下溢标志在帧开始时置 1。如果 USART_CR3 寄存器中的 EIE 位置 1，则会触发下溢错误中断。

通过将 USART_ICR 寄存器中的位 UDRCF 置 1 来清零下溢错误标志。

发生下溢错误时，仍然可以对 TDR 寄存器进行写操作。清除下溢错误将允许发送新数据。

如果发生下溢错误且没有新数据被写入 TDR 中，则 TC 标志在帧结束时置 1。

注意： *如果向 USART_TDR 写入数据的时间点过于接近第一个 CK 发送边沿，则可能会发生下溢错误。为避免发生此下溢错误，应在第一个 CK 边沿的 3 个* usart_ker_ck *周期之前对 USART_TDR 进行写操作。*

24.5.15 USART 单线半双工通信

通过将 USART_CR3 寄存器中的 HDSEL 位置 1 来选择单线半双工模式。在此模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 和 CLKEN 位，
- USART_CR3 寄存器中的 SCEN 和 IREN 位。

USART 可以配置为遵循单线半双工协议，其中 TX 和 RX 线路从内部相连接。使用 USART_CR3 寄存器中的控制位 HDSEL 可在半双工通信和全双工通信间进行选择。

向 HDSEL 位写入“1”后：

- TX 和 RX 线路从内部相连接。
- 不再使用 RX 引脚。
- 无数据传输时，TX 引脚始终处于释放状态。因此，它在空闲状态或接收过程中用作标准 I/O。这意味着，必须对 I/O 进行配置，以便将 TX 配置为复用功能开漏并外接上拉电阻。

除此之外，通信协议与正常 USART 模式下的通信协议相似。此线路上的任何冲突都必须由软件管理（例如，使用中央仲裁器）。尤其要注意，发送过程永远不会被硬件封锁，只要数据是在 TE 位置 1 的情况下写入，发送就会持续进行。

24.5.16 USART 接收器超时

接收器超时功能可通过将 USART_CR2 控制寄存器中的 RTOEN 位置 1 来使能。

超时间隔通过 USART_RTOR 寄存器中的 RTO 位域进行编程。

接收器超时计数器遵循以下规则开始计数：

- STOP =“00”或 STOP =“11”时从停止位结束时开始计数。
- STOP =“10”时从第二个停止位结束时开始计数。
- STOP =“01”时从停止位开始时开始计数。

经过超时间隔后，USART_ISR 寄存器中的 RTOF 标志置 1。如果 USART_CR1 寄存器中的 RTOIE 位置 1，则会产生超时。

24.5.17 USART 智能卡模式

仅在支持智能卡模式时才涉及本节内容。请参见*第 608 页的第 24.4 节：USART 实现*。

通过将 USART_CR3 寄存器中的 SCEN 位置 1 选择智能卡模式。在智能卡模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 位，
- USART_CR3 寄存器中的 HDSEL 和 IREN 位。

此外，还可将 CLKEN 位置 1，以便为智能卡提供时钟。

智能卡接口支持符合 ISO 7816-3 标准的异步智能卡协议。同时支持 T = 0（字符模式）和 T = 1（块模式）。

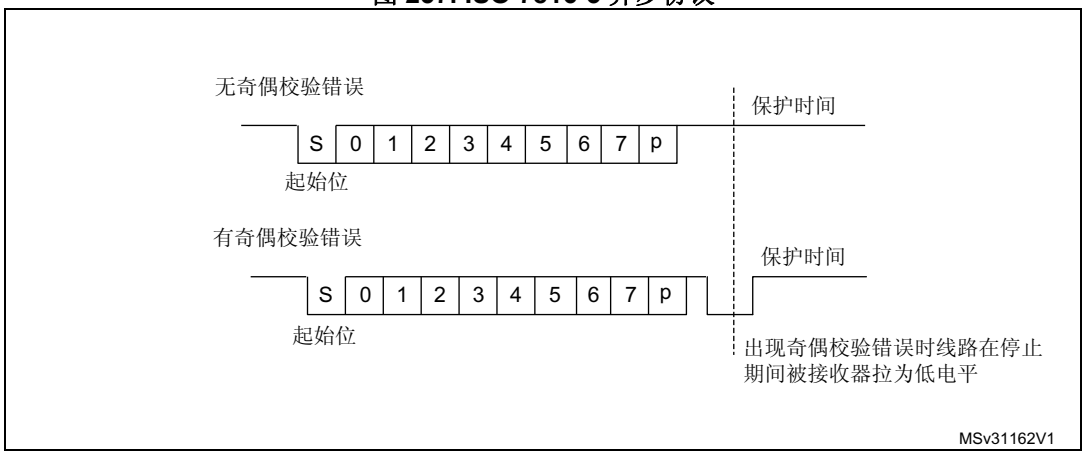
USART 应如下所示进行配置：

- 8 个位加奇偶校验：当 USART_CR1 寄存器中的 M = 1 且 PCE = 1 时
- 发送和接收数据时使用 1.5 个停止位：USART_CR2 寄存器中的 STOP =“11”。接收时也可以选择 0.5 个停止位。

在 T = 0（字符）模式下，奇偶校验错误在保护时间周期内的每个字符结束时指示。

图 257 显示了有奇偶校验错误和无奇偶校验错误时数据线上情况的示例。

图 257. ISO 7816-3 异步协议



连接到智能卡时，USART 的 TX 输出会驱动一条双向线（它也由智能卡驱动）。必须将 TX 引脚配置为开漏引脚。

智能卡模式采用单线半双工通信协议。

- 从发送移位寄存器发送数据会经过至少 1/2 个时钟周期的延迟。正常工作时，已满的发送移位寄存器会在下一个波特时钟边沿开始移位。在智能卡模式下，此发送过程还会进一步经过 1/2 波特时钟周期的延迟。
- 发送时，如果智能卡检测到奇偶校验错误，它会通过将线路驱动为低电平 (NACK) 告知 USART 此状态。此 NACK 信号（将发送线拉低 1 个时钟周期）会导致发送器端（配置为 1.5 个停止位）出现帧错误。USART 可根据协议自动重新发送数据。重试次数在 SCARCNT 位域中编程。如果经过编程次数的重试后，USART 继续收到 NACK，USART 会停止发送并将该错误以帧错误形式发出。TXE 位（使能 FIFO 模式时为 TXFNF 位）可使用 USART_RQR 寄存器中的 TXFRQ 位置 1。
- 发送时智能卡自动重试：在 USART 检测 NACK 与重复字符的起始位之间插入 2.5 个波特率周期的延迟。接收最后一个重复字符后，立即将 TC 位置 1（无保护时间）。如果软件要重复此操作，必须确保标准要求的最短 2 个波特率周期。
- 如果在接收一个使用 1.5 个停止位编程的帧期间检测到奇偶校验错误，则在完成接收帧后，发送线会被拉低一个时钟周期。这是为了向智能卡指出发送到 USART 的数据尚未正确接收。如果 NACK 控制位置 1，接收器会向奇偶校验错误发送“NACK”信号；否则不会发送 NACK 信号（将在 T = 1 模式下会使用）。如果接收到的字符错误，则不会激活 RXNE（使能 FIFO 模式时为 RXFNE）/接收 DMA 请求。根据协议规范，智能卡必须重新发送相同的字符。如果经过 SCARCNT 位域中指定的最大重试次数后接收到的字符仍然错误，USART 会停止发送 NACK 信号，并将错误以奇偶校验错误的形式发出。
- 接收时智能卡自动重试：如果 USART 向智能卡发送 NACK 信号，但智能卡不重复字符，则 BUSY 标志将保持置 1。

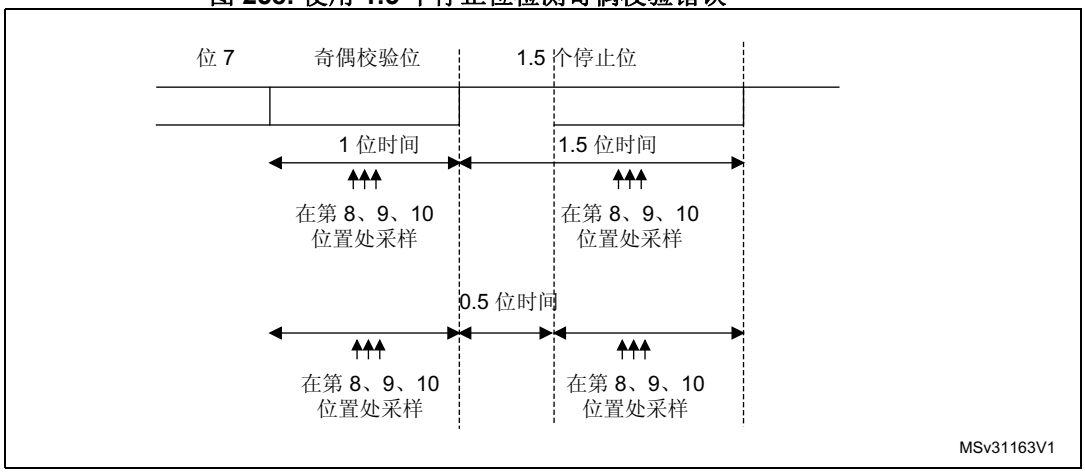
- 发送时，USART 会在两个连续字符之间插入保护时间（按照保护时间寄存器中编程的值）。由于保护时间在前一个字符的停止位后测量，因此必须将 GT[7:0] 寄存器编程为所需 CGT（字符保护时间，如 7816-3 规范所定义）减去 12（一个字符的持续时间）。
- 通过对保护时间寄存器进行编程，可以延迟 TC 标志的置位。正常工作时，当发送移位寄存器为空且没有新的发送请求出现时，会对 TC 标志进行置位。在智能卡模式下，空的发送移位寄存器会触发保护时间计数器，使其递增计数至保护时间寄存器中的值。在此期间，TC 标志被强制为低电平。当保护时间计数器达到设置值时，TC 置位为高电平。TCBGT 标志可用于检测数据传输是否结束，而无需等待保护时间结束。该标志在帧发送结束之后且未从智能卡接收到 NACK 时置 1。
- TC 标志的释放不受智能卡模式的影响。
- 如果在发送端检测到帧错误（由来自接收器的 NACK 信号引起），则发送端的接收块不会将 NACK 作为起始位进行检测。根据 ISO 协议，接收到的 NACK 信号的持续时间可以是 1 或 2 个时钟周期。
- 在接收端，如果检测到奇偶校验错误并发送了 NACK 信号，则接收端不会将 NACK 作为起始位进行检测。

注意： *中断字符在智能卡模式下无效。带有帧错误的 0x00 数据被视为数据，而非中断。*

当翻转 TE 位时，不会发送空闲帧。空闲帧（在其他配置中进行了定义）在 ISO 协议中未进行定义。

图 258 介绍了 USART 如何对 NACK 信号采样。在本例中，USART 正在发送数据并配置了 1.5 个停止位。USART 的接收部分已被使能，以检查数据的完整性和 NACK 信号。

图 258. 使用 1.5 个停止位检测奇偶校验错误



USART 可以通过 CK 输出为智能卡提供时钟。在智能卡模式下，CK 仅通过一个 5 位预分频器由内部外设输入时钟提供。分频比在 USART_GTPR 寄存器中进行配置。CK 频率可在 usart_ker_ck_pres/2 到 usart_ker_ck_pres/62 之间进行编程，其中 usart_ker_ck_pres 为经过编程的预分频器分频的外设输入时钟。

块模式 (T = 1)

在 T = 1（块）模式下，通过将 USART_CR3 寄存器中的 NACK 位清零可停用奇偶校验错误发送。

在块模式下请求从智能卡执行读操作时，软件必须将 RTOR 寄存器编程为 BWT（块等待时间）- 11。如果在经过此时间段后未从智能卡接收到应答，将生成超时中断。如果在该时间段超时之前接收到第一个字符，则通过 RXNE/RXFNE 中断发出信号指示。

注意： *即使在使用 DMA 模式下的 USART 从块模式下的智能卡读取时，也必须使能 RXNE/RXFNE 中断。同时，只有在接收到第一个字节后才可使能 DMA。*

接收到第一个字符（RXNE/RXFNE 中断）后，为允许自动检查两个连续字符间的最长等待时间，必须将 RTO 寄存器编程为 CWT（字符等待时间 - 值 11）。此时间以波特率时间单位表示。前一个字符结束后，如果智能卡未在小于 CWT 的时间段内发送新字符，USART 将通过 RTOF 标志和中断（当 RTOIE 位置 1 时）向软件指示此情况。

注意： *按照智能卡协议定义，BWT/CWT 的值应从最后一个字符开始（起始位）时定义。必须将 RTO 寄存器分别编程为 BWT -11 或 CWT -11，并考虑最后一个字符本身的长度。*

块长度计数器用于对 USART 接收到的所有字符进行计数。当 USART 进行发送时，此计数器复位。块长度由智能卡在块的第三个字节（起始字段）中传达。必须将此值编程到 USART_RTOR 寄存器中的 BLEN 字段。使用 DMA 模式时，在块开始之前，必须将该寄存器位域编程为最小值 (0x0)。对于该值，在接收到第四个字符后生成中断。软件必须读取 LEN 字段（第三个字节），其值必须从接收缓存区中读取。

在中断驱动接收模式下，块长度可以由软件或者通过编程 BLEN 值来检查。但是在块开始前，可以编程 BLEN 的最大值 (0xFF)。接收到第三个字符后，将编程实际值。

如果块使用 LRC（纵向冗余校验，1 个结尾字节），则 BLEN = LEN。如果块使用 CRC 机制（2 个结尾字节），则必须编程 BLEN = LEN+1。块总长度（包括起始字段、结尾字段和信息字段）等于 BLEN+4。块结束的信号通过 EOBF 标志和中断（EOBIE 位置 1 时）发送给软件。

如果块长度出现错误，则通过 RTO 中断（字符等待时间上溢）发送块结束信号。

注意： *错误检查代码 (LRC/CRC) 必须通过软件计算/验证。*

正向约定和反向约定

智能卡协议定义了两种约定：正向约定和反向约定。

正向约定定义为：LSB 在前，逻辑位值 1 对应于线路的 H 状态，奇偶校验为偶校验。要使用此约定，必须编程以下控制位：MSBFIRST = 0，DATAINV = 0（默认值）。

反向约定定义为：MSB 在前，逻辑位值 1 对应于信号线路的 L 状态，奇偶校验为偶校验。要使用此约定，必须编程以下控制位：MSBFIRST = 1，DATAINV = 1。

注意： *将逻辑数据值取反（0 = H，1 = L）时，奇偶校验位将同样取反。*

为识别智能卡约定，智能卡会将初始字符 TS 作为 ATR（复位应答）的第一个字符发送。TS 支持两种格式：LHHL LLL LLH 和 LHHL HHH LLH。

- (H) LHHL LLL LLH 建立反向约定：状态 L 编码为值 1，时间分量 2 传送最高有效位（MSB 在前）。按反向约定解码时，传送的字节等于“3F”。
- (H) LHHL HHH LLH 建立正向约定：状态 H 编码为值 1，时间分量 2 传送最低有效位（LSB 在前）。按正向约定解码时，传送的字节等于“3B”。

在 2 到 10 的九个时间分量中，如果有偶数个位设置为 1，则字符的奇偶校验正确。

由于 USART 不了解智能卡使用哪种约定，因此 USART 需要能够识别任意一种模式并相应操作。模式识别不在硬件中完成，而是通过软件序列完成。此外，假设以正向约定配置 USART（默认）而智能卡以反向约定 (TS = LHHL LLL LLH) 应答，则 USART 接收到的字节将为 03，奇偶校验将为奇校验。

因此，有两种方法可用于识别 TS 模式：

方法 1

将 USART 编程为标准智能卡模式/正向约定。这种情况下，TS 模式接收会向智能卡生成奇偶校验错误中断和错误信号。

- 奇偶校验错误中断通知软件智能卡未以正向约定正确应答。之后，软件重新将 USART 编程为反向约定。
- 为响应错误信号，智能卡会重试同一 TS 字符，此时重新编程后的 USART 将正确接收该字符。

或者，为应答奇偶校验错误中断，软件可决定重新编程 USART，同时向智能卡生成新的复位命令，然后再次等待 TS。

方法 2

将 USART 编程为 9 位/无奇偶校验模式，无位反向。在此模式下，按如下方式接收两种 TS 模式中的任一种：

(H) LHHL LLL LLH = 0x103：选择反向约定

(H) LHHL HHH LLH = 0x13B：选择正向约定

软件根据这两种模式检查接收到的字符，如果其中任意一种匹配，则相应编程 USART 以接收下一个字符。

如果两种都未被识别，则可能复位智能卡以重新开始协商。

24.5.18 USART IrDA SIR ENDEC 模块

仅在支持 IrDA 模式时才涉及本节内容。请参见*第 608 页的第 24.4 节：USART 实现*。

通过将 USART_CR3 寄存器中的 IREN 位置 1 来选择 IrDA 模式。在 IrDA 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN、STOP 和 CLKEN 位，
- USART_CR3 寄存器中的 SCEN 和 HDSEL 位。

IrDA SIR 物理层规定使用反相归零 (RZI) 调制方案，它以红外光脉冲表示逻辑 0（参见*图 259*）。

SIR 发送编码器用于调制 USART 发出的非归零 (NRZ) 位流。输出脉冲流会发送到外部输出驱动器和红外线 LED。USART 支持的 SIR ENDEC 比特率最高为 115.2 kbaud。在正常模式下，所发送的脉冲宽度规定为一个位周期的 3/16。

SIR 接收解码器用于解调由红外探测器发出的归零位流，并将接收到的 NRZ 串行位流输出到 USART。在空闲状态下，解码器输入通常为高电平（标记状态）。发送编码器输出的极性与解码器输入相反。当解码器输入为低电平时，会检测到起始位。

- IrDA 是一个半双工通信协议。如果发送器忙（USART 正在向 IrDA 编码器发送数据时），则 IrDA 解码器会忽略 IrDA 接收线上的所有数据；如果接收器忙（USART 正在接收来自 USART 的解码数据时），则 IrDA 不会对 USART 发送到 IrDA 的 TX 上的数据进行编码。在接收数据时，应避免同时进行发送，因为这样做可能会破坏要发送的数据。
- “0”作为高电平脉冲发送，而“1”作为“0”发送。在正常模式下，脉冲宽度规定为所选位周期的 3/16（参见*图 260*）。
- SIR 解码器用于将兼容 IrDA 的接收信号转换为 USART 的位流。
- SIR 接收逻辑将高电平状态视为逻辑“1”，将低电平脉冲视为逻辑“0”。
- 发送编码器输出的极性与解码器输入相反。SIR 输出在空闲时处于低电平状态。

- IrDA 规范要求脉冲容忍值要大于 1.41 µs。可接受的脉冲宽度可通过寄存器设置。接收器端的干扰检测逻辑会滤除宽度小于 2 个 PSC 周期的脉冲（PSC 是在 USART_GTPR 中编程的预分频器值）。宽度小于 1 个 PSC 周期的脉冲都将被拒绝，但宽度大于 1 个而小于 2 个周期的脉冲可能被接受也可能被拒绝，而宽度大于 2 个周期的脉冲将被接受作为有效脉冲。当 PSC = 0 时，IrDA 编码器/解码器不工作。
- 接收器能够与低功耗发送器进行通信。
- 在 IrDA 模式下，USART_CR2 寄存器中的停止位必须配置为“1 个停止位”。

IrDA 低功耗模式

- 发送器

  在低功耗模式下，脉冲宽度不再保持为位周期的 3/16。此时的脉冲宽度为低功耗波特率的 3 倍，最小可为 1.42 MHz。通常此值是 1.8432 MHz (1.42 MHz < PSC < 2.12 MHz)。低功耗模式下的可编程分频器会对系统时钟进行分频，以达到此值。
- 接收器

  在低功耗模式下接收与在正常模式下接收类似。为进行干扰检测，USART 应丢弃持续时间短于 1/PSC 的脉冲。只有当持续时间大于 2 个 IrDA 低功耗波特时钟周期（USART_GTPR 寄存器中的 PSC 值）时，才是有效低电平。

注意： *宽度小于两个但大于一个 PSC 周期的脉冲可能被接受，也可能被拒绝。*

接收器的建立时间应由软件进行管理。IrDA 物理层规范规定发送和接收之间至少要经过 10 ms 的延迟（IrDA 是一个半双工协议）。

图 259. IrDA SIR ENDEC 框图

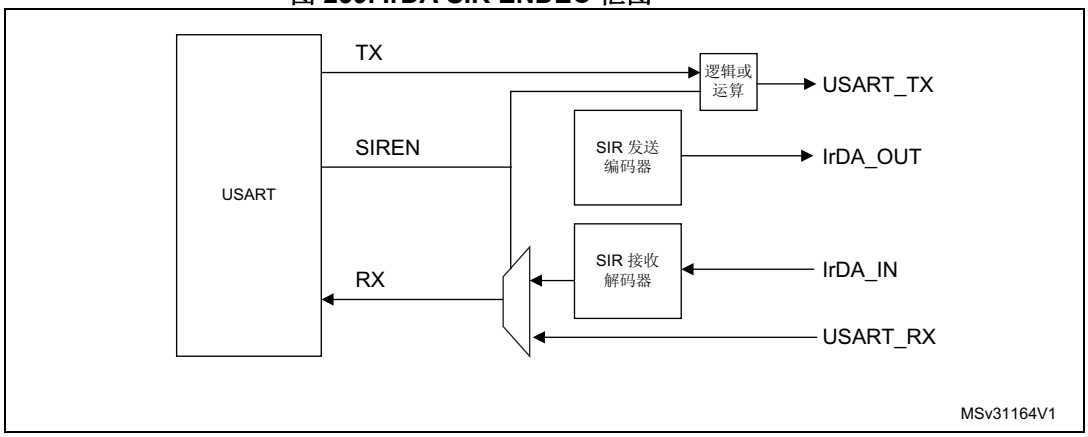
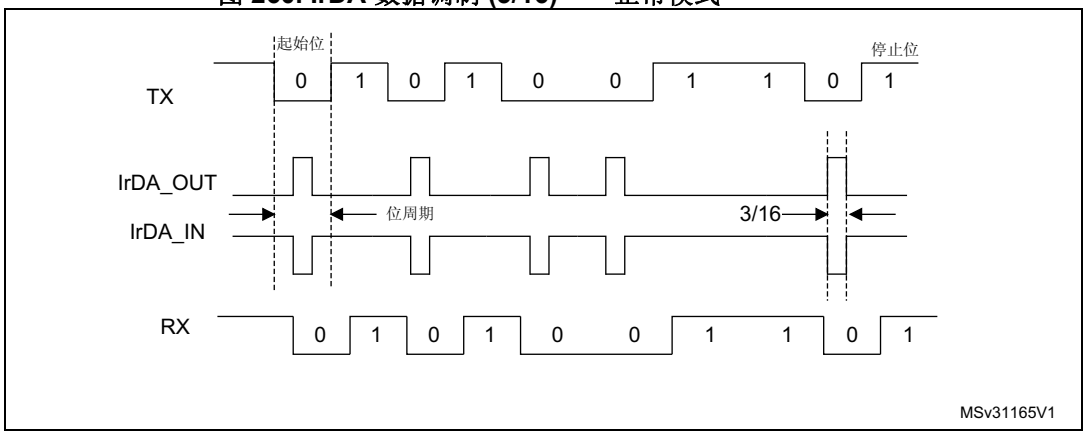


图 260. IrDA 数据调制 (3/16)——正常模式



24.5.19 使用 USART 和 DMA 进行连续通信

USART 能够使用 DMA 进行连续通信。接收缓冲区和发送缓冲区的 DMA 请求是独立的。

注意： *要确定是否支持 DMA 模式，请参见第 608 页的第 24.4 节：USART 实现。如果不支持 DMA 模式，请按照第 24.5.6 节中的说明使用 USART。可以将 USART_ISR 寄存器中的 TXE/RXNE 标志清零，从而在禁止 FIFO 时实现连续通信。*

使用 DMA 进行发送

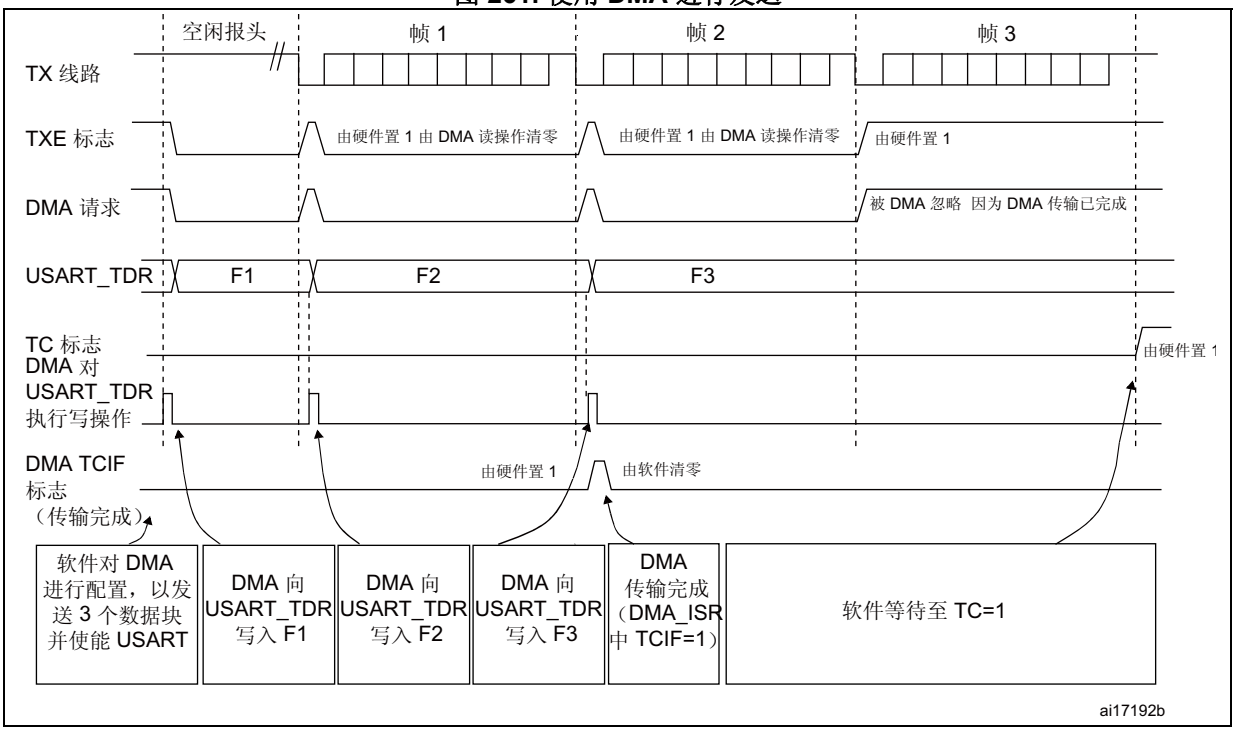
将 USART_CR3 寄存器中的 DMAT 位置 1 可以使能 DMA 模式进行发送。当 TXE 标志（使能 FIFO 模式时为 TXFNF 标志）置 1 时，可使用 DMA 外设（请参见相应的*直接存储器访问控制器*部分）将数据从配置的 SRAM 区域加载到 USART_TDR 寄存器。要映射一个 DMA 通道以进行 USART 发送，请按以下步骤操作（x 表示通道编号）：

1. 在 DMA 控制寄存器中写入 USART_TDR 寄存器地址，将其配置为传输的目标地址。每次发生 TXE（使能 FIFO 模式时为 TXFNF）事件后，数据都从存储器移动到此地址。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的源地址。每次发生 TXE（使能 FIFO 模式时为 TXFNF）事件后，数据都从该存储区域加载到 USART_TDR 寄存器中。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 寄存器中配置通道优先级
5. 根据应用的需求，在完成一半或全部传输后产生 DMA 中断。
6. 通过将 USART_ICR 寄存器中的 TCCF 位置 1，将 USART_ISR 寄存器中的 TC 标志清零。
7. 在 DMA 寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

在发送模式下，DMA 对所有要发送的数据执行了写操作（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 TC 标志进行监视，以确保 USART 通信已完成。在禁止 USART 之前或系统进入低功耗模式之前（禁止外设时钟时）必须执行此步骤，以避免损坏最后一次发送。软件必须等待直到 TC = 1。TC 标志在所有数据发送期间都保持清零状态，然后在最后一帧发送结束时由硬件置 1。

图 261. 使用 DMA 进行发送



注意： *使能 FIFO 管理时，DMA 请求由发送 FIFO 未满（即 TXFNF = 1）触发。*

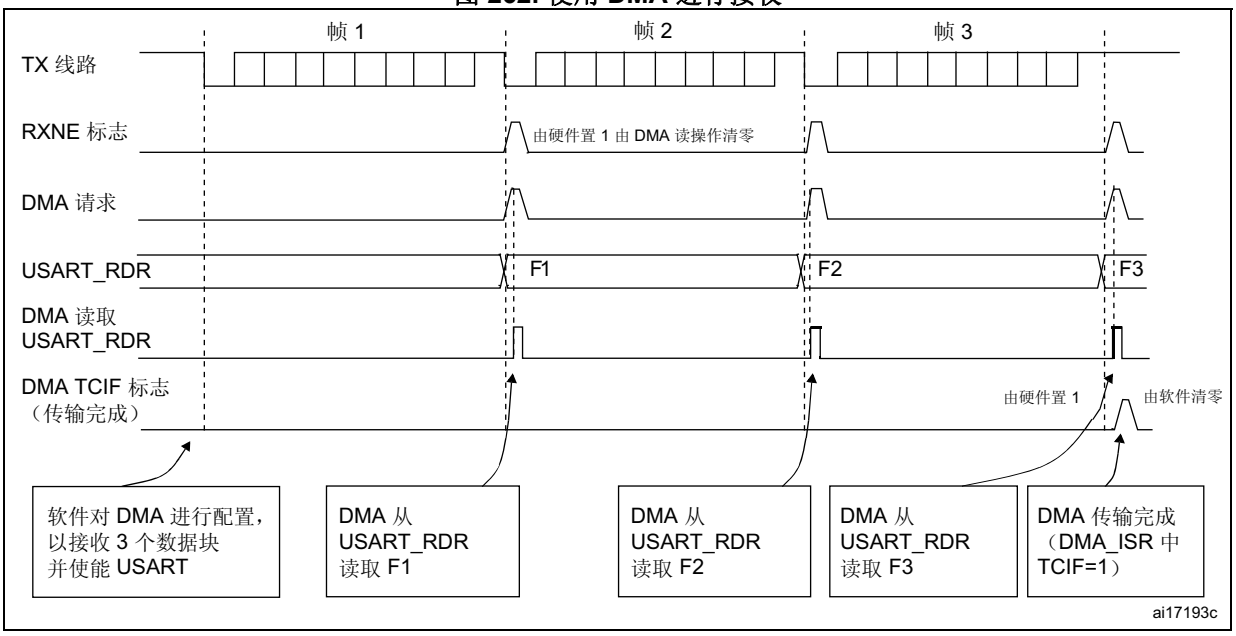
使用 DMA 进行接收

将 USART_CR3 寄存器中的 DMAR 位置 1 可以使能 DMA 模式进行接收。接收数据字节时，可使用 DMA 外设（请参见相应的*直接存储器访问控制器*部分）将数据从 USART_RDR 寄存器加载到配置的 SRAM 区域中。要映射一个 DMA 通道以进行 USART 接收，请按以下步骤操作：

1. 在 DMA 控制寄存器中写入 USART_RDR 寄存器地址，将其配置为传输的源地址。每次发生 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据都从该地址移动到存储器。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的目标地址。每次发生 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据都从 USART_RDR 加载到此存储区域。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 控制寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生中断。
6. 在 DMA 控制寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

图 262. 使用 DMA 进行接收



注意： *使能 FIFO 管理时，DMA 请求由接收 FIFO 非空（即 RXFNE = 1）触发。*

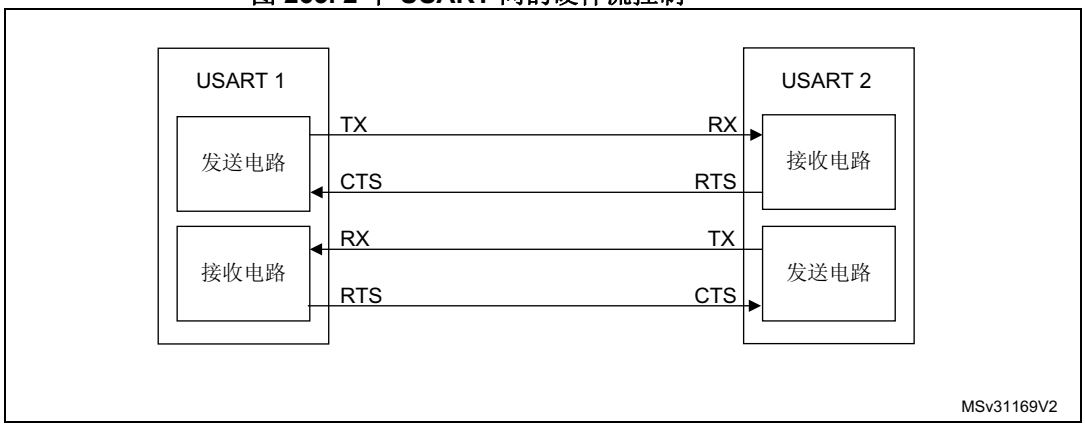
多缓冲区通信中的错误标志和中断生成

在多缓冲区通信模式下，如果事务中发生任何错误，都会在当前字节后将相应错误标志置 1。如果中断使能标志置 1，则会产生中断。在单字节接收过程中，与 RXNE（使能 FIFO 模式时为 RXFNE）一同置位的帧错误、上溢错误和噪声标志具有单独的错误标志中断使能位（USART_CR3 寄存器中的 EIE 位）；如果该位置 1，在发生其中任何一个错误时，都会在当前字节后使能中断。

24.5.20 RS232 硬件流控制和 RS485 驱动器使能

使用 CTS 输入和 RTS 输出可以控制 2 个器件间的串行数据流。*图 263* 显示了在这种模式下如何连接 2 个器件：

图 263. 2 个 USART 间的硬件流控制

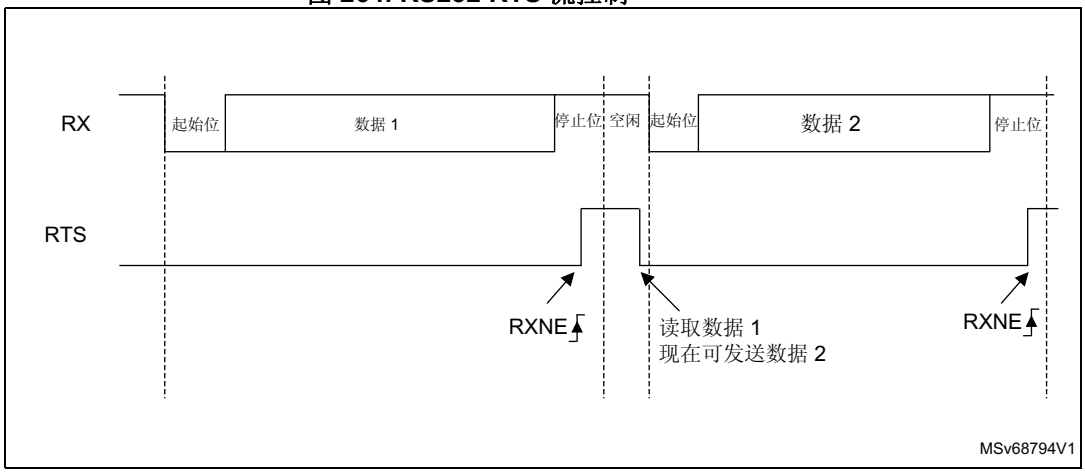


向 USART_CR3 寄存器中的 RTSE 位和 CTSE 位写入“1”，可分别使能 RS232 RTS 和 CTS 流控制。

RS232 RTS 流控制

如果使能 RTS 流控制 (RTSE = 1)，只要 USART 接收器准备好接收新数据，便会将 RTS 变为无效（连接到低电平）。当接收寄存器已满时，会将 RTS 变为有效，表明发送过程会在当前帧结束后停止。*图 264* 显示了在使能 RTS 流控制的情况下进行通信的示例。

图 264. RS232 RTS 流控制



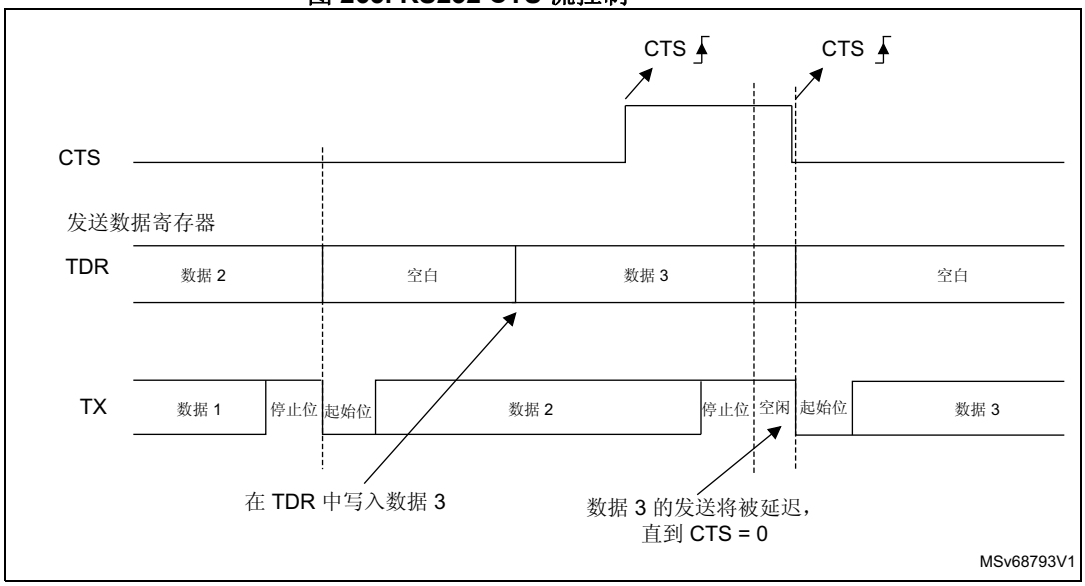
注意： *如果使能 FIFO 模式，则仅当 RXFIFO 已满时，RTS 才会变为有效。*

RS232 CTS 流控制

如果使能 CTS 流控制 (CTSE = 1)，则发送器会在发送下一帧前检查 CTS。如果 CTS 无效（连接到低电平），则会发送下一数据（假设数据已准备好发送，即 TXE/TXFE = 0）；否则不会进行发送。如果在发送过程中 CTS 变为有效，则当前发送完成之后，发送器停止。

当 CTSE = 1 时，只要 CTS 发生变化，CTSIF 状态位便会由硬件自动置 1。这指示接收器是否已准备好进行通信。如果 USART_CR3 寄存器中的 CTSIE 位置 1，则会产生中断。*图 265* 显示了在使能 CTS 流控制的情况下进行通信的示例。

图 265. RS232 CTS 流控制



注意： *为正常运行，必须在当前字符结束前至少 3 个 USART 时钟源周期内将 CTS 拉低。此外还应注意，当脉冲短于 2 个 PCLK 周期时无法将 CTSCF 标志置 1。*

RS485 驱动器使能

驱动器使能功能可通过将 USART_CR3 控制寄存器中的 DEM 位置 1 使能。这样用户便可通过 DE（驱动器使能）信号激活外部收发器控制。使能时间为激活 DE 信号与 START 位开始间的时间。可以通过 USART_CR1 控制寄存器中的 DEAT [4:0] 位域编程禁止时间。禁止时间为发送的消息中最后一个停止位结束与取消激活 DE 信号间的时间。可以通过 USART_CR1 控制寄存器中的 DEDT [4:0] 位域编程禁止时间。DE 信号的极性可使用 USART_CR3 控制寄存器中的 DEP 位配置。

在 USART 中，DEAT 和 DEDT 以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

24.5.21 USART 低功耗管理

USART 具有高级低功耗模式功能，即使禁止 usart_pclk 时钟，也能正常传输数据。

UESM 位置 1 时，USART 能够将 MCU 从低功耗模式唤醒。

当 usart_pclk 被关闭时，如果某些特定操作需要激活 **usart_pclk** 时钟，则 USART 会提供唤醒中断 (**usart_wkup**)：

- 禁止 FIFO 模式时

  必须激活 usart_pclk 时钟以清空 USART 数据寄存器。

  在这种情况下，usart_wkup 中断源为 RXNE 置“1”。RXNEIE 位必须在进入低功耗模式之前置 1。

- 使能 FIFO 模式时

  必须激活 usart_pclk 时钟以：

  – 填充 TXFIFO

  – 或清空 RXFIFO

  在这种情况下，usart_wkup 中断源可以为：

  – RXFIFO 非空。此时，RXFNEIE 位必须在进入低功耗模式之前置 1。

  – RxFIFO 已满。此时，RXFFIE 位必须在进入低功耗模式之前置 1，接收到的数据量对应于 RXFIFO 大小，并且 RXFF 标志未置 1。

  – TXFIFO 为空。此时，TXFEIE 位必须在进入低功耗模式之前置 1。

  这允许在低功耗模式下发送/接收 TXFIFO/RXFIFO 中的数据。

  为了避免发生上溢/下溢错误以及在低功耗模式下发送/接收数据，usart_wkup 中断源可以是以下事件之一：

  – 达到 TXFIFO 阈值。此时，TXFTIE 位必须在进入低功耗模式之前置 1。

  – 达到 RXFIFO 阈值。此时，RXFTIE 位必须在进入低功耗模式之前置 1。

  例如，如果唤醒时间短于在线路上接收单个字节所需的时间，则应用可将阈值设为最大 RXFIFO 大小。

  使用 RXFIFO 已满、TXFIFO 为空、RXFIFO 非空和 RXFIFO/TXFIFO 阈值中断将 MCU 从低功耗模式唤醒时，可在低功耗模式下尽可能多地进行 USART 传输，这样有助于优化功耗。

或者，也可通过 WUS 位域选择特定 **usart_wkup** 中断。

检测到唤醒事件后，WUF 标志会由硬件置 1 并在 WUFIE 位置 1 时生成一个 **usart_wkup** 中断。

注意： *在进入低功耗模式之前，请确保未进行任何 USART 传输。检查 BUSY 标志不能确保在进行数据接收时绝不会进入低功耗模式。*

WUF 标志在检测到唤醒事件时置 1，而与 MCU 处于低功耗模式还是工作模式无关。

如果在初始化和使能接收器后立即进入低功耗模式，则必须检查 REACK 位以确保 USART 确实已使能。

当 DMA 用于接收时，它必须在进入低功耗模式前禁止，并在退出低功耗模式后重新使能。

如果使能 FIFO，则只有在使能静默模式的情况下才能在地址匹配时从低功耗模式唤醒。

使用静默模式和低功耗模式

如果 USART 在进入低功耗模式前处于静默模式：

- 不得使用空闲检测来从静默模式唤醒，因为空闲检测无法在低功耗模式下工作。
- 如果使用地址匹配来从静默模式唤醒，则低功耗模式唤醒源也必须是地址匹配。如果 RXNE 标志在进入低功耗模式时置 1，则接口将在地址匹配时和从低功耗模式唤醒时保持静默模式。

注意： *使能 FIFO 管理时，静默模式可与从低功耗模式唤醒搭配使用，而且没有任何限制（即，上述关于静默和低功耗模式的两点仅在 FIFO 管理禁止时有效）。*

USART 内核时钟 (usart_ker_ck) 在低功耗模式下关闭时从低功耗模式唤醒

在低功耗模式下，如果在 USART 接收线上检测到下降沿时 usart_ker_ck 时钟处于关闭状态，则 USART 接口会借助 usart_ker_ck_req 信号请求开启 usart_ker_ck 时钟。usart_ker_ck 随后用来接收帧。

如果唤醒事件通过验证，则 MCU 将从低功耗模式唤醒，并会正常进行数据接收。

如果唤醒事件未通过验证，则 usart_ker_ck 会再次关闭，MCU 不会被唤醒并保持在低功耗模式下，且内核时钟请求被释放。

以下示例显示了将唤醒事件编程为“地址匹配检测”并禁止 FIFO 管理的情况。

图 266 所示为唤醒事件通过验证时的 USART 行为。

图 266. 唤醒事件通过验证（唤醒事件 = 地址匹配，禁止 FIFO）

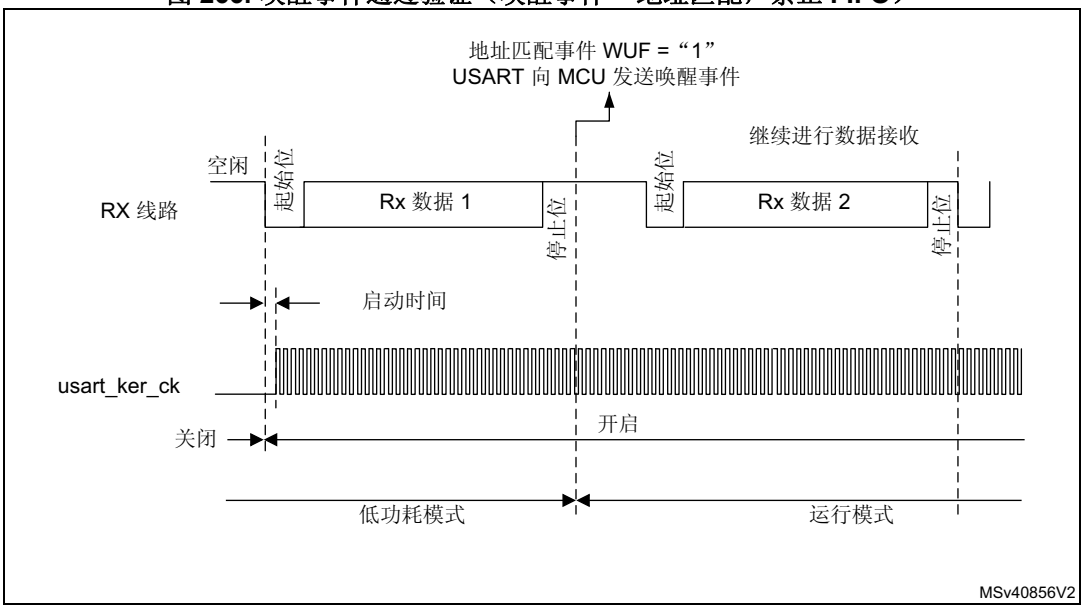
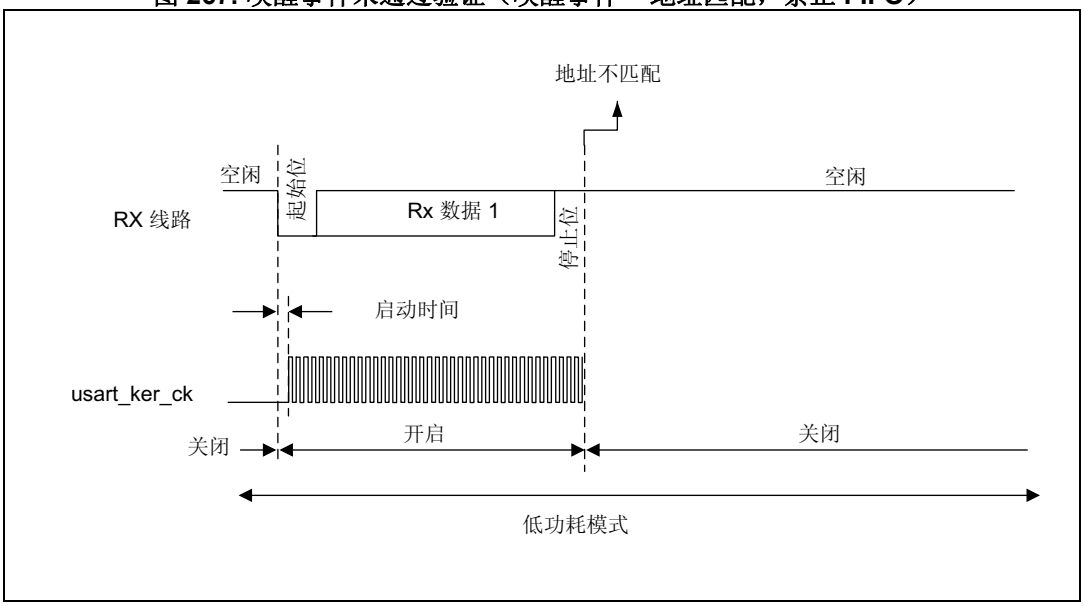


图 267 所示为唤醒事件未通过验证时的 USART 行为。

图 267. 唤醒事件未通过验证（唤醒事件 = 地址匹配，禁止 FIFO）



注意： *将地址匹配或任一接收的帧用作唤醒事件时，如上两图适用。如果唤醒事件为起始位检测，则 USART 会在起始位结束时向 MCU 发送唤醒事件。*

确定允许从低功耗模式正确唤醒器件的最大 USART 波特率

允许从低功耗模式正确唤醒器件的最大波特率取决于唤醒时间参数（请参见器件数据手册）和 USART 接收器容差（请参见*第 24.5.8 节：USART 接收器对时钟偏差的容差*）。

举例来说：OVER8 = 0，M 位 = “01”，ONEBIT = 0，BRR [3:0] = 0000。

在这些条件下，根据*表 108：BRR [3:0] = 0000 时的 USART 接收器容差*，USART 接收器的容差为 3.41%。

DTRA + DQUANT + DREC + DTCL + DWU < USART 接收器的容差

$D_{WUmax} = t_{WUUSART} / (11 \times T_{bit\ Min})$

$T_{bit\ Min} = t_{WUUSART} / (11 \times D_{WUmax})$

其中 $t_{WUUSART}$ 为从低功耗模式唤醒的时间。

考虑一种理想情况：参数 DTRA、DQUANT、DREC 和 DTCL 为 0%，则 DWU 的最大值为 3.41%。实际上，我们至少需要考虑 usart_ker_ck 不精确的情况。

例如，如果将 HSI48 用作 usart_ker_ck，HSI48 的不精确度为 1%，则可以得到：

$t_{WUUSART}$ = 3 µs（仅提供数值作为参考；有关正确数值，请参见器件数据手册）。

D_{WUmax} = 3.41% - 1% = 2.41%

$T_{bit\ min}$ = 3 µs/ (11 x 2.41%) = 11.32 µs。

因此，允许从低功耗模式正确唤醒的最大波特率为：1/11.32 µs = 88.36 Kbaud。

24.6 处于低功耗模式的 USART

表 111. 低功耗模式对 USART 的影响

模式	说明
睡眠	无影响。USART 中断可使器件退出睡眠模式。
停止(1)	USART 寄存器的内容会被保持。 当 USART 的时钟由停止模式下可用的振荡器提供时，USART 能够将微控制器从停止模式唤醒。
待机	USART 外设掉电，退出待机模式后必须重新初始化。

1. 请参见*第 24.4 节：USART 实现*以了解给定外设实例是否支持从停止模式唤醒。如果某个实例在给定的停止模式下不起作用，则必须在进入此停止模式之前将其禁用。

24.7 USART 中断

有关所有 USART 中断请求的详细说明，请参见*表 112*。

表 112. USART 中断请求

中断向量	中断事件	事件标志	使能控制位	中断清除方法	退出睡眠模式	退出停止(1)模式	退出待机模式
USART 或 UART	发送数据寄存器为空	TXE	TXEIE	写入 TDR	是	否	否
	发送 FIFO 未满	TXFNF	TXFNFIE	TXFIFO 已满		否	
	发送 FIFO 为空	TXFE	TXFEIE	写入 TDR 或向 TXFRQ 中写入 1		是	
	达到发送 FIFO 阈值	TXFT	TXFTIE	写入 TDR		是	
	CTS 中断	CTSIF	CTSIE	向 CTSCF 中写入 1		否	
	发送完成	TC	TCIE	写入 TDR 或向 TCCF 中写入 1		否	
	保护时间前发送完成	TCBGT	TCBGTIE	写入 TDR 或向 TCBGT 中写入 1		否	

表 112. USART 中断请求（续）

中断向量	中断事件	事件标志	使能控制位	中断清除方法	退出睡眠模式	退出停止(1)模式	退出待机模式
USART 或 UART	接收数据寄存器不为空（已准备好读取数据）	RXNE	RXNEIE	读取 RDR 或向 RXFRQ 中写入 1	是	是	否
	接收 FIFO 非空	RXFNE	RXFNEIE	读取 RDR 直到 RXFIFO 为空或向 RXFRQ 中写入 1		是	
	接收 FIFO 已满	RXFF(2)	RXFFIE	读取 RDR		是	
	达到接收 FIFO 阈值	RXFT	RXFTIE	读取 RDR		是	
	检测到溢出错误	ORE	RXNEIE/ RXFNEIE	向 ORECF 中写入 1		否	
	检测到空闲线路	IDLE	IDLEIE	向 IDLECF 中写入 1		否	
	奇偶校验错误	PE	PEIE	向 PECF 中写入 1		否	
	LIN 中断	LBDF	LBDIE	向 LBDCF 中写入 1		否	
	多缓冲区通信中的噪声错误	NE	EIE	向 NFCF 中写入 1		否	
	多缓冲区通信中的上溢错误	ORE(3)		向 ORECF 中写入 1		否	
	多缓冲区通信中的帧错误	FE		向 FECF 中写入 1		否	
	字符匹配	CMF	CMIE	向 CMCF 中写入 1		否	
	接收器超时	RTOF	RTOFIE	向 RTOCCF 中写入 1		否	
	块结束	EOBF	EOBIE	向 EOBCF 中写入 1		否	
	从低功耗模式唤醒	WUF	WUFIE	向 WUC 中写入 1		是	
	SPI 从器件下溢错误	UDR	EIE	向 UDRCF 中写入 1		否	

1. 仅当外设实例支持从停止模式唤醒功能时，USART 才能将器件从停止模式唤醒。关于所支持的停止模式的列表，请参见*第 24.4 节：USART 实现*。
2. 如果 USART 接收 n+1 个数据（n 表示 RXFIFO 大小，RXFIFO 中有 n 个数据，USART_RDR 中有 1 个数据），则 RXFF 标志置 1。在停止模式下，未向 USART_RDR 提供时钟。因此，不会对该寄存器进行写操作，一旦有 n 个数据被接收并写入到 RXFIFO，RXFF 中断便会生效（RXFF 标志未置 1）。
3. 当 OVRDIS = 0 时。

24.8 USART 寄存器

有关寄存器说明中使用的缩写，请参见*第 33 页的第 1.2 节*。

外设寄存器必须按字（32 位）进行访问。

24.8.1 USART 控制寄存器 1 (USART_CR1)

USART control register 1

偏移地址：0x00

复位值：0x0000 0000

同一寄存器可用于使能 FIFO 模式（本节）和禁止 FIFO 模式（下一节）的情况。

使能 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXFFIE	TXFEIE	FIFOEN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXFNFIE	TCIE	RXFNEIE	IDLEIE	TE	RE	UESM	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **RXFFIE**：RXFIFO 变满时中断使能 (RXFIFO full interrupt enable)
该位由软件置 1 和清零。
0：禁止中断
1：当 USART_ISR 寄存器中的 RXFF = 1 时，生成 USART 中断

位 30 **TXFEIE**：TXFIFO 为空时中断使能 (TXFIFO empty interrupt enable)
该位由软件置 1 和清零。
0：禁止中断
1：当 USART_ISR 寄存器中的 TXFE = 1 时，生成 USART 中断

位 29 **FIFOEN**：FIFO 模式使能 (FIFO mode enable)
该位由软件置 1 和清零。
0：禁止 FIFO 模式。
1：使能 FIFO 模式。
只有在禁止 USART (UE = 0) 时才能写入该位域。
注意：FIFO 模式只能在标准 UART 通信、SPI 主/从器件模式和智能卡模式下使用，不得在 IrDA 和 LIN 模式下使能。

位 28 **M1**：字长 (Word length)
该位必须与位 12 (M0) 搭配使用来确定字长度。该位由软件置 1 或清零。
M[1:0] = “00”：1 个起始位，8 个数据位，n 个停止位
M[1:0] = “01”：1 个起始位，9 个数据位，n 个停止位
M[1:0] = “10”：1 个起始位，7 个数据位，n 个停止位
只有在禁止 USART (UE = 0) 时才能写入该位。
注意：在 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。

位 27 **EOBIE**：块结束中断使能 (End-of-block interrupt enable)
该位由软件置 1 和清零。
0：禁止中断
1：USART_ISR 寄存器中的 EOBF 标志置 1 时生成 USART 中断
注意：如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 26 **RTOIE**：接收器超时中断使能 (Receiver timeout interrupt enable)
该位由软件置 1 和清零。
0：禁止中断
1：USART_ISR 寄存器中的 RTOF 位置 1 时生成 USART 中断。
注意：如果 USART 不支持接收器超时功能，该位保留且必须保持复位值。第 608 页的第 24.4 节：USART 实现。

位 25:21 **DEAT[4:0]**：驱动器使能有效时间 (Driver enable assertion time)
该 5 位值用于定义激活 DE（启动器使能）信号与起始位开始间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。
只有在禁止 USART (UE = 0) 时才能写入该位域。
注意：如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 20:16 **DEDT[4:0]**：驱动器使能无效时间 (Driver enable deassertion time)

该 5 位值用于定义发送的消息中最后一个停止位结束与取消激活 DE（驱动器使能）信号间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

如果在 DEDT 时间内对 USART_TDR 寄存器执行写操作，则新数据仅在经过 DEDT 和 DEAT 时间后才会发送。

只有在禁止 USART (UE = 0) 时才能写入该位域。

注意：如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 15 **OVER8**：过采样模式 (Oversampling mode)

0：16 倍过采样

1：8 倍过采样

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：在 LIN、IrDA 和智能卡模式下，该位必须保持清零。

位 14 **CMIE**：字符匹配中断使能 (Character match interrupt enable)

该位由软件置 1 和清零。

0：禁止中断

1：USART_ISR 寄存器中的 CMF 位置 1 时生成 USART 中断。

位 13 **MME**：静默模式使能 (Mute mode enable)

该位可使能 USART 静默模式功能。该位置 1 时，USART 会按照 WAKE 位的定义在工作模式与静默模式之间切换。该位由软件置 1 和清零。

0：接收器永久处于活动模式

1：接收器可在静默模式和活动模式之间切换。

位 12 **M0**：字长 (Word length)

该位与位 28 (M1) 搭配使用来确定字长度。它由软件置 1 或清零（请参见位 28 (M1) 的说明）。

只有在禁止 USART (UE = 0) 时才能写入该位。

位 11 **WAKE**：接收器唤醒方法 (Receiver wake-up method)

该位用于确定 USART 静默模式的唤醒方法。该位由软件置 1 或清零。

0：空闲线路

1：地址标记

只有在禁止 USART (UE = 0) 时才能写入该位域。

位 10 **PCE**：奇偶校验控制使能 (Parity control enable)

该位选择硬件奇偶校验控制（生成和检测）。使能奇偶校验控制时，计算出的奇偶校验位被插入到 MSB 位置（如果 M = 1，则为第 9 位；如果 M = 0，则为第 8 位），并对接收到的数据检查奇偶校验位。该位由软件置 1 和清零。一旦该位置 1，PCE 在当前字节的后面处于活动状态（在接收和发送时）。

0：禁止奇偶校验控制

1：使能奇偶校验控制

只有在禁止 USART (UE = 0) 时才能写入该位域。

位 9 **PS**：奇偶校验选择 (Parity selection)

该位用于在使能奇偶校验生成/检测（PCE 位置 1）时选择奇校验或偶校验。该位由软件置 1 和清零。在当前字节的后面选择奇偶校验。

0：偶校验

1：奇校验

只有在禁止 USART (UE = 0) 时才能写入该位域。

位 8 **PEIE**：PE 中断使能 (PE interrupt enable)

该位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 PE = 1 时，生成 USART 中断

位 7 **TXFNFIE**：TXFIFO 未满中断使能 (TXFIFO not-full interrupt enable)

该位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 TXFNF = 1 时，生成 USART 中断

位 6 **TCIE**：传送完成中断使能 (Transmission complete interrupt enable)

该位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 TC = 1 时，生成 USART 中断

位 5 **RXFNEIE**：RXFIFO 非空中断使能 (RXFIFO not empty interrupt enable)

该位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 ORE = 1 或 RXFNE = 1 时，生成 USART 中断

位 4 **IDLEIE**：IDLE 中断使能 (IDLE interrupt enable)

该位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 IDLE = 1 时，生成 USART 中断

位 3 **TE**：发送器使能 (Transmitter enable)

该位使能发送器。该位由软件置 1 和清零。

0：禁止发送器

1：使能发送器

注意：除了在智能卡模式下以外，传送期间 TE 位上的低电平脉冲（“0”后紧跟的是“1”）会在当前字的后面发送一个报头（空闲线路）。为生成空闲字符，TE 不能立即写入“1”。为确保所需的持续时间，软件可轮询 USART_ISR 寄存器中的 TEACK 位。

在智能卡模式下，当 TE 置 1 时，在发送开始前存在 1 位的时间延迟。

位 2 **RE**：接收器使能 (Receiver enable)

该位使能接收器。该位由软件置 1 和清零。

0：禁止接收器

1：使能接收器并开始搜索起始位

位 1 **UESM**：低功耗模式下的 USART 使能 (USART enable in low-power mode)

当该位清零时，USART 无法将 MCU 从低功耗模式唤醒。

当该位置 1 时，USART 可将 MCU 从低功耗模式唤醒。

该位由软件置 1 和清零。

0：USART 无法将 MCU 从低功耗模式唤醒。

1：USART 能够将 MCU 从低功耗模式唤醒。

注意：建议在进入低功耗模式前将 UESM 位置 1，并在退出低功耗模式时将其清零。

如果 USART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 0 **UE**：USART 使能 (USART enable)

该位清零后，USART 预分频器和输出将立即停止，并丢弃所有当前操作。USART 配置会保留，而所有的 USART_ISR 状态标志均会复位。该位由软件置 1 和清零。

0：禁止 USART 预分频器和输出，低功耗模式

1：使能 USART

注意：为进入低功耗模式而不在线路上生成错误，之前必须复位 TE 位，并且软件必须等待 USART_ISR 中的 TC 位置 1 后才能复位 UE 位。

UE = 0 时也会复位 DMA 请求，因此必须在复位 UE 位前禁止 DMA 通道。

在智能卡模式下 (SCEN = 1)，无论 UE 位值为何，当 CLKEN = 1 时，CK 始终可用。

24.8.2 USART 控制寄存器 1 [备用] (USART_CR1)

USART control register 1 [alternate]

偏移地址：0x00

复位值：0x0000 0000

同一寄存器可用于使能 FIFO 模式（上一节）和禁止 FIFO 模式（本节）的情况。

禁止 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	FIFO EN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]				
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29 **FIFOEN**：FIFO 模式使能 (FIFO mode enable)

该位由软件置 1 和清零。

0：禁止 FIFO 模式。

1：使能 FIFO 模式。

只有在禁止 USART (UE = 0) 时才能写入该位域。

注意：FIFO 模式只能在标准 UART 通信、SPI 主/从器件模式和智能卡模式下使用，不得在 IrDA 和 LIN 模式下使能。

位 28 **M1**：字长 (Word length)

该位必须与位 12 (M0) 搭配使用来确定字长度。该位由软件置 1 或清零。

M[1:0] = “00”：1 个起始位，8 个数据位，n 个停止位

M[1:0] = “01”：1 个起始位，9 个数据位，n 个停止位

M[1:0] = “10”：1 个起始位，7 个数据位，n 个停止位

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：在 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。

位 27 **EOBIE**：块结束中断使能 (End of Block interrupt enable)

该位由软件置 1 和清零。

0：禁止中断

1：USART_ISR 寄存器中的 EOBF 标志置 1 时生成 USART 中断

注意：如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 26 **RTOIE**：接收器超时中断使能 (Receiver timeout interrupt enable)

该位由软件置 1 和清零。

0：禁止中断

1：USART_ISR 寄存器中的 RTOF 位置 1 时生成 USART 中断。

注意：如果 USART 不支持接收器超时功能，该位保留且必须保持复位值。第 608 页的第 24.4 节：USART 实现。

位 25:21 **DEAT[4:0]**：驱动器使能有效时间 (Driver enable assertion time)

该 5 位值用于定义激活 DE（启动器使能）信号与起始位开始间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

只有在禁止 USART (UE = 0) 时才能写入该位域。

注意：如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 20:16 **DEDT[4:0]**：驱动器使能无效时间 (Driver enable deassertion time)

该 5 位值用于定义发送的消息中最后一个停止位结束与取消激活 DE（驱动器使能）信号间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

如果在 DEDT 时间内对 USART_TDR 寄存器执行写操作，则新数据仅在经过 DEDT 和 DEAT 时间后才会发送。

只有在禁止 USART (UE = 0) 时才能写入该位域。

注意：如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 15 **OVER8**：过采样模式 (Oversampling mode)

0：16 倍过采样

1：8 倍过采样

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：在 LIN、IrDA 和智能卡模式下，该位必须保持清零。

位 14 **CMIE**：字符匹配中断使能 (Character match interrupt enable)

该位由软件置 1 和清零。

0：禁止中断

1：USART_ISR 寄存器中的 CMF 位置 1 时生成 USART 中断。

位 13 **MME**：静音模式使能 (Mute mode enable)

该位可使能 USART 静默模式功能。该位置 1 时，USART 会按照 WAKE 位的定义在工作模式与静默模式之间切换。该位由软件置 1 和清零。

0：接收器永久处于活动模式

1：接收器可在静默模式和活动模式之间切换。

位 12 **M0**：字长 (Word length)

该位与位 28 (M1) 搭配使用来确定字长度。它由软件置 1 或清零（请参见位 28 (M1) 的说明）。

只有在禁止 USART (UE = 0) 时才能写入该位。

位 11 **WAKE**：接收器唤醒方法 (Receiver wake-up method)

该位用于确定 USART 静默模式的唤醒方法。该位由软件置 1 或清零。

0：空闲线路

1：地址标记

只有在禁止 USART (UE = 0) 时才能写入该位域。

位 10 **PCE**：奇偶校验控制使能 (Parity control enable)

该位选择硬件奇偶校验控制（生成和检测）。使能奇偶校验控制时，计算出的奇偶校验位被插入到 MSB 位置（如果 M = 1，则为第 9 位；如果 M = 0，则为第 8 位），并对接收到的数据检查奇偶校验位。该位由软件置 1 和清零。一旦该位置 1，PCE 在当前字节的后面处于活动状态（在接收和发送时）。

0：禁止奇偶校验控制

1：使能奇偶校验控制

只有在禁止 USART (UE = 0) 时才能写入该位域。

位 9 **PS**：奇偶校验选择 (Parity selection)

该位用于在使能奇偶校验生成/检测（PCE 位置 1）时选择奇校验或偶校验。该位由软件置 1 和清零。在当前字节的后面选择奇偶校验。

0：偶校验

1：奇校验

只有在禁止 USART (UE = 0) 时才能写入该位域。

位 8 **PEIE**：PE 中断使能 (PE interrupt enable)

该位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 PE = 1 时，生成 USART 中断

位 7 **TXEIE**：发送数据寄存器为空

该位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 TXE = 1 时，生成 USART 中断

位 6 **TCIE**：传送完成中断使能 (Transmission complete interrupt enable)

该位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 TC = 1 时，生成 USART 中断

位 5 **RXNEIE**：接收数据寄存器非空

该位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 ORE = 1 或 RXNE = 1 时，生成 USART 中断

位 4 **IDLEIE**：IDLE 中断使能 (IDLE interrupt enable)

该位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 IDLE = 1 时，生成 USART 中断

位 3 **TE**：发送器使能 (Transmitter enable)

该位使能发送器。该位由软件置 1 和清零。

0：禁止发送器

1：使能发送器

注意：除了在智能卡模式下以外，传送期间 TE 位上的低电平脉冲（“0”后紧跟的是“1”）会在当前字的后面发送一个报头（空闲线路）。为生成空闲字符，TE 不能立即写入“1”。为确保所需的持续时间，软件可轮询 USART_ISR 寄存器中的 TEACK 位。

在智能卡模式下，当 TE 置 1 时，在发送开始前存在 1 位的时间延迟。

位 2 **RE**：接收器使能 (Receiver enable)

该位使能接收器。该位由软件置 1 和清零。

0：禁止接收器

1：使能接收器并开始搜索起始位

位 1 **UESM**：低功耗模式下的 USART 使能 (USART enable in low-power mode)

当该位清零时，USART 无法将 MCU 从低功耗模式唤醒。

当该位置 1 时，USART 可将 MCU 从低功耗模式唤醒。

该位由软件置 1 和清零。

0：USART 无法将 MCU 从低功耗模式唤醒。

1：USART 能够将 MCU 从低功耗模式唤醒。

注意：建议在进入低功耗模式前将 UESM 位置 1，并在退出低功耗模式时将其清零。

如果 USART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 0 **UE**：USART 使能 (USART enable)

该位清零后，USART 预分频器和输出将立即停止，并丢弃所有当前操作。USART 配置会保留，而所有的 USART_ISR 状态标志均会复位。该位由软件置 1 和清零。

0：禁止 USART 预分频器和输出，低功耗模式

1：使能 USART

注意：为进入低功耗模式而不在线路上生成错误，之前必须复位 TE 位，并且软件必须等待 USART_ISR 中的 TC 位置 1 后才能复位 UE 位。

UE = 0 时也会复位 DMA 请求，因此必须在复位 UE 位前禁止 DMA 通道。

在智能卡模式下 (SCEN = 1)，无论 UE 位值为何，当 CLKEN = 1 时，CK 引脚始终可用。

24.8.3 USART 控制寄存器 2 (USART_CR2)

USART control register 2

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
ADD[7:0]								RTOEN	ABRMOD[1:0]		ABREN	MSBFIRST	DATAINV	TXINV	RXINV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	DIS_NSS	Res.	Res.	SLVEN
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw			rw

位 31:24 **ADD[7:0]**：USART 节点的地址

这些位用于指定静默模式下 USART 节点的地址或低功耗模式/运行模式下要识别的字符代码。

– 在静默模式下：它们用于在多处理器通信时通过 4 位/7 位地址标记检测从静默模式唤醒。发送器发送字符的 MSB 应为 1。在 4 位地址标记检测中，只使用 ADD[3:0] 位。

– 在低功耗模式下：它们用于在字符匹配时从低功耗模式唤醒。
如果将 WUS[1:0] 编程为 0b00（WUF 在地址匹配时激活），并且已通过将 WUFIE 位置 1 使能 WUF 中断，则当收到的字符与通过 ADD[6:0] 或 ADD[3:0] 位域（取决于 ADDM7 位）编程的字符匹配时会从低功耗模式唤醒。发送器发送字符的 MSB 应为 1。

– 在静默模式无效时的运行模式下（例如，ModBus 协议中的块结束检测）：接收到的整个字符（8 位）将与 ADD[7:0] 值进行比较，如果匹配，CMF 标志将置 1。如果 CMIE 位置 1，则会生成中断。

仅在禁止接收 (RE = 0) 或禁止 USART (UE = 0) 时才能写入这些位。

位 23 **RTOEN**：接收器超时使能 (Receiver timeout enable)

该位由软件置 1 和清零。

0：禁止接收器超时功能。

1：使能接收器超时功能。

使能此功能后，如果 RX 线路在 RTOR（接收器超时寄存器）中编程的持续时间内处于空闲状态（无接收），则 USART_ISR 寄存器中的 RTOF 标志置 1。

注意：如果 USART 不支持接收器超时功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 22:21 **ABRMOD[1:0]**：自动波特率模式 (Auto baud rate mode)

这些位将由软件置 1 和清零。

00：通过测量起始位检测波特率。

01：下降沿到下降沿的测量（接收到的帧必须以一个等于 1 的位开头，即帧 = 10xxxxxx）

10：0x7F 帧检测。

11：0x55 帧检测。

仅在 ABREN = 0 时或禁止 USART (UE = 0) 时才能写入该位域。

注意：如果 DATAINV = 1 且/或 MSBFIRST = 1，这些模式必须与在线路上时相同，例如 MSBFIRST 的 0xAA。

如果 USART 不支持自动波特率功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 20 **ABREN**：自动波特率检测使能 (Auto baud rate enable)

该位由软件置 1 和清零。

0：禁止自动波特率检测。

1：使能自动波特率检测。

注意：如果 USART 不支持自动波特率功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 19 **MSBFIRST**：最高有效位在前 (Most significant bit first)

该位由软件置 1 和清零。

0：发送/接收数据时位 0 在前，后跟起始位。

1：发送/接收数据时 MSB（位 7/8）在前，后跟起始位。

只有在禁止 USART (UE = 0) 时才能写入该位域。

位 18 **DATAINV**：二进制数据反向 (Binary data inversion)

该位由软件置 1 和清零。

0：按正/正向逻辑发送/接收数据寄存器中的逻辑数据。（1 = H，0 = L）。

1：按负/反向逻辑发送/接收数据寄存器中的逻辑数据。（1 = L，0 = H）。奇偶校验位也取反。

只有在禁止 USART (UE = 0) 时才能写入该位域。

位 17 **TXINV**：TX 引脚有效电平反向 (TX pin active level inversion)

该位由软件置 1 和清零。

0：TX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作。

1：TX 引脚信号值取反（V_{DD} = 0/标记，Gnd = 1/空闲）。

允许在 TX 线路上使用外部反相器。

只有在禁止 USART (UE = 0) 时才能写入该位域。

位 16 **RXINV**：RX 引脚有效电平反向 (RX pin active level inversion)

该位由软件置 1 和清零。

0：RX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作。

1：RX 引脚信号值取反（V_{DD} = 0/标记，Gnd = 1/空闲）。

允许在 RX 线路上使用外部反相器。

只有在禁止 USART (UE = 0) 时才能写入该位域。

位 15 **SWAP**：Swap TX/RX 引脚 (Swap TX/RX pins)

该位由软件置 1 和清零。

0：按标准引脚排列定义使用 TX/RX 引脚

1：交换 TX 和 RX 引脚功能。允许在与另一个 UART 的交叉连接时工作。

只有在禁止 USART (UE = 0) 时才能写入该位域。

位 14 **LINEN**：LIN 模式使能 (LIN mode enable)

该位由软件置 1 和清零。

0：禁止 LIN 模式

1：使能 LIN 模式

LIN 模式可以使用 USART_CR1 寄存器中的 SBKRQ 位发送 LIN 同步中断（13 个低位），并可检测 LIN 同步中断。

只有在禁止 USART (UE = 0) 时才能写入该位域。

注意：如果 USART 不支持 LIN 模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 13:12 **STOP[1:0]**：停止位

这些位用于编程停止位。

00：1 个停止位

01：0.5 个停止位

10：2 个停止位

11：1.5 个停止位

只有在禁止 USART (UE = 0) 时才能写入该位域。

位 11 **CLKEN**：时钟使能 (Clock enable)

该位允许用户使能 CK 引脚。

0：禁止 CK 引脚

1：使能 CK 引脚

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果既不支持同步模式，也不支持智能卡模式，则该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

在智能卡模式下，为了向智能卡正确提供 CK 时钟，必须按以下步骤操作：

UE = 0

SCEN = 1

GTPR 配置

CLKEN= 1

UE = 1

位 10 **CPOL**：时钟极性 (Clock polarity)

该位允许用户在同步模式下选择 CK 引脚上时钟输出的极性。它与 CPHA 位结合使用可获得所需的时钟/数据关系

0：空闲时 CK 引脚为低电平

1：空闲时 CK 引脚为高电平

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持同步模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 9 **CPHA**：时钟相位

该位用于在同步模式下选择 CK 引脚上时钟输出的相位。它与 CPOL 位结合使用可获得所需的时钟/数据关系（请参见*图 247* 和*图 248*）

0：从第一个时钟边沿开始采样数据

1：从第二个时钟边沿开始采样数据

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持同步模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 8 **LBCL**：最后一个位时钟脉冲 (Last bit clock pulse)

该位用于在同步模式下选择与发送的最后一个数据位 (MSB) 关联的时钟脉冲是否必须在 CK 引脚上输出。

0：最后一个数据位的时钟脉冲不在 CK 引脚上输出

1：最后一个数据位的时钟脉冲在 CK 引脚上输出

小心： 最后一位为发送的第 7 个、第 8 个或第 9 个数据位，具体取决于 USART_CR1 寄存器中 M 位所选择的 7 位、8 位或 9 位格式。

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持同步模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 7 保留，必须保持复位值。

位 6 **LBDIE**：LIN 中断检测中断使能 (LIN break detection interrupt enable)

中断中断屏蔽（使用中断分隔符进行中断检测）。

0：禁止中断

1：当 USART_ISR 寄存器中 LBDF = 1 时，生成中断

注意：如果不支持 LIN 模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 5 **LBDL**：LIN 中断检测长度 (LIN break detection length)

该位用于选择 11 位中断检测或 10 位中断检测。

0：10 位中断检测

1：11 位中断检测

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持 LIN 模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 4 **ADDM7**：7 位地址检测/4 位地址检测 (7-bit address detection/4-bit address detection)

该位用于选择 4 位地址检测或 7 位地址检测。

0：4 位地址检测

1：7 位地址检测（在 8 位数据模式下）

只有在禁止 USART (UE = 0) 时才能写入该位

注意：在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。

位 3 **DIS_NSS**：NSS 引脚使能 (NSS pin enable)

当 DIS_NSS 位置 1 时，忽略 NSS 引脚输入。

0：SPI 从器件选择取决于 NSS 输入引脚。

1：始终选择 SPI 从器件，忽略 NSS 输入引脚。

注意：不支持 SPI 从器件模式时，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 2:1 保留，必须保持复位值。

位 0 **SLVEN**：同步从模式使能 (Synchronous Slave mode enable)

SLVEN 位置 1 时，使能同步从模式。

0：禁止从模式。

1：使能从模式。

注意：不支持 SPI 从器件模式时，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

注意：使能发送器时不应对 CPOL、CPHA 和 LBCL 位进行写操作。

24.8.4 USART 控制寄存器 3 (USART_CR3)

USART control register 3

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
TXFTCFG[2:0]			RXF TIE	RXFTCFG[2:0]			TCBG TIE	TXFTIE	WUFIE	WUS[1:0]		SCARCNT[2:0]			Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVR DIS	ONE 位	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HD SEL	IRLP	IREN	EIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 **TXFTCFG[2:0]**：TXFIFO 阈值配置 (TXFIFO threshold configuration)

000：TXFIFO 达到其深度的 1/8
001：TXFIFO 达到其深度的 1/4
010：TXFIFO 达到其深度的 1/2
011：TXFIFO 达到其深度的 3/4
100：TXFIFO 达到其深度的 7/8
101：TXFIFO 变空
其余组合：保留

位 28 **RXFTIE**：RXFIFO 阈值中断使能 (FIFO threshold interrupt enable)

该位由软件置 1 和清零。
0：禁止中断
1：当接收 FIFO 达到 RXFTCFG 中编程的阈值时，生成 USART 中断。

位 27:25 **RXFTCFG[2:0]**：接收 FIFO 阈值配置 (Receive FIFO threshold configuration)

000：接收 FIFO 达到其深度的 1/8
001：接收 FIFO 达到其深度的 1/4
010：接收 FIFO 达到其深度的 1/2
011：接收 FIFO 达到其深度的 3/4
100：接收 FIFO 达到其深度的 7/8
101：接收 FIFO 已满
其余组合：保留

位 24 **TCBGTIE**：保护时间前发送完成中断使能 (Transmission complete before guard time, interrupt enable)

该位由软件置 1 和清零。
0：禁止中断
1：当 USART_ISR 寄存器中的 TCBGT=1 时，生成 USART 中断

注意：如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 23 **TXFTIE**：TXFIFO 阈值中断使能 (TXFIFO threshold interrupt enable)

该位由软件置 1 和清零。
0：禁止中断
1：当 TXFIFO 达到 TXFTCFG 中编程的阈值时，生成 USART 中断。

位 22 **WUFIE**：从低功耗模式唤醒中断使能 (Wake-up from low-power mode interrupt enable)

该位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 WUF = 1 时，生成 USART 中断

注意：WUFIE 必须在进入低功耗模式前置 1。

如果 USART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 21:20 **WUS[1:0]**：从低功耗模式唤醒中断标志选择 (Wake-up from low-power mode interrupt flag selection)

该位域用于指定激活 WUF（从低功耗模式唤醒标志）的事件。

00：WUF 在地址匹配时激活（按 ADD[7:0] 和 ADDM7 所定义）

01：保留。

10：WUF 在起始位检测时激活

11：WUF 在 RXNE/RXFNE 时激活。

只有在禁止 USART (UE = 0) 时才能写入该位域。

如果 USART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 19:17 **SCARCNT[2:0]**：智能卡自动重试计数 (Smartcard auto-retry count)

该位域用于指定智能卡模式下发送和接收的重试次数。

在发送模式下，该位域用于指定生成发送错误（FE 位置 1）前自动重新发送的重试次数。

在接收模式下，该位域用于指定生成接收错误（RXNE/RXFNE 位和 PE 位置 1）前错误接收尝试的次数。

只有在禁止 USART (UE = 0) 时才能编程该位域。

使能 USART (UE = 1) 时，该位域只能写入 0x0，以停止重新发送。

0x0：禁止重新发送——发送模式下不会自动重新发送。

0x1 到 0x7：自动重新发送的尝试次数（发出错误信号前）

注意：如果不支持智能卡模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 16 保留，必须保持复位值。

位 15 **DEP**：驱动器使能极性选择 (Driver enable polarity selection)

0：DE 信号高电平有效。

1：DE 信号低电平有效。

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 14 **DEM**：驱动器使能模式 (Driver enable mode)

该位用于通过 DE 信号激活外部收发器控制。

0：禁止 DE 功能。

1：使能 DE 功能。DE 信号在 RTS 引脚上输出。

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持驱动器使能功能，该位保留且必须保持复位值。第 608 页的第 24.4 节：USART 实现。

位 13 **DDRE**：接收出错时的 DMA 禁止 (DMA Disable on reception error)

0：接收出错时不禁止 DMA。相应的错误标志置 1，但 RXNE 保持为 0 以防止上溢。因此，将不使能 DMA 请求，从而不会传送错误数据（无 DMA 请求），但会传送接收到的下一个正确数据（用于智能卡模式）。

1：接收出错后禁止 DMA。相应的错误标志以及 RXNE 均置 1。屏蔽 DMA 请求，直到错误标志清零。这意味着软件必须首先禁止 DMA 请求 (DMAR = 0) 或者将 RXNE（使能 FIFO 模式时为 RXFNE）清零，然后再将错误标志清零。

只有在禁止 USART (UE=0) 时才能写入该位。

注意：接收错误包括：奇偶校验错误、帧错误或噪声错误。

位 12 **OVRDIS**：上溢禁止 (Overrun disable)

该位用于禁止接收上溢检测。

0：接收新数据前未读取已接收的数据时，上溢错误标志 ORE 置 1。

1：禁止上溢功能。如果在 RXNE 标志仍置 1 时接收到新数据，

则 ORE 标志不会置 1，且新接收的数据会覆盖 USART_RDR 寄存器之前的内容。使能 FIFO 模式时，RXFIFO 将被旁路，数据将直接写入 USART_RDR 寄存器中。即使在使能 FIFO 管理时，也将使用 RXNE 标志。

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：此控制位用于检查通信流而不会读取数据

位 11 **ONEBIT**：一个采样位方法使能 (One sample bit method enable)

该位允许用户选择采样方法。选择一个采样位方法后，将禁止噪声检测标志 (NE)。

0：三个采样位方法

1：一个采样位方法

只有在禁止 USART (UE = 0) 时才能写入该位。

位 10 **CTSIE**：CTS 中断使能 (CTS interrupt enable)

0：禁止中断

1：当 USART_ISR 寄存器中 CTSIF = 1 时，生成中断

注意：如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 9 **CTSE**：CTS 使能 (CTS enable)

0：禁止 CTS 硬件流控制

1：使能 CTS 模式，仅当 CTS 输入无效（连接到 0）时才发送数据。如果在发送数据时使 CTS 输入有效，会在停止之前完成发送。如果使 CTS 有效时数据已写入数据寄存器，则将延迟发送，直到 CTS 无效。

只有在禁止 USART (UE = 0) 时才能写入该位

注意：如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 8 **RTSE**：RTS 使能 (RTS enable)

0：禁止 RTS 硬件流控制

1：使能 RTS 输出，仅当接收缓冲区中有空间时才会请求数据。发送完当前字符后应停止发送数据。可以接收数据时使 RTS 输出无效（拉至 0）。

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 7 **DMAT**：DMA 使能发送器 (DMA enable transmitter)

该位由软件置 1/复位。

1：针对发送使能 DMA 模式

0：针对发送禁止 DMA 模式

位 6 **DMAR**：DMA 使能接收器 (DMA enable receiver)

该位由软件置 1/复位。

1：针对接收使能 DMA 模式

0：针对接收禁止 DMA 模式

位 5 **SCEN**：智能卡模式使能 (Smartcard mode enable)

该位用于使能智能卡模式。

0：禁止智能卡模式

1：使能智能卡模式

只有在禁止 USART (UE = 0) 时才能写入该位域。

注意：如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 4 **NACK**：智能卡 NACK 使能 (Smartcard NACK enable)

0：出现奇偶校验错误时禁止 NACK 发送

1：出现奇偶校验错误时使能 NACK 发送

只有在禁止 USART (UE = 0) 时才能写入该位域。

注意：如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 3 **HDSEL**：半双工选择 (Half-duplex selection)

选择单线半双工模式

0：未选择半双工模式

1：选择半双工模式

只有在禁止 USART (UE = 0) 时才能写入该位。

位 2 **IRLP**：IrDA 低功耗 (IrDA low-power)

该位用于选择正常模式和低功耗 IrDA 模式

0：正常模式

1：低功耗模式

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持 IrDA 模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 1 **IREN**：IrDA 模式使能 (IrDA mode enable)

该位由软件置 1 和清零。

0：禁止 IrDA

1：使能 IrDA

只有在禁止 USART (UE = 0) 时才能写入该位。

注意：如果不支持 IrDA 模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 0 **EIE**：错误中断使能 (Error interrupt enable)

如果出现帧错误、上溢错误、噪声标志或 SPI 从器件下溢错误（USART_ISR 寄存器中的 FE = 1、ORE = 1、NE = 1 或 UDR = 1），则需要使用错误中断使能位来使能中断生成。

0：禁止中断

1：USART_ISR 寄存器中的 FE = 1、ORE = 1、NE = 1 或 UDR = 1（在 SPI 从器件模式下）时，生成中断。

24.8.5 USART 波特率寄存器 (USART_BRR)

USART baud rate register

只有在禁止 USART (UE = 0) 时才能写入此寄存器。在自动波特率检测模式下，该位由硬件自动更新。

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

位 31:16 保留，必须保持复位值。

位 15:0 **BRR[15:0]**：USART 波特率 (USART baud rate)

BRR[15:4]

BRR[15:4] = USARTDIV[15:4]

BRR[3:0]

当 OVER8 = 0 时，BRR[3:0] = USARTDIV[3:0]。

当 OVER8 = 1 时：

BRR[2:0] = USARTDIV[3:0] 右移 1 位。

BRR[3] 必须保持清零。

24.8.6 USART 保护时间和预分频器寄存器 (USART_GTPR)

USART guard time and prescaler register

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

位 31:16 保留，必须保持复位值。

位 15:8 **GT[7:0]**：保护时间值 (Guard time value)

该位域用于编程保护时间值（以波特时钟周期数为单位）。

该位用于智能卡模式。经过此保护时间后，发送完成标志置 1。

只有在禁止 USART (UE = 0) 时才能写入该位域。

注意：如果不支持智能卡模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 7:0 **PSC[7:0]**：预分频器值 (Prescaler value)

在 IrDA 低功耗和正常的 IrDA 模式下：

PSC[7:0] = IrDA 正常和低功耗波特率

PSC[7:0] 用于编程预分频器，进行 USART 源时钟分频以获得低功耗频率：使用寄存器中给出的值（8 个有效位）对源时钟进行分频：

在智能卡模式下：

PSC[4:0] = 预分频器值

PSC[4:0] 用于编程预分频器，进行 USART 源时钟分频以提供智能卡时钟。将寄存器中给出的值（5 个有效位）乘以 2 得出源时钟频率的分频系数：

00000：保留 - 不编程此值
00001：源时钟 1 分频（IrDA 模式）/2 分频（智能卡模式）
00010：源时钟 2 分频（IrDA 模式）/4 分频（智能卡模式）
00011：源时钟 3 分频（IrDA 模式）/6 分频（智能卡模式）
...
11111：源时钟 31 分频（IrDA 模式）/62 分频（智能卡模式）
0010 0000：源时钟 32 分频（IrDA 模式）
...
1111 1111：源时钟 255 分频（IrDA 模式）

只有在禁止 USART (UE = 0) 时才能写入该位域。

注意：如果使用智能卡模式，则位 [7:5] 必须保持清零。
不支持智能卡和 IrDA 模式时，该位域保留并由硬件强制清零。请参见第 608 页的第 24.4 节：USART 实现。

24.8.7 USART 接收器超时寄存器 (USART_RTOR)

USART receiver timeout register

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
BLEN[7:0]								RTO[23:16]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTO[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **BLEN[7:0]**：块长度

该位域用于提供智能卡 T = 1 接收下的块长度。其值等于信息字符数 + 结尾字段的长度 (1-LEC/2-CRC) - 1。

例如：

BLEN = 0：0 个信息字符 + LEC

BLEN = 1：0 个信息字符 + CRC

BLEN = 255：254 个信息字符 + CRC（总共 256 个字符）

在智能卡模式下，块长度计数器在 TXE = 0（使能 FIFO 模式时为 TXFE = 0）时复位。

该位域也可用于其他模式。在这种情况下，块长度计数器在 RE = 0（禁止接收器）和/或 EOBCF 位写入 1 时复位。

注意：块接收开始后可编程此值（使用起始字段中 LEN 字符中的数据）。每个接收到的块只能对此值编程一次。

位 23:0 **RTO[23:0]**：接收器超时值 (Receiver timeout value)

该位域根据 RX 线上没有活动的位数分配接收器超时值。

在标准模式下，如果在接收到最后一个字符后，在 RTO 值对应的时间内未检测到新的起始位，则 RTOF 标志置 1。

在智能卡模式下，此值用于实施 CWT 和 BWT。有关更多详细信息，请参见智能卡章节。在标准模式下，从接收到的最后一个字符的起始位开始执行 CWT/BWT 测量。

注意：每个接收到的字符只能对此值编程一次。

注意：可以实时写入 RTOR。如果新值小于或等于计数器的值，RTOF 标志置 1。

如果不支持接收器超时功能，此寄存器保留并由硬件强制为“0x00000000”。请参见第 608 页的第 24.4 节：USART 实现。

24.8.8 USART 请求寄存器 (USART_RQR)

USART request register

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ
											w	w	w	w	w

位 31:5 保留，必须保持复位值。

位 4 **TXFRQ**：发送数据刷新请求 (Transmit data flush request)

禁止 FIFO 模式时，向该位写入“1”会将 TXE 标志置 1。这可丢弃发送数据。由于错误 (NACK) 而未发送数据且 USART_ISR 寄存器中的 FE 标志有效时，只能在智能卡模式下使用该位。如果 USART 不支持智能卡模式，该位保留且必须保持复位值。

使能 FIFO 时，TXFRQ 位置 1 以清空整个 FIFO。这会将 TXFE 标志（发送 FIFO 为空，USART_ISR 寄存器中的位 23）置 1。在 UART 模式和智能卡模式下都支持清空发送 FIFO。

注意：在 FIFO 模式下，TXFNF 标志在清空请求期间复位，直到 TxFIFO 为空，以确保数据寄存器中没有写入数据。

位 3 **RXFRQ**：接收数据刷新请求 (Receive data flush request)

向该位写入 1 将清空整个接收 FIFO（即，将 RXFNE 位清零）。

这可以丢弃接收的数据而不对其执行读取操作，并避免发生上溢情况。

位 2 **MMRQ**：静默模式请求 (Mute mode request)

向该位写入 1 可将 USART 置于静默模式，并将 RWU 标志复位。

位 1 **SBKRQ**：发送中断请求 (Send break request)

向该位写入 1 可将 SBKF 标志置 1 并在发送设备可用后立即请求在线路上发送 BREAK。

注意：如果应用需要在之前插入的所有数据（包括尚未发送的数据）后发送中断字符，软件应等到 TXE 标志使能后将 SBKRQ 位置 1。

位 0 **ABRRQ**：自动波特率请求 (Auto baud rate request)

向该位写入 1 可复位 USART_ISR 中的 ABRF 和 ABRE 标志，并请求对下一个接收到的数据帧进行自动波特率测量。

注意：如果 USART 不支持自动波特率功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

24.8.9 USART 中断和状态寄存器 (USART_ISR)

USART interrupt and status register

偏移地址：0x1C

复位值：0x0X80 00C0

使能 FIFO/智能卡模式时，X = 2

使能 FIFO 模式且禁止智能卡模式时，X = 0

同一寄存器可用于使能 FIFO 模式（本节）和禁止 FIFO 模式（下一节）的情况。

使能 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXFT	RXFT	TCBGT	RXFF	TXFE	RE ACK	TE ACK	WUF	RWU	SBKF	CMF	BUSY
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXFNF	TC	RXFNE	IDLE	ORE	NE	FE	PE
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27 **TXFT**：TXFIFO 阈值标志 (FIFO threshold flag)

当 TXFIFO 达到在 USART_CR3 寄存器的 TXFTCFG 中编程的阈值时（即 TXFIFO 包含 TXFTCFG 个空位置），该位由硬件置 1。如果 USART_CR3 寄存器中的 TXFTIE 位 = 1（位 31），则会生成中断。

0：TXFIFO 未达到编程的阈值。

1：TXFIFO 已达到编程的阈值。

位 26 **RXFT**：RXFIFO 阈值标志 (RXFIFO threshold flag)

达到在 USART_CR3 寄存器的 RXFTCFG 中编程的阈值时，该位由硬件置 1。这意味着，接收 FIFO 中有 (RXFTCFG - 1) 个数据，USART_RDR 寄存器中有一个数据。如果 USART_CR3 寄存器中的 RXFTIE 位 = 1（位 27），则会生成中断。

0：接收 FIFO 未达到编程的阈值。

1：接收 FIFO 已达到编程的阈值。

注意：当 RXFTCFG 阈值配置为“101”时，如果存在 16 个数据（即 RXFIFO 中有 15 个数据，USART_RDR 中有 1 个数据），则 RXFT 标志将置 1。因此，接收到的第 17 个数据不会导致上溢错误。接收到第 18 个数据后会发生上溢错误。

位 25 **TCBGT**：保护时间前发送完成标志 (Transmission complete before guard time flag)

当写入到 USART_TDR 中的最后一个数据已正确从移位寄存器中发出时，该位置 1。

如果包含数据的帧完成发送，并且智能卡未发回任何 NACK，则该位在智能卡模式下由硬件置 1。如果 USART_CR3 寄存器中的 TCBGTIE = 1，则会生成中断。

该位由软件清零，方法是向 USART_ICR 寄存器中的 TCBGTCF 写入 1 或向 USART_TDR 寄存器执行写操作。

0：发送未完成或发送未成功完成（即，从智能卡接收到 NACK）

1：发送成功完成（在保护时间结束之前完成，智能卡未发送 NACK）。

注意：如果 USART 不支持智能卡模式，该位保留且保持复位值。如果 USART 支持智能卡模式并使能智能卡模式，则 TCBGT 复位值为“1”。请参见第 608 页的第 24.4 节：USART 实现。

位 24 **RXFF**：RxFIFO 已满 (RXFIFO full)

当接收到的数据量对应于 RXFIFO 大小 + 1（RXFIFO 已满 + USART_RDR 寄存器中的 1 个数据）时，该位由硬件置 1。

如果 USART_CR1 寄存器中 RXFFIE 位 = 1，则会生成中断。

0：RXFIFO 未满。

1：RxFIFO 已满。

位 23 **TXFE**：TXFIFO 为空 (TXFIFO empty)

当 TXFIFO 为空时，该位由硬件置 1。当 TXFIFO 包含至少一个数据时，该标志被清零。也可以通过向 USART_RQR 寄存器中的位 TXFRQ（位 4）写入 1 将 TXFE 标志置 1。

如果 USART_CR1 寄存器中的 TXFEIE 位 = 1（位 30），则会生成中断。

0：TXFIFO 非空。

1：TXFIFO 为空。

位 22 **REACK**：接收使能确认标志 (Receive enable acknowledge flag)

USART 采用接收使能值时，通过硬件将该位置 1/复位。

该位可用于验证 USART 是否准备好在进入低功耗模式前接收数据。

注意：如果 USART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 21 **TEACK**：发送使能确认标志 (Transmit enable acknowledge flag)

USART 采用发送使能值时，通过硬件将该位置 1/复位。

通过写入 TE = 0 生成空闲帧请求，然后在 USART_CR1 寄存器中写入 TE = 1 以遵循 TE = 0 最短周期时，可使用该位。

位 20 **WUF**：从低功耗模式唤醒标志 (Wake-up from low-power mode flag)

当检测到唤醒事件时，该位由硬件置 1。事件通过 WUS 位域定义。该位由软件清零，方法是向 USART_ICR 寄存器中的 WUCF 写入 1。

如果 USART_CR3 寄存器中 WUFIE = 1，则会生成中断。

注意：当 UESM 清零时，WUF 标志也清零。

如果 USART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 19 **RWU**：接收器从静默模式唤醒 (Receiver wake-up from Mute mode)

该位指示 USART 是否处于静默模式。当识别出唤醒/静默序列时，该位由硬件清零/置 1。静默模式控制序列（地址或 IDLE）通过 USART_CR1 寄存器中的 WAKE 位进行选择。

当选择 IDLE 模式下唤醒时，该位只能通过用软件向 USART_RQR 寄存器中的 MMRQ 位写 1 的方式置 1。

0：接收器处于活动模式

1：接收器处于静默模式

注意：如果 USART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 18 **SBKF**：发送中断标志 (Send break flag)

该位指示已请求发送中断字符。通过将 1 写入 USART_CR3 寄存器中的 SBKRQ 位，该位由软件置 1。该位在中断发送的停止位期间由硬件自动复位。

0：中断字符已发送

1：通过将 USART_RQR 寄存器中的 SBKRQ 位置 1 请求中断字符

位 17 **CMF**：字符匹配标志 (Character match flag)

接收到由 ADD[7:0] 定义的字符后由硬件将该位置 1。通过向 USART_ICR 寄存器中的 CMCF 写入 1，该位由软件清零。

如果 USART_CR1 寄存器中 CMIE = 1，则会生成中断。

0：未检测到字符匹配

1：检测到字符匹配

位 16 **BUSY**：忙标志 (Busy flag)

该位由硬件置 1 和复位。当 RX 线路上正在进行通信（成功检测到起始位）时有效。在接收结束（成功或失败）时复位。

0：USART 处于空闲状态（无接收）

1：正在接收

位 15 **ABRF**：自动波特率标志 (Auto baud rate flag)

已设置自动波特率（RXFNE 也置 1，并在 RXFNEIE = 1 时生成中断），或者自动波特率操作未成功完成时，该位由硬件置 1 (ABRE = 1)（此时，ABRE、RXFNE 和 FE 也置 1）

为请求新的自动波特率检测，通过向 USART_RQR 寄存器中的 ABRRQ 写入 1，该位由软件清零。

注意：如果 USART 不支持自动波特率功能，该位保留且保持复位值。

位 14 **ABRE**：自动波特率错误 (Auto baud rate error)

如果波特率测量失败（波特率超出范围或字符比较失败），该位由硬件置 1。

通过将 1 写入 USART_RQR 寄存器中的 ABRRQ 位，该位由软件清零。

注意：如果 USART 不支持自动波特率功能，该位保留且保持复位值。

位 13 **UDR**：SPI 从器件下溢错误标志 (SPI slave underrun error flag)

在从发送模式下，如果在软件尚未将任何值加载到 USART_TDR 之前出现第一个数据发送时钟脉冲，此标志将置 1。通过将 USART_ICR 寄存器中的 UDRCF 位置 1 来复位此标志。

0：无下溢错误

1：存在下溢错误

注意：如果 USART 不支持 SPI 从器件模式，该位保留且保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 12 **EOBF**：块结束标志 (End of block flag)

接收到完整块后，该位由硬件置 1（例如 T = 1 智能卡模式）。接收到的字节数（从块的起始处，包括起始字段）等于或大于 BLEN + 4 时执行检测。

如果 USART_CR1 寄存器中 EOBIE = 1，则会生成中断。

通过向 USART_ICR 寄存器中的 EOBCF 写入 1，该位由软件清零。

0：未达到块结束

1：已达到块结束（字符数）

注意：如果不支持智能卡模式，该位保留且保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 11 **RTOF**：接收器超时

已经过在 RTOR 寄存器中编程的超时值后，如果无任何通信，该位由硬件置 1。通过向 USART_ICR 寄存器中的 RTOCF 写入 1，该位由软件清零。

如果 USART_CR2 寄存器中 RTOIE = 1，则会生成中断。

在智能卡模式下，该超时对应于 CWT 或 BWT 时间。

0：未达到超时值

1：已达到超时值，未接收到任何数据

注意：如果 RTOR 寄存器中编程的时间值将 2 个字符隔开，则 RTOF 不置 1。如果此时间大于该值 + 2 个采样时间（2/16 或 2/8，具体取决于过采样方法），则 RTOF 标志置 1。

即使 RE = 0，计数器仍会计数，但 RTOF 仅在 RE = 1 时置 1。如果 RE 置 1 时已经超时，则 RTOF 会置 1。

如果 USART 不支持接收器超时功能，该位保留且保持复位值。

位 10 **CTS**：CTS 标志 (CTS flag)

该位由硬件置 1/复位。该位是对 CTS 输入引脚的状态取反。

0：CTS 线置 1

1：CTS 线复位

注意：如果不支持硬件流控制功能，该位保留且保持复位值。

位 9 **CTSIF**：CTS 中断标志 (CTS interrupt flag)

如果 CTSE 位置 1，当 CTS 输入切换时，该位由硬件置 1。通过将 1 写入 USART_ICR 寄存器中的 CTSCF 位，该位由软件清零。

如果 USART_CR3 寄存器中 CTSIE = 1，则会生成中断。

0：CTS 状态线上未发生变化

1：CTS 状态线上发生变化

注意：如果不支持硬件流控制功能，该位保留且保持复位值。

位 8 **LBDF**：LIN 中断检测标志 (LIN break detection flag)

检测到 LIN 中断时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 LBDCF 写入 1，该位由软件清零。

如果 USART_CR2 寄存器中 LBDIE = 1，则会生成中断。

0：未检测到 LIN 中断

1：检测到 LIN 中断

注意：如果 USART 不支持 LIN 模式，该位保留且保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 7 **TXFNF**：TXFIFO 未满 (TXFIFO not full)

TXFNF 会在 TXFIFO 未满时由硬件置 1，表示可向 USART_TDR 中写入数据。每次对 USART_TDR 进行写操作都会将数据置于 TXFIFO 中。该标志会保持置 1，直到 TXFIFO 已满。当 TXFIFO 已满时，该标志清零，表示不能向 USART_TDR 中写入数据。

如果 USART_CR1 寄存器中 TXFNFIE 位 = 1，则会生成中断。

0：发送 FIFO 已满

1：发送 FIFO 未满

注意：在清空请求期间，TXFNF 保持复位，直到 TXFIFO 为空。发送清空请求（通过将 TXFRQ 位置 1）后，应先检查 TXFNF 标志，然后再写入 TXFIFO（TXFNF 和 TXFE 会同时置 1）。

单缓冲区发送期间使用该位。

位 6 **TC**：发送完成 (Transmission complete)

该位指示写入到 USART_TDR 中的最后一个数据已从移位寄存器中发出。

如果已完成对包含数据的帧的发送并且 TXFE 置 1，则该位由硬件置 1。

如果 USART_CR1 寄存器中 TCIE = 1，则会生成中断。

TC 位由软件清零，方法是向 USART_ICR 寄存器中的 TCCF 写入 1 或向 USART_TDR 寄存器执行写操作。

0：传送未完成

1：传送已完成

注意：如果 TE 位复位且无任何发送正在进行，TC 位会立即置 1。

位 5 **RXFNE**：RXFIFO 非空 (RXFIFO not empty)

RXFIFO 非空时，RXFNE 位由硬件置 1，这表示可以从 USART_RDR 寄存器读取数据。每次对 USART_RDR 进行读操作都会在 RXFIFO 中释放一个位置。

RXFNE 在 RXFIFO 为空时清零。也可以通过向 USART_RQR 寄存器中的 RXFRQ 位写入 1 将 RXFNE 标志位清零。

如果 USART_CR1 寄存器中 RXFNEIE = 1，则会生成中断。

0：未接收到数据

1：已准备好读取接收到的数据

位 4 **IDLE**：检测到空闲线路

检测到空闲线路时，该位由硬件置 1。如果 USART_CR1 寄存器中 IDLEIE = 1，则会生成中断。通过向 USART_ICR 寄存器中的 IDLECF 写入 1，该位由软件清零。

0：未检测到空闲线路

1：检测到空闲线路

注意：直到 RXFNE 位已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1。

使能静默模式 (MME = 1) 后，如果 USART 未静默 (RWU = 0)，则 IDLE 置 1，无论是否通过 WAKE 位选择了静默模式。如果 RWU = 1，IDLE 不置 1。

位 3 **ORE**：上溢错误

在 RXFF = 1 的情况下，当移位寄存器中当前正在接收的数据

准备好传输到 USART_RDR 寄存器时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 ORECF 写入 1，该位由软件清零。

如果 USART_CR1 寄存器中 RXFNEIE = 1 或 USART_CR3 寄存器中 EIE = 1，则会生成中断。

0：无上溢错误

1：检测到上溢错误

注意：当该位置 1 时，USART_RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。EIE 位置 1 后，如果在多缓冲区通信中 ORE 标志置 1，则会生成中断。

USART_CR3 寄存器中的 OVRDIS 位置 1 时，该位将被永久强制清零（无上溢检测）。

位 2 **NE**：噪声检测标志 (Noise detection flag)

当在接收的帧上检测到噪声时，该位由硬件置 1。该位由软件清零，方法是向 USART_ICR 寄存器中的 NECF 位写入 1。

0：未检测到噪声

1：检测到噪声

注意：该位不会生成中断，因为该位出现的时间与本身生成中断的 RXFNE 位出现的时间相同。EIE 位置 1 后，如果在多缓冲区通信中 NE 标志置 1，则会生成中断。

当线路无噪声时，可以通过将 ONEBIT 位编程为 1 提高 USART 对偏差的容差来禁止 NE 标志（请参见第 624 页的第 24.5.8 节：USART 接收器对时钟偏差的容差）。

此错误与 USART_RDR 中的字符相关联。

位 1 **FE**：帧错误 (Framing error)

当检测到同步丢失、过度的噪声或中断字符时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 FECF 写入 1，该位由软件清零。

在智能卡模式下发送数据时，如果在达到最大发送尝试次数后仍未成功（智能卡向数据帧发送 NACK 信号），则该位置 1。

如果 USART_CR3 寄存器中 EIE = 1，则会生成中断。

0：未检测到帧错误

1：检测到帧错误或中断字符

注意：此错误与 USART_RDR 中的字符相关联。

位 0 **PE**：奇偶校验错误 (Parity error)

当在接收器模式下发生奇偶校验错误时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 PECF 写入 1，该位由软件清零。

如果 USART_CR1 寄存器中的 PEIE = 1，则会生成中断。

0：无奇偶校验错误

1：奇偶校验错误

注意：此错误与 USART_RDR 中的字符相关联。

24.8.10 USART 中断和状态寄存器 [备用] (USART_ISR)

USART interrupt and status register [alternate]

偏移地址：0x1C

复位值：0x0000 00C0

同一寄存器可用于使能 FIFO 模式（上一节）和禁止 FIFO 模式（本节）的情况。

禁止 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	TCBGT	Res.	Res.	RE ACK	TE ACK	WUF	RWU	SBKF	CMF	BUSY
						r			r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:26 保留，必须保持复位值。

位 25 **TCBGT**：保护时间前发送完成标志 (Transmission complete before guard time flag)

当写入到 USART_TDR 中的最后一个数据已正确从移位寄存器中发出时，该位置 1。

如果包含数据的帧完成发送，并且智能卡未发回任何 NACK，则该位在智能卡模式下由硬件置 1。如果 USART_CR3 寄存器中的 TCBGTIE = 1，则会生成中断。

该位由软件清零，方法是向 USART_ICR 寄存器中的 TCBGTCF 写入 1 或向 USART_TDR 寄存器执行写操作。

0：发送未完成或发送未成功完成（即，从智能卡接收到 NACK）

1：发送成功完成（在保护时间结束之前完成，智能卡未发送 NACK）。

注意：如果 USART 不支持智能卡模式，该位保留且保持复位值。如果 USART 支持智能卡模式并使能智能卡模式，则 TCBGT 复位值为“1”。请参见第 608 页的第 24.4 节：USART 实现。

位 24:23 保留，必须保持复位值。

位 22 **REACK**：接收使能确认标志 (Receive enable acknowledge flag)

USART 采用接收使能值时，通过硬件将该位置 1/复位。

该位可用于验证 USART 是否准备好在进入低功耗模式前接收数据。

注意：如果 USART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 21 **TEACK**：发送使能确认标志 (Transmit enable acknowledge flag)

USART 采用发送使能值时，通过硬件将该位置 1/复位。

通过写入 TE = 0 生成空闲帧请求，然后在 USART_CR1 寄存器中写入 TE = 1 以遵循 TE = 0 最短周期时，可使用该位。

位 20 **WUF**：从低功耗模式唤醒标志 (Wake-up from low-power mode flag)

当检测到唤醒事件时，该位由硬件置 1。事件通过 WUS 位域定义。该位由软件清零，方法是向 USART_ICR 寄存器中的 WUCF 写入 1。

如果 USART_CR3 寄存器中 WUFIE = 1，则会生成中断。

注意：当 UESM 清零时，WUF 标志也清零。

如果 USART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 19 **RWU**：*接收器从静默模式唤醒 (Receiver wake-up from Mute mode)*

该位指示 USART 是否处于静默模式。当识别出唤醒/静默序列时，该位由硬件清零/置 1。静默模式控制序列（地址或 IDLE）通过 USART_CR1 寄存器中的 WAKE 位进行选择。

当选择 IDLE 模式下唤醒时，该位只能通过用软件向 USART_RQR 寄存器中的 MMRQ 位写 1 的方式置 1。

0：接收器处于活动模式

1：接收器处于静默模式

注意：如果 USART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 18 **SBKF**：发送中断标志 (Send break flag)

该位指示已请求发送中断字符。通过将 1 写入 USART_CR3 寄存器中的 SBKRQ 位，该位由软件置 1。该位在中断发送的停止位期间由硬件自动复位。

0：发送中断字符

1：通过将 USART_RQR 寄存器中的 SBKRQ 位置 1 请求中断字符

位 17 **CMF**：字符匹配标志 (Character match flag)

接收到由 ADD[7:0] 定义的字符后由硬件将该位置 1。通过向 USART_ICR 寄存器中的 CMCF 写入 1，该位由软件清零。

如果 USART_CR1 寄存器中 CMIE = 1，则会生成中断。

0：未检测到字符匹配

1：检测到字符匹配

位 16 **BUSY**：忙标志 (Busy flag)

该位由硬件置 1 和复位。当 RX 线路上正在进行通信（成功检测到起始位）时有效。在接收结束（成功或失败）时复位。

0：USART 处于空闲状态（无接收）

1：正在接收

位 15 **ABRF**：自动波特率标志 (Auto baud rate flag)

已设置自动波特率（RXNE 也置 1，在 RXNEIE = 1 时生成中断），或者自动波特率操作未成功完成时，该位由硬件置 1 (ABRE = 1)（此时，ABRE、RXNE 和 FE 也置 1）

为请求新的自动波特率检测，通过向 USART_RQR 寄存器中的 ABRRQ 写入 1，该位由软件清零。

注意：如果 USART 不支持自动波特率功能，该位保留且保持复位值。

位 14 **ABRE**：自动波特率错误 (Auto baud rate error)

如果波特率测量失败（波特率超出范围或字符比较失败），该位由硬件置 1。

通过将 1 写入 USART_RQR 寄存器中的 ABRRQ 位，该位由软件清零。

注意：如果 USART 不支持自动波特率功能，该位保留且保持复位值。

位 13 **UDR**：SPI 从器件下溢错误标志 (SPI slave underrun error flag)

在从发送模式下，如果在软件尚未将任何值加载到 USART_TDR 之前出现第一个数据发送时钟脉冲，此标志将置 1。通过将 USART_ICR 寄存器中的 UDRCF 位置 1 来复位此标志。

0：无下溢错误

1：存在下溢错误

注意：如果 USART 不支持 SPI 从器件模式，该位保留且保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 12 **EOBF**：块结束标志 (End of block flag)

接收到完整块后，该位由硬件置 1（例如 T = 1 智能卡模式）。接收到的字节数（从块的起始处，包括起始字段）等于或大于 BLEN + 4 时执行检测。

如果 USART_CR1 寄存器中 EOBIE = 1，则会生成中断。

通过向 USART_ICR 寄存器中的 EOBCF 写入 1，该位由软件清零。

0：未达到块结束

1：已达到块结束（字符数）

注意：如果不支持智能卡模式，该位保留且保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 11 **RTOF**：接收器超时

已经过在 RTOR 寄存器中编程的超时值后，如果无任何通信，该位由硬件置 1。通过向 USART_ICR 寄存器中的 RTOCF 写入 1，该位由软件清零。

如果 USART_CR2 寄存器中 RTOIE = 1，则会生成中断。

在智能卡模式下，该超时对应于 CWT 或 BWT 时间。

0：未达到超时值

1：已达到超时值，未接收到任何数据

注意：如果 RTOR 寄存器中编程的时间值将 2 个字符隔开，则 RTOF 不置 1。如果此时间大于该值 + 2 个采样时间（2/16 或 2/8，具体取决于过采样方法），则 RTOF 标志置 1。

即使 RE = 0，计数器仍会计数，但 RTOF 仅在 RE = 1 时置 1。如果 RE 置 1 时已经超时，则 RTOF 会置 1。

如果 USART 不支持接收器超时功能，该位保留且保持复位值。

位 10 **CTS**：CTS 标志 (CTS flag)

该位由硬件置 1/复位。该位是对 CTS 输入引脚的状态取反。

0：CTS 线置 1

1：CTS 线复位

注意：如果不支持硬件流控制功能，该位保留且保持复位值。

位 9 **CTSIF**：CTS 中断标志 (CTS interrupt flag)

如果 CTSE 位置 1，当 CTS 输入切换时，该位由硬件置 1。通过将 1 写入 USART_ICR 寄存器中的 CTSCF 位，该位由软件清零。

如果 USART_CR3 寄存器中 CTSIE = 1，则会生成中断。

0：CTS 状态线上未发生变化

1：CTS 状态线上发生变化

注意：如果不支持硬件流控制功能，该位保留且保持复位值。

位 8 **LBDF**：LIN 中断检测标志 (LIN break detection flag)

检测到 LIN 中断时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 LBDCF 写入 1，该位由软件清零。

如果 USART_CR2 寄存器中 LBDIE = 1，则会生成中断。

0：未检测到 LIN 中断

1：检测到 LIN 中断

注意：如果 USART 不支持 LIN 模式，该位保留且保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 7 **TXE**：发送数据寄存器为空

当 USART_TDR 寄存器的内容已传输到移位寄存器时，TXE 由硬件置 1。通过对 USART_TDR 寄存器执行写操作将该位清零。为丢弃数据（仅在智能卡 T = 0 模式下出现发送故障时），也可以通过向 USART_RQR 寄存器中的 TXFRQ 写入 1 来将 TXE 标志置 1。

如果 USART_CR1 寄存器中 TXEIE 位 = 1，则会生成中断。

0：数据寄存器已满

1：数据寄存器未满

位 6 **TC**：发送完成 (Transmission complete)

该位指示写入到 USART_TDR 中的最后一个数据已从移位寄存器中发出。

如果已完成对包含数据的帧的发送并且 TXE 置 1，则该位由硬件置 1。

如果 USART_CR1 寄存器中 TCIE = 1，则会生成中断。

TC 位由软件清零，方法是向 USART_ICR 寄存器中的 TCCF 写入 1 或向 USART_TDR 寄存器执行写操作。

0：传送未完成

1：传送已完成

注意：如果 TE 位复位且无任何发送正在进行，TC 位会立即置 1。

位 5 **RXNE**：读取数据寄存器不为空 (Read data register not empty)

当 USART_RDR 移位寄存器的内容已传输到 USART_RDR 寄存器时，RXNE 位由硬件置 1。通过对 USART_RDR 寄存器执行读取操作将该位清零。也可以通过将 USART_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE 标志位清零。

如果 USART_CR1 寄存器中 RXNEIE = 1，则会生成中断。

0：未接收到数据

1：已准备好读取接收到的数据

位 4 **IDLE**：检测到空闲线路

检测到空闲线路时，该位由硬件置 1。如果 USART_CR1 寄存器中 IDLEIE = 1，则会生成中断。通过向 USART_ICR 寄存器中的 IDLECF 写入 1，该位由软件清零。

0：未检测到空闲线路

1：检测到空闲线路

注意：直到 RXNE 位已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1。

使能静默模式 (MME = 1) 后，如果 USART 未静默 (RWU = 0)，则 IDLE 置 1，无论是否通过 WAKE 位选择了静默模式。如果 RWU = 1，IDLE 不置 1。

位 3 **ORE**：上溢错误

在 RXNE = 1 的情况下，当移位寄存器中当前正在接收的数据

准备好传输到 USART_RDR 寄存器时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 ORECF 写入 1，该位由软件清零。

如果 LPUART_CR1 寄存器中的 RXNEIE = 1 或 EIE = 1，或者 USART_CR3 寄存器中的 EIE = 1，则会生成中断。

0：无上溢错误

1：检测到上溢错误

注意：当该位置 1 时，USART_RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。EIE 位置 1 后，如果在多缓冲区通信中 ORE 标志置 1，则会生成中断。

USART_CR3 寄存器中的 OVRDIS 位置 1 时，该位将被永久强制清零（无上溢检测）。

位 2 **NE**：噪声检测标志 (Noise detection flag)

当在接收的帧上检测到噪声时，该位由硬件置 1。该位由软件清零，方法是向 USART_ICR 寄存器中的 NECF 位写入 1。

0：未检测到噪声

1：检测到噪声

注意：该位不会生成中断，因为该位出现的时间与本身生成中断的 RXNE 位出现的时间相同。EIE 位置 1 后，如果在多缓冲区通信中 NE 标志置 1，则会生成中断。

当线路无噪声时，可以通过将 ONEBIT 位编程为 1 提高 USART 对偏差的容差来禁止 NE 标志（请参见第 624 页的第 24.5.8 节：USART 接收器对时钟偏差的容差）。

位 1 **FE**：帧错误 (Framing error)

当检测到去同步化、过度的噪声或中断字符时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 FECF 写入 1，该位由软件清零。

在智能卡模式下发送数据时，如果在达到最大发送尝试次数后仍未成功（智能卡向数据帧发送 NACK 信号），则该位置 1。

如果 USART_CR3 寄存器中 EIE = 1，则会生成中断。

0：未检测到帧错误

1：检测到帧错误或中断字符

位 0 **PE**：奇偶校验错误 (Parity error)

当在接收器模式下发生奇偶校验错误时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 PECF 写入 1，该位由软件清零。

如果 USART_CR1 寄存器中的 PEIE = 1，则会生成中断。

0：无奇偶校验错误

1：奇偶校验错误

24.8.11 USART 中断标志清零寄存器 (USART_ICR)

USART interrupt flag clear register

偏移地址：0x20

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	UDRCF	EOBCF	RTOCF	Res.	CTSCF	LBDCF	TCBGTCF	TCCF	TXFECF	IDLECF	ORECF	NECF	FECF	PECF
		w	w	w		w	w	w	w	w	w	w	w	w	w

位 31:21 保留，必须保持复位值。

位 20 **WUCF**：从低功耗模式唤醒清零标志 (Wake-up from low-power mode clear flag)

将 1 写入该位时，USART_ISR 寄存器中 WUF 标志将清零。

注意：如果 USART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 19:18 保留，必须保持复位值。

位 17 **CMCF**：字符匹配清零标志 (Character match clear flag)

将 1 写入该位时，USART_ISR 寄存器中 CMF 标志将清零。

位 16:14 保留，必须保持复位值。

位 13 **UDRCF**：SPI 从器件下溢清零标志 (SPI slave underrun clear flag)

将 1 写入该位时，USART_ISR 寄存器中 UDRF 标志将清零。

注意：如果 USART 不支持 SPI 从器件模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 12 **EOBCF**：块结束清零标志 (End of block clear flag)

将 1 写入该位时，USART_ISR 寄存器中 EOBF 标志将清零。

注意：如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 11 **RTOCF**：接收器超时清零标志 (Receiver timeout clear flag)
将 1 写入该位时，USART_ISR 寄存器中 RTOF 标志将清零。
注意：如果 USART 不支持接收器超时功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 10 保留，必须保持复位值。

位 9 **CTSCF**：CTS 清零标志 (CTS clear flag)
将 1 写入该位时，USART_ISR 寄存器中 CTSIF 标志将清零。
注意：如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 8 **LBDCF**：LIN 中断检测清零标志 (LIN break detection clear flag)
将 1 写入该位时，USART_ISR 寄存器中 LBDF 标志将清零。
注意：如果不支持 LIN 模式，该位保留且必须保持复位值。请参见第 608 页的第 24.4 节：USART 实现。

位 7 **TCBGTCF**：保护时间前发送完成清零标志 (Transmission complete before Guard time clear flag)
将 1 写入该位时，USART_ISR 寄存器中 TCBGT 标志将清零。

位 6 **TCCF**：发送完成清零标志 (Transmission complete clear flag)
将 1 写入该位时，USART_ISR 寄存器中 TC 标志将清零。

位 5 **TXFECF**：TXFIFO 为空清零标志 (TXFIFO empty clear flag)
将 1 写入该位时，USART_ISR 寄存器中 TXFE 标志将清零。

位 4 **IDLECF**：检测到空闲线路清零标志 (Idle line detected clear flag)
将 1 写入该位时，USART_ISR 寄存器中 IDLE 标志将清零。

位 3 **ORECF**：上溢错误清零标志 (Overrun error clear flag)
将 1 写入该位时，USART_ISR 寄存器中 ORE 标志将清零。

位 2 **NECF**：检测到噪声清零标志 (Noise detected clear flag)
将 1 写入该位时，USART_ISR 寄存器中 NE 标志将清零。

位 1 **FECF**：帧错误清零标志 (Framing error clear flag)
将 1 写入该位时，USART_ISR 寄存器中 FE 标志将清零。

位 0 **PECF**：奇偶校验错误清零标志 (Parity error clear flag)
将 1 写入该位时，USART_ISR 寄存器中 PE 标志将清零。

24.8.12 USART 接收数据寄存器 (USART_RDR)

USART receive data register

偏移地址：0x24

复位值：0x0000 0000

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

位 31:9 保留，必须保持复位值。

位 8:0 **RDR[8:0]**：接收数据值 (Receive data value)

包含接收到的数据字符。

RDR 寄存器在输入移位寄存器和内部总线之间提供了并行接口（请参见*图 241*）。

在使能奇偶校验位的情况下进行接收时，从 MSB 位中读取的值为接收到的奇偶校验位。

24.8.13 USART 发送数据寄存器 (USART_TDR)

USART transmit data register

偏移地址：0x28

复位值：0x0000 0000

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

位 31:9 保留，必须保持复位值。

位 8:0 **TDR[8:0]**：发送数据值 (Transmit data value)

包含要发送的数据字符。

USART_TDR 寄存器在内部总线和输出移位寄存器之间提供了并行接口（请参见*图 241*）。

在使能奇偶校验位的情况下（USART_CR1 寄存器中的 PCE 位被置 1）进行发送时，由于 MSB 的写入值（位 7 或位 8，具体取决于数据长度）会被奇偶校验位所取代，因此该值不起任何作用。

注意：只能在 TXE/TXFNF = 1 时写入此寄存器。

24.8.14 USART 预分频器寄存器 (USART_PRESC)

USART prescaler register

只有在禁止 USART (UE = 0) 时才能写入此寄存器。

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

位 31:4 保留，必须保持复位值。

位 3:0 **PRESCALER[3:0]**：时钟预分频器

USART 输入时钟可通过预分频系数进行分频：

0000：输入时钟未分频
0001：输入时钟 2 分频
0010：输入时钟 4 分频
0011：输入时钟 6 分频
0100：输入时钟 8 分频
0101：输入时钟 10 分频
0110：输入时钟 12 分频
0111：输入时钟 16 分频
1000：输入时钟 32 分频
1001：输入时钟 64 分频
1010：输入时钟 128 分频
1011：输入时钟 256 分频
其余组合：保留

注意：为 PRESCALER 编程不允许的值时，编程的预分频值为 1011，即输入时钟除以 256。

24.8.15 USART 寄存器映射

下表提供了 USART 寄存器映射和复位值。

表 113. USART 寄存器映射和复位值

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	USART_CR1 FIFO enabled	RXFFIE	TXFEIE	FIFOEN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]					OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXFNFIE	TCIE	RXFNEIE	IDLEIE	TE	RE	UESM	UE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00	USART_CR1 FIFO disabled	Res.	Res.	FIFOEN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]					OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	USART_CR2	ADD[7:0]								RTOEN	ABRMOD[1:0]		ABREN	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	LINEN	STOP [1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	DIS_NSS	Res.	Res.	SLVEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0			0
0x08	USART_CR3	TXFTCFG[2:0]			RXFTIE	RXFTCFG[2:0]			TCBGTIE	TXFTIE	WUFIE	WUS [1:0]	SCAR CNT [2:0]				Res.	DEP	DEM	DDRE	OVRDIS	ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	USART_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	USART_GTPR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	GT[7:0]								PSC[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	USART_RTOR	BLEN[7:0]								RTO[23:0]																							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	USART_RQR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ
	Reset value																												0	0	0	0	0
0x1C	USART_ISR FIFO mode enabled	Res.	Res.	Res.	Res.	TXFT	RXFT	TCBGT	RXFF	TXFE	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXFNF	TC	RXFNE	IDLE	ORE	NE	FE	PE
	Reset value					X	X	X	X	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
0x1C	USART_ISR FIFO mode disabled	Res.	Res.	Res.	Res.	Res.	Res.	TCBGT	Res.	Res.	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
	Reset value							0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
0x20	USART_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.	Res.	Res.	UDRCF	EOBCF	RTOCF	Res.	CTSCF	LBDCF	TCBGTCF	TCCF	TXFECF	IDLECF	ORECF	NECF	FECF	PECF
	Reset value												0			0				0	0	0		0	0	0	0	0	0	0	0	0	0
0x24	USART_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0

表 113. USART 寄存器映射和复位值（续）

偏移	寄存器名称 复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x28	USART_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x2C	USART_ PRESC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3:0]			
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 2.2 节：存储器结构*。

25 串行外设接口/集成电路内置音频总线 (SPI/I2S)

25.1 简介

SPI/I²S 接口可用于使用 SPI 协议或 I^2S 音频协议与外部器件进行通信。SPI 或 I^2S 模式可通过软件进行选择。器件复位后默认选择 SPI Motorola 模式。

串行外设接口 (SPI) 协议支持与外部器件进行半双工、全双工和单工同步串行通信。该接口可配置为主模式，在这种情况下，它可为外部从器件提供通信时钟 (SCK)。该接口还能够在多主模式配置下工作。

集成电路内置音频总线 (I^2S) 也是同步串行通信接口。它可以在从模式或主模式下进行半双工通信。它可满足四种不同音频标准的要求，包括 Philips I^2S 标准、MSB 和 LSB 对齐标准以及 PCM 标准。

25.2 SPI 主要特性

- 主模式或从模式操作
- 基于三条线的全双工同步传输
- 基于双线的半双工同步传输，其中一条可作为双向数据线
- 基于双线的单工同步传输，其中一条可作为单向数据线
- 4 位到 16 位数据位宽大小选择
- 多主模式功能
- 8 个主模式波特率预分频器，可达 $f_{PCLK}/2$。
- 从模式频率可达 $f_{PCLK}/2$。
- 对于主模式和从模式都可通过硬件或软件进行 NSS 管理：动态切换主/从操作
- 可编程的时钟极性和相位
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 可触发中断的专用发送和接收标志
- SPI 总线忙状态标志
- 支持 SPI Motorola 模式
- 用于确保可靠通信的硬件 CRC 功能：
  - 在发送模式下可将 CRC 值作为最后一个字节发送
  - 根据收到的最后一个字节自动进行 CRC 错误校验
- 可触发中断的主模式故障和上溢标志
- CRC 错误标志
- 具有 DMA 功能的两个 32 位内置 Rx 和 Tx FIFO
- 增强型 TI 和 NSS 脉冲模式支持

25.3 I2S 主要特性

- 半双工通信（仅作为发送器或接收器）
- 主模式或从模式操作
- 8 位可编程线性预分频器，可实现精确的音频采样频率（从 8 kHz 到 192 kHz）
- 数据格式可以是 16 位、24 位或 32 位
- 数据包帧由音频通道固定为 16 位（可容纳 16 位数据帧）或 32 位（可容纳 16 位、24 位、32 位数据帧）
- 可编程的时钟极性（就绪时的电平状态）
- 从发送模式下的下溢标志、接收模式下的上溢标志（主模式和从模式），以及接收和发送模式下的帧错误标志（仅从模式）
- 发送和接收使用同一个 16 位数据寄存器
- 支持的 I^2S 协议：
  - I^2S Philips 标准
  - MSB 对齐标准（左对齐）
  - LSB 对齐标准（右对齐）
  - PCM 标准（在 16 位通道帧或扩展为 32 位通道帧的 16 位数据帧上进行短帧和长帧同步）
- 数据方向始终为 MSB 在前
- 用于发送和接收的 DMA 功能（16 位宽）
- 可输出主时钟以驱动外部音频元件。其频率在所有 I2S 模式下固定为 256 × f_s，而在所有 PCM 模式下则固定为 128 x f_s（其中，f_s 为音频信号采样频率）。

25.4 SPI/I2S 实现

下表列出了器件内置的所有 SPI 外设的特性。

表 114. STM32C0x1 SPI 和 SPI/I2S 实现

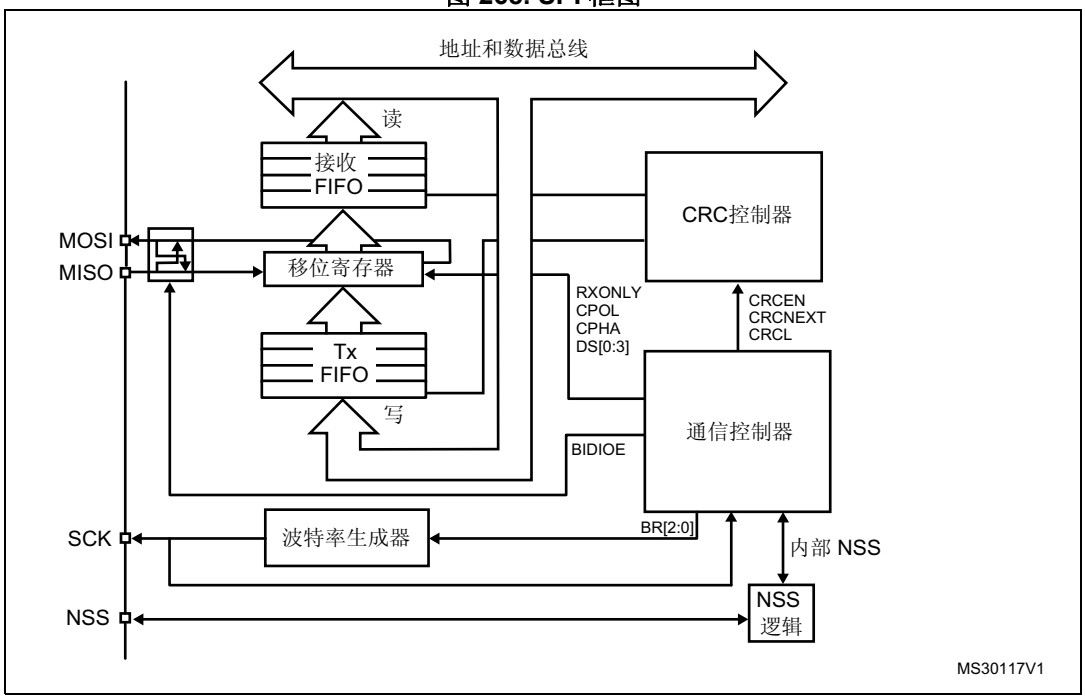
SPI 特性	SPI1/I2S1
增强型 NSSP 和 TI 模式	有
I2S 支持	有
硬件 CRC 计算	有
数据大小配置	4 到 16 位
Rx/Tx FIFO 大小	32 位
从低功耗睡眠模式唤醒	有

25.5 SPI 功能描述

25.5.1 一般说明

SPI 支持在 MCU 与外部器件之间进行同步串行通信。应用软件可通过轮询状态标志或使用专用 SPI 中断对通信进行管理。SPI 的主要组件及其交互方式如以下框图（*图 268*）所示。

图 268. SPI 框图



四个 I/O 引脚专用于与外部器件进行 SPI 通信。

- **MISO**：主输入 / 从输出数据。通常情况下，此引脚用于在从模式下发送数据和在主模式下接收数据。
- **MOSI**：主输出 / 从输入数据。通常情况下，此引脚用于在主模式下发送数据和在从模式下接收数据。
- **SCK**：SPI 主器件的串行时钟输出引脚以及 SPI 从器件的串行时钟输入引脚。
- **NSS**：从器件选择引脚。根据 SPI 和 NSS 设置，该引脚可用于：
  - 选择单个从器件以进行通信
  - 同步数据帧或
  - 检测多个主器件之间是否存在冲突

  详细信息，请参见*第 25.5.5 节：从器件选择 (NSS) 引脚管理*。

SPI 总线支持一个主器件与一个或多个从器件之间进行通信。该总线至少由两条线构成——一条用于时钟信号，另一条用于同步数据传输。其他信号可以根据 SPI 节点间的数据交换及其从器件选择信号管理进行添加。

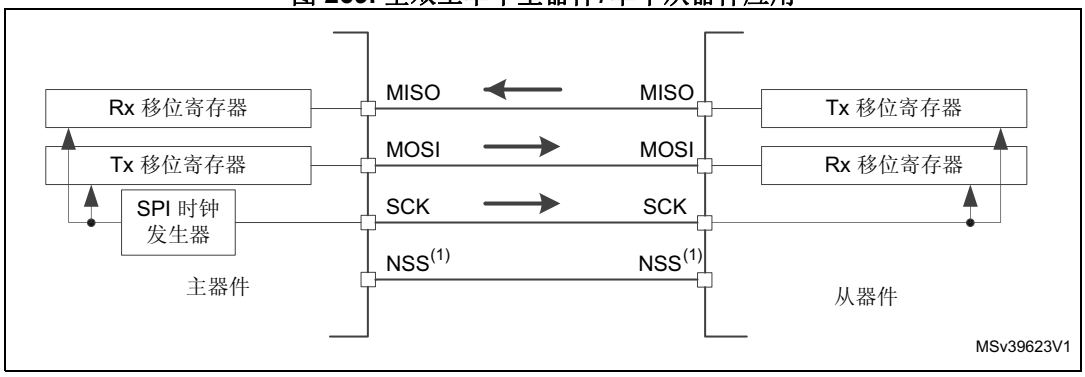
25.5.2 一个主器件和一个从器件之间的通信

SPI 支持 MCU 基于目标器件和应用要求使用不同的配置进行通信。这些配置使用 2 条或 3 条线（通过软件 NSS 管理），也可以使用 3 条或 4 条线（通过硬件 NSS 管理）。通信始终由主器件发起。

全双工通信

默认情况下，SPI 配置为进行全双工通信。在这种配置下，主器件和从器件的移位寄存器通过 MOSI 和 MISO 引脚之间的两条单向线连接。在 SPI 通信过程中，数据随主器件提供的 SCK 时钟边沿同步移位。主器件通过 MOSI 线将待发送的数据发送给从器件，通过 MISO 线从从器件接收数据。当数据帧传输完成时（所有位均移出），主器件和从器件之间即完成信息交换。

图 269. 全双工单个主器件/单个从器件应用

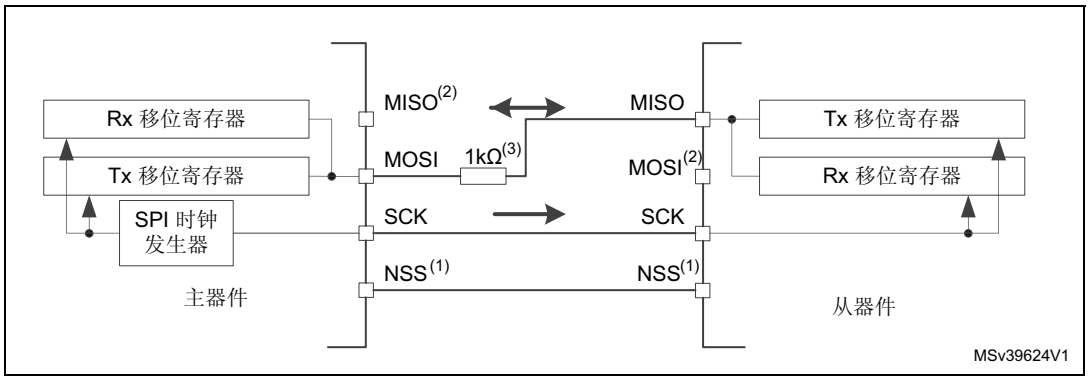


1. NSS 引脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些引脚。之后，必须在内部为主器件和从器件处理硬件控制流。有关更多详细信息，请参见*第 25.5.5 节：从器件选择 (NSS) 引脚管理*。

半双工通信

通过将 SPIx_CR1 寄存器的 BIDIMODE 位置 1，SPI 可采用半双工模式进行通信。在这种配置下，使用一条交叉连接线将主器件和从器件的移位寄存器连接起来。在此通信过程中，数据随 SCK 时钟边沿在移位寄存器之间进行移位，传输方向由主器件和从器件通过各自 SPIx_CR1 寄存器中的 BDIOE 位进行选择。在这种配置下，主器件的 MISO 引脚和从器件的 MOSI 引脚空闲，可在其他应用中用作 GPIO。

图 270. 半双工单个主器件/单个从器件应用



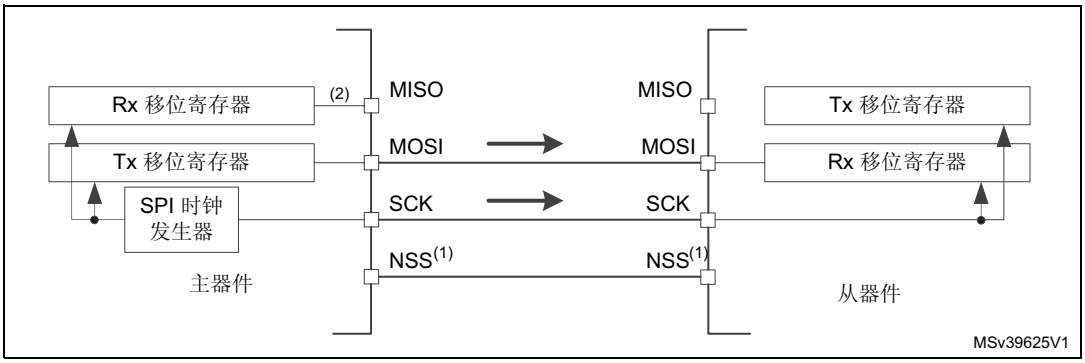
1. NSS 引脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些引脚。之后，必须在内部为主器件和从器件处理硬件控制流。有关更多详细信息，请参见*第 25.5.5 节：从器件选择 (NSS) 引脚管理*。
2. 在这种配置下，主器件的 MISO 引脚和从器件的 MOSI 引脚可用作 GPIO。
3. 当以双向模式工作的两个节点间的通信方向不是同步变化时，会出现临界情况，新发送器访问共用数据线，而前一个发送器仍保持线路上的相反值（值取决于 SPI 配置和通信数据）。两个节点会出现冲突，在共用线上短暂提供相反的输出电平，直到下一个节点也相应地改变其方向设置。建议此模式下在 MISO 和 MOSI 引脚之间插入串行电阻以在这种情况下保护输出并限制电流在二者之间流过。

单工通信

通过 SPIx_CR1 寄存器中的 RXONLY 位将 SPI 设置为只发送模式或只接收模式，可使 SPI 以单工模式进行通信。在这种配置下，仅使用一条线在主器件和从器件的移位寄存器之间进行传输。其余 MISO 和 MOSI 引脚对不用于通信，可用作标准 GPIO。

- **只发送模式 (RXONLY=0)**：配置设置与全双工设置相同。应用必须忽略在未使用的输入引脚上捕获的信息。该引脚可以用作标准 GPIO。
- **只接收模式 (RXONLY=1)**：应用可通过将 RXONLY 位置 1 来禁止 SPI 输出功能。在从器件配置下，MISO 输出被禁止，该引脚可用作 GPIO。当从器件选择信号有效时，从器件继续从 MOSI 引脚接收数据（请参见 *25.5.5：从器件选择 (NSS) 引脚管理*）。基于数据缓冲区的配置产生接收数据事件。在主器件配置下，MOSI 输出被禁止，该引脚可用作 GPIO。只要 SPI 处于使能状态，便不断生成时钟信号。停止时钟的唯一方式是将 RXONLY 位或 SPE 位清零，直至来自 MISO 引脚的传入模式结束，然后基于相应配置填充数据缓冲区结构。

图 271. 单工单个主器件/单个从器件应用（主器件为只发送模式/从器件为只接收模式）



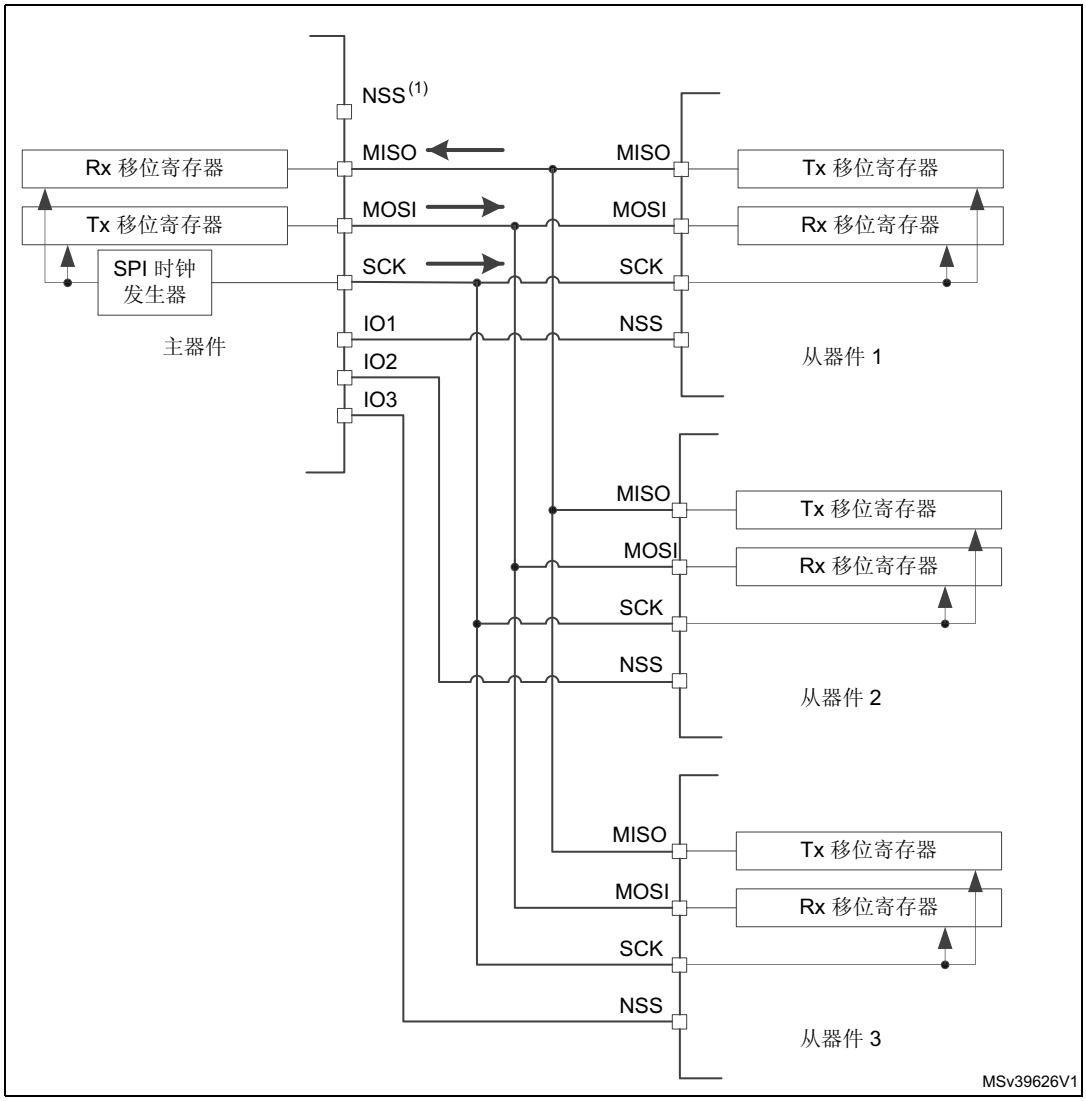
1. NSS 引脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些引脚。之后，必须在内部为主器件和从器件处理硬件控制流。有关更多详细信息，请参见*第 25.5.5 节：从器件选择 (NSS) 引脚管理*。
2. 在发送器 Rx 移位寄存器的输入上捕获意外输入信息。标准只发送模式下必须忽略与发送器接收流相关的所有事件（例如 OVR 标志）。
3. 在这种配置下，两个 MISO 引脚均可用作 GPIO。

注意： *任何单工通信都可以通过把半双工模式中的方向设置固定来实现（使能双向模式，同时 BDIO 位保持不变）。*

25.5.3 标准多从器件通信

在具有两个或多个独立从器件的配置下，主器件使用 GPIO 引脚来管理每个从器件的片选线（请参见*图 272*）。主器件必须通过拉低与从器件 NSS 输入相连的 GPIO 的电平来单独选择一个从器件。执行该操作后，便建立了标准主器件与专用从器件之间的通信。

图 272. 主器件和三个独立的从器件



1. 此配置的主器件侧不使用 NSS 引脚。该引脚必须在内部管理（SSM=1，SSI=1）以避免任何 MODF 错误。
2. 由于从器件的 MISO 引脚连在一起，所有从器件 MISO 引脚的 GPIO 配置必须设置为复用功能的开漏模式（请参见 I/O 复用功能输入/输出部分 (GPIO)）。

25.5.4 多主器件通信

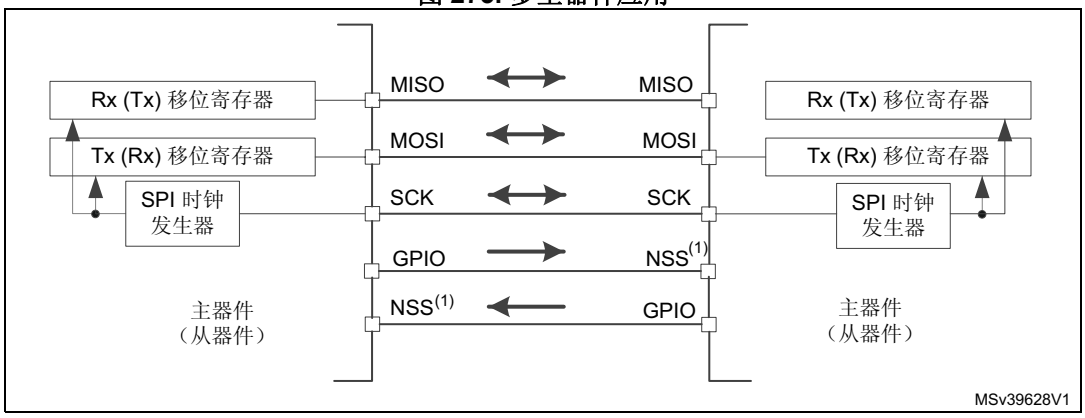
如果 SPI 总线未用于多主功能，用户可使用内置功能来检测尝试同时控制总线的两个节点间是否存在潜在冲突。对于该检测，NSS 引脚配置为硬件输入模式。

由于此时只有一个结点可将其输出施加到公用数据线上，因此无法连接超过两个以此模式工作的 SPI 节点。

当节点无效时，默认情况下均保持从模式。一旦一个节点要接管对总线的控制，它会将自身切换到主模式，然后通过专用 GPIO 引脚向其他节点的从器件选择输入施加有效电平。会话完成后，有效的从器件选择信号将被释放，控制总线的节点会短暂切换回被动从模式，等待下一个会话开始。

如果两个节点同时发出各自的控制请求，则会出现总线冲突（请参见模式故障 MODF 事件）。随后，用户可应用某个简单的仲裁过程（例如，在两个节点上施加不同的预定义超时来推迟下一个尝试）。

图 273. 多主器件应用



1. 在两个节点上，NSS 引脚配置为硬件输入模式。当无效节点配置为从器件时，其有效电平将使能 MISO 线输出控制。

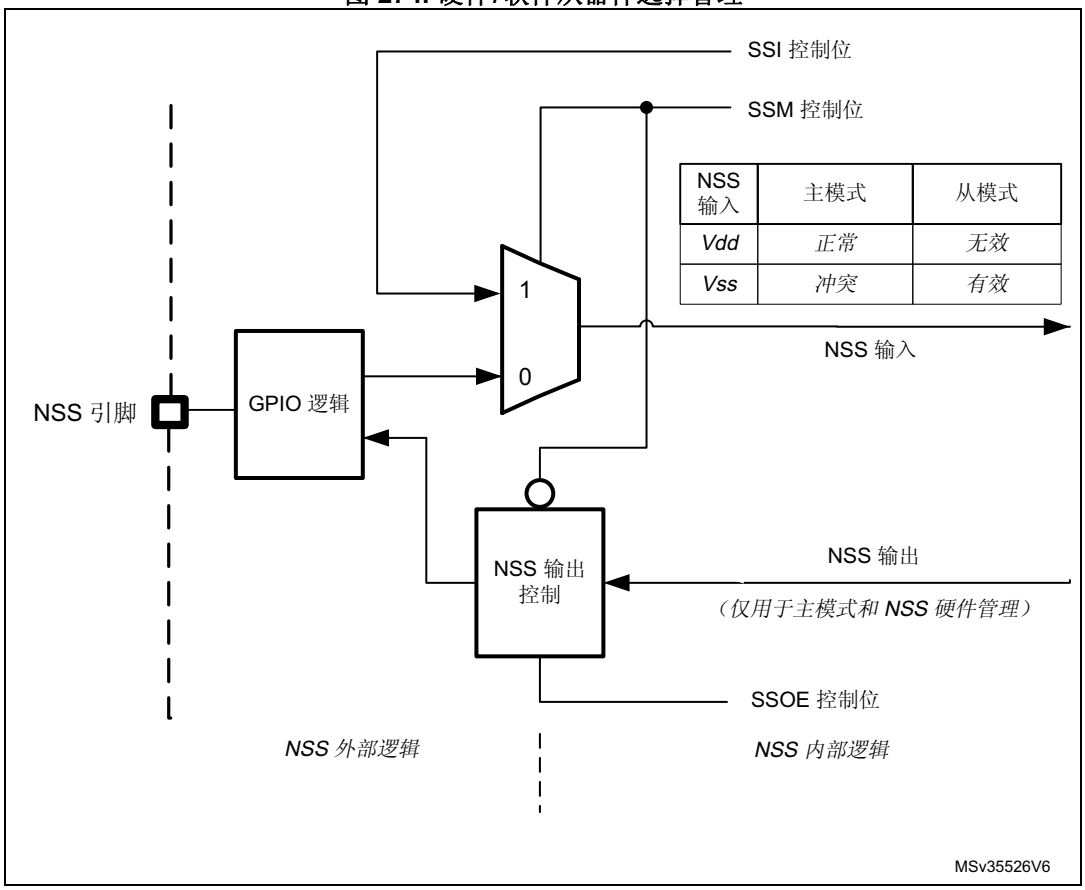
25.5.5 从器件选择 (NSS) 引脚管理

在从模式下，NSS 用作标准的“片选”输入，使从器件与主器件进行通信。在主模式下，NSS 可用作输出或输入。用作输入时，可防止多主模式总线冲突；用作输出时，可驱动单个从器件的从器件选择信号。

可以使用 SPIx_CR1 寄存器中的 SSM 位设置硬件或软件从器件选择管理：

- **软件 NSS 管理 (SSM = 1)**：在这种配置下，由 SPIx_CR1 寄存器中的 SSI 位的值内部驱动从器件选择信息。外部 NSS 引脚空闲，可供其他应用使用。
- **硬件 NSS 管理 (SSM = 0)**：在这种情况下，可行的配置有两种：所用配置取决于 NSS 输出配置（SPIx_CR1 寄存器中的 SSOE 位）。
  - **NSS 输出使能（SSM=0 且 SSOE = 1）**：仅在将 MCU 设置为主器件时才使用该配置。NSS 引脚由硬件管理。只要在主模式下使能 SPI (SPE=1)，NSS 信号便会被驱动为低电平，并且会一直保持低电平状态，直至禁止 SPI (SPE =0)。如果激活 NSS 脉冲模式 (NSSP=1)，连续通信间会生成脉冲。SPI 无法在采用此 NSS 设置的多主模式配置下工作。
  - **NSS 输出禁止（SSM=0 且 SSOE = 0）**：如果微控制器在总线上用作主器件，此配置可实现多主模式功能。如果在该模式下将 NSS 引脚拉至低电平，SPI 将进入主模式故障状态，器件将在从模式下自动进行重新配置。在从模式下，NSS 引脚用作标准的“片选”输入，当 NSS 线为低电平时将选择从器件。

图 274. 硬件/软件从器件选择管理



25.5.6 通信格式

SPI 通信过程中，将同时执行接收和发送操作。串行时钟 (SCK) 对数据线上的信息的移位和采样进行同步。通信格式取决于时钟相位、时钟极性和数据帧格式。为了能够在彼此间进行通信，主器件和从器件必须遵循相同的通信格式。

时钟相位和极性控制

通过 SPIx_CR1 寄存器中的 CPOL 和 CPHA 位，可以用软件选择四种可能的时序关系。CPOL（时钟极性）位控制不传输任何数据时时钟的空闲状态值。该位对主器件和从器件都有效。如果复位 CPOL，SCK 引脚在空闲状态处于低电平。如果将 CPOL 置 1，SCK 引脚在空闲状态处于高电平。

如果将 CPHA 位置 1，则会在 SCK 引脚的第二个边沿捕获传输的第一个数据位（如果复位 CPOL 位，则为下降沿；如果将 CPOL 位置 1，则为上升沿）。即，在每次出现该时钟边沿时锁存数据。如果将 CPHA 位复位，则会在 SCK 引脚的第一个边沿捕获传输的第一个数据位（如果将 CPOL 位置 1，则为下降沿；如果将 CPOL 位复位，则为上升沿）。即，在每次出现该时钟边沿时锁存数据。

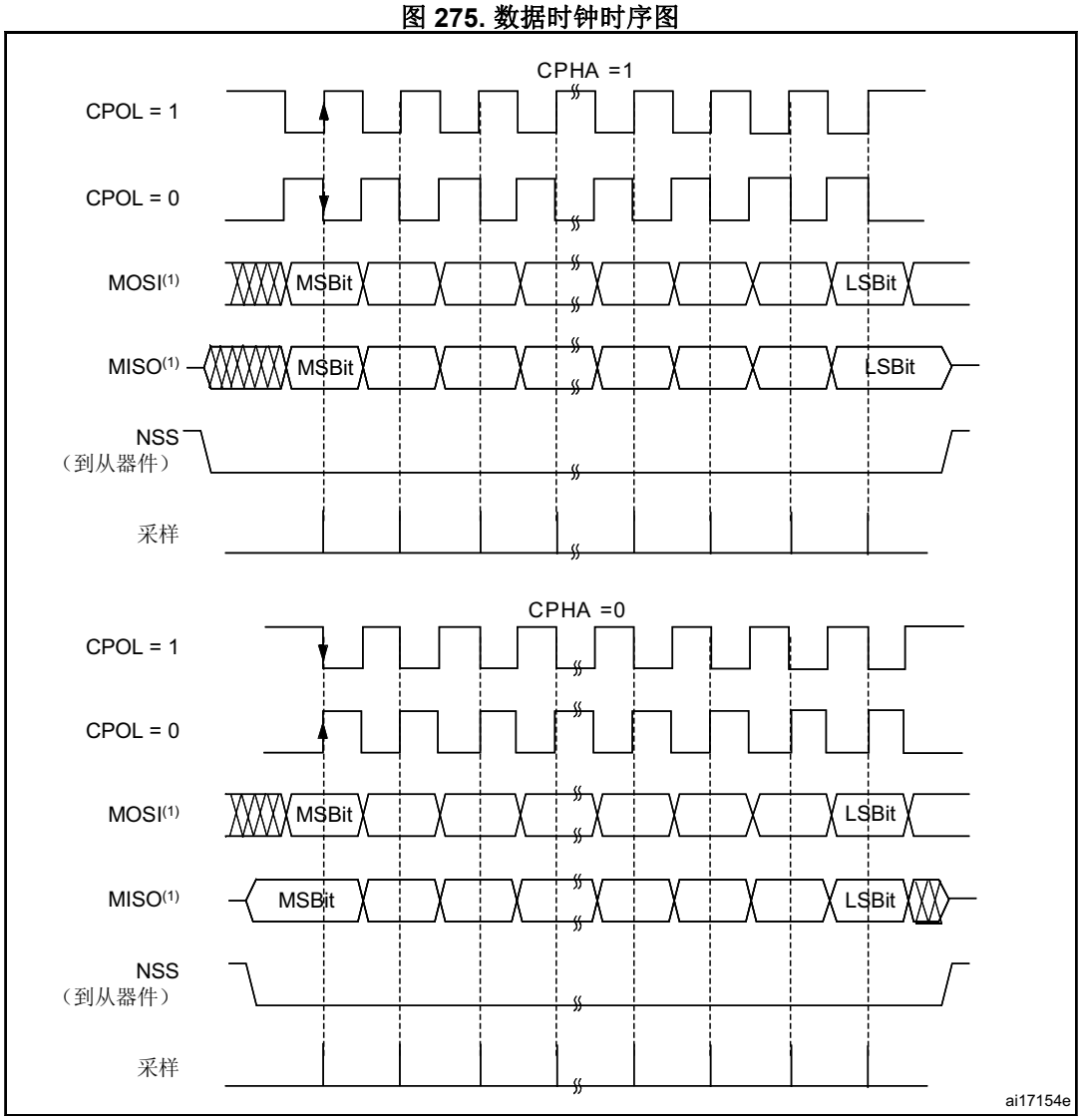
CPOL（时钟极性）和 CPHA（时钟相位）位的组合用于选择数据捕获时钟边沿。

图 275 给出了在 CPHA 和 CPOL 位的四种组合下的 SPI 全双工传输。

注意： *在切换 CPOL/CPHA 位之前，必须通过复位 SPE 位来禁止 SPI。*

SCK 的空闲状态必须与 SPIx_CR1 寄存器中选择的极性相对应（如果 CPOL=1，则上拉 SCK；如果 CPOL=0，则下拉 SCK）。

图 275. 数据时钟时序图

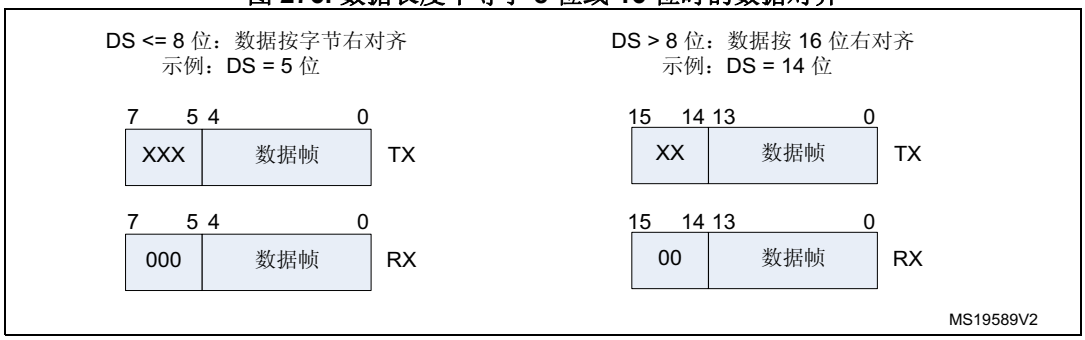


1. 数据位的顺序取决于 LSBFIRST 位的设置。

数据帧格式

SPI 移位寄存器可设置为以 MSB 在前或 LSB 在前的方式移出数据，具体取决于 LSBFIRST 位的值。数据帧的长度通过 DS 位进行选择。数据帧的长度可设置为 4 位到 16 位，且此设置对于发送和接收均适用。无论所选的数据帧长度为何，必须依照 FRXTH 电平对 FIFO 进行读访问。访问 SPIx_DR 寄存器时，数据帧始终按字节（数据不超过一个字节时）或半字进行右对齐（请参见*图 276*）。在通信过程中，只会为数据帧内的位提供时钟并传输这些位。

图 276. 数据长度不等于 8 位或 16 位时的数据对齐



注意： *数据长度至少为 4 位。如果所选的数据长度不足 4 位，则会将数据帧长度强制为 8 位。*

25.5.7 SPI 配置

主器件和从器件的配置步骤几乎相同。对于具体的模式设置，请遵从相应部分的内容。若要对标准通信进行初始化，请执行以下步骤：

1. 对相应的 GPIO 寄存器执行写操作：将 MOSI、MISO 和 SCK 引脚配置为 GPIO。
2. 对 SPI_CR1 寄存器执行写操作：
   a) 通过 BR[2:0] 位配置串行时钟波特率（注：4）。
   b) 配置 CPOL 和 CPHA 位的组合，定义数据传输和串行时钟之间的关系（四种关系中的一种）（NSSP 模式下必须将 CPHA 清零）。（注：2——在 TI 模式下使能 CRC 的情况除外）。
   c) 通过配置 RXONLY 或 BIDIMODE 和 BIDIOE 来选择单工或半双工模式（RXONLY 和 BIDIMODE 不可同时置 1）。
   d) 配置 LSBFIRST 位以定义帧格式（注：2）。
   e) 如果需要 CRC，请配置 CRCL 和 CRCEN 位（SCK 时钟信号处于空闲状态时）。
   f) 配置 SSM 和 SSI（注：2 和 3）。
   g) 配置 MSTR 位（在多主模式 NSS 配置下，如果主器件配置为防止发生 MODF 错误，则应避免 NSS 上出现状态冲突）。
3. 对 SPI_CR2 寄存器执行写操作：
   a) 配置 DS[3:0] 位，选择传输的数据长度。
   b) 配置 SSOE（注：1、2 和 3）。
   c) 如果需要使用 TI 协议，请将 FRF 位置 1（TI 模式下将 NSSP 位保持清零状态）。
   d) 如果在两个数据单元之间需要 NSS 脉冲模式，请将 NSSP 位置 1（NSSP 模式下将 CHPA 和 TI 位保持清零状态）。
   e) 配置 FRXTH 位。RXFIFO 阈值必须与 SPIx_DR 寄存器的读访问大小相符。
   f) 如果封装模式下使用 DMA，请初始化 LDMA_TX 和 LDMA_RX 位。
4. 对 SPI_CRCPR 寄存器执行写操作：需要时配置 CRC 多项式。
5. 对相应的 DMA 寄存器执行写操作：如果使用 DMA 数据流，请在 DMA 寄存器中配置 SPI Tx 和 Rx 专用的 DMA 数据流。

注意： *(1) 从模式下无需此步骤。*

(2) TI 模式下无需此步骤。

(3) NSSP 模式下无需此步骤。

(4) 从模式下无需此步骤，但从器件在 TI 模式下工作时除外。

25.5.8 使能 SPI 的步骤

建议在主器件发送时钟前使能 SPI 从器件。否则，数据传输可能会不正常。从器件的数据寄存器必须包含待发送的数据才能开始与主器件通信（在通信时钟的第一个边沿；如果时钟信号连续，则是在正在进行的通信结束前）。使能 SPI 从器件前，SCK 信号必须稳定为所选极性对应的空闲状态电平。

当 SPI 处于使能状态，且 TXFIFO 不为空或者对 TXFIFO 执行下一次写操作时，全双工模式（或任何只发送模式）下的主器件开始通信。

在任何主器件只接收模式（RXONLY=1 或 BIDIMODE=1 且 BIDIOE=0）下，使能 SPI 后，主器件立即开始通信且时钟立即开始运行。

要处理 DMA，请遵从相应部分的内容。

25.5.9 数据发送和接收过程

RXFIFO 和 TXFIFO

所有 SPI 数据交互均经由 32 位内置 FIFO。这使得 SPI 能够以连续流工作，并能防止在数据帧长度较短时发生上溢。每个方向都有其自身的 FIFO，分别被称为 TXFIFO 和 RXFIFO。这两个 FIFO 可在所有 SPI 模式下使用，但已使能 CRC 计算的只接收模式（从器件或主器件）除外（请参见*第 25.5.14 节：CRC 计算*）。

FIFO 的处理取决于数据交换模式（双工和单工）、数据帧格式（帧中的位数）、FIFO 数据寄存器中的访问大小（8 位或 16 位）以及访问 FIFO 时是否使用数据封装（请参见*第 25.5.13 节：TI 模式*）。

对 SPIx_DR 寄存器执行读访问时，会返回尚未读取的 RXFIFO 中存储的最早的值。对 SPIx_DR 执行写访问时，会在发送队列结束时将写入的数据存储到 TXFIFO 中。读访问必须始终符合由 SPIx_CR2 寄存器中的 FRXTH 位配置的 RXFIFO 阈值。FTLVL[1:0] 和 FRLVL[1:0] 位用于指示两个 FIFO 当前的占用水平。

SPIx_DR 寄存器的读访问必须通过 RXNE 事件进行管理。当数据存储到 RXFIFO 中且达到阈值（由 FRXTH 位定义）时会触发该事件。当 RXNE 清零时，RXFIFO 被视为空。同样地，待发送数据帧的写访问通过 TXE 事件进行管理。当 TXFIFO 占用水平小于或等于其容量的一半时会触发该事件。否则，TXE 清零，TXFIFO 被视为已满。通过这种方式，当数据帧格式不超过 8 位时，RXFIFO 最多可存储四个数据帧，而 TXFIFO 最多只能存储三个数据帧。当软件尝试向 TXFIFO 中写入更多 16 位模式的数据时，这种差异能够防止 TXFIFO 中已存储的 3 个 8 位数据帧出现损坏的情况。TXE 和 RXNE 事件均可通过中断进行轮询或处理。请参见*图 278* 到*图 281*。

另一种管理数据交换的方式是使用 DMA（请参见*使用 DMA（直接存储器寻址）进行通信*）。

如果在 RXFIFO 已满时收到下一个数据，将发生上溢事件（请参见*第 25.5.10 节：SPI 状态标志*中的 OVR 标志说明）。上溢事件可通过中断来轮询或处理。

BSY 位被置 1 表示正在处理当前数据帧交互。当时钟信号连续运行时，在主器件中，BSY 标志在数据帧之间保持置 1 状态，但在从器件中，BSY 标志在数据帧传输之间变为低电平并持续最短的一段时间（一个 SPI 时钟）。

序列处理

可通过一个序列传送一些数据帧从而完成一条消息。使能发送后，序列即开始，只要主器件的 TXFIFO 中存在数据便一直继续。时钟信号由主器件持续提供，直至 TXFIFO 变为空，之后时钟信号停止，等待其他数据。

在只接收模式、半双工模式（BIDIMODE=1 且 BIDIOE=0）或单工模式（BIDIMODE=0 且 RXONLY=1）下，当使能 SPI 并激活只接收模式后，主器件将立即启动序列。时钟信号由主器件提供，且直至主器件禁止 SPI 或只接收模式才会停止。在此之前，主器件会连续接收数据帧。

当主器件能够以连续模式（SCK 信号连续）提供所有交互时，任何时候都必须根据从器件功能来处理数据流及其内容。必要时，主器件必须降低通信速度，提供较慢的时钟或带有足够延时的单独帧或数据段。请注意，SPI 模式下不存在主器件或从器件的下溢错误信号，来自从器件的数据始终由主器件处理，即使从器件无法及时正确地准备数据也是如此。从器件最好使用 DMA，尤其是数据帧较短而总线速率较高时。

在多从器件系统中，每个序列必须通过 NSS 脉冲进行控制，从而只选择其中一个从器件进行通信。在单个从器件系统中，无需通过 NSS 来控制从器件，但此时提供此脉冲通常会更好，以在每个数据序列开始时同步从器件。NSS 可通过软件和硬件进行管理（请参见 *第 25.5.5 节：从器件选择 (NSS) 引脚管理*）。

当 BSY 位置 1 时，表示正在处理数据帧事务。当所进行的帧交互完成时，RXNE 标志将置 1。最后一位采样后，整个数据帧会存储到 RXFIFO 中。

禁止 SPI 的步骤

当禁止 SPI 时，必须按照本段中介绍的禁止步骤进行操作。当外设时钟停止时，在系统进入低功耗模式前做到这一点是十分重要的。否则会损坏正在进行的交互。在某些模式下，禁止步骤是停止所进行的连续通信的唯一方式。

当处于全双工或只发送模式下的主器件停止提供待发送的数据时，可结束任何事务。在这种情况下，时钟在最后一个数据传输后停止。当处理奇数数量的数据帧时，必须特别注意封装模式以防止交换一些空字节（请参见*数据封装*部分）。在这些模式下禁止 SPI 之前，用户必须按照标准的禁止步骤进行操作。在主发送器上禁止 SPI，而同时正在处理帧交互或已将下一个数据帧存储到 TXFIFO 中时，SPI 行为无法保证。

只要主器件处于只接收模式，停止连续时钟的唯一方式就是通过 SPE=0 来禁止外设。这必须在最后一个数据帧传输内的特定时间段，即第一位采样与最后一位传输开始之间完成（以便接收全部数量的预期数据帧并防止在最后一个有效数据帧后读取任何其他的“空”数据）。在该模式下禁止 SPI 时必须遵循特定步骤。

禁止 SPI 后，已接收但未读取的数据始终存储在 RXFIFO 中，这些数据必须在下次使能 SPI 后进行处理，然后才能启动新序列。为防止存在未读取的数据，需确保禁止 SPI 时 RXFIFO 为空，可通过正确的禁止步骤来禁止 SPI，也可以通过控制外设复位专用的特定寄存器以软件复位的方式来初始化所有 SPI 寄存器从而禁止 SPI（请参见 RCC_APBiRSTR 寄存器中的 SPIiRST 位）。

标准禁止步骤通过轮询 BSY 状态以及 FTLVL[1:0] 来检查发送会话是否完全结束。还可以在必须识别正在处理的传输是否结束的特定情况下完成这种检查，例如：

- 当 NSS 信号由软件管理且主器件必须为从器件提供 NSS 脉冲结束时，或者
- 最后一个数据帧或 CRC 帧传输仍在外设总线中处理时，来自 DMA 或 FIFO 的传输数据流完成。

正确的禁止步骤如下（使用只接收模式时除外）：

1. 等待至 FTLVL[1:0] = 00（无需发送更多数据）。
2. 等待至 BSY=0（最后一个数据帧已处理完）。
3. 禁止 SPI (SPE = 0)。
4. 读取数据，直至 FRLVL[1:0] = 00（读取接收的所有数据）。

某些只接收模式的正确禁止步骤如下：

1. 当最后一个数据帧正在处理时，通过在特定时间窗口内禁止 SPI (SPE=0) 来中断接收流。
2. 等待至 BSY=0（最后一个数据帧已处理完）。
3. 读取数据，直至 FRLVL[1:0] = 00（读取接收的所有数据）。

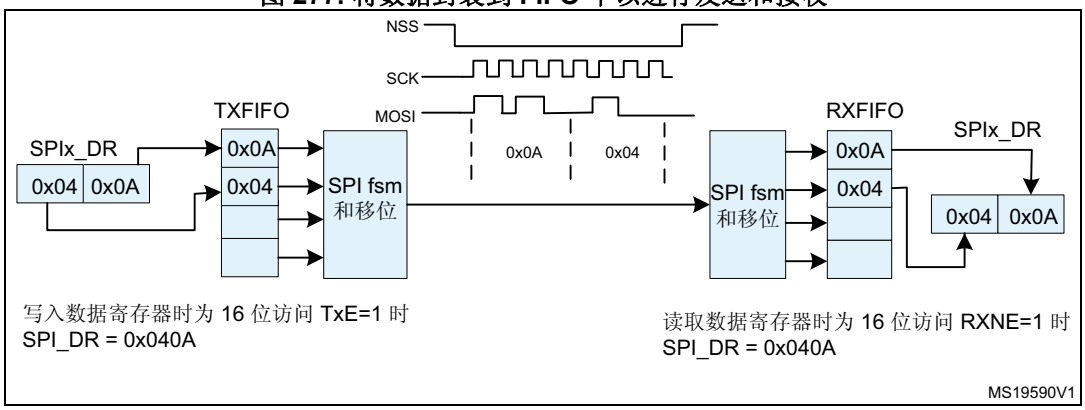
注意： *如果使用封装模式并且必须接收奇数数量的数据帧且数据帧格式为小于或等于 8 位（不超过一个字节），则 FRXTH 必须在 FRLVL[1:0] = 01 时置 1，以便生成 RXNE 事件从而读取最后的奇数编号数据帧并且使 FIFO 指针保持正确对齐。*

数据封装

若数据帧长度不足一个字节（小于或等于 8 位），当 SPIx_DR 寄存器上执行任何 16 位读写访问时将自动使用数据封装。在这种情况下将并行处理双数据帧模式。最初，SPI 以所访问字的 LSB 中存储的模式工作，然后以 MSB 中存储的另一种模式来工作。*图 277* 给出了数据封装模式序列处理的示例。在对发送器的 SPIx_DR 寄存器执行单次 16 位访问后发送两个数据帧。如果 RXFIFO 阈值设置为 16 位 (FRXTH=0)，该序列只会在接收器中生成一个 RXNE 事件。接收器随后必须通过对 SPIx_DR 执行单次 16 位读访问来访问这两个数据帧，从而响应该单个 RXNE 事件。RxFIFO 阈值设置和后续读访问必须始终与接收器侧保持一致，因为若不一致，则会丢失数据。

如果必须处理奇数数量的此类“不超过一个字节”的数据帧，则会出现特定问题。在发送器侧，只需将任意奇序列的最后一个数据帧以 8 位访问的方式写入 SPIx_DR 即可。接收器必须针对所接收的奇序列中的最后一个数据帧更改 Rx_FIFO 阈值大小，以生成 RXNE 事件。

图 277. 将数据封装到 FIFO 中以进行发送和接收



1. 在本示例中：数据大小 DS[3:0] 配置为 4 位、CPOL=0、CPHA=1 且 LSBFIRST =0。数据存储始终采用右对齐方式，同时有效位仅在总线上执行，总线上 LSB 字节的内容在前，未使用的位在发送器侧无需考虑，在接收器侧则用零进行填充。

使用 DMA（直接存储器寻址）进行通信

为了以最大速度工作并且为了促进避免上溢所需的数据寄存器读/写过程，SPI 提供了 DMA 功能，该功能采用了简单的请求/应答协议。

将 SPIx_CR2 寄存器中的使能位 TXDMAEN 或 RXDMAEN 置 1 时，将请求 DMA 访问。必须向发送缓冲区和接收缓冲区发出单独的请求。

- 在发送过程中，每次 TXE 位置 1 都会发出 DMA 请求。然后，DMA 将对 SPIx_DR 寄存器执行写操作。
- 在接收过程中，每次 RXNE 位置 1 都会发出 DMA 请求。然后，DMA 将对 SPIx_DR 寄存器执行读操作。

请参见*图 278* 到*图 281*。

当 SPI 仅用于发送数据时，可以只使能 SPI Tx DMA 通道。在这种情况下，OVR 标志会置 1，因为未读取接收的数据。当 SPI 仅用于接收数据时，可以只使能 SPI Rx DMA 通道。

在发送模式下，DMA 写入所有要发送的数据（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 BSY 标志进行监视，以确保 SPI 通信已完成。在禁止 SPI 或进入停止模式前必须执行此步骤，以避免损坏最后一次发送。软件必须首先等待 FTLVL[1:0]=00，再等待 BSY=0。

通过 DMA 开始通信时，为防止 DMA 通道管理引发错误事件，必须按顺序执行以下步骤：

1. 如果使用 DMA Rx，通过 SPI_CR2 寄存器中的 RXDMAEN 位来使能 DMA 接收缓冲区。
2. 如果使用数据流，通过 DMA 寄存器来使能 Tx 和 Rx 的 DMA 数据流。
3. 如果使用 DMA Tx，通过 SPI_CR2 寄存器中的 TXDMAEN 位来使能 DMA 发送缓冲区。
4. 通过将 SPE 位置 1 使能 SPI。

要关闭通信，必须按顺序执行以下步骤：

1. 如果使用了 Tx 和 Rx 的 DMA 数据流，通过 DMA 寄存器来禁止这些数据流。
2. 通过后续 SPI 禁止步骤来禁止 SPI。
3. 如果使用 DMA Tx 和/或 DMA Rx，通过将 SPI_CR2 寄存器中的 TXDMAEN 和 RXDMAEN 位清零来禁止 DMA 发送缓冲区和接收缓冲区。

使用 DMA 时的数据封装

如果由 DMA 管理传输（SPIx_CR2 寄存器中的 TXDMAEN 和 RXDMAEN 位置 1），则将根据为 SPI TX 和 SPI RX 的 DMA 通道配置的 PSIZE 值自动使能/禁止封装模式。如果 DMA 通道的 PSIZE 值等于 16 位，且 SPI 数据大小小于或等于 8 位，则将使能封装模式。然后，DMA 将自动管理对 SPIx_DR 寄存器的写操作。

如果使用数据封装模式，且要传输的数据数量不是 2 的倍数，则 LDMA_TX/LDMA_RX 位必须置 1。然后，SPI 会认为最后一次 DMA 传输时只发送或接收一个数据（有关详细信息，请参见*第 695 页的数据封装*）。

通信图

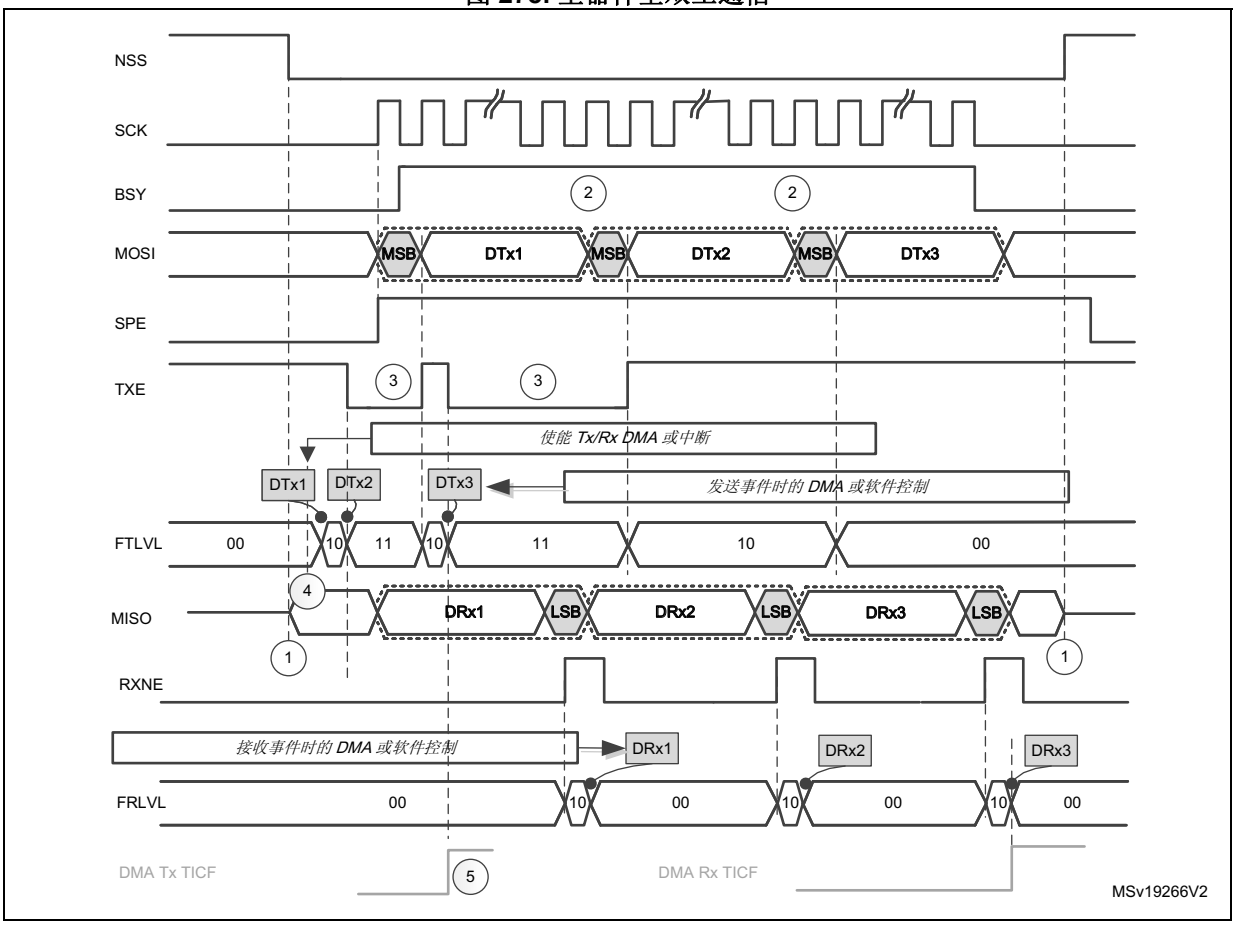
本部分将介绍一些典型的时序图。无论 SPI 事件是通过轮询、中断还是 DMA 进行处理，这些时序图均有效。为简单起见，此处均假设 LSBFIRST=0、CPOL=0 和 CPHA=1。不提供 DMA 数据流的完整配置。

以下带编号的注释对*第 697 页的图 278* 到*第 700 页的图 281* 均适用。

1. 激活 NSS 并使能 SPI 后，从器件开始控制 MISO 线，而当其中一个条件不成立时，从器件将与 MISO 线断开。必须为从器件提供充足的时间，以便在传输开始前准备好主器件专用的数据。
   在主器件上，只有使能 SPI 后，SPI 外设才会控制 MOSI 和 SCK 信号（偶尔还会控制 NSS 信号）。如果禁止 SPI，SPI 外设会与 GPIO 逻辑断开，因此这些线上的电平只取决于 GPIO 设置。
2. 在主器件上，如果通信（时钟信号）连续，BSY 在数据帧之间保持有效。在从器件上，BSY 信号在数据帧之间始终会保持至少一个时钟周期的低电平状态。
3. 只有 TXFIFO 已满时，TXE 信号才会清零。
4. DMA 仲裁过程在 TXDMAEN 位置 1 后立即开始。TXE 中断在 TXEIE 置 1 后立即生成。TXE 信号处于有效电平时，开始向 TxFIFO 传输数据，直至 TxFIFO 已满或 DMA 传输完成。
5. 如果要发送的所有数据可装入 TxFIFO，则 DMA Tx TCIF 标志甚至会在 SPI 总线上的通信开始前置 1。SPI 传输完成前，该标志始终为高电平状态。

6. 数据包的 CRC 值是通过 SPIx_TXCRCR 和 SPIx_RXCRCR 寄存器一帧一帧连续生成的，完成整个数据封装后，CRC 信息可通过 DMA 自动处理（Tx 通道必须设置为要处理的数据帧数），也可由软件处理（用户必须在处理最后一个数据帧的过程中处理 CRCNEXT 位）。
   在发送端 SPIx_TXCRCR 所计算出的 CRC 值只是简单的由发送器发送出去，而接收端收到的 CRC 信息会被加载到 RxFIFO 与 SPIx_RXCRCR 寄存器内容进行比较（如果不一致则 CRC 错误标志位会置位）。因此用户必须注意刷新 FIFO 中的相关信息，可以通过软件读出 RxFIFO 中存储的所有内容的方式来实现，若已针对 Rx 通道预设置了适当数量的数据帧（数据帧数 + CRC 帧数），也可通过 DMA 的方式来实现（请参见示例假设中的设置）。
7. 在数据封装模式下，TxE 和 RxNE 事件成对出现，对 FIFO 的每次读/写访问都为 16 位宽，直至数据帧数为偶数。如果 TxFIFO 处于 ¾ 满状态，则 FTLVL 将保持 FIFO 满时对应的状态。因此在 TxFIFO 变为 ½ 满状态前，无法存储最后一个奇数编号的数据帧。该帧可通过软件或自动由 DMA（LDMA_TX 控制置 1 时）对其进行 8 位访问的方式存储在 TxFIFO 中。
8. 要在封装模式下接收最后一个奇数编号的数据帧，必须在处理最后一个数据帧后，将 Rx 阈值更改为 8 位，这可通过软件（设置为 FRXTH=1）实现或由 DMA 内部信号在 LDMA_RX 置 1 时自动实现。

图 278. 主器件全双工通信



主器件全双工通信示例的假设条件如下：

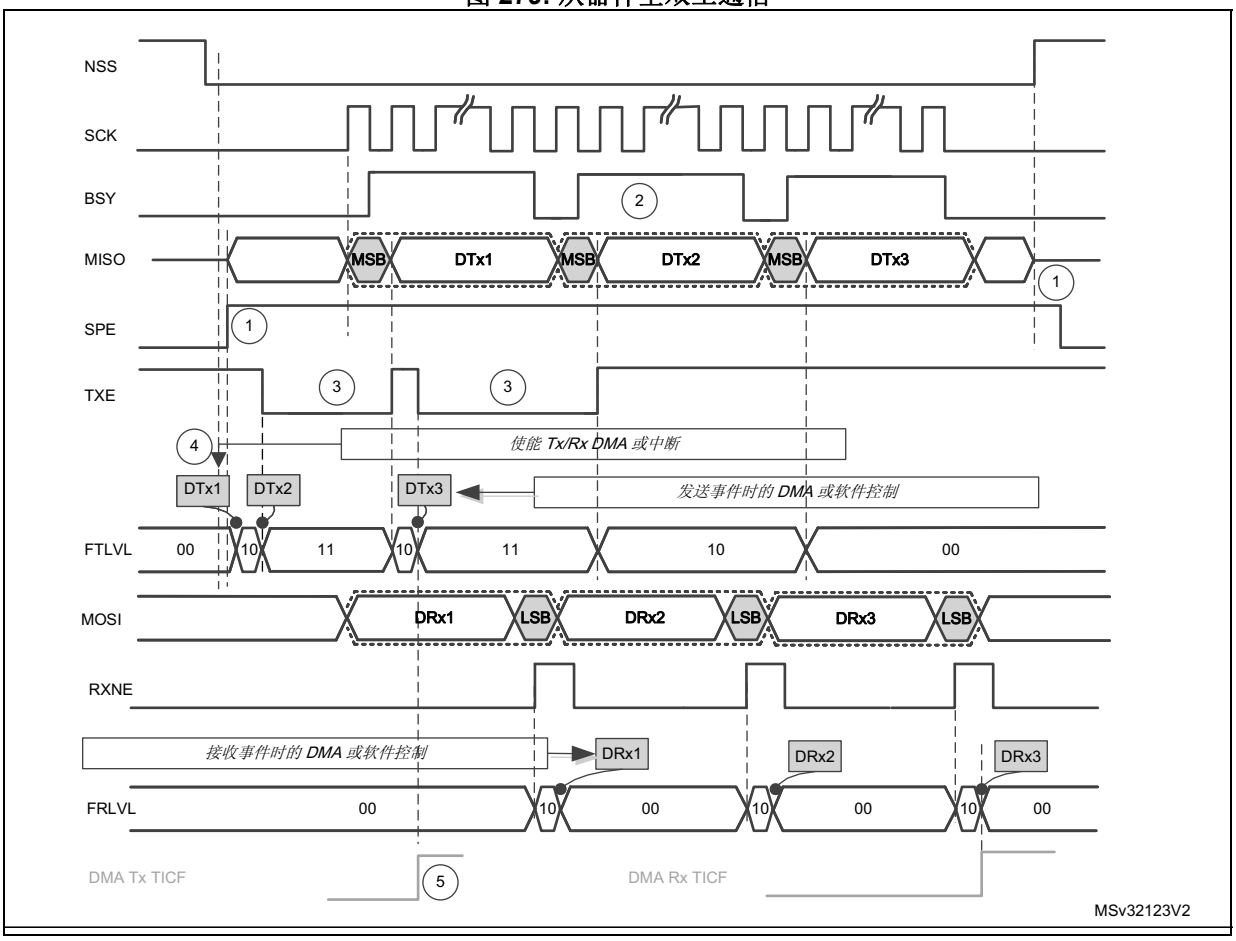
- 数据大小 > 8 位

如果使用 DMA：

- 由 DMA 处理的 Tx 帧的数量设置为 3
- 由 DMA 处理的 Rx 帧的数量设置为 3

有关通用假设条件和注释的详细信息，另请参见*第 696 页的通信图*一节。

图 279. 从器件全双工通信



从器件全双工通信示例的假设条件如下：

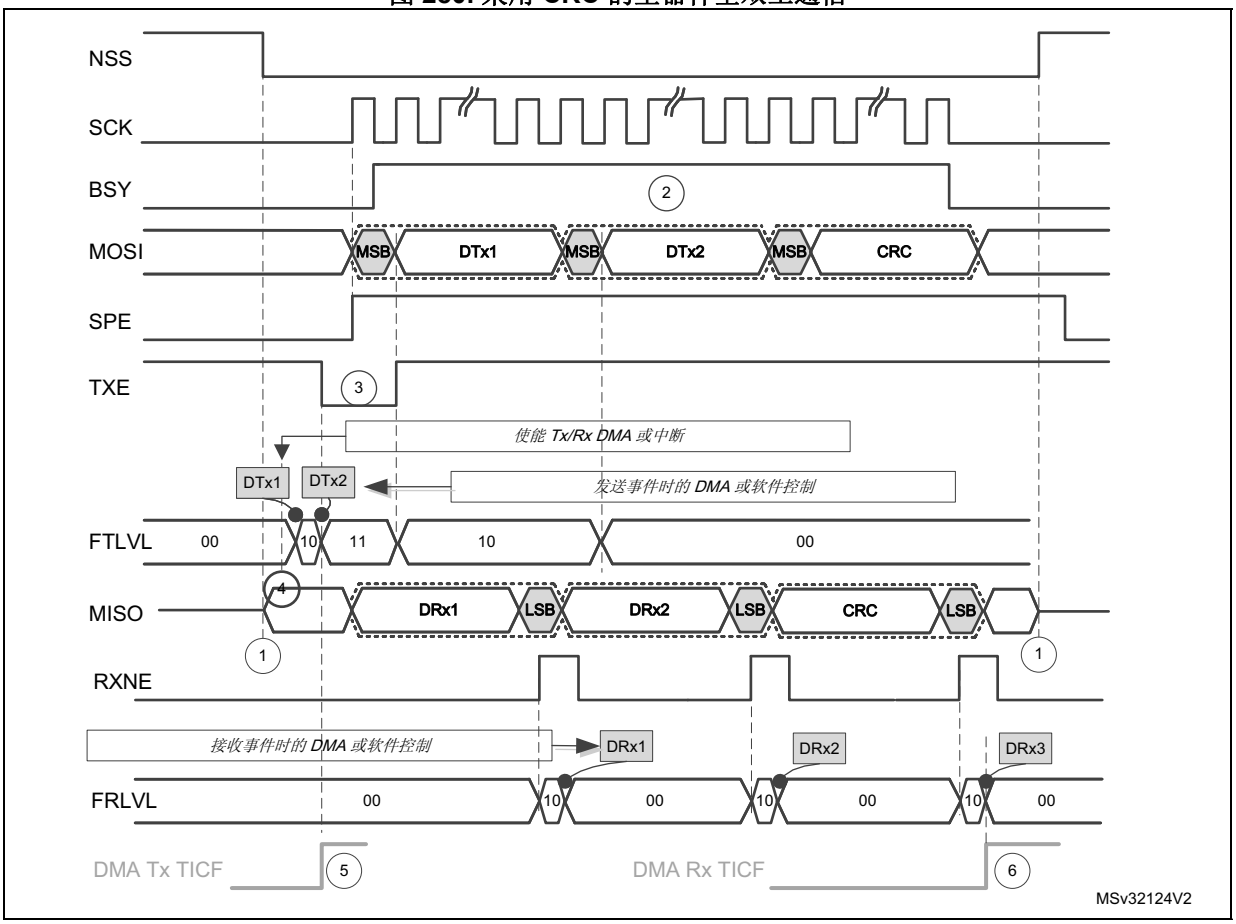
- 数据大小 > 8 位

如果使用 DMA：

- 由 DMA 处理的 Tx 帧的数量设置为 3
- 由 DMA 处理的 Rx 帧的数量设置为 3

有关通用假设条件和注释的详细信息，另请参见*第 696 页的通信图*。

图 280. 采用 CRC 的主器件全双工通信



带有 CRC 的主器件全双工通信的假设条件如下：

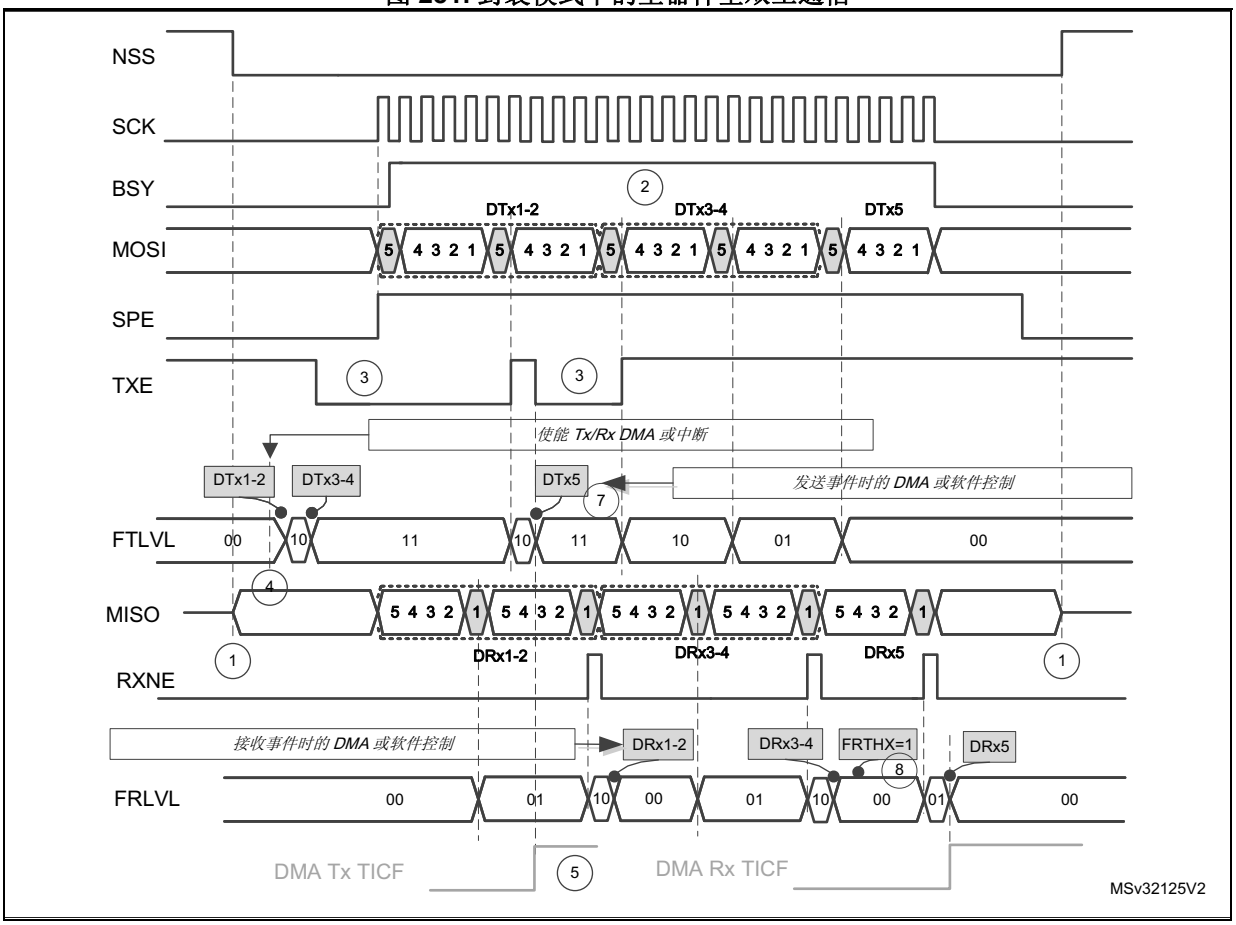
- 数据大小 = 16 位
- CRC 已使能

如果使用 DMA：

- 由 DMA 处理的 Tx 帧的数量设置为 2
- 由 DMA 处理的 Rx 帧的数量设置为 3

有关通用假设条件和注释的详细信息，另请参见*第 696 页的通信图*一节。

图 281. 封装模式下的主器件全双工通信



封装模式下的主器件全双工通信示例的假设条件如下：

- 数据大小 = 5 位
- 主要通过 16 位访问的方式来读/写 FIFO
- FRXTH=0

如果使用 DMA：

- 要由 DMA 处理的 Tx 帧的数量设置为 3
- 要由 DMA 处理的 Rx 帧的数量设置为 3
- Tx 和 Rx 的 DMA 通道的 PSIZE 均设置为 16 位
- LDMA_TX=1 且 LDMA_RX=1

有关通用假设条件和注释的详细信息，另请参见*第 696 页的通信图*一节。

25.5.10 SPI 状态标志

应用可通过三种状态标志监视 SPI 总线的状态。

发送缓冲区为空 (TXE)

当发送 TXFIFO 有足够的空间来存储要发送的数据时，TXE 标志将置 1。TXE 标志与 TXFIFO 占用水平相关。该标志变为高电平后将一直保持高电平状态，直至 TXFIFO 占用水平小于或等于 FIFO 深度的 1/2。如果 SPIx_CR2 寄存器中的 TXEIE 位置 1，可产生中断。当 TXFIFO 占用水平超过 1/2 时，该位将自动清零。

接收缓冲区非空 (RXNE)

RXNE 标志根据 SPIx_CR2 寄存器中 FRXTH 位的值进行设置：

- 如果 FRXTH 置 1，RXNE 将变为高电平并一直保持高电平状态，直至 RXFIFO 占用水平大于或等于 1/4（8 位）。
- 如果 FRXTH 清零，RXNE 将变为高电平并一直保持高电平状态，直至 RXFIFO 占用水平大于或等于 1/2（16 位）。

如果 SPIx_CR2 寄存器中的 RXNEIE 位置 1，可产生中断。

当上述条件不再为真时，RXNE 将由硬件自动清零。

忙标志 (BSY)

BSY 标志由硬件置 1 和清零（写入此标志没有任何作用）。

当 BSY 置 1 时，表示 SPI 上正在进行数据传输（SPI 总线繁忙）。

某些模式下可以使用 BSY 标志来检测传输是否结束，以便软件在进入低功耗模式（该模式下不提供外设时钟）前禁止 SPI 或其外设时钟。这可避免破坏最后一个数据的传输。

BSY 标志还可用于避免在多主模式系统中发生写冲突。

在以下任意一种条件下，BSY 标志将清零：

- 正确禁止 SPI 时
- 在主模式下检测到故障时（MODF 位置 1）
- 在主模式下，完成了数据发送并且不准备发送任何新数据时
- 在从模式下，BSY 标志在各传输之间的至少一个 SPI 时钟周期内置为“0”时。

注意： *当主器件可以立即处理下一次发送时（例如，如果主器件处于只接收模式或其发送 FIFO 不为空），在主器件侧的传输之间，通信连续且 BSY 标志始终置“1”。尽管从器件并非如此，但建议始终使用 TXE 和 RXNE 标志（而非 BSY 标志）来处理数据发送或接收操作。*

25.5.11 SPI 错误标志

如果以下其中一个错误标志置 1 且已通过将 ERRIE 位置 1 使能了中断，则将生成 SPI 中断。

上溢标志 (OVR)

当主器件或从器件接收了数据但 RXFIFO 没有足够的空间来存储接收的数据时，将出现上溢的情况。如果软件或 DMA 没有足够的时间来读取之前接收的数据（存储在 RXFIFO 中）或数据存储空间受限（例如在只接收模式下使能 CRC 时 RXFIFO 不可用，在这种情况下，接收缓冲区便限制为一个数据帧缓冲区），会发生这种情况（请参见*第 25.5.14 节：CRC 计算*）。

当出现上溢的情况时，新接收的值不会覆盖 RXFIFO 中之前的值。新接收的值将被丢弃，之后发送的所有数据都将丢失。要将 OVR 位清零，应首先对 SPI_DR 寄存器执行读访问，然后再对 SPI_SR 寄存器执行读访问。

模式故障 (MODF)

当主器件的内部 NSS 信号（NSS 硬件模式下为 NSS 引脚，NSS 软件模式下为 SSI 位）被拉低时，将发生模式故障。这会自动将 MODF 位置 1。主模式故障会在以下几方面影响 SPI 接口：

- 如果 ERRIE 位置 1，MODF 位将置 1，并生成 SPI 中断。
- SPE 位清零。这将关闭器件的所有输出，并禁止 SPI 接口。
- MSTR 位清零，从而强制器件进入从模式。

使用以下软件序列将 MODF 位清零：

1. 在 MODF 位置 1 时，对 SPIx_SR 寄存器执行读或写访问。
2. 然后，对 SPIx_CR1 寄存器执行写操作。

为避免包含多个 MCU 的系统中发生多从模式冲突，必须在 MODF 位清零序列期间将 NSS 引脚拉高。在该清零序列后，可以将 SPE 和 MSTR 位恢复到原始状态。安全起见，硬件不允许在 MODF 位置 1 时将 SPE 和 MSTR 位置 1。在从器件中，MODF 位不可置 1，但由前一次多主模式冲突引起时除外。

CRC 错误 (CRCERR)

当 SPIx_CR1 寄存器中的 CRCEN 位置 1 时，此标志用于验证接收数据的有效性。如果移位寄存器中接收的值与 SPIx_RXCRCR 的值不匹配，SPIx_SR 寄存器中的 CRCERR 标志将置 1。该标志由软件清零。

TI 模式帧格式错误 (FRE)

如果 SPI 在从模式下工作，并配置为符合 TI 模式协议，则在通信进行期间出现 NSS 脉冲时，将检测到 TI 模式帧格式错误。出现此错误时，SPIx_SR 寄存器中的 FRE 标志将置 1。发生错误时不会禁止 SPI，但会忽略 NSS 脉冲，并且 SPI 会等待下一个 NSS 脉冲，然后再开始新的传输。由于错误检测可能导致丢失两个数据字节，因此数据可能会损坏。

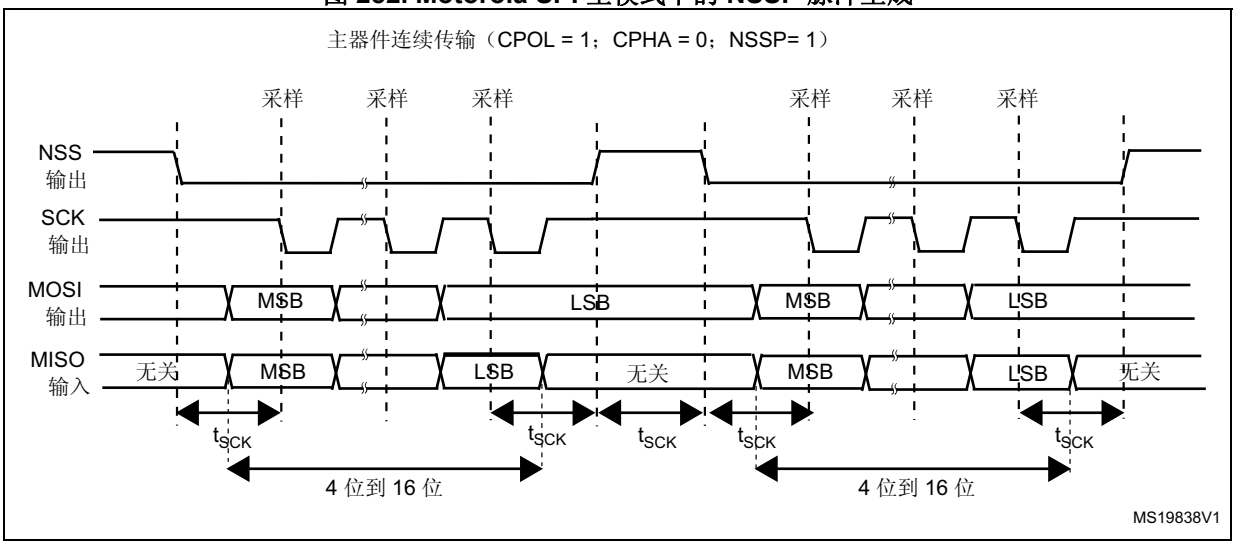
读取 SPIx_SR 寄存器时，将清零 FRE 标志。如果 ERRIE 位置 1，则检测到 NSS 错误时将生成中断。在这种情况下，由于无法保证数据的一致性，应禁止 SPI，并在重新使能从 SPI 后，由主器件重新发起通信。

25.5.12 NSS 脉冲模式

该模式通过 SPIx_CR2 寄存器中的 NSSP 位来激活，只有将 SPI 接口配置为 Motorola SPI 主模式 (FRF=0) 且在第一个边沿捕捉时，该模式才起作用（SPIx_CR1 CPHA = 0，CPOL 设置忽略）。激活后，当 NSS 至少保持一个时钟周期的高电平状态时，两个连续的数据帧传输间将生成 NSS 脉冲。该模式下，从器件可以锁存数据。NSSP 脉冲模式旨在用于具有一个主器件-从器件对的应用。

图 282 给出了使能 NSSP 脉冲模式后的 NSS 引脚管理情况。

图 282. Motorola SPI 主模式下的 NSSP 脉冲生成



注意： 当 CPOL = 0 时会出现类似行为。在这种情况下，采样边沿为 SCK 的*上升*沿，NSS 的使能和禁止均参考该采样边沿。

25.5.13 TI 模式

主模式下的 TI 协议

SPI 接口与 TI 协议兼容。可以使用 SPIx_CR2 寄存器的 FRF 位来配置 SPI，以兼容此协议。

时钟极性和相位都被强制为遵循 TI 协议，和 SPIx_CR1 中的设置无关。NSS 管理也特定于 TI 协议，在这种情况下，无法通过 SPIx_CR1 和 SPIx_CR2 寄存器（SSM、SSI 和 SSOE）来对 NSS 管理进行配置。

在从模式下，SPI 波特率预分频器用于控制在当前传输完成时 MISO 引脚切换为高阻态的时刻（请参见*图 283*）。可以使用任意波特率，因此可以非常灵活地确定此时刻。但是，波特率通常设置为外部主时钟波特率。MISO 信号变为高阻态的延时 ($t_{release}$) 取决于内部重新同步以及通过 SPIx_CR1 寄存器的 BR[2:0] 位设置的波特率值。具体公式如下：

$$\frac{t_{baud\_rate}}{2} + 4 \times t_{pclk} < t_{release} < \frac{t_{baud\_rate}}{2} + 6 \times t_{pclk}$$

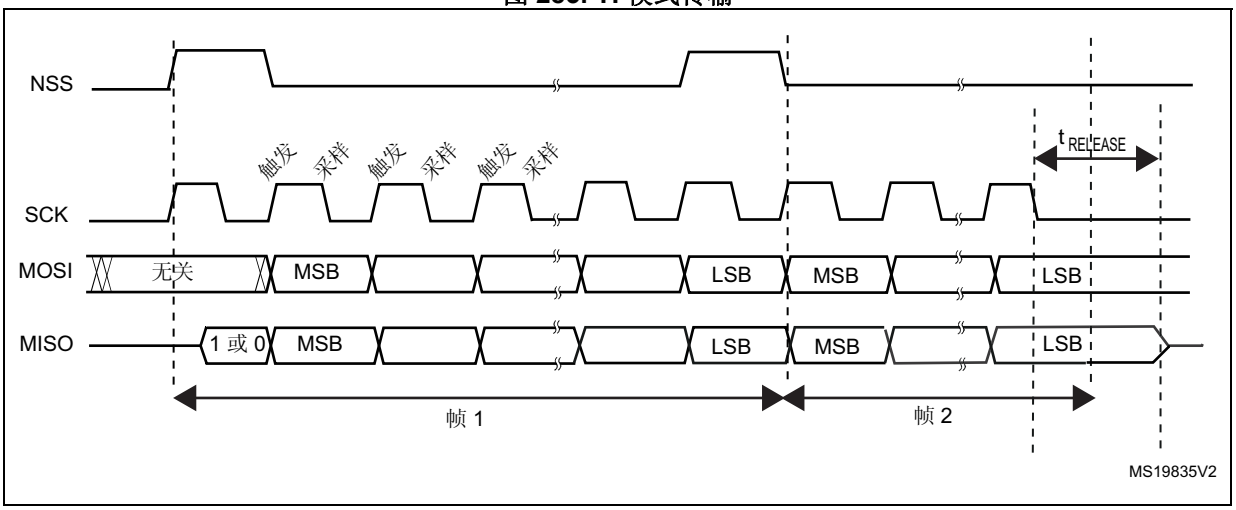
如果从器件在数据帧传输期间检测到错位的 NSS 脉冲，TIFRE 标志将置 1。

如果数据大小等于 4 位或 5 位，处于全双工模式或只发送模式的主器件将使用在 LSB 后再添加一个空数据位的协议。每个周期内，将在该空位（而非 LSB）时钟周期生成 TI NSS 脉冲。

此特性不适用于 Motorola SPI 通信（FRF 位为 0）。

图 283：TI 模式传输 给出了选择 TI 模式时的 SPI 通信波形。

图 283. TI 模式传输



25.5.14 CRC 计算

为检查发送数据和接收数据的可靠性，使用两个独立的 CRC 计算器。SPI 可提供 CRC8 或 CRC16 计算，而与固定为 8 位或 16 位的数据帧长度无关。对于所有其他的数据帧长度，CRC 均不适用。

CRC 原理

在使能 SPI (SPE = 1) 前，通过将 SPIx_CR1 寄存器中的 CRCEN 位置 1 来使能 CRC 计算。使用值为奇数的可编程多项式对每个位计算 CRC 值。在由 SPIx_CR1 寄存器中的 CPHA 位和 CPOL 位定义的采样时钟边沿进行计算。所计算的 CRC 值在数据块末尾自动进行校验，以及针对由 CPU 或 DMA 管理的传输进行校验。当检测到所接收数据内部计算的 CRC 与发送器发送的 CRC 不匹配时，CRCERR 标志将置 1 以指示存在数据损坏错误。CRC 计算的正确处理步骤取决于 SPI 配置和所选的传输管理。

注意： *多项式值只应为奇数。不支持任何偶数值。*

CPU 管理的 CRC 传输

通信开始后将一直持续到必须发送或接收 SPIx_DR 寄存器中的最后一个数据帧时。之后，SPIx_CR1 寄存器中的 CRCNEXT 位必须置 1，以指示当前处理的数据帧传输后将处理 CRC 帧传输。CRCNEXT 位必须在最后一个数据帧传输结束前置 1。在 CRC 传输期间，CRC 计算将冻结。

所接收的 CRC 以数据字节或数据字的形式存储在 RXFIFO 中。因此仅在 CRC 模式下，接收缓冲区才必须被视为一个 16 位缓冲区且一次仅接收一个数据帧。

CRC 帧的传输通常在数据序列结束时再传输一个数据帧。然而，在设置通过 16 位 CRC 校验的 8 位数据帧时，还需要另外两个帧才能发送完整的 CRC。

接收最后一个 CRC 数据后，将执行自动校验，将接收的值与 SPIx_RXCRC 寄存器中的值进行比较。软件必须校验 SPIx_SR 寄存器中的 CRCERR 标志，以确定数据传输是否损坏。软件通过向 CRCERR 标志写入“0”来将其清零。

接收 CRC 后，CRC 值存储到 RXFIFO 中，且必须在 SPIx_DR 寄存器中进行读取，以将 RXNE 标志清零。

DMA 管理的 CRC 传输

当使能的 SPI 通信支持 CRC 通信和 DMA 模式时，在通信结束时会自动发送和接收 CRC（在只接收模式下读取 CRC 数据时除外）。CRCNEXT 位不是一定要通过软件来处理。SPI 发送 DMA 通道计数器必须设置为要发送的数据帧数，其中不包括 CRC 帧。在接收器侧，接收的 CRC 值在传输结束时通过 DMA 自动处理，但用户必须注意刷新 RXFIFO 中接收的 CRC 信息（因为该信息始终加载到其中）。在全双工模式下，接收 DMA 通道计数器可设置为要接收的数据帧数，其中包括 CRC，这意味着在通过 16 位 CRC 校验的 8 位数据帧的特殊情况下：

DMA_RX = Numb_of_data + 2

在只接收模式下，DMA 接收通道计数器应仅包含已传输的数据量，而不包括 CRC 计算。之后，基于通过 DMA 实现的完整传输，必须通过软件从 FIFO 中读回所有 CRC 值，因为 FIFO 在该模式下用作单个缓冲区。

如果传输过程中出现损坏，则在数据和 CRC 传输结束时，SPIx_SR 寄存器中的 CRCERR 标志将置 1。

如果使用封装模式，则 LDMA_RX 位需要管理数据数是否为奇数。

复位 SPIx_TXCRC 和 SPIx_RXCRC 值

在 CRC 阶段后对新数据进行采样时，SPIx_TXCRC 和 SPIx_RXCRC 值将自动清零。借此，可使用 DMA 循环模式（只接收模式下不适用）从而不间断地传输数据（中间 CRC 校验阶段会涉及一些数据块）。

如果在通信过程中禁止 SPI，必须按以下顺序进行操作：

1. 禁止 SPI
2. 将 CRCEN 位清零
3. 使能 CRCEN 位
4. 使能 SPI

注意： *当 SPI 接口配置为从模式时，一旦 CRCNEXT 信号被释放，NSS 内部信号需要在 CRC 阶段事务期间保持低电平。这就是当 NSS 硬件模式正常应用于从器件时，不能在 NSS 脉冲模式下使用 CRC 计算的原因。*

在 TI 模式下，尽管时钟相位和时钟极性设置固定并且与 SPIx_CR1 寄存器无关，但如果应用 CRC，则 SPIx_CR1 寄存器中必须保持相应的设置（CPOL = 0，CPHA = 1）。此外，CRC 计算必须通过 SPI 禁止序列在会话之间复位，方法是在主器件和从器件两侧重新使能上述 CRCEN 位，否则 CRC 计算可能在该特定模式下损坏。

25.6 SPI 中断

在 SPI 通信过程中，中断可由以下事件产生：

- 发送 TXFIFO 准备就绪，可以装载数据
- 接收 RXFIFO 中接收了数据
- 主模式故障
- 上溢错误
- TI 帧格式错误
- CRC 协议错误

中断可分别进行使能和禁止。

表 115. SPI 中断请求

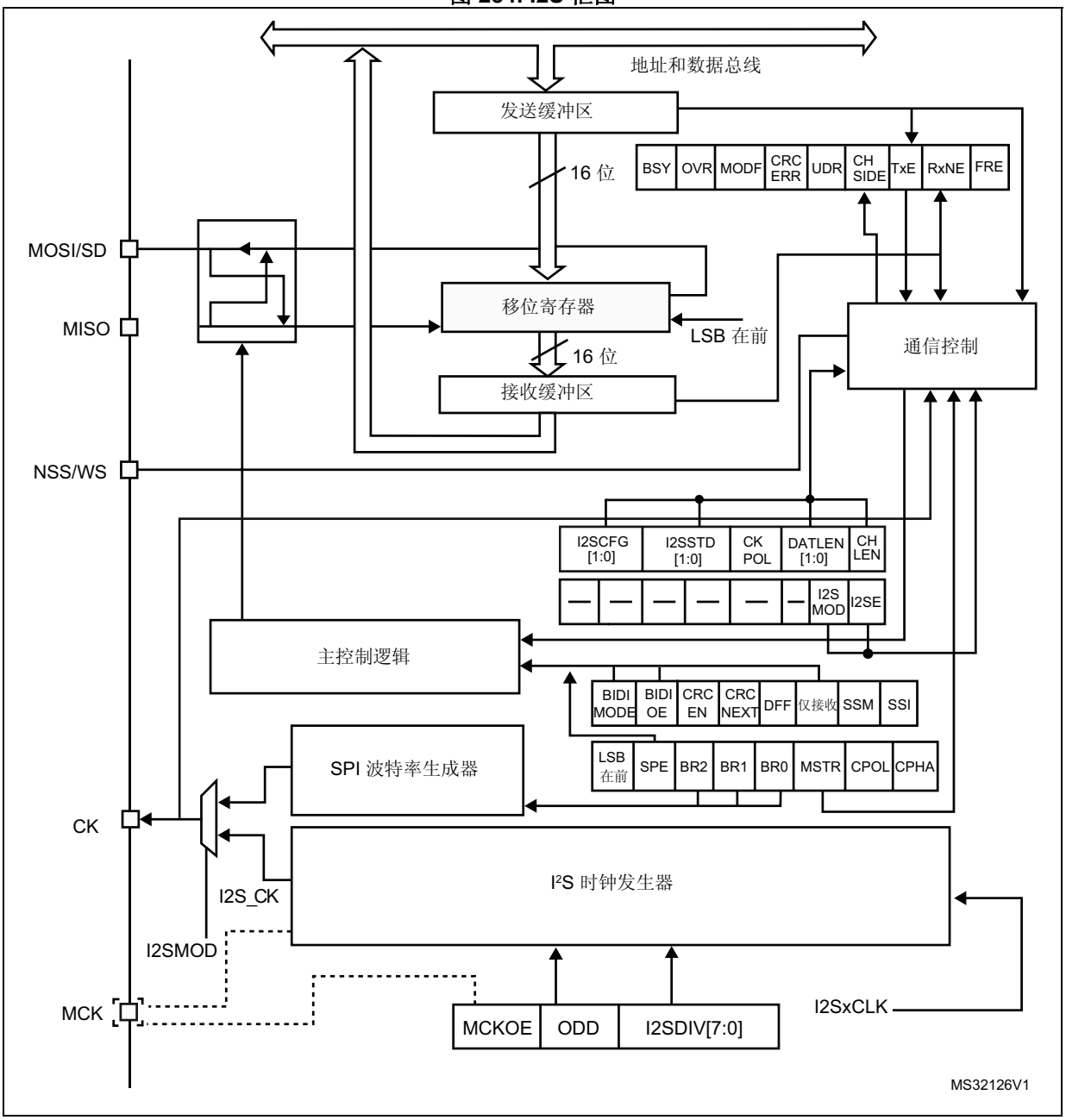
中断事件	事件标志	使能控制位
发送 TXFIFO 准备就绪，可以装载数据	TXE	TXEIE
RXFIFO 中接收了数据	RXNE	RXNEIE
主模式故障事件	MODF	ERRIE
上溢错误	OVR	
TI 帧格式错误	FRE	
CRC 协议错误	CRCERR	

25.7 I2S 功能描述

25.7.1 I2S 一般说明

I2S 的框图如*图 284* 所示。

图 284. I2S 框图



1. MCK 映射到 MISO 引脚上。

当使能 I2S 功能（将 SPIx_I2SCFGR 寄存器中的 I2SMOD 位置 1）时，SPI 可用作音频 I2S 接口。此接口使用几乎与 SPI 相同的引脚、标志和中断。

I2S 与 SPI 共用以下三个引脚：

- SD：串行数据（映射到 MOSI 引脚），用于发送或接收两个时分复用的数据通道上的数据（仅半双工模式）。
- WS：字选择（映射到 NSS 引脚），是主模式下的数据控制信号输出以及从模式下的数据控制信号输入。
- CK：串行时钟（映射到 SCK 引脚），是主模式下的串行时钟输出以及从模式下的串行时钟输入。

当某些外部音频设备需要使用主时钟输出时，可以使用其他引脚：

- MCK：当 I2S 配置为主模式（并且 SPIx_I2SPR 寄存器中的 MCKOE 位置 1）时，使用主时钟（单独映射）输出此附加时钟，该时钟以 256 × f_S（对于所有 I2S 模式）或 128 x f_S（对于所有 PCM 模式）的预配置频率生成，其中 f_S 为音频信号采样频率。

I2S 在主模式下使用自身的时钟发生器生成通信时钟。此时钟发生器也是主时钟输出的源。在 I^2S 模式下可以使用两个额外的寄存器。一个是时钟发生器配置寄存器 SPIx_I2SPR，另一个是通用 I2S 配置寄存器 SPIx_I2SCFGR（音频标准、从/主模式、数据格式、数据包帧、时钟极性等）。

在 I^2S 模式下不使用 SPIx_CR1 寄存器和所有 CRC 寄存器。同样，也不使用 SPIx_CR2 寄存器中的 SSOE 位以及 SPIx_SR 中的 MODF 和 CRCERR 位。

I2S 使用和 SPIx 相同的寄存器 (SPI_DR) 进行 16 位数据传输。

25.7.2 支持的音频协议

三线总线仅需要处理通常在左右两个通道上时分复用的音频数据。但是，只有一个 16 位寄存器进行发送和接收。所以，需由软件将与每个通道对应的值写入数据寄存器，或者从数据寄存器中读取数据，并通过检查 SPIx_SR 寄存器中的 CHSIDE 位来识别对应的通道。始终先发送左通道数据，而后再发送右通道数据（对于 PCM 协议来说，CHSIDE 没有意义）。

数据和帧格式组合有四种，可采用下列格式发送数据：

- 将 16 位数据封装在 16 位帧中
- 将 16 位数据封装在 32 位帧中
- 将 24 位数据封装在 32 位帧中
- 将 32 位数据封装在 32 位帧中

当使用 16 位数据扩展到 32 位数据帧的格式时，前 16 位 (MSB) 为有效位，16 位 LSB 被强制清零，无需任何软件操作或 DMA 请求（只需一个读/写操作）。

24 位和 32 位数据帧需要对 SPIx_DR 寄存器执行两次 CPU 读取或写入操作，或者当采样 DMA 时，则需要两次 DMA 操作。对于 24 位数据帧，硬件会在低 8 位填充 8 个 0 扩展到 32 位。

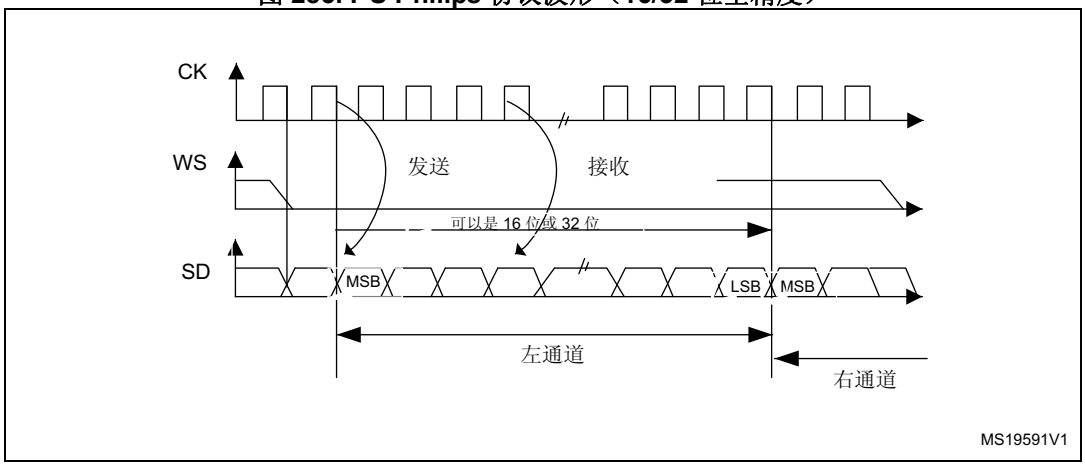
对于所有数据格式和通信标准而言，始终会先发送最高有效位（MSB 优先）。

I^2S 接口支持四种音频标准，可使用 SPIx_I2SCFGR 寄存器中的 I2SSTD[1:0] 和 PCMSYNC 位对其进行配置。

I^2S Philips 标准

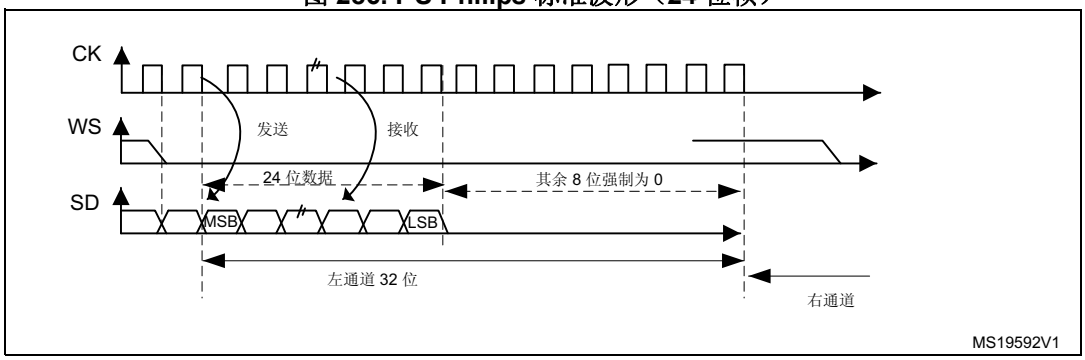
使用 WS 信号来指示当前正在发送的数据所属的通道。该信号从当前通道数据的第一个位 (MSB) 之前的一个时钟开始有效。

图 285. I²S Philips 协议波形（16/32 位全精度）



发送方在时钟信号 (CK) 的下降沿改变数据，接收方在上升沿读取数据。WS 信号也在 CK 的下降沿变化。

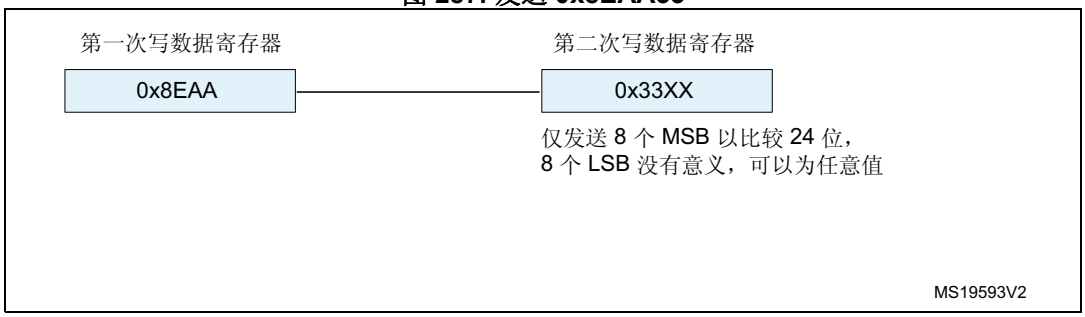
图 286. I²S Philips 标准波形（24 位帧）



该模式需要对 SPIx_DR 寄存器执行两次写入或读取操作。

- 在发送模式下：
  如果需要发送 0x8EAA33（24 位）：

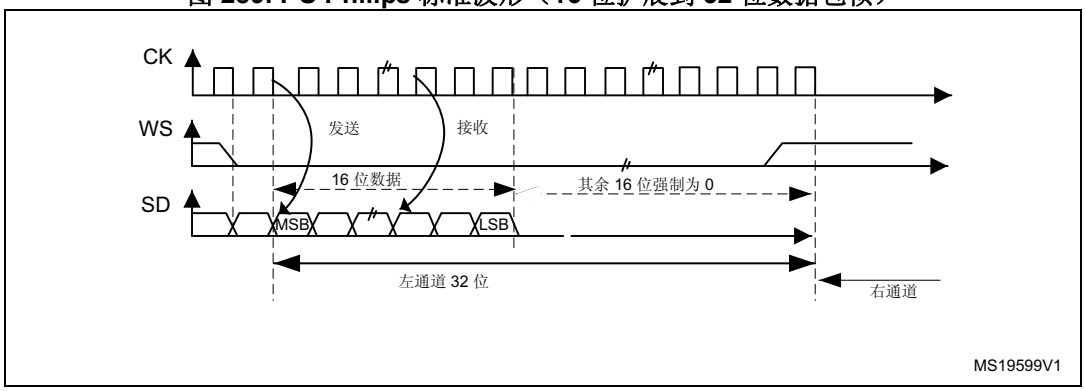
图 287. 发送 0x8EAA33



- 在接收模式下：
  如果接收数据 0x8EAA33：

图 288. 接收 0x8EAA33

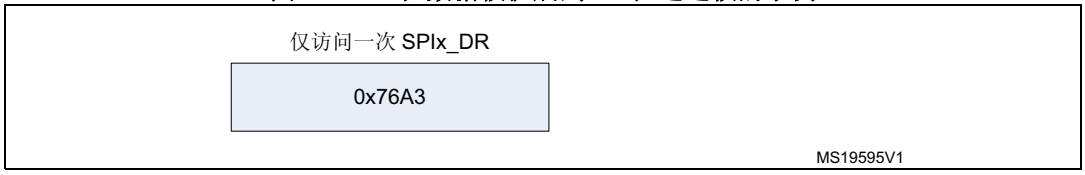
图 289. I²S Philips 标准波形（16 位扩展到 32 位数据包帧）



如果在 I2S 配置阶段选择将 16 位数据帧扩展到 32 位通道帧，则只需要访问一次 SPIx_DR 寄存器。扩展到 32 位中高半字（16 位 MSB）被硬件置为 0x0000。

如果要发送的数据或已接收的数据为 0x76A3（0x76A30000 扩展为 32 位），则需要执行 *图 290* 中显示的操作。

图 290. 16 位数据帧扩展到 32 位通道帧的示例



发送时，每次将 MSB 写入 SPIx_DR，TXE 标志就会置 1，并在中断使能的情况下触发中断，以将要发送的新数据加载到 SPIx_DR 寄存器。即使硬件填充的低 16 位 0x0000 还未发送，也会如此，因为低 16 位是由硬件发送。

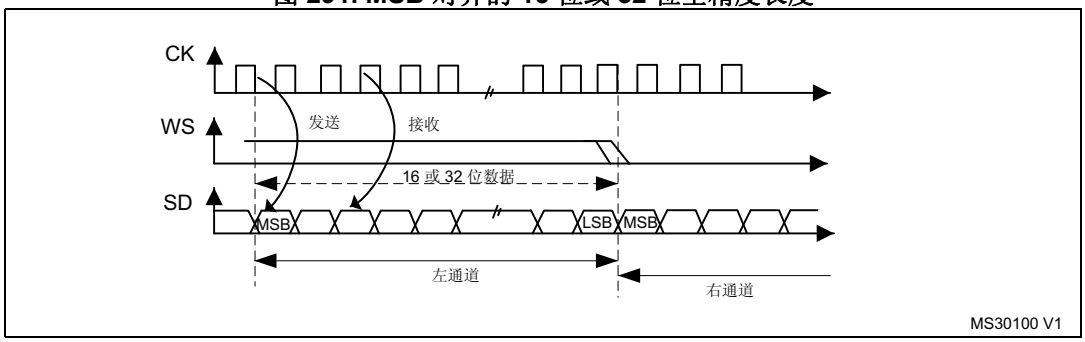
接收时，接收到第一个半字（高 16 位），则硬件将 RXNE 标志置 1，并在中断使能的情况下触发中断。

这样，就延长了两个写入或读取操作之间的时间间隔，从而可防止出现下溢或上溢情况（具体取决于数据传输方向）。

MSB 对齐标准

此标准同时生成 WS 信号和第一个数据位（即 MSBit）。

图 291. MSB 对齐的 16 位或 32 位全精度长度



发送方在时钟信号的下降沿改变数据；接收方在上升沿读取数据。

图 292. MSB 对齐的 24 位帧长度

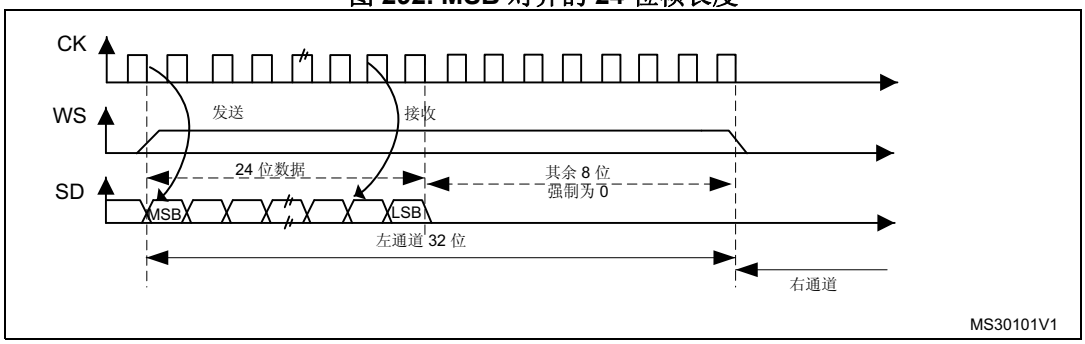
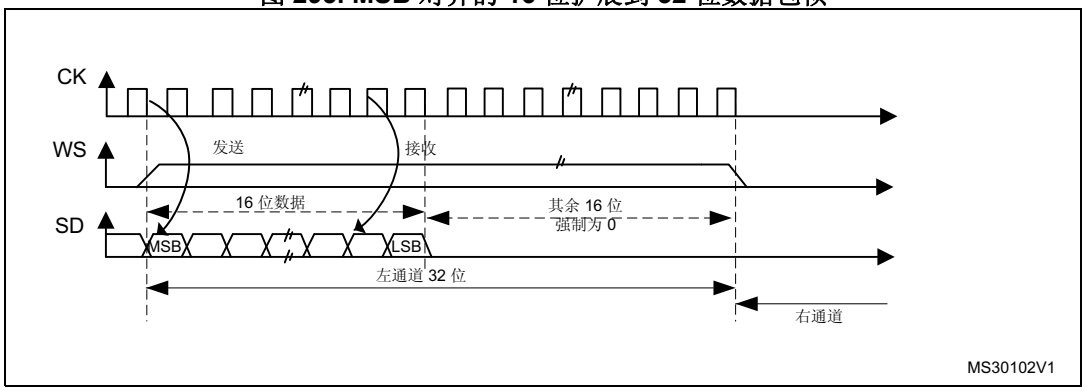


图 293. MSB 对齐的 16 位扩展到 32 位数据包帧



LSB 对齐标准

该标准与 MSB 对齐标准类似（对于 16 位和 32 位全精度帧格式，没有任何不同）。

输入信号和输出信号的采样符合 I²S Philips 标准。

图 294. LSB 对齐的 16 位或 32 位全精度长度

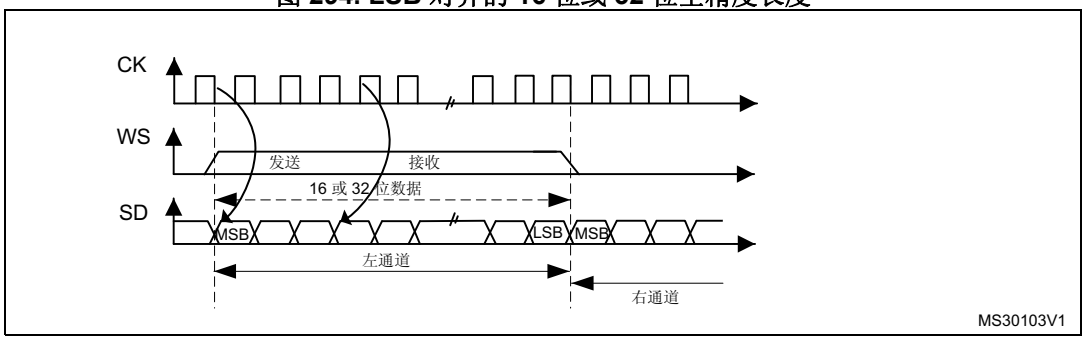
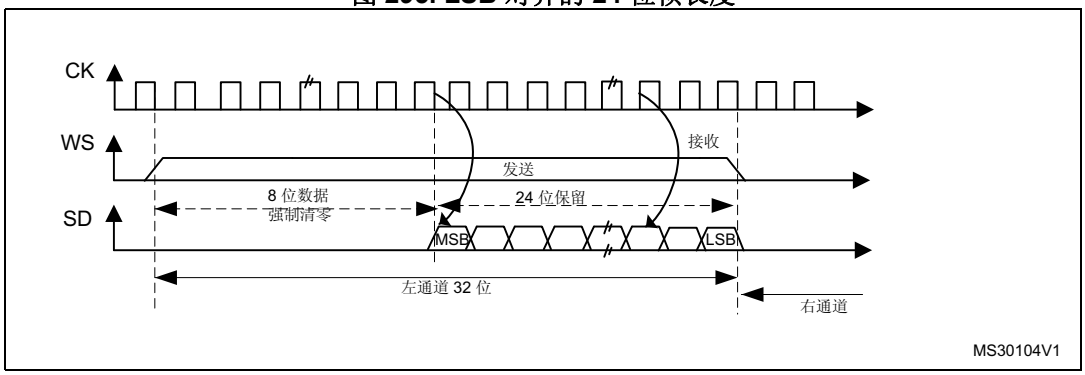


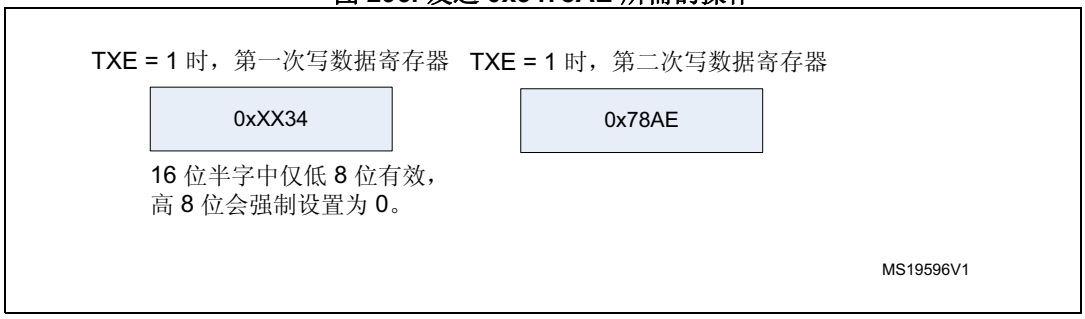
图 295. LSB 对齐的 24 位帧长度



- 在发送模式下：

  如果需要发送数据 0x3478AE，则需要通过软件或 DMA 对 SPIx_DR 寄存器执行两次写入操作。操作如下所示。

图 296. 发送 0x3478AE 所需的操作



- 在接收模式下：
  如果接收到数据 0x3478AE，则在每个 RXNE 事件时需要对 SPIx_DR 寄存器执行两次连续的读取操作。

图 297. 接收 0x3478AE 时所需的操作

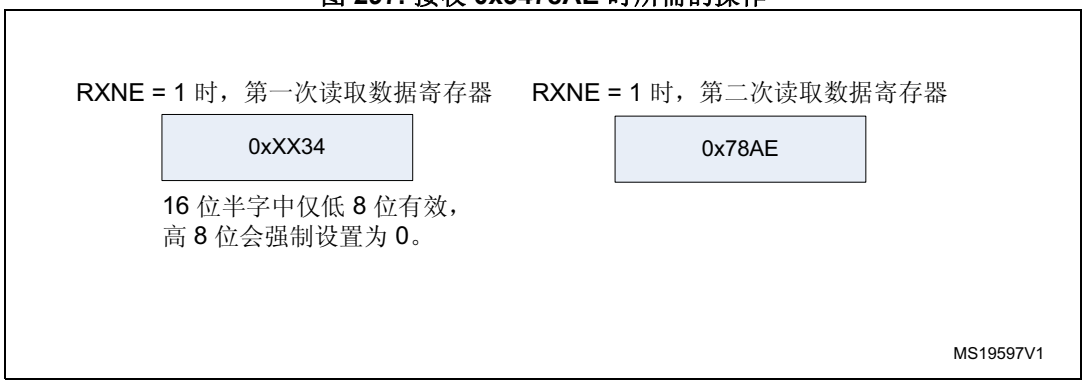
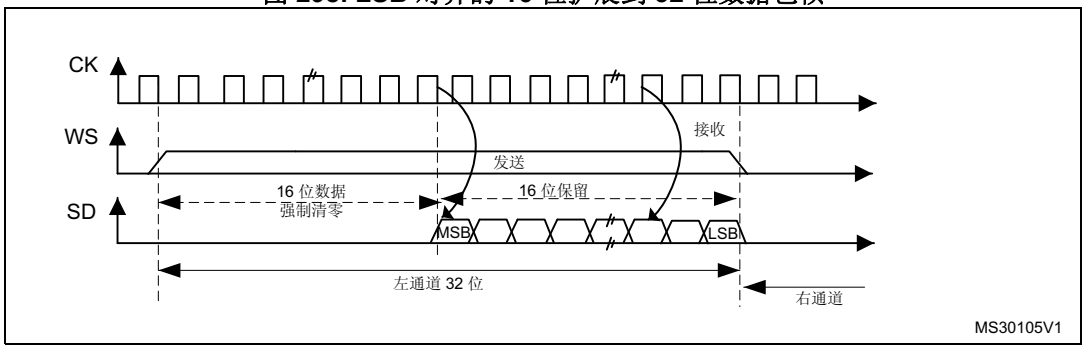


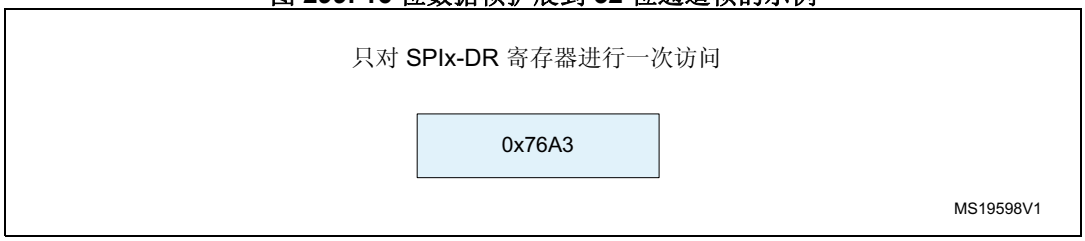
图 298. LSB 对齐的 16 位扩展到 32 位数据包帧



如果在 I2S 配置阶段选择将 16 位数据帧扩展到 32 位通道帧，则只需要访问一次 SPIx_DR 寄存器。扩展到 32 位中高半字（16 位 MSB）被硬件置为 0x0000。在这种情况下，其对应于半字 MSB。

如果要发送的数据或已接收的数据为 0x76A3（0x0000 76A3 扩展为 32 位），则需要执行 *图 299* 中显示的操作。

图 299. 16 位数据帧扩展到 32 位通道帧的示例



在发送模式下，发生 TXE 事件时，应用程序需要写入要发送的数据（此例中，为 0x76A3）。首先发送 0x000 字段（扩展到 32 位）。有效数据 (0x76A3) 发送到 SD 后，TXE 标志会被再次置 1。

在接收模式下，当接收到有效半字后（而非 0x0000 字段），即会置位 RXNE。

这样，就延长了两个写入或读取操作之间的时间间隔，以防止出现下溢或上溢情况。

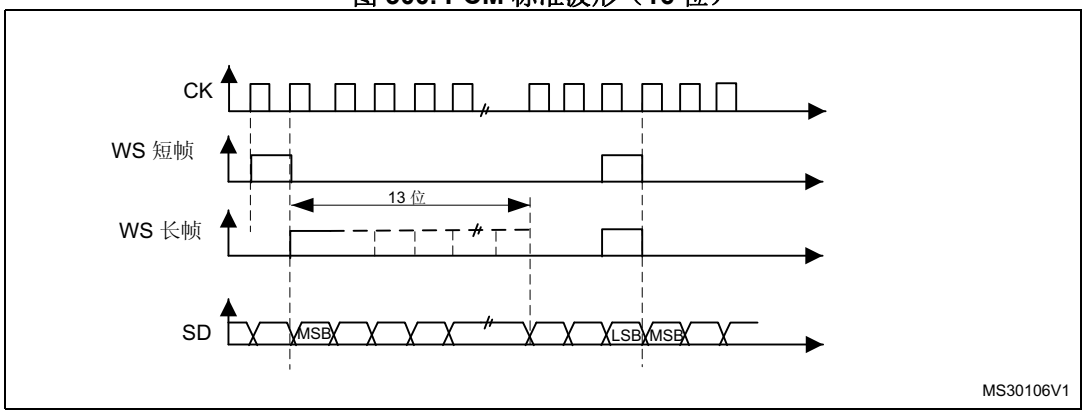
PCM 标准

对于 PCM 标准，无需使用通道信息。可使用两种 PCM 模式（短帧和长帧），并且可使用 SPIx_I2SCFGR 寄存器中的 PCMSYNC 位来配置。

在 PCM 模式下，输出信号（WS、SD）在 CK 信号的上升沿采样。输入信号（WS、SD）在 CK 的下降沿捕获。

请注意，CK 和 WS 在主模式下配置为输出。

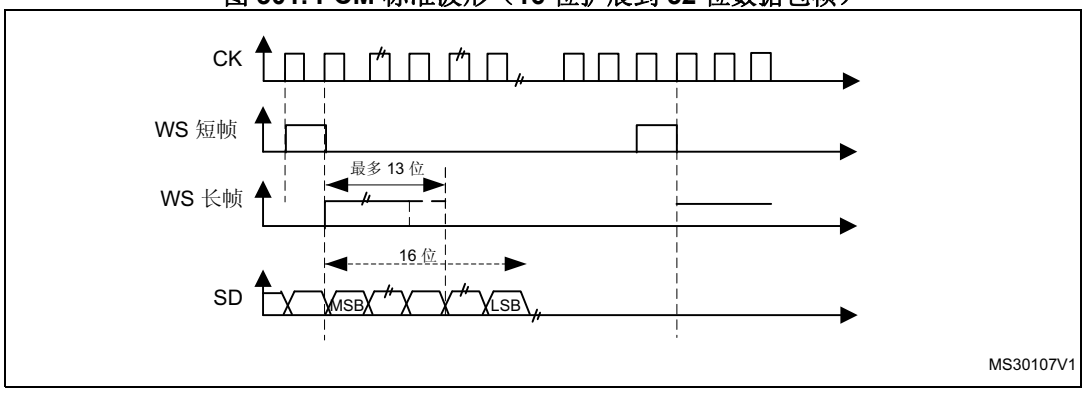
图 300. PCM 标准波形（16 位）



对于长帧同步，在主模式下会将 WS 信号持续 13 个周期。

对于短帧同步，WS 同步信号的持续时间仅为一个周期。

图 301. PCM 标准波形（16 位扩展到 32 位数据包帧）

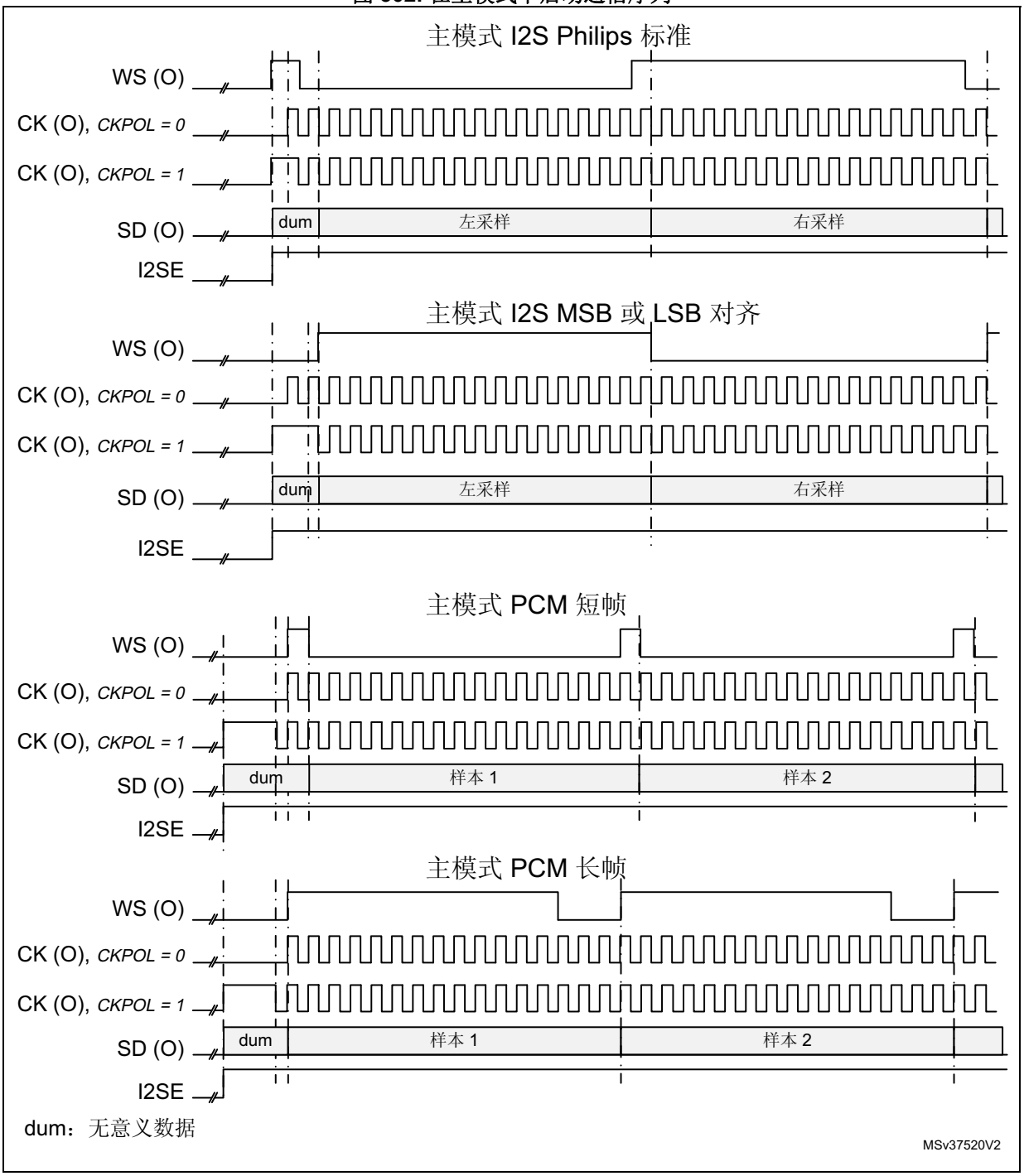


注意： *对于两种模式（主/从模式）和两种同步（短/长同步），即使在从模式下，也需要指定两组连续数据（以及两个同步信号）之间位的个数（SPIx_I2SCFGR 寄存器中的 DATLEN 位和 CHLEN 位）。*

25.7.3 启动说明

图 302 显示了使能 SPI/I2S 时（通过 I2SE 位），如何在主模式下处理串行接口。其中还显示了 CKPOL 对所生成信号的影响。

图 302. 在主模式下启动通信序列



在从模式下，检测帧同步的方式取决于 ASTRTEN 位的值。

如果 ASTRTEN = 0，当使能音频接口时 (I2SE = 1)，硬件使用 CK 信号等待传入 WS 信号发生适当的转换。

使用 I^2S Philips 标准时，适当的边沿为 WS 信号的下降沿；使用其他标准时，则为上升沿。先后采样到 WS 从 1 变为 0 即表示检测到下降沿，先后采样到 WS 从 0 变为 1 则表示检测到上升沿。

如果 ASTRTEN = 1，用户必须在 WS 激活之前使能音频接口。这意味着当 WS = 1（对于 I^2S Philips 标准）或 WS = 0（对于其他标准）时，必须将 I2SE 位置 1。

25.7.4 时钟发生器

I^2S 比特率用来确定 I^2S 数据线上的数据流和 I^2S 时钟信号频率。

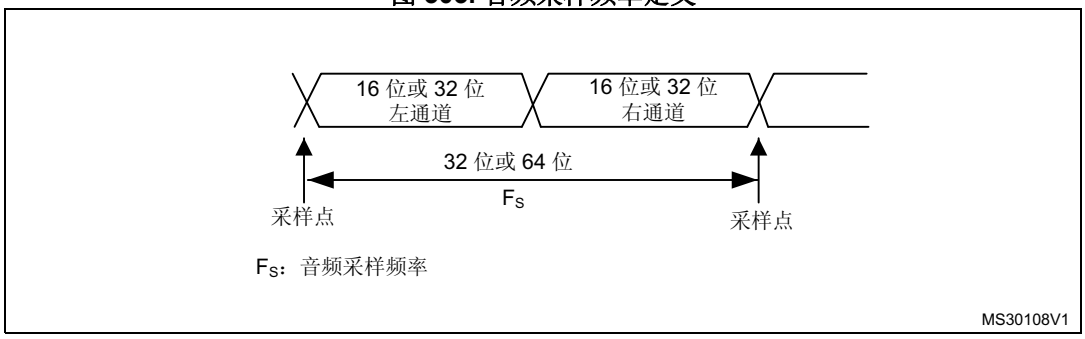
I^2S 比特率 = 每个通道的位数 × 通道数 × 音频采样频率

对于 16 位双通道音频，I^2S 比特率的计算公式如下：

I^2S 比特率 = 16 × 2 × f_S

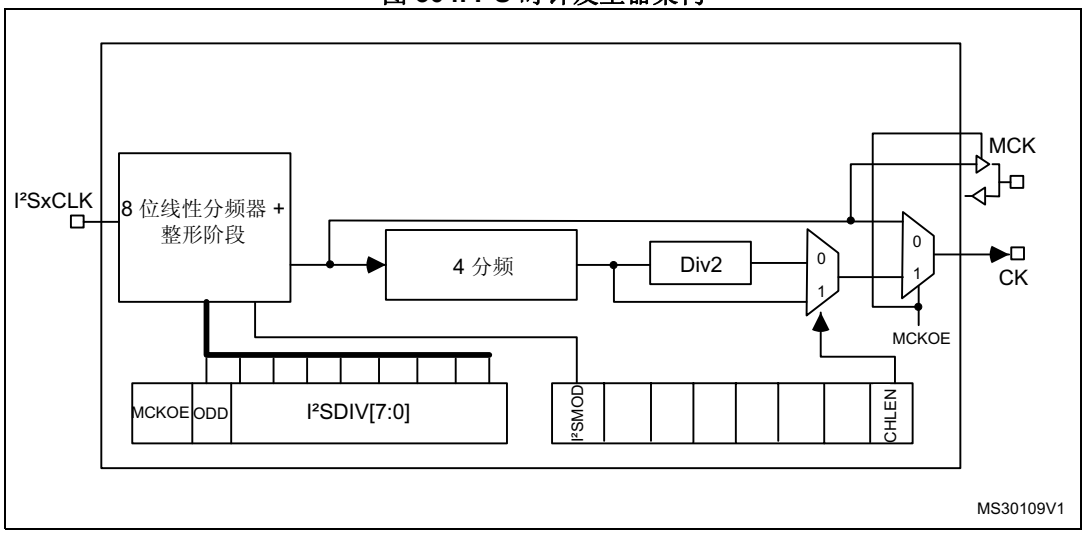
或者：如果数据包为 32 位宽，则 I^2S 比特率 = 32 x 2 x f_S。

图 303. 音频采样频率定义



配置主模式时，需要正确地对线性分频器进行设置，以便采用所需的音频频率进行通信。

图 304. I^2S 时钟发生器架构



1. 其中 x 可以是 2 或 3。

图 304 展示了通信时钟架构。I2SxCLK 时钟由产品的复位和时钟控制器 (RCC) 提供。I2SxCLK 时钟可以与 SPI/I2S APB 时钟异步。

警告： **此外，必须使 I2SxCLK 频率始终高于或等于 SPI/I2S 模块所使用的 APB 时钟。如果不满足这一条件，SPI/I2S 将无法正常工作。**

音频采样频率可以是 192 kHz、96 kHz、48 kHz、44.1 kHz、32 kHz、22.05 kHz、16 kHz、11.025 kHz 或 8 kHz（或此范围内的任何其他值）。

为达到所需频率，需要根据以下公式对线性分频器进行编程：

对于 I²S 模式：

输出主时钟（SPIx_I2SPR 寄存器中的 MCKOE 置 1）时：

$$Fs = \frac{F_{I2SxCLK}}{256 \times ((2 \times I2SDIV) + ODD)}$$

禁止主时钟输出（MCKOE 位清零）时：

$$Fs = \frac{F_{I2SxCLK}}{32 \times (CHLEN + 1) \times ((2 \times I2SDIV) + ODD)}$$

当通道帧宽度为 16 位时，CHLEN = 0；
当通道帧宽度为 32 位时，CHLEN = 1。

对于 PCM 模式：

输出主时钟（SPIx_I2SPR 寄存器中的 MCKOE 置 1）时：

$$Fs = \frac{F_{I2SxCLK}}{128 \times ((2 \times I2SDIV) + ODD)}$$

禁止主时钟输出（MCKOE 位清零）时：

$$Fs = \frac{F_{I2SxCLK}}{16 \times (CHLEN + 1) \times ((2 \times I2SDIV) + ODD)}$$

当通道帧宽度为 16 位时，CHLEN = 0；
当通道帧宽度为 32 位时，CHLEN = 1。

其中，F_S 表示音频采样频率，$F_{I2SxCLK}$ 表示提供给 SPI/I2S 模块的内核时钟的频率。

注意： *I2SDIV 必须绝对大于 1。*

下表提供了针对不同时钟配置的示例精度值。

注意： *还可以采用其他配置以达到更好的时钟精度。*

表 116. 使用源自 HSE 的 48 MHz 时钟时的音频频率精度(1)

SYSCLK (MHz)	数据长度	I2SDIV	I2SODD	MCLK	目标 fs (Hz)	实际 fs (kHz)	误差
48	16	8	0	无	96000	93750	2.3438%
48	32	4	0	无	96000	93750	2.3438%
48	16	15	1	无	48000	48387.0968	0.8065%
48	32	8	0	无	48000	46875	2.3438%
48	16	17	0	无	44100	44117.647	0.0400%
48	32	8	1	无	44100	44117.647	0.0400%
48	16	23	1	无	32000	31914.8936	0.2660%
48	32	11	1	无	32000	32608.696	1.9022%
48	16	34	0	无	22050	22058.8235	0.0400%
48	32	17	0	无	22050	22058.8235	0.0400%
48	16	47	0	无	16000	15957.4468	0.2660%
48	32	23	1	无	16000	15957.447	0.2660%
48	16	68	0	无	11025	11029.4118	0.0400%
48	32	34	0	无	11025	11029.412	0.0400%
48	16	94	0	无	8000	7978.7234	0.2660%
48	32	47	0	无	8000	7978.7234	0.2660%
48	16	2	0	有	48000	46875	2.3430%
48	32	2	0	有	48000	46875	2.3430%
48	16	2	0	有	44100	46875	6.2925%
48	32	2	0	有	44100	46875	6.2925%
48	16	3	0	有	32000	31250	2.3438%
48	32	3	0	有	32000	31250	2.3438%
48	16	4	1	有	22050	20833.333	5.5178%
48	32	4	1	有	22050	20833.333	5.5178%
48	16	6	0	有	16000	15625	2.3438%
48	32	6	0	有	16000	15625	2.3438%
48	16	8	1	有	11025	11029.4118	0.0400%
48	32	8	1	有	11025	11029.4118	0.0400%
48	16	11	1	有	8000	8152.17391	1.9022%
48	32	11	1	有	8000	8152.17391	1.9022%

1. 该表格仅给出了不同时钟配置的示例值。还可以采用其他配置以达到更好的时钟精度。

25.7.5 I^2S 主模式

I2S 可配置为主模式。I2S 工作在主模式，串行时钟由引脚 CK 输出，字选信号由引脚 WS 产生。可以通过设置寄存器 SPIx_I2SPR 的 MCKOE 位来控制输出或者不输出主时钟 (MCK)。

步骤

1. 设置 SPIx_I2SPR 寄存器的 I2SDIV[7:0] 位，以定义串行时钟波特率，从而达到适当的音频采样频率。SPIx_I2SPR 寄存器的 ODD 位也需要设置。
2. 设置 CKPOL 位，定义时钟在空闲时的电平状态。如果需要为外部 DAC/ADC 音频组件提供主时钟 MCK，则将 SPIx_I2SPR 寄存器的 MCKOE 位置 1（I2SDIV 和 ODD 值应根据 MCK 输出的状态进行计算。有关详细信息，请参见*第 25.7.4 节：时钟发生器*）。
3. 将 SPIx_I2SCFGR 寄存器中的 I2SMOD 位置 1 以激活 I2S 功能，通过 I2SSTD[1:0] 和 PCMSYNC 位选择 I^2S 标准，通过 DATLEN[1:0] 位选择数据长度并通过配置 CHLEN 位选择每个通道的位数。此外，通过 SPIx_I2SCFGR 寄存器的 I2SCFG[1:0] 位选择 I^2S 主模式和方向（发送器或接收器）。
4. 如果需要，通过对 SPIx_CR2 寄存器执行写操作来选择所有可能的中断源和 DMA 功能。
5. SPIx_I2SCFGR 寄存器的 I2SE 位必须置 1。

WS 和 CK 配置为输出模式。如果 SPIx_I2SPR 的 MCKOE 位置 1，则 MCK 也是输出。

发送序列

将半字写入发送缓冲区后，发送序列随即开始。

假设写入发送缓冲区的第一个数据对应于左通道数据。数据从发送缓冲区传输到移位寄存器时，TXE 置 1，并且必须将对应于右通道的数据写入发送缓冲区。CHSIDE 标志指示将发送的数据对应的通道。TXE 标志置 1 时，CHSIDE 标志有意义，因为该标志在 TXE 变为高电平时进行更新。

一个完整帧表示先进行左通道数据发送再进行右通道数据发送。不存在仅发送左通道的部分帧。

首位发送期间，数据按半字并行加载到 16 位移位寄存器中，然后以串行方式移位并输出到 MOSI/SD 引脚（MSB 在前）。每次数据从发送缓冲区传输到移位寄存器后，TXE 标志都将置 1，如果 SPIx_CR2 寄存器的 TXEIE 位置 1，将产生中断。

有关各种 I^2S 标准模式的写操作的更多详细信息，请参见*第 25.7.2 节：支持的音频协议*。

为确保连续进行音频数据发送，必须在当前数据发送结束前将下一个要发送的数据写入 SPIx_DR 寄存器。

要通过将 I2SE 清零来关闭 I2S，必须等待 TXE = 1 且 BSY = 0。

接收序列

此工作模式与发送模式相同，只有第 3 点存在不同（请参见*第 25.7.5 节：I²S 主模式*所述的步骤），即通过 I2SCFG[1:0] 位设置主器件接收模式。

无论数据或通道长度如何，音频数据始终按 16 位数据包进行接收。这意味着，每当接收缓冲区满时，RXNE 标志即置 1，并且如果 SPIx_CR2 寄存器的 RXNEIE 位置 1，还将产生中断。所接收的右通道或左通道的音频值可通过一次或两次接收操作进入接收缓冲区，具体取决于数据和通道长度配置。

读取 SPIx_DR 寄存器即会使 RXNE 位清零。

CHSIDE 在每次接收后进行更新。它由 I2S 单元所产生的 WS 信号触发。

有关各种 I²S 标准模式中读操作的更多详细信息，请参见*第 25.7.2 节：支持的音频协议*。

如果在先前收到的数据尚未读取时又接收到新数据，将产生上溢错误并将 OVR 标志置 1。如果 SPIx_CR2 寄存器的 ERRIE 位置 1，将产生中断以指示该错误。

要关闭 I2S，需要执行特定操作来确保 I2S 正确完成传输周期而不启动新的数据传输。该序列取决于数据和通道长度的配置，以及所选的音频协议模式。在以下情况下：

- 32 位通道长度上扩展的 16 位数据长度（DATLEN = 00 且 CHLEN = 1），使用 LSB 对齐模式 (I2SSTD = 10)
  a) 等待倒数第二个 RXNE = 1 (n – 1)
  b) 然后等待 17 个 I2S 时钟周期（使用软件循环）
  c) 禁止 I2S (I2SE = 0)
- 32 位通道长度上扩展的 16 位数据长度（DATLEN = 00 且 CHLEN = 1），使用 MSB 对齐、I²S 或 PCM 模式（分别为 I2SSTD = 00、I2SSTD = 01 或 I2SSTD = 11）
  a) 等待最后一个 RXNE
  b) 然后等待 1 个 I2S 时钟周期（使用软件循环）
  c) 禁止 I2S (I2SE = 0)
- 对于 DATLEN 和 CHLEN 的所有其他组合，无论通过 I2SSTD 位选择何种音频模式，都将执行以下序列来关闭 I2S：
  a) 等待倒数第二个 RXNE = 1 (n – 1)
  b) 然后等待 1 个 I2S 时钟周期（使用软件循环）
  c) 禁止 I2S (I2SE = 0)

注意： *传输期间，BSY 标志保持低电平。*

25.7.6 I²S 从模式

对于从配置而言，I2S 可配置为发送模式或接收模式。此工作模式所遵循的规则与 I²S 主模式配置基本相同。在从模式下，I2S 接口不产生时钟。时钟和 WS 信号从 I2S 接口所连接的外部主器件输入。这样，用户便不需要配置时钟。

应遵循如下配置步骤：

1. 将 SPIx_I2SCFGR 寄存器的 I2SMOD 位置 1 以选择 I²S 模式，通过 I2SSTD[1:0] 位选择 I²S 标准，通过 DATLEN[1:0] 位选择数据长度并通过配置 CHLEN 位选择帧中每个通道的位数。此外，通过 SPIx_I2SCFGR 寄存器的 I2SCFG[1:0] 位选择从器件的模式（发送或接收）。
2. 如果需要，通过对 SPIx_CR2 寄存器执行写操作来选择所有可能的中断源和 DMA 功能。
3. SPIx_I2SCFGR 寄存器的 I2SE 位必须置 1。

发送序列

当外部主器件发送时钟并且通过 NSS_WS 信号请求传输数据时，发送序列开始。必须首先使能从器件，然后外部主器件才能开始通信。主器件开始通信前，从器件还必须加载 I2S 数据寄存器。

对于 I2S、MSB 对齐和 LSB 对齐模式，要写入数据寄存器的第一个数据项对应于左通道的数据。通信开始时，数据从发送缓冲区传输到移位寄存器。TXE 标志随即置 1，以请求将右通道的数据写入 I2S 数据寄存器。

CHSIDE 标志指示将发送的数据对应的通道。与主发送模式相比，在从模式下，CHSIDE 由来自外部主器件的 WS 信号触发。这意味着，从器件需要首先为发送第一个数据做好准备，然后主器件才能产生时钟。WS 置位意味着首先发送左通道数据。

注意： *必须要在主器件发出的第一个时钟出现在 CK 线上至少 2 个 PCLK 周期之前置位 I2SE。*

首位发送期间，数据按半字从内部总线并行加载到 16 位移位寄存器中，然后以串行方式移位并输出到 MOSI/SD 引脚（MSB 在前）。每次数据从发送缓冲区传输到移位寄存器后，TXE 标志都将置 1，如果 SPIx_CR2 寄存器的 TXEIE 位置 1，将产生中断。

请注意，仅当 TXE 标志为 1 时，才可以尝试向发送缓冲区写入数据。

有关各种 I^{2}S 标准模式中写操作的更多详细信息，请参见*第 25.7.2 节：支持的音频协议*。

为确保连续进行音频数据发送，必须在当前数据发送结束前将下一个要发送数据写入 SPIx_DR 寄存器。如果在数据尚未写入 SPIx_DR 寄存器时下一个数据通信的首个时钟边沿到来，下溢标志将置 1 并可能产生中断。通过这种方式，软件可以获知所传输的数据不正确。如果 SPIx_CR2 寄存器的 ERRIE 位置 1，则当 SPIx_SR 寄存器中的 UDR 标志变为 1 时，将产生中断。这种情况下，必须关闭 I2S 并从左通道开始重新启动数据传输。

要通过将 I2SE 位清零来关闭 I2S，必须等待 TXE = 1 且 BSY = 0。

接收序列

此工作模式与发送模式相同，只有第 1 点存在不同（请参见*第 25.7.6 节：I^{2}S 从模式*所述的步骤），即通过 SPIx_I2SCFGR 寄存器的 I2SCFG[1:0] 位设置从器件接收模式。

无论数据长度或通道长度如何，音频数据始终按 16 位数据包进行接收。这意味着，每当接收缓冲区填满时，SPIx_SR 寄存器中的 RXNE 标志即置 1，并且如果 SPIx_CR2 寄存器的 RXNEIE 位置 1，还将产生中断。所接收的右通道或左通道的音频值可能通过一次或两次接收操作进入接收缓冲区，具体取决于数据长度和通道长度配置。

每次接收要从 SPIx_DR 寄存器读取的数据时，CHSIDE 标志都将更新。该标志由外部主器件所管理的外部 WS 线路触发。

读取 SPIx_DR 寄存器即会使 RXNE 位清零。

有关各种 I^{2}S 标准模式的读操作的更多详细信息，请参见*第 25.7.2 节：支持的音频协议*。

如果在先前收到的数据尚未读取时又接收到新数据，将产生上溢错误，OVR 标志将置 1。如果 SPIx_CR2 寄存器的 ERRIE 位置 1，将产生中断以指示该错误。

要在接收模式下关闭 I2S，必须在接收到最后一个 RXNE = 1 后立即将 I2SE 清零。

注意： *外部主器件应能够通过音频通道以 16 位或 32 位数据包发送/接收数据。*

25.7.7 I2S 状态标志

应用程序可通过三个状态标志来全面监视 I2S 总线的状态。

忙标志 (BSY)

BSY 标志由硬件置 1 和清零（写入此标志没有任何作用）。该标志表示 I2S 通信层的状态。

BSY 置 1 时，表示 I2S 正忙于通信。在主接收模式 (I2SCFG = 11) 中，BSY 标志的情况例外，该标志在接收期间仍保持低电平。

如果软件需要禁止 I2S，可使用 BSY 标志检测传输是否结束。这可避免破坏最后一个数据的传输。为此，必须严格遵循下述步骤。

在传输开始时（I2S 处于主接收器模式时除外），硬件将 BSY 标志置 1。

出现以下情况时，BSY 标志被硬件清零：

- 传输完成时（主发送模式除外，在该模式下通信是连续的）
- 禁止 I2S 时

当通信连续时：

- 在主发送模式下，BSY 标志在所有传输期间均保持高电平
- 在从模式下，BSY 标志在各传输之间的一个 I2S 时钟周期内变为低电平

注意： *请勿使用 BSY 标志处理每次数据发送或接收，最好改用 TXE 标志和 RXNE 标志。*

发送缓冲区为空 (TXE)

如果此标志置 1，表示发送缓冲区为空，可将要发送的下一个数据加载到其中。发送缓冲区已包含要发送的数据时，TXE 标志复位。禁止 I2S（I2SE 位复位）时，该标志也会复位。

接收缓冲区非空 (RXNE)

此标志置 1 时，表示接收缓冲区中存在有效的已接收数据。读取 SPIx_DR 寄存器时，该标志复位。

通道方向 (CHSIDE)

在发送模式下，此标志将在 TXE 变为高电平时进行刷新。此标志指示 SD 上要传输的数据所属的通道。如果在从发送模式下发生下溢错误事件，此标志将不可靠，在恢复通信前需要关闭并重新开启 I2S。

在接收模式下，此标志将在数据接收到 SPIx_DR 中时进行刷新。此标志指示已接收的数据所属的通道。请注意，如果发生错误（例如 OVR），此标志将失去意义，应通过关闭并重新使能 I2S（根据需要更改配置）来将其复位。

此标志在 PCM 标准中没有意义（短帧和长帧模式）。

当 SPIx_SR 中的 OVR 或 UDR 标志置 1，并且 SPIx_CR2 中的 ERRIE 位也置 1 时，将产生中断。中断源被清除后，可通过读取 SPIx_SR 状态寄存器来将此中断清除。

25.7.8 I2S 错误标志

I2S 单元共有三个错误标志。

下溢标志 (UDR)

在从发送模式下，如果在软件尚未将任何值加载到 SPIx_DR 之前出现第一个数据发送时钟，此标志将置 1。SPIx_I2SCFGR 寄存器中的 I2SMOD 位置 1 时，可以使用此标志。如果 SPIx_CR2 寄存器中的 ERRIE 位置 1，可产生中断。
UDR 位通过 SPIx_SR 寄存器上的读操作进行清零。

上溢标志 (OVR)

如果在尚未从 SPIx_DR 寄存器读取上一个数据时又接收到新数据，此标志将置 1。因此，传入的数据将丢失。如果 SPIx_CR2 寄存器中的 ERRIE 位置 1，可产生中断。

这种情况下，将不会用接收到的新数据更新接收缓冲区的内容。对 SPIx_DR 寄存器执行的读操作将返回先前正确接收的数据。主器件后续发送的所有其他半字都将丢失。

要将 OVR 位清零，应首先对 SPIx_DR 寄存器执行读操作，然后再对 SPIx_SR 寄存器进行读访问。

帧错误标志 (FRE)

仅当 I2S 配置为从模式时，此标志才可由硬件置 1。如果外部主器件没有按照从器件期望的那样改变 WS 线，则此标志将置 1。如果失去同步，要从此状态中恢复并将外部主器件与 I2S 从器件重新同步，需执行下列步骤：

1. 禁止 I2S。
2. 在 WS 线上检测到正确的电平时将其重新使能（WS 线在 I^2S 模式下为高电平，在 MSB 对齐、LSB 对齐或 PCM 模式下为低电平）。

主器件与从器件之间的同步失效可能是由于 CK 通信时钟或 WS 帧同步信号线上存在噪音干扰。如果 ERRIE 位置 1，可产生错误中断。读取状态寄存器时，同步失效标志 (FRE) 由软件清零。

25.7.9 DMA 特性

在 I^2S 模式下，DMA 的工作方式与在 SPI 模式下完全相同。除了由于不存在数据传输保护机制，I^2S 模式下没有 CRC 功能外，没有其他差别。

25.8 I2S 中断

表 117 为 I2S 中断的列表。

表 117. I2S 中断请求

中断事件	事件标志	使能控制位
发送缓冲区为空	TXE	TXEIE
接收缓冲区非空	RXNE	RXNEIE
上溢错误	OVR	ERRIE
下溢错误	UDR	
帧错误	FRE	

25.9 SPI 和 I2S 寄存器

外设寄存器可按半字（16 位）或字（32 位）访问。此外，SPI_DR 可支持 8 位访问。

25.9.1 SPI 控制寄存器 1 (SPIx_CR1)

SPI control register 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
BIDI MODE	BIDIOE	CRC EN	CRCN EXT	CRCL	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15 **BIDIMODE**：双向通信数据模式使能 (Bidirectional data mode enable)

该位通过一条共用的双向数据线来支持半双工通信。双向模式激活时，使 RXONLY 位保持清零状态。

0：选择双线单向通信数据模式

1：选择单线双向通信数据模式

注意：该位不适用于 I^2S 模式。

位 14 **BIDIOE**：双向通信模式下的输出使能 (Output enable in bidirectional mode)

该位与 BIDIMODE 位结合，用于选择双向模式下的传输方向。

0：禁止输出（只接收模式）

1：使能输出（只发送模式）

注意：在主模式下，使用 MOSI 引脚；在从模式下，使用 MISO 引脚。
该位不适用于 I^2S 模式。

位 13 **CRCEN**：硬件 CRC 计算使能 (Hardware CRC calculation enable)

0：禁止 CRC 计算

1：使能 CRC 计算

注意：为确保正确操作，只应在禁止 SPI（SPE = “0”）时对该位执行写操作。
该位不适用于 I^2S 模式。

位 12 **CRCNEXT**：发送下一个 CRC (Transmit CRC next)

0：下一个发送值来自发送缓冲区。

1：下一个发送值来自发送 CRC 寄存器。

注意：向 SPIx_DR 寄存器写入最后一个数据后，必须立即对该位执行写操作。
该位不适用于 I^2S 模式。

位 11 **CRCL**：CRC 长度 (CRC length)

该位由软件置 1 和清零，用以选择 CRC 长度。

0：8 位 CRC 长度

1：16 位 CRC 长度

注意：为确保正确操作，只应在禁止 SPI（SPE = “0”）时对该位执行写操作。
该位不适用于 I^2S 模式。

位 10 **RXONLY**：只接收模式使能 (Receive only mode enabled)

该位用于使能通过一条双向线专门接收数据的单工通信。当只接收模式激活时，将 BIDIMODE 位保持清零状态。该位也适用于多从模式系统，在此类系统中，不会访问特定从器件，也不会损坏访问的从器件的输出。

0：全双工（发送和接收）

1：禁止输出（只接收模式）

注意：该位不适用于 I^2S 模式。

位 9 **SSM**：软件从器件管理 (Software slave management)

当 SSM 位置 1 时，NSS 引脚输入替换为 SSI 位的值。

0：禁止软件从器件管理

1：使能软件从器件管理

注意：该位不适用于 I^2S 模式 和 SPI TI 模式。

位 8 **SSI**：内部从器件选择 (Internal slave select)

仅当 SSM 位置 1 时，该位才有效。该位的值将作用到 NSS 引脚上，并忽略 NSS 引脚的 I/O 值。

注意：该位不适用于 I^2S 模式 和 SPI TI 模式。

位 7 **LSBFIRST**：帧格式 (Frame format)

0：发送/接收数据时 MSB 在前

1：发送/接收数据时 LSB 在前

注意：1. 正在通信时不应更改该位。

2. 该位不适用于 I^2S 模式 和 SPI TI 模式。

位 6 **SPE**：SPI 使能 (SPI enable)

0：禁止外设

1：使能外设

注意：禁止 SPI 时，请按照第 694 页的禁止 SPI 的步骤中所述的步骤操作。

该位不适用于 I^2S 模式。

位 5:3 **BR[2:0]**：波特率控制 (Baud rate control)

000：$f_{PCLK}/2$

001：$f_{PCLK}/4$

010：$f_{PCLK}/8$

011：$f_{PCLK}/16$

100：$f_{PCLK}/32$

101：$f_{PCLK}/64$

110：$f_{PCLK}/128$

111：$f_{PCLK}/256$

注意：正在通信时不应更改这些位。

这些位不适用于 I^2S 模式。

位 2 **MSTR**：主模式选择 (Master selection)

0：从配置

1：主配置

注意：正在通信时不应更改该位。

该位不适用于 I^2S 模式。

位 1 **CPOL**：时钟极性 (Clock polarity)
0：空闲状态时，CK 保持低电平
1：空闲状态时，CK 保持高电平
注意：正在通信时不应更改该位。
除了在 TI 模式下应用 CRC 的情况外，该位不会用于 I^2S 模式和 SPI TI 模式。

位 0 **CPHA**：时钟相位
0：从第一个时钟边沿开始采样数据
1：从第二个时钟边沿开始采样数据
注意：正在通信时不应更改该位。
除了在 TI 模式下应用 CRC 的情况外，该位不会用于 I^2S 模式和 SPI TI 模式。

25.9.2 SPI 控制寄存器 2 (SPIx_CR2)

SPI control register 2

偏移地址：0x04

复位值：0x0700

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LDMA_TX	LDMA_RX	FRXTH	DS[3:0]				TXEIE	RXNEIE	ERRIE	FRF	NSSP	SSOE	TXDMAEN	RXDMAEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15 保留，必须保持复位值。

位 14 **LDMA_TX**：发送的最后一次 DMA 传输 (Last DMA transfer for transmission)
该位用于数据封装模式，用于定义通过 DMA 发送的数据总数为奇数还是偶数。只有 SPIx_CR2 寄存器中的 TXDMAEN 位置 1 且使用封装模式（数据长度 =< 8 位，对 SPIx_DR 的写访问为 16 位宽）时，该位才有意义。当 SPI 禁止时（SPIx_CR1 寄存器中的 SPE = 0），必须对其执行写操作。
0：待传输数据数量为偶数
1：待传输数据数量为奇数
注意：如果 CRCEN 位置 1，请参见第 694 页的禁止 SPI 的步骤。
该位不适用于 I^2S 模式。

位 13 **LDMA_RX**：接收的最后一次 DMA 传输 (Last DMA transfer for reception)
该位用于数据封装模式下，用于定义通过 DMA 接收的数据总数为奇数还是偶数。只有 SPIx_CR2 寄存器中的 RXDMAEN 位置 1 且使用封装模式（数据长度 =< 8 位，对 SPIx_DR 的写访问为 16 位宽）时，该位才有意义。当 SPI 禁止时（SPIx_CR1 寄存器中的 SPE = 0），必须对其执行写操作。
0：待传输数据数量为偶数
1：待传输数据数量为奇数
注意：如果 CRCEN 位置 1，请参见第 694 页的禁止 SPI 的步骤。
该位不适用于 I^2S 模式。

位 12 **FRXTH**：FIFO 接收阈值 (FIFO reception threshold)
该位用于设置触发 RXNE 事件的 RXFIFO 阈值
0：如果 FIFO 占用水平大于或等于 1/2（16 位），将生成 RXNE 事件
1：如果 FIFO 占用水平大于或等于 1/4（8 位），将生成 RXNE 事件
注意：该位不适用于 I^2S 模式。

位 11:8 **DS[3:0]**：数据大小 (Data size)

这些位用于配置 SPI 传输的数据位宽。

0000：未使用
0001：未使用
0010：未使用
0011：4 位
0100：5位
0101：6 位
0110：7 位
0111：8 位
1000：9 位
1001：10位
1010：11位
1011：12 位
1100：13 位
1101：14 位
1110：15 位
1111：16 位

如果软件尝试写入其中一个“未使用”值，这些位将被强制设为“0111”。（8 位）

注意：这些位不适用于 I^2S 模式。

位 7 **TXEIE**：发送缓冲区空中断使能 (Tx buffer empty interrupt enable)

0：屏蔽 TXE 中断
1：使能 TXE 中断。TXE 标志置 1 时产生中断请求。

位 6 **RXNEIE**：接收缓冲区非空中断使能 (RX buffer not empty interrupt enable)

0：屏蔽 RXNE 中断
1：使能 RXNE 中断。RXNE 标志置 1 时产生中断请求。

位 5 **ERRIE**：错误中断使能 (Error interrupt enable)

该位用于在出现错误条件（SPI 模式下的 CRCERR、OVR 和 MODF；TI 模式下的 FRE；以及 I^2S 模式下的 UDR、OVR 和 FRE）时控制中断的生成。

0：屏蔽错误中断
1：使能错误中断

位 4 **FRF**：帧格式 (Frame format)

0：SPI Motorola 模式
1：SPI TI 模式

注意：只有在禁止 SPI (SPE=0) 后才能对该位执行写操作。
该位不适用于 I^2S 模式。

位 3 **NSSP**：NSS 脉冲管理 (NSS pulse management)

该位仅用于主模式。连续传输时，该位允许 SPI 在两个连续数据间生成 NSS 脉冲。单次数据传输时，该位强制 NSS 引脚在传输后变为高电平。

如果 CPHA =“1”或 FRF =“1”，该位无意义。

0：未生成 NSS 脉冲
1：生成 NSS 脉冲

注意：1. 只有在禁止 SPI (SPE=0) 后才能对该位执行写操作。
2. 该位不适用于 I^2S 模式 和 SPI TI 模式。

位 2 **SSOE**：SS 输出使能 (SS output enable)

0：在主模式下禁止 SS 输出，SPI 接口可在多主模式配置下工作
1：在主模式下以及使能 SPI 接口后使能 SS 输出。SPI 接口不能在多主模式环境下工作。

注意：该位不适用于 I^2S 模式 和 SPI TI 模式。

位 1 **TXDMAEN**：发送缓冲区 DMA 使能 (Tx buffer DMA enable)
当该位置 1 时，每当 TXE 标志置 1，即产生 DMA 请求。
0：关闭发送缓冲区 DMA
1：使能发送缓冲区 DMA

位 0 **RXDMAEN**：接收缓冲区 DMA 使能 (Rx buffer DMA enable)
当该位置 1 时，每当 RXNE 标志置 1，即产生 DMA 请求。
0：关闭接收缓冲区 DMA
1：使能接收缓冲区 DMA

25.9.3 SPI 状态寄存器 (SPIx_SR)

SPI status register

偏移地址：0x08

复位值：0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	FTLVL[1:0]		FRLVL[1:0]		FRE	BSY	OVR	MODF	CRCERR	UDR	CHSIDE	TXE	RXNE
			r	r	r	r	r	r	r	r	rc_w0	r	r	r	r

位 15:13 保留，必须保持复位值。

位 12:11 **FTLVL[1:0]**：FIFO 发送级别 (FIFO transmission level)
这些位将由硬件置 1 和清零。
00：FIFO 为空
01：1/4 FIFO
10：1/2 FIFO
11：FIFO 为满（当 FIFO 阈值大于 1/2 时即视为满）
注意：该位不适用于 I^2S 模式。

位 10:9 **FRLVL[1:0]**：FIFO 接收级别 (FIFO reception level)
这些位将由硬件置 1 和清零。
00：FIFO 为空
01：1/4 FIFO
10：1/2 FIFO
11：FIFO 已满
注意：使能 CRC 计算时，这些位不适用于 I²S 模式和 SPI 仅接收模式。

位 8 **FRE**：帧格式错误 (Frame format error)
该标志在 TI 从模式和 I^2S 从模式下用于 SPI。请参见*第 25.5.11 节：SPI 错误标志*和*第 25.7.8 节：I2S 错误标志*。
此标志由硬件置 1，在读取 SPIx_SR 时由软件复位。
0：未发生帧格式错误
1：发生帧格式错误

位 7 **BSY**：忙标志 (Busy flag)
0：SPI（或 I2S）不繁忙
1：SPI（或 I2S）忙于通信或者发送缓冲区不为空
此标志由硬件置 1 和清零。
注意：BSY 标志必须谨慎使用：请参见第 25.5.10 节：SPI 状态标志和第 694 页的禁止 SPI 的步骤。

位 6 **OVR**：上溢标志 (Overrun flag)
0：未发生上溢
1：发生上溢
此标志由硬件置 1，由软件序列复位。有关软件序列，请参见*第 723 页的I2S 错误标志*。

位 5 **MODF**：模式故障 (Mode fault)
0：未发生模式故障
1：发生模式故障
此标志由硬件置 1，由软件序列复位。有关软件序列，请参见*第 702 页的模式故障 (MODF)*一节。
注意：该位不适用于 I^2S 模式。

位 4 **CRCERR**：CRC 错误标志
0：接收到的 CRC 值与 SPIx_RXCRCR 值匹配
1：接收到的 CRC 值与 SPIx_RXCRCR 值不匹配
注意：此标志由硬件置 1，通过软件写入 0 来清零。
该位不适用于 I^2S 模式。

位 3 **UDR**：下溢标志 (Underrun flag)
0：未发生下溢
1：发生下溢
此标志由硬件置 1，由软件序列复位。有关软件序列，请参见*第 723 页的I2S 错误标志*。
注意：该位不适用于 SPI 模式。

位 2 **CHSIDE**：通道信息 (Channel side)
0：发送或接收左通道信息
1：发送或接收右通道信息
注意：该位不适用于 SPI 模式。该位在 PCM 模式下无意义。

位 1 **TXE**：发送缓冲区为空 (Transmit buffer empty)
0：发送缓冲区非空
1：发送缓冲区为空

位 0 **RXNE**：接收缓冲区非空 (Receive buffer not empty)
0：接收缓冲区为空
1：接收缓冲区非空

25.9.4 SPI 数据寄存器 (SPIx_DR)

SPI data register

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
DR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **DR[15:0]**：数据寄存器 (Data register)
已接收或者要发送的数据
数据寄存器用于连接 Rx 和 Tx FIFO。读取数据寄存器时，将访问 RxFIFO；而写入数据寄存器时，将访问 TxFIFO（请参见*第 25.5.9 节：数据发送和接收过程*）。
注意：数据始终右对齐。写入寄存器时将忽略未使用位，读取寄存器时会将未使用位读为 0。Rx 阈值设置必须始终与当前使用的读访问相符。

25.9.5 SPI CRC 多项式寄存器 (SPIx_CRCPR)

SPI CRC polynomial register

偏移地址：0x10

复位值：0x0007

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

位 15:0 **CRCPOLY[15:0]**：CRC 多项式寄存器 (CRC polynomial register)

此寄存器包含用于 CRC 计算的多项式。

CRC 多项式 (0x0007) 是此寄存器的复位值。可根据需要配置另一个多项式。

注意：多项式值只应为奇数。不支持任何偶数值。

25.9.6 SPI 接收 CRC 寄存器 (SPIx_RXCRCR)

SPI Rx CRC register

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
RXCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 15:0 **RXCRC[15:0]**：接收 CRC 寄存器 (Rx CRC register)

使能 CRC 计算后，RXCRC[15:0] 位包含接收字节序列的 CRC 值。当 SPIx_CR1 寄存器中的 CRCEN 位写入 1 时，此寄存器复位。CRC 通过 SPIx_CRCPR 寄存器中编程的多项式连续计算。

CRC 帧格式设置为 8 位长度（SPIx_CR1 的 CRCL 位清零）时，仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。

选择 16 位 CRC 帧格式（SPIx_CR1 寄存器的 CRCL 位置 1）时，考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC16 标准进行。

注意：当 BSY 标志置 1 时，读取此寄存器可能返回一个不正确的值。

这些位不适用于 I^2S 模式。

25.9.7 SPI 发送 CRC 寄存器 (SPIx_TXCRCR)

SPI Tx CRC register

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
TXCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 15:0 **TXCRC[15:0]**：发送 CRC 寄存器 (Tx CRC register)

使能 CRC 计算后，TXCRC[7:0] 位将包含发送字节序列的 CRC 值。当 SPIx_CR1 寄存器中的 CRCEN 位写入 1 时，此寄存器复位。CRC 通过 SPIx_CRCPR 寄存器中编程的多项式连续计算。

CRC 帧格式设置为 8 位长度（SPIx_CR1 的 CRCL 位清零）时，仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。

选择 16 位 CRC 帧格式（SPIx_CR1 寄存器的 CRCL 位置 1）时，考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC16 标准进行。

注意：当 BSY 标志置 1 时，读取此寄存器可能返回一个不正确的值。

这些位不适用于 I^2S 模式。

25.9.8 SPIx_I2S 配置寄存器 (SPIx_I2SCFGR)

SPIx_I2S configuration register

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.	Res.	Res.	ASTRTEN	I2SMOD	I2SE	I2SCFG[1:0]		PCMSYNC	Res.	I2SSTD[1:0]		CKPOL	DATLEN[1:0]		CHLEN
			rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位 15:13 保留，必须保持复位值。

位 12 **ASTRTEN**：异步启动使能 (Asynchronous start enable)

0：禁止异步启动。

在从模式下使能 I2S 后，若接收到 I2S 时钟并且在 WS 信号上检测到适当的边沿，硬件将启动传输。

1：使能异步启动。

在从模式下使能 I2S 后，若接收到 I2S 时钟并且在 WS 信号上检测到适当的电平，硬件将启动传输。

*注意：使用 I^2S Philips 标准时，适当的**边沿**为 WS 信号的下降沿；使用其他标准时，则为上升沿。*

*使用 I^2S Philips 标准时，适当的**电平**为 WS 信号的低电平；使用其他标准时，则为高电平。*

更多相关信息，请参见第 25.7.3 节：启动说明。

位 11 **I2SMOD**：I2S 模式选择 (I2S mode selection)

0：选择 SPI 模式

1：选择 I2S 模式

注意：应在 SPI 禁止时配置该位。

位 10 **I2SE**：I2S 使能 (I2S enable)

0：禁止 I2S 外设

1：使能 I2S 外设

注意：该位不适用于 SPI 模式。

位 9:8 **I2SCFG[1:0]**：I2S 配置模式 (I2S configuration mode)

00：从模式 - 发送

01：从模式 - 接收

10：主模式 - 发送

11：主模式 - 接收

注意：应在 I2S 禁止时配置这些位。

这些位不适用于 SPI 模式。

位 7 **PCMSYNC**：PCM 帧同步 (PCM frame synchronization)
0：短帧同步
1：长帧同步
注意：只有在 I2SSTD = 11（使用 PCM 标准）时，该位才有意义。
该位不适用于 SPI 模式。

位 6 保留，必须保持复位值。

位 5:4 **I2SSTD[1:0]**：I2S 标准选择 (I2S standard selection)
00：I^2S Philips 标准
01：MSB 对齐标准（左对齐）
10：LSB 对齐标准（右对齐）
11：PCM 标准
有关 I^2S 标准的详细信息，请参见*第 708 页的第 25.7.2 节*
注意：为确保正确运行，应在 I2S 禁止时配置这些位。
这些位不适用于 SPI 模式。

位 3 **CKPOL**：空闲状态时钟极性 (Inactive state clock polarity)
0：空闲状态时钟为低电平
1：空闲状态时钟为高电平
注意：为确保正确运行，应在 I2S 禁止时配置该位。
该位不适用于 SPI 模式。
CKPOL 位不影响用于接收或发送 SD 和 WS 信号的 CK 边沿有效性。

位 2:1 **DATLEN[1:0]**：传输的数据长度 (Data length to be transferred)
00：16 位数据长度
01：24 位数据长度
10：32 位数据长度
11：不允许
注意：为确保正确运行，应在 I2S 禁止时配置这些位。
这些位不适用于 SPI 模式。

位 0 **CHLEN**：通道长度（每个音频通道的位数）(Channel length (number of bits per audio channel))
0：16 位
1：32 位
只有在 DATLEN = 00 时，该位的写操作才有意义，否则无论写入何值，通道长度始终由硬件固定为 32 位。
注意：为确保正确运行，应在 I2S 禁止时配置该位。
该位不适用于 SPI 模式。

25.9.9 SPIx_I2S 预分频器寄存器 (SPIx_I2SPR)

SPIx_I2S prescaler register

偏移地址：0x20

复位值：0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	MCKOE	ODD	I2SDIV[7:0]							
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:10 保留，必须保持复位值。

位 9 **MCKOE**：主时钟输出使能 (Master clock output enable)

0：禁止主时钟输出

1：使能主时钟输出

注意：应在 I2S 禁止时配置该位。只有在 I2S 为主模式时，才会使用该位。
该位不适用于 SPI 模式。

位 8 **ODD**：预分频器的奇数因子 (Odd factor for the prescaler)

0：实际分频值为 = I2SDIV *2

1：实际分频值为 = (I2SDIV * 2) + 1

请参见*第 715 页的第 25.7.3 节*。

注意：应在 I2S 禁止时配置该位。只有在 I2S 为主模式时，才会使用该位。
该位不适用于 SPI 模式。

位 7:0 **I2SDIV[7:0]**：I2S 线性预分频器 (I2S linear prescaler)

I2SDIV [7:0] = 0 或 I2SDIV [7:0] = 1 为禁用值。

请参见*第 715 页的第 25.7.3 节*。

注意：应在 I2S 禁止时配置这些位。只有在 I2S 为主模式时，才会使用这些位。
这些位不适用于 SPI 模式。

25.9.10 SPI/I2S 寄存器映射

表 118 给出了 SPI/I2S 寄存器映射和复位值。

表 118. SPI/I2S 寄存器映射和复位值

偏移	寄存器名称 复位值	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SPIx_CR1	BIDIMODE	BIDIOE	CRCEN	CRCNEXT	CRCL	RXONLY	SSM	SSI	LSBFIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	SPIx_CR2	Res.	LDMA_TX	LDMA_RX	FRXTH	DS[3:0]				TXEIE	RXNEIE	ERRIE	FRF	NSSP	*SSOE*	TXDMAEN	RXDMAEN
	Reset value		0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
0x08	SPIx_SR	Res.	Res.	Res.	FTLVL[1:0]		FRLVL[1:0]		FRE	BSY	OVR	MODF	CRCERR	UDR	CHSIDE	TXE	RXNE
	Reset value				0	0	0	0	0	0	0	0	0	0	0	1	0
0x0C	SPIx_DR	DR[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	SPIx_CRCPR	CRCPOLY[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0x14	SPIx_RXCRCR	RXCRC[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	SPIx_TXCRCR	TXCRC[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	SPIx_I2SCFGR	Res.	Res.	Res.	ASTRTEN	I2SMOD	I2SE	I2SCFG[1:0]		PCMSYNC	Res.	I2SSTD		CKPOL	DATLEN[1:0]		CHLEN
	Reset value				0	0	0	0	0	0		0	0	0	0	0	0
0x20	SPIx_I2SPR	Res.	Res.	Res.	Res.	Res.	Res.	MCKOE	ODD	I2SDIV[7:0]							
	Reset value							0	0	0	0	0	0	0	0	1	0

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

26 调试支持 (DBG)

26.1 概述

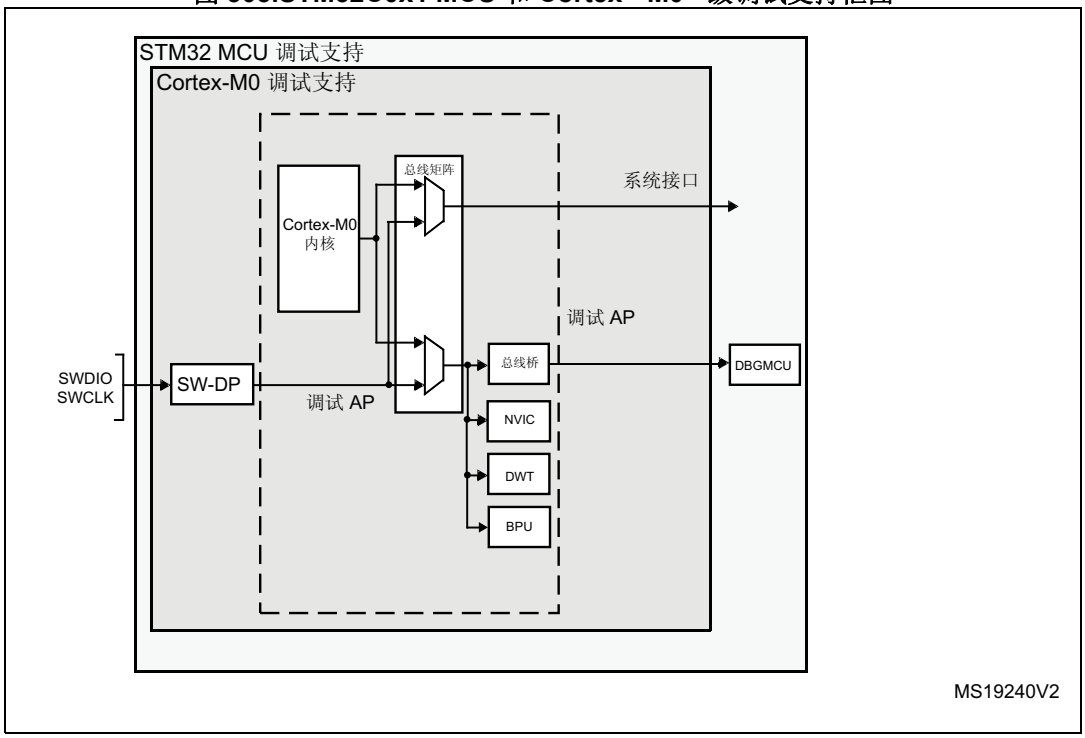
STM32C0x1 器件的内核是 Cortex®-M0+，该内核包含用于高级调试功能的硬件扩展。调试扩展允许内核在取指（指令断点）或取访问数据（数据断点）时停止内核。内核停止时，可以查询内核的内部状态和系统的外部状态。查询完成后，将恢复内核和系统，并继续程序的执行。

当调试器主机与 STM32C0x1 MCU 相连，并进行调试时，将使用调试功能。

提供一个调试接口：

- 串行接口

图 305.STM32C0x1 MCU 和 Cortex®-M0+ 级调试支持框图



1. Cortex®-M0+ 内核中内置的调试功能是 Arm CoreSight 设计套件的一部分。

Arm Cortex®-M0+ 内核提供集成片上调试支持。它包括：

- SW-DP：串行接口
- BPU：断点单元
- DWT：数据断点触发

它还包括专用于 STM32C0x1 的调试功能：

- 灵活调试引脚分配
- MCU 调试盒（支持低功耗模式和对外设时钟的控制等）

注意： *有关 Arm Cortex®-M0+ 内核支持的调试功能的详细信息，请参见 Cortex®-M0+ 技术参考手册（请参见第 26.2 节：Arm 参考文档）。*

26.2 Arm 参考文档

- Cortex®-M0+ 技术参考手册 (TRM)，可从 http://infocenter.arm.com 获取
- Arm 调试接口 V5
- Arm CoreSight 设计套件版本 r1p1 技术参考手册

26.3 引脚排列和调试端口引脚

STM32C0x1 MCU 的不同封装有不同的有效引脚数。

26.3.1 SWD 端口引脚

两个引脚被用作 SW-DP 的输出，作为通用 I/O 的复用功能。所有封装都提供这些引脚。

表 119. SW 调试端口引脚

SW-DP 引脚名称	SW 调试端口		引脚分配
	类型	调试分配	
SWDIO	I/O	串行线数据输入/输出	PA13
SWCLK	I	串行线时钟	PA14

26.3.2 SW-DP 引脚分配

复位（SYSRESETn 或 PORESETn）后，用于 SW-DP 的引脚将分配为可由调试器主机立即使用的专用引脚。

但是，MCU 可以禁止 SWD 端口，进而可释放相关引脚以用作通用 I/O (GPIO)。有关如何禁止 SW-DP 端口引脚的更多详细信息，请参见*第 137 页的第 6.3.2 节：I/O 引脚复用功能复用器和映射*。

26.3.3 SWD 引脚上的内部上拉和下拉

用户软件释放 SW I/O 后，GPIO 控制器便会控制这些引脚。GPIO 控制寄存器的复位状态会将 I/O 置于等效的状态：

- SWDIO：输入上拉
- SWCLK：输入下拉

由于带有上拉和下拉电阻，因此无需添加外部电阻。

26.4 ID 代码和锁定机制

MCU 中提供了一些 ID 代码。ST 强烈建议工具制造商（例如 Keil、IAR、Raisonance）使用地址 0x40015800 处的 MCU 器件 ID 锁定其调试工具。

调试软件/编程工具只能将 DEV_ID[15:0] 用于识别芯片（不得考虑版本 ID）。

26.5 SWD 端口

26.5.1 SWD 协议简介

此同步串行协议使用两个引脚：

- SWCLK：从主机到目标的时钟
- SWDIO：双向

利用该协议，可以同时读取和写入两组寄存器组（DPACC 寄存器组和 APACC 寄存器组）。

传输数据时，LSB 在前。

对于 SWDIO 双向管理，必须在电路板上对线路进行上拉（Arm 建议采用 100 kΩ）。

每次在协议中更改 SWDIO 的方向时，都会插入转换时间，此时线路即不受主机驱动也不受目标驱动。默认情况下，此转换时间为一个位的时间，但可以通过配置 SWCLK 频率来调整。

26.5.2 SWD 协议序列

每个序列包括三个阶段：

1. 主机发送的数据包请求（8 位）
2. 目标发送的确认响应（3 位）
3. 主机或目标发送的数据传输阶段（33 位）

表 120. 数据包请求（8 位）

位	名称	说明
0	启动	必须为 1
1	APnDP	0：DP 访问 1：AP 访问
2	RnW	0：写请求 1：读请求
4:3	A[3:2]	DP 或 AP 寄存器的地址位域（请参见*第 739 页的表 124*）
5	奇偶校验	前面几位的单位奇偶校验
6	停止	0
7	驻留	不受主机驱动。由于存在上拉，因此必须由目标读为 1

有关 DPACC 和 APACC 寄存器的详细说明，请参见 Cortex®-M0+ *TRM*。

数据包请求的后面始终为转换时间（默认 1 位），此时主机和目标都不会驱动线路。

表 121. ACK 响应（3 位）

位	名称	说明
0..2	ACK	001：FAULT 010：WAIT 100：OK

仅当发生 READ 事务或者接收到 WAIT 或 FAULT 确认时，ACK 响应后才必须是转换时间。

表 122. DATA 传输（33 位）

位	名称	说明
0..31	WDATA 或 RDATA	写入或读取数据
32	奇偶校验	32 个数据位的单奇偶校验

仅当发生 READ 事务时，DATA 传输后才必须是转换时间。

26.5.3 SW-DP 状态机（复位、空闲状态、ID 代码）

SW-DP 的状态机有一个用于标识 SW-DP 的内部 ID 代码。此代码遵循 JEP-106 标准。此 ID 代码是默认的 Arm 代码，设置为 **0x0BB11477**（相当于 Cortex®-M0+）。

注意： *请注意，在目标读取此 ID 代码前，SW-DP 状态机是不工作的。*

- 在上电复位后或者线路处于高电平超过 50 个周期后，SW-DP 状态机处于复位状态。
- 如果在复位状态后线路处于低电平至少两个周期，SW-DP 状态机处于空闲状态。
- 复位状态后，该状态机**必须**首先进入空闲状态，然后对 DP-SW ID CODE 寄存器执行读访问。否则，目标将在另一个事务上发出 FAULT 确认响应。

有关 SW-DP 状态机的更多详细信息，请参见 *Cortex®-M0+ TRM* 和 *CoreSight 设计套件 r1p0 TRM*。

26.5.4 DP 和 AP 读/写访问

- 对 DP 的不延迟读访问：可以立即发送目标响应（如果 ACK=OK），也可以延迟发送目标响应（如果 ACK=WAIT）。
- 对 AP 的延迟读访问。这意味着会在下次传输时返回访问结果。如果要执行的下次访问不是 AP 访问，则必须读取 DP-RDBUFF 寄存器来获取结果。
  每次进行 AP 读访问或 RDBUFF 读请求时都会更新 DP-CTRL/STAT 寄存器的 READOK 标志，以便了解 AP 读访问是否成功。
- SW-DP 有写缓冲区（用于 DP 或 AP 写入），这样即使在其他操作仍未完成时，也可以接受写入操作。如果写缓冲区已满，则目标确认响应为 WAIT。但 IDCODE 读取、CTRL/STAT 读取或 ABORT 写入除外，这几项操作在写缓冲区已满时也会被接受。
- 由于存在异步时钟域 SWCLK 和 HCLK，因此写操作后（奇偶校验位后）还需要两个额外的 SWCLK 周期，以使写入操作在内部生效。应在将线路驱动为低电平时（空闲状态）应用这些周期。
  这一点在通过写 CTRL/STAT 寄存器以产生一个上电请求时，就显得特别重要。否则下一个操作（在内核上电后才有效的操作）会立即执行，这将导致失败。

26.5.5 SW-DP 寄存器

当 APnDP=0 时能够访问这些寄存器。

表 123. SW-DP 寄存器

A[3:2]	R/W	SELECT 寄存器的 CTRLSEL 位	寄存器	注释
00	读		IDCODE	制造商代码设置为 Cortex®-M0+ 的默认 Arm 代码：**0x0BC11477**（标识 SW-DP）
00	写		ABORT	
01	读/写	0	DP-CTRL/STAT	目的： – 请求系统或调试上电 – 配置 AP 访问的传输操作 – 控制比较和验证操作。 – 读取一些状态标志（上溢和上电确认）
01	读/写	1	WIRE CONTROL	用于配置物理串行端口协议（如转换时间的持续时间）
10	读		READ RESEND	允许从已损坏的调试软件传输中恢复读取数据，无需重复执行原始 AP 传输。
10	写		SELECT	用于选择当前访问端口和活动的 4 字寄存器窗口
11	读/写		READ BUFFER	由于已发出 AP 访问，因此该读缓冲区非常有用（在执行下个 AP 事务时提供读取 AP 请求的结果）。 此读取缓冲区捕获 AP 中的数据，显示为前一次读取的结果，无需启动新操作

26.5.6 SW-AP 寄存器

当 APnDP=1 时能够访问这些寄存器

有多个 AP 寄存器，这些寄存器按以下组合进行寻址：

- 移位值 A[3:2]
- DP SELECT 寄存器的当前值。

表 124. 32 位调试端口寄存器，通过移位值 A[3:2] 进行寻址

地址	A[3:2] 值	说明
0x0	00	保留，必须保持复位值。
0x4	01	DP CTRL/STAT 寄存器。用于： – 请求系统或调试上电 – 配置 AP 访问的传输操作 – 控制比较和验证操作。 – 读取一些状态标志（上溢和上电确认）

表 124. 32 位调试端口寄存器，通过移位值 A[3:2] 进行寻址（续）

地址	A[3:2] 值	说明
0x8	10	DP SELECT 寄存器：用于选择当前访问端口和活动的 4 字寄存器窗口。 – 位 31:24：APSEL：选择当前 AP – 位 23:8：保留 – 位 7:4：APBANKSEL：在当前 AP 上选择活动的 4 字寄存器窗口 – 位 3:0：保留
0xC	11	DP RDBUFF 寄存器：用于通过调试软件在执行一系列操作后获取最后结果（无需请求新的 JTAG-DP 操作）

26.6 内核调试

通过内核调试寄存器调试内核。通过调试访问端口调试访问这些寄存器。它由四个寄存器组成：

表 125. 内核调试寄存器

寄存器	说明
DHCSR	*32 位调试停止控制和状态寄存器* 此寄存器提供有关处理器状态的信息，能够使内核进入调试停止状态并提供处理器步进功能。
DCRSR	*17 位调试内核寄存器选择器寄存器：* 此寄存器选择需要进行读写操作的处理器寄存器。
DCRDR	*32 位调试内核寄存器数据寄存器：* 此寄存器保存在寄存器与 DCRSR（选择器）寄存器选择的处理器之间读取和写入的数据。
DEMCR	*32 位调试异常和监视控制寄存器：* 此寄存器提供向量捕获和调试监视控制。

这些寄存器在系统复位时不复位。它们只能通过上电复位来复位。有关更多详细信息，请参见 Cortex®-M0+ TRM。

为了在复位后立即使内核进入调试停止状态，必须：

- 使能调试和异常监视控制寄存器的位 0 (VC_CORRESET)
- 使能调试停止控制和状态寄存器的位 0 (C_DEBUGEN)

26.7 BPU（断点单元）

Cortex®-M0+ BPU 实现提供四个断点寄存器。BPU 是 Armv7-M（Cortex-M3 和 Cortex-M4）中提供的 Flash 补丁和断点 (FPB) 模块的一部分。

26.7.1 BPU 功能

处理器断点实现了基于 PC 的断点功能。

有关 BPU CoreSight 标识寄存器及其地址和访问类型的更多信息，请参见 Armv6-M Arm 和 Arm CoreSight 组件技术参考手册。

26.8 DWT（数据观察点）

Cortex®-M0+ DWT 实现提供了两个观察点寄存器组。

26.8.1 DWT 功能

处理器观察点实现了数据地址和基于 PC 的观察点功能（即 PC 采样寄存器），并支持比较器地址掩码，如 *Armv6-M Arm* 中所述。

26.8.2 DWT 程序计数器采样寄存器

实现数据观察点单元的处理器还实现了 Armv6-M 可选 *DWT 程序计数器采样寄存器* (DWT_PCSR)。此寄存器允许调试程序定期采样 PC，无需停止处理器。这可提供粗略分析。有关更多信息，请参见 *Armv6-M Arm*。

Cortex®-M0+ DWT_PCSR 记录通过条件代码和指令以及未通过条件代码的指令。

26.9 MCU 调试组件 (DBG)

MCU 调试组件帮助调试器为以下各项提供支持：

- 低功耗模式
- 断点期间的定时器、看门狗和 I2C 的时钟控制

26.9.1 对低功耗模式的调试支持

CPU 需要激活 FCLK 或 HCLK 时钟才能允许进行任何调试。

默认情况下，停止、待机和关断这三种低功耗模式会禁用 FCLK 和 HCLK，从而阻止调试功能。但在睡眠模式下，器件会使 FCLK 和 HCLK 始终保持激活状态。

为了使 FCLK 或 HCLK 时钟保持激活状态，从而在停止模式、待机模式和关断模式下保留调试功能，调试主机必须在进入这些低功耗模式之前将 DBG_CR 寄存器的 DBG_STOP 位（对于停止模式）或 DBG_STANDBY 位（对于待机模式和关断模式）置 1。

26.9.2 对定时器、看门狗和 I2C 的调试支持

断点期间，必须选择定时器和看门狗的计数器的行为方式：

- 在产生断点时，计数器继续计数。例如，当 PWM 控制电机时，通常需要这种方式。
- 在产生断点时，计数器停止计数。用于看门狗时需要这种方式。

对于 I2C 外设，用户可以选择在断点期间阻止 SMBUS 超时。

26.10 DBG 寄存器

器件集成了 ID 代码，用于标识器件及其晶片版本。

可通过软件调试端口（两个引脚）或用户软件访问此代码。

26.10.1 DBG 器件 ID 代码寄存器 (DBG_IDCODE)

DBG device ID code register

偏移地址：0x00

复位值：请参见*表 126*

仅支持 32 位访问。

该寄存器是只读的，用于标识器件及其晶片版本。可通过软件调试端口（两个引脚）或用户软件访问此寄存器。

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

位 31:16 **REV_ID[15:0]**：版本标识符 (Revision identifier)
该位域指示器件的版本。请参见*表 126*。

位 15:12 保留，必须保持复位值。
读取这些保留位后会返回 0b0110。

位 11:0 **DEV_ID[11:0]**：器件标识符 (Device identifier)
该位域指示器件 ID。请参见*表 126*。

表 126. DEV_ID 和 REV_ID 位域值

器件	DEV_ID	版本代码	版本号	REV_ID
STM32C011xx	0x443	A	1.0	0x1000
		Z	1.1	0x1001
STM32C031xx	0x453	A	1.0	0x1000
		Z	1.1	0x1001

26.10.2 DBG 配置寄存器 (DBG_CR)

DBG configuration register

偏移地址：0x0000 0004

复位值：0x0000 0000（上电复位）

仅支持 32 位访问。

该寄存器用于在 MCU 处于调试状态时为其配置低功耗模式。
它通过 POR 异步复位，但不受系统复位的影响。可在系统复位状态下用调试器写入该寄存器。
如果调试主机不支持该功能，仍然可以通过用户软件写入该寄存器。

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

位 31:3 保留，必须保持复位值。

位 2 **DBG_STANDBY**：调试待机模式和关断模式 (Debug Standby and Shutdown modes)

待机模式或关断模式下的调试选项。

0：数字部分上电。从软件角度来看，退出待机模式和关断模式相当于取复位向量（指示 MCU 退出待机状态的几个状态位除外）

1：数字部分上电，FCLK 和 HCLK 运行（这两个时钟由仍保持活动状态的内部 RC 振荡器提供）。MCU 会生成系统复位，以便在退出待机模式和关断模式时与从复位启动具有相同的效果。

位 1 **DBG_STOP**：调试停止模式 (Debug Stop mode)

停止模式下的调试选项。

0：禁止所有时钟，包括 FCLK 和 HCLK。退出停止模式后，CPU 时钟由 HSI 内部 RC 振荡器提供。

1：FCLK 和 HCLK 运行（这两个时钟由仍保持活动状态的内部 RC 振荡器提供）。如果使能 Systick，则可生成周期性中断和唤醒事件。

退出停止模式后，必须通过软件重新配置所需的时钟。

位 0 保留，必须保持复位值。

26.10.3 DBG APB 冻结寄存器 1 (DBG_APB_FZ1)

DBG APB freeze register 1

偏移地址：0x08

复位值：0x0000 0000（上电复位）

仅支持 32 位访问。

该寄存器用于在 MCU 处于调试状态时为其定时器、RTC、IWDG、WWDG 和 I2C SMBUS 外设配置时钟：
该寄存器通过 POR 异步复位，但不受系统复位的影响。可在系统复位状态下用调试器写入该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_I2C1_SMBUS_TIMEOUT	Res.	Res.	Res.	Res.	Res.
										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_RTC_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM3_STOP	Res.
			rw	rw	rw									rw	

位 31:22 保留，必须保持复位值。

位 21 **DBG_I2C1_SMBUS_TIMEOUT**：内核停止时的 SMBUS 超时 (SMBUS timeout when core is halted)

0：行为方式与正常模式下相同

1：冻结 SMBUS 超时

位 20:13 保留，必须保持复位值。

位 12 **DBG_IWDG_STOP**：内核停止时 IWDG 计数器的时钟 (Clocking of IWDG counter when the core is halted)

该位用于在内核停止时使能/禁止 IWDG 计数器的时钟：

0：使能

1：禁用

位 11 **DBG_WWDG_STOP**：内核停止时 WWDG 计数器的时钟 (Clocking of WWDG counter when the core is halted)

该位用于在内核停止时使能/禁止 WWDG 计数器的时钟：

0：使能

1：禁用

位 10 **DBG_RTC_STOP**：内核停止时 RTC 计数器的时钟 (Clocking of RTC counter when the core is halted)

该位用于在内核停止时使能/禁止 RTC 计数器的时钟：

0：使能

1：禁用

位 9:2 保留，必须保持复位值。

位 1 **DBG_TIM3_STOP**：内核停止时 TIM3 计数器的时钟 (Clocking of TIM3 counter when the core is halted)

该位用于在内核停止时使能/禁止 TIM3 计数器的时钟：

0：使能

1：禁用

位 0 保留，必须保持复位值。

26.10.4 DBG APB 冻结寄存器 2 (DBG_APB_FZ2)

DBG APB freeze register 2

偏移地址：0x0C

复位值：0x0000 0000（上电复位）

仅支持 32 位访问。

该寄存器用于在 MCU 处于调试状态时配置定时器计数器的时钟。
它通过 POR 异步复位，但不受系统复位的影响。可在系统复位状态下用调试器写入该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM17_STOP	DBG_TIM16_STOP	Res.
													rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBG_TIM14_STOP	Res.	Res.	Res.	DBG_TIM1_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw				rw											

位 31:19 保留，必须保持复位值。

位 18 **DBG_TIM17_STOP**：内核停止时 TIM17 计数器的时钟 (Clocking of TIM17 counter when the core is halted)

该位用于在内核停止时使能/禁止 TIM17 计数器的时钟：
0：使能
1：禁用

位 17 **DBG_TIM16_STOP**：内核停止时 TIM16 计数器的时钟 (Clocking of TIM16 counter when the core is halted)

该位用于在内核停止时使能/禁止 TIM16 计数器的时钟：
0：使能
1：禁用

位 16 保留，必须保持复位值。

位 15 **DBG_TIM14_STOP**：内核停止时 TIM14 计数器的时钟 (Clocking of TIM14 counter when the core is halted)

该位用于在内核停止时使能/禁止 TIM14 计数器的时钟：
0：使能
1：禁用

位 14:12 保留，必须保持复位值。

位 11 **DBG_TIM1_STOP**：内核停止时 TIM1 计数器的时钟 (Clocking of TIM1 counter when the core is halted)

该位用于在内核停止时使能/禁止 TIM1 计数器的时钟：
0：使能
1：禁用

位 10:0 保留，必须保持复位值。

26.10.5 DBG 寄存器映射

下表对 DBG 寄存器进行了汇总。

表 127. DBG 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DBG_IDCODE	REV_ID																Res.	Res.	Res.	Res.	DEV_ID											
	Reset value[1]	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X
0x04	DBG_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_STANDBY	DBG_STOP	Res.
	Reset value																														0	0	
0x08	DBG_APB_FZ1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_I2C1_SMBUS_TIMEOUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_RTC_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM3_STOP	Res.
	Reset value											0									0	0	0									0	
0x0C	DBG_APB_FZ2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM17_STOP	DBG_TIM16_STOP	Res.	DBG_TIM14_STOP	Res.	Res.	Res.	DBG_TIM1_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value														0	0		0				0											

1. 复位值与产品有关。有关详细信息，请参见*第 26.10.1 节：DBG 器件 ID 代码寄存器 (DBG_IDCODE)*。

有关寄存器边界地址的信息，请参见*第 37 页的第 2.2 节*。

27 器件电子签名

器件电子签名存储在 Flash 模块的系统存储区中，可以使用调试接口或 CPU 对其进行读取。它包含出厂前编程的标识和校准数据，这些数据允许用户固件或其他外部器件与 STM32C0x1 微控制器的特性自动匹配。

27.1 唯一器件 ID 寄存器（96 位）

Unique device ID register (96 bits)

唯一器件标识符最适合：

- 用作序列号（例如 USB 字符串序列号或其他终端应用程序）
- 在对内部 Flash 进行编程前将唯一 ID 与软件加密原语和协议结合使用时用作安全密钥的一部分以提高 Flash 中代码的安全性
- 激活安全启动进程等

96 位的唯一器件标识符提供了一个对于任何器件和任何上下文都唯一的参考号码。用户不能更改这些位。

基址：0x1FFF 7550

偏移地址：0x00

复位值：0xXXXX XXXX，其中 X 是出厂前编程的

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

位 31:0 **UID[31:0]**：以 BCD 格式表示的晶圆上的 X 和 Y 坐标 (X and Y coordinates on the wafer expressed in BCD format)

偏移地址：0x04

复位值：0xXXXX XXXX，其中 X 是出厂前编程的

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[63:48]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[47:32]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:8 **UID[63:40]**：LOT_NUM[23:0]
批号（ASCII 编码）

位 7:0 **UID[39:32]**：WAF_NUM[7:0]
晶圆编号（8 位无符号数）

偏移地址：0x08

复位值：0xXXXX XXXX，其中 X 是出厂前编程的

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[95:80]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[79:64]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **UID[95:64]**：LOT_NUM[55:24]
批号（ASCII 编码）

27.2 Flash 大小数据寄存器

Flash memory size data register

基址：0x1FFF 75A0

偏移地址：0x00

复位值：0xXXXX，其中 X 是出厂前编程的

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

位 15:0 **FLASH_SIZE[15:0]**：Flash 大小 (Flash memory size)
此处的位域指示以 KB 表示的器件 Flash 的大小。
例如，0x040 对应于 64 KB。

27.3 封装数据寄存器

Package data register

基址：0x1FFF 7500

偏移地址：0x00

复位值：0xXXXX，其中 X 是出厂前编程的

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

位 15:4 保留

位 3:0 **PKG[3:0]**：封装类型

条件：STM32C031xx

0001：保留
0010：TSSOP20
0011：UFQFPN28
0100：UFQFPN32 / LQFP32
0101：UFQFPN48 / LQPF48
其他值：保留

条件：STM32C011xx

0001：SO8
0010：WLCSP12
0011：UFQFPN20
0100：TSSOP20
其他值：保留

28 重要安全说明

意法半导体集团公司 (ST) 高度重视产品安全性，因此，本文档中所述的 ST 产品可能已通过各种安全认证机构的认证和/或可能已实施我们内部的安全措施（如本文所述）。但是，任何级别的安全认证和/或 ST 内部的安全措施都不能保证 ST 产品能够抵御所有形式的攻击。因此，ST 的每位客户都需自行负责确定 ST 产品中提供的安全级别是否满足自身的需求，而且除了 ST 产品本身的安全级别之外，还需要考虑在最终产品或应用中与其他组件和/或软件结合使用时的情况。特别要注意以下几点：

- ST 产品可能已通过一家或多家安全认证机构的认证，例如平台安全架构 (www.psacertified.org) 和/或物联网平台安全评估标准 (www.trustcb.com)。有关此处引用的 ST 产品是否已获得此类安全认证以及认证级别和当前认证状态的详细信息，请访问相关认证标准网站或访问 *www.st.com* 上的相关产品页以获取最新信息。由于 ST 产品的安全认证状态和/或级别可能会不时发生变化，客户应根据需要随时检查 ST 产品的安全认证状态/级别。如果未显示 ST 产品已获得特定安全标准认证，客户不应主观认为 ST 产品已通过认证。
- 认证机构有权评估、授予和撤销与 ST 产品相关的安全认证。因此，这些认证机构独立负责授予或撤销 ST 产品的安全认证，针对认证机构对任何 ST 产品所做的误判、评估、测试或其他活动，ST 不承担任何责任。
- 行业通用的加密算法（如 AES、DES 或 MD5）以及其他可与 ST 产品结合使用的开放标准技术均基于非 ST 开发的标准。对于此类加密算法或开放技术所存在的任何缺陷，或者可能甚至已经开发出的能够绕过、解密或破解此类算法或技术的任何方法，ST 不承担任何责任。
- 尽管 ST 产品已完成可靠的安全测试，但任何级别的认证都无法绝对保证抵御所有形式的攻击，其中包括未经测试的高级攻击、新的或未识别形式的攻击、在规范或预期用途之外使用 ST 产品时存在的任何形式攻击，或者与客户用来创建最终产品或应用的其他组件或软件结合使用时存在的攻击等。对于此类攻击，ST 不承担任何责任。因此，无论集成的安全功能和/或 ST 可能提供的任何信息或支持如何，每位客户都需独自负责确定测试的攻击抵御级别是否满足自身的需求，而且除了 ST 产品本身的安全级别之外，还需要考虑将 ST 产品整合到最终产品或应用中的情况。
- ST 产品（包括任何硬件、软件、文档等）的所有安全功能，包括但不限于 ST 添加的任何增强安全功能，均按“原样”提供。因此，除非适用的书面和签署的合同条款另有明确规定，否则在适用法律允许的范围内，ST 不提供任何明示或暗示的担保，包括但不限于对适销性或特定用途适用性的暗示担保。

29 版本历史

表 128. 文档版本历史

日期	版本	变更
2022 年 4 月 12 日	1	初始版本。
2022 年 7 月 21 日	2	*快速编程*部分——更正了行大小。 *表 10：选项字节的构成*现在包含指向选项寄存器的链接，并且删除了重复的描述。 更新和/或更正了选项寄存器的复位值格式。 更新了 *FLASH 安全寄存器 (FLASH_SECR)* 的位 16 的注释。 更新了*第 15.3.18 节：发生外部事件时清除 OCxREF 信号*。 更新了*第 15.4.8 节：TIM1 捕获/比较模式寄存器 1 [复用] (TIM1_CCMR1)*、*第 16.4.8 节：TIM3 捕获/比较模式寄存器 1 [复用] (TIM3_CCMR1)*、*第 17.4.6 节：TIM14 捕获/比较模式寄存器 1 [复用] (TIM14_CCMR1)* 和*第 18.4.7 节：TIMx 捕获/比较模式寄存器 1 [复用] (TIMx_CCMR1)（x = 16 到 17）*中的 OC1M[3:0] 位域说明。 更正了*表 19：不同工作模式下使能的器件资源*中的 USART2 信息。 删除了*第 5.2.6 节：时钟安全系统 (CSS)* 中并不存在的“TIM15”。 在 WWDG *第 21.3.5 节：调试模式*中增加了 DBG 的交叉引用。 更正了*第 22.2 节：RTC 主要特性*（删除了不合逻辑的“或通过入侵事件”）。
2022 年 12 月 3 日	3	首次公开发布。 *第 4.3.2 节：低功耗模式*，增加了有关待机模式的内容。 *第 5.2 节：时钟*，增加了有关 TIMx 的内容。 *图 9：时钟树*。 *第 5.3 节：低功耗模式*，删除了“USART2”。 *将电源相关的 ADC 测量值转换为绝对电压值*部分。 *图 98：刹车事件和刹车 2 电路概述*——更正了图片下方的注释。 *第 15.3.25 节：连接霍尔传感器*——删除了“TIM4”。 *图 201：刹车电路概述*——更正了图片下方的注释。

索引

P

R

S

T

U

W