保密级别：外部使用

UM32M153 用户手册

版本：V1.0

广芯微电子（广州）股份有限公司

http://www.unicmicro.com/

条款协议

1. 本文档中所记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本文档中的电路、软件和相关信息，请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失，广芯微电子不承担任何责任。
2. 在准备本文档所记载的信息的过程中，广芯微电子已尽量做到合理注意，但是，广芯微电子并不保证这些信息都是准确无误的。用户因本文档中所记载的信息的错误或遗漏而遭受的任何损失，广芯微电子不承担任何责任。
3. 对于因使用本文档中的广芯微电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为，广芯微电子不承担任何责任。本文档所记载的内容不应视为对广芯微电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
4. 使用本文档中记载的广芯微电子产品时，应在广芯微电子指定的范围内，特别是在最大额定值、电源工作电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用广芯微电子产品而产生的故障或损失，广芯微电子不承担任何责任。
5. 虽然广芯微电子一直致力于提高广芯微电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，广芯微电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当广芯微电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。

目录

图目录

表目录

1 文档说明

1.1 术语和缩写

表 1-1：术语和缩写表

术语/缩写	全称	描述
rw	read/write	软件能读写此位
r	read-only	软件只能读此位
w	write-only	软件只能写此位，读此位将返回复位值
rc_w1	read/clear	软件可以读此位，也可以通过写‘1’清除此位，写‘0’对此位无影响。
rc_w0	read/clear	软件可以读此位，也可以通过写‘0’清除此位，写‘1’对此位无影响
rc_r	read/clear by read	软件可以读此位，读取此位自动清除此位，写‘0’对此位无影响
rs	read/set	软件可以读此位及对此位写‘1’，写‘0’对此位无影响
rt_w	read-only write trigger	软件可以读此位，对此位写‘0’或‘1’可以触发事件，但不影响此位的值。
t	toggle	软件只能通过写‘1’来翻转此位，写‘0’对此位无影响。
Res.	Reserved	保留位，必须保持默认值不变
SWD	Serial Wire Debug	串行线调试接口，2 线
JTAG	Joint Test Action Group	4 线仿真接口
Word	-	字，32 位长的数据或指令长度
Half-word	-	半字，16 位长的数据或指令长度
B/Byte	-	字节，8 位数据长度
ICP	In Circuit Programming	在电路编程，用户可以通过仿真接口直接将程序下载至芯片中
IAP	In Application Programming	在应用编程，用户可以通过程序对芯片本身进行编程
Option bytes	-	选项字节，用于存储 MCU 的用户配置字
AHB	Advanced High-performance Bus	先进高性能总线
APB	Advanced Peripheral Bus	先进外设总线

1.2 可用的外设

有关本产品系列的全部型号，具体的外设存在与否及数目，请查阅对应的数据手册。

2 存储器和总线架构

2.1 系统架构

系统主要由以下几个模块组成:

- 1 个主单元:
  - Cortex-M0 内核及系统总线（S-bus）
- 4 个从单元:
  - 内部 SRAM
  - 内部闪存存储器
  - AHB 到 APB 的桥，连接所有的 APB 设备
  - AHB 设备总线，专门用于连接 GPIO、RCC、CRC、COPROC 等模块

这些单元内部由一个多级 AHB 总线结构互联，如图 2-1 所示。

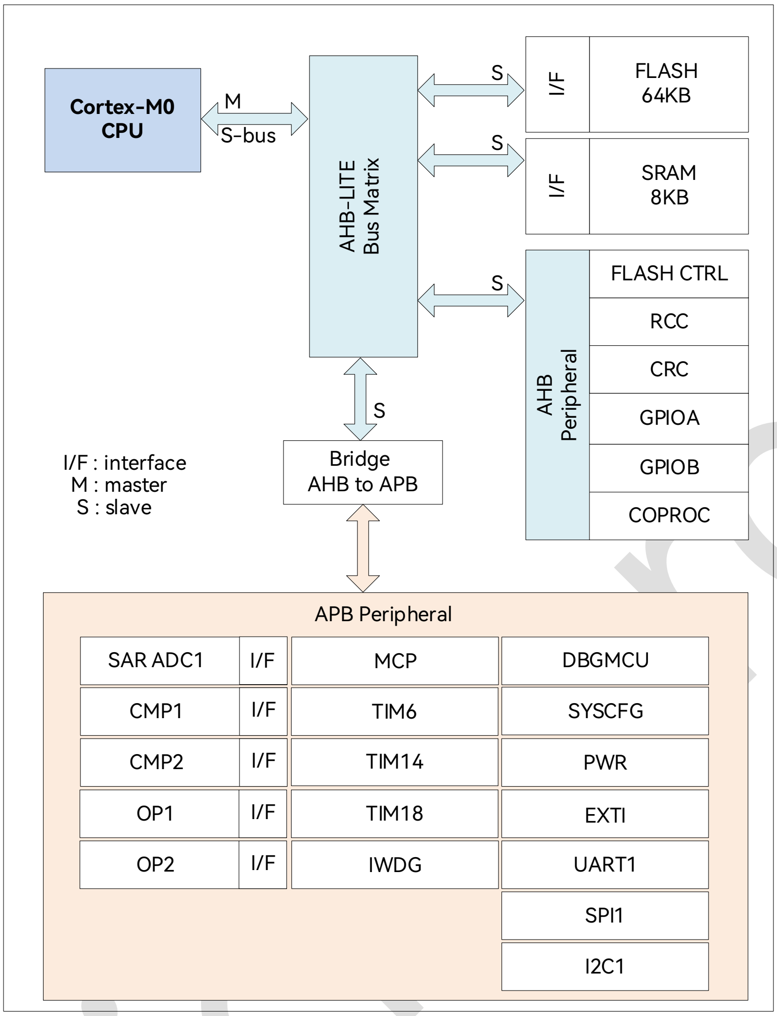


图 2-1：系统架构框图

- **系统总线**

此总线连接 Cortex-M0 内核的系统总线到总线矩阵。

- **总线矩阵（BusMatrix）**

总线矩阵由 1 个主单元总线（系统总线）及 4 个从单元总线（FLASH、SRAM、AHB 设备总线和 AHB2APB 桥）组成。

AHB 外设通过总线矩阵与系统总线相连。

- **AHB 到 APB 桥（AHB2APB bridges—APB）**

AHB 到 APB 桥在 AHB 与 APB 总线间提供同步连接。有关连接到桥的不同外设的地址映射请参考存储器映射章节。

在每次复位之后，所有的外设时钟都处于关闭状态（除 SRAM 及 FLASH 外）。使用外设前，用户必

须打开 RCC_AHBENR、RCC_APB2ENR 或 RCC_APB1ENR 寄存器中相应的时钟使能位。

注：当对 APB 寄存器进行 8 位或者 16 位访问时，该访问会被自动转换成 32 位的访问：AHB2APB 桥会自动将 16 位或者 8 位的数据扩展以配合 32 位的位宽。

2.2 存储器

2.2.1 概述

程序存储器，数据存储器，寄存器及 I/O 口统一编址，其线性地址空间达到 4G。

数据字节以小端格式存放在存储器中，即一个字里的最低地址字节被认为是该字的最低有效字节，而最高地址字节是最高有效字节。

寻址空间分成 8 块，每块 512MB。

其他所有没有分配给片上存储器和外设的存储器空间都是保留的地址空间。详细请参考存储器映像和寄存器编址章节和外设章节。

2.2.2 存储器映像及寄存器编址

表 2-1：存储器映像表

总线	编址范围	大小	外设	备注
AHB	0xE000 0000 - 0xE00F FFFF	1MB	Cortex M0 内部外设	
	0x4800 0800 - 0x5FFF FFFF	~384 MB	Reserved	
	0x4800 0400 - 0x4800 07FF	1KB	GPIOB	
	0x4800 0000 - 0x4800 03FF	1KB	GPIOA	
	0x4002 4400 - 0x47FF FFFF	~128 MB	Reserved	
	0x4002 3400 - 0x4002 37FF	1 KB	COPROC	
	0x4002 3000 - 0x4002 33FF	1 KB	CRC	
	0x4002 2400 - 0x4002 2FFF	3 KB	Reserved	
	0x4002 2000 - 0x4002 23FF	1 KB	FLASH 接口	
	0x4002 1400 - 0x4002 1FFF	3 KB	Reserved	
	0x4002 1000 - 0x4002 13FF	1 KB	RCC	
	0x4002 0000 - 0x4002 0FFF	4 KB	Reserved	
APB	0x4001 5C00 - 0x4001 FFFF	41 KB	Reserved	

总线	编址范围	大小	外设	备注
	0x4001 5800 – 0x4001 5BFF	1 KB	DBGMCU	
	0x4001 5000 – 0x4001 57FF	2 KB	Reserved	
	0x4001 4C00 – 0x4001 4FFF	1 KB	TIM18	
	0x4001 4400 – 0x4001 4BFF	2 KB	Reserved	
	0x4001 4000 – 0x4001 43FF	1 KB	OPAMP	
	0x4001 3C00 – 0x4001 3FFF	1 KB	CMP	
	0x4001 3800 – 0x4001 3BFF	1 KB	UART1	
	0x4001 3400 – 0x4001 37FF	1 KB	Reserved	
	0x4001 3000 – 0x4001 33FF	1 KB	SPI1	
	0x4001 2C00 – 0x4001 2FFF	1 KB	Reserved	
	0x4001 2800 – 0x4001 2BFF	1 KB	MCP	
	0x4001 2400 – 0x4001 27FF	1 KB	ADC	
	0x4001 0800 – 0x4001 23FF	7 KB	Reserved	
	0x4001 0400 – 0x4001 07FF	1 KB	EXTI	
	0x4001 0000 – 0x4001 03FF	1 KB	SYSCFG	
	0x4000 7400 – 0x4000 FFFF	35 KB	Reserved	
	0x4000 7000 – 0x4000 73FF	1 KB	PWR	
	0x4000 5800 – 0x4000 6FFF	6 KB	Reserved	
	0x4000 5400 – 0x4000 57FF	1 KB	I2C1	
	0x4000 3800 – 0x4000 53FF	7 KB	Reserved	
	0x4000 3400 – 0x4000 37FF	1KB	TIM6	
	0x4000 3000 – 0x4000 33FF	1 KB	IWDG	
	0x4000 2400 – 0x4000 2FFF	3 KB	Reserved	
	0x4000 2000 – 0x4000 23FF	1 KB	TIM14	
	0x4000 0000 – 0x4000 1FFF	8 KB	Reserved	
-	0x2000 2000 – 0x3FFF FFFF	~512 MB	Reserved	
SRAM	0x2000 0000 – 0x2000 1FFF	8 KB	SRAM	
-	0x1FFF FC00 – 0x1FFF FFFF	1 KB	Reserved	
Info	0x1FFF F800 – 0x1FFF F9FF	512bytes	Option bytes	
	0x1FFF E800 – 0x1FFF F7FF	4 KB	System memory	
-	0x0800 8000 – 0x1FFF EBFF	~384 MB	Reserved	
EEPROM	0x0808 0000 – 0x0808 07FF	2KB	EEPROM	
FLASH	0x0800 0000 – 0x0800 FFFF	64 KB	Main Flash memory	
-	0x0000 8000 – 0x07FF FFFF	128 MB	Reserved	
-	0x0000 000 – 0x0000 FFFF	64 KB	主闪存存储器，系统存储器或是 SRAM @依赖 BOOT 的配置	

2.2.3 SRAM

最大 8KB 的内置静态 SRAM。它可以以字节（8 位）、半字（16 位）或字（32 位）进行访问。CPU 可以用最快的系统时钟且不插入任何等待访问 SRAM。

2.2.3.1 校验检查 Parity check

用户可以使用用户选项字节(Option byte)中的选项位 RAM_PARITY_CHECK 来启用奇偶校验。

数据总线宽度为 36 位，其中 32 位为数据，4 位用于每字节的奇偶校验位（1bit/byte）。以此来增强数据存储的鲁棒性，以适应欧洲 Class B 及 SIL 规范的数据安全要求。

当写入 SRAM 时，奇偶校验位被计算和存储。当读取时 MCU 自动校验。如果 1 bit 失败，SYSCFG_CFGR2 寄存器中的 SRAM_PEF 位会被置 1，同时产生 NMI 中断。当配置寄存器 SYSCFG_CFGR2 中的 SRAM_PARITY_LOCK 控制位为 1 时，此错误还可以连接到 MCP 的 BKIN 输入。SRAM 校验错误标志（SRAM_PEF）可以通过 SYSCFG_CFGR2 寄存器进行访问，当被硬件置 1 后，可以通过软件对该位写 1 清零。

注：当启用 SRAM 奇偶校验时，建议在代码开始时通过软件初始化整个 RAM 内存，以避免在读取未初始化位置时出现奇偶校验错误。

2.2.4 FLASH 概述

FLASH 存储器有两个不同存储区域：

- 主闪存在储块（Main flash block），它包括应用程序和用户数据区（若需要时）
- 信息块（Information block），其包含两个部分：
  - 选项字节（Option byte），内含硬件及存储保护用户配置选项。
  - 系统存储区（System memory），其包含专有 boot loader 代码，参见<3 嵌入式闪存>章节。

闪存接口基于 AHB 协议执行指令和数据存取。

2.2.5 启动配置

在启动时，通过自举引脚和自举选项控制字可以选择三种自举模式中的一种：

表 2-2：启动配置

启动模式选择		nBOOT_SEL	nBOOT0 bit	启动模式	说明
nBOOT1 bit	BOOT0 pin				
x	0	0	x	主闪存存储器	主闪存存储器选为启动区域
1	1	0	x	系统存储器	系统存储器选为启动区域
0	1	0	x	内置 SRAM	内置 SRAM 选为启动区域
x	x	1	1	主闪存存储器	主闪存存储器选为启动区域
1	x	1	0	系统存储器	系统存储器选为启动区域
0	x	1	0	内置 SRAM	内置 SRAM 选为启动区域

BOOT0 的值根据 nBOOT_SEL 位选择由 nBOOT0 bit 或 BOOT0 Pin 的电平值决定。当器件复位后，在复位的初始阶段，芯片会先判断 nBOOT_SEL 的值，当 nBOOT_SEL 为 1 时，根据 nBOOT0 bit 和 nBOOT1 bit 的值选择启动模式。当 nBOOT_SEL 为 0 时，芯片先采样 BOOT0 Pin 的值，再与 nBOOT1 bit 一起选择启动模式。

当芯片在仿真状态时，硬件会忽略 BOOT0 Pin 的采样值，BOOT0 Pin 固定以低电平与其他条件组合以确定启动条件。

在启动延迟之后，CPU 从地址 0x0000 0000 获取堆栈栈顶的地址，并从启动存储器的 0x0000 0004 指示的地址开始执行代码。

根据选定的启动模式，主闪存存储器，系统存储器或 SRAM 按照以下说明访问：

- 从主闪存存储器启动：主闪存存储器被映射到启动存储空间（0x0000 0000），但仍然能从原有的地址空间（0x0800 0000）访问。即闪存存储器的内容可从两个地址开始访问，0x0000 0000 或 0x0800 0000。
- 从系统存储器启动：系统存储器被映射到启动空间（0x0000 0000），但仍然能够在它原有的地址空间（0x1FFF EC00）访问。
- 从内置的 SRAM 启动：SRAM 映射到启动空间（0x0000 0000），但仍然能够在它原有的地址空间（0x2000 0000）访问。

2.2.5.1 物理重映射（Physical remap）

一旦选择了引导模式，应用程序软件就可以修改代码区域中可访问的内存。这个修改是通过编程 SYSCFG 配置寄存器 1（SYSCFG_CFGR1）中的 MEM_MODE 位来执行的。与 Cortex®M3 和 M4 不同，M0 CPU 不支持向量表的重定位。对位于与 0x0800 0000 不同地址的应用程序代码，必须添加一些额外的代码，以便能够为应用程序中断提供服务。

一个解决方案是通过软件将矢量表重新定位到内部 SRAM：从闪存（映射在应用程序加载地址的基址上）复制向量表到 SRAM 的基址 0x2000 0000。重新映射 SRAM 地址 0x0000 0000，使用 SYSCFG_CFGR1 寄存器。一旦中断发生，Cortex-M0 处理器将从 SRAM 中重新定位的矢量表中获取中断处理程序的起始地址，然后它将跳转到 FLASH 中执行中断处理程序。

注：这个操作应该在应用程序的初始化阶段完成。请参考 IAP 应用例程，以了解更多细节。

2.2.5.2 内嵌的自举程序（Boot loader）

内嵌的自举程序存储在系统存储器，由原厂在生产时写入。该程序可以通过 UART1 对闪存进行重新编程。

3 嵌入式闪存（Flash）

3.1 特性

- 高达 64K 字节闪存存储器
- 存储器结构：
  - 主闪存模块（main flash block）：16K 字（16K×32 位）
  - EEPROM：2Kbytes
  - 信息模块（information block）：5Kbytes

闪存接口的特性为：

- 选项字节加载器
- 闪存编程/擦除操作
- 闪存保护
- 类 EEPROM 编程/擦除操作
- 低功耗模式

3.2 功能描述

3.2.1 闪存结构

闪存空间由 32 位宽的存储单元组成，既可以存代码又可以存数据。主闪存块按 64 页（每页 1K 字节）或 16 扇区（每扇区 4K 字节）分块。

表 3-1：FLASH 模块结构

模块	Flash 存储器地址	大小（byte）	名称	描述
主存储块 Main flash block	0x0800 0000 – 0x0800 03FF	1K	页 0	扇区 0
	0x0800 0400 – 0x0800 07FF	1K	页 1	
	0x0800 0800 – 0x0800 0BFF	1K	页 2	

模块	Flash 存储器地址	大小（byte）	名称	描述
	0x0800 0C00 - 0x0800 0FFF	1K	页 3	
	……	……	……	……
	0x0800 F000 - 0x0800 F3FF	1K	页 60	扇区 16
	0x0800 F400 - 0x0800 F7FF	1K	页 61	
	0x0800 F800 - 0x0800 FBFF	1K	页 62	
	0x0800 FC00 - 0x0800 FFFF	1K	页 63	
信息块 Infomation block	0x1FFF E800 - 0x1FFF F7FF	4K	-	系统存储器
	0x1FFF F800 - 0x1FFF F80F	16	-	Option byte
闪存接口寄存器	0x4002 2000 - 0x4002 2003	4	-	FLASH_ACR
	0x4002 200C - 0x4002 200F	4	-	FLASH_SR
	0x4002 2010 - 0x4002 2013	4	-	FLASH_CR
	0x4002 2014 - 0x4002 2017	4	-	FLASH_AR
	0x4002 2018 - 0x4002 201B	4	-	保留
	0x4002 201C - 0x4002 201F	4	-	FLASH_OBR
	0x4002 2020 - 0x4002 2023	4	-	FLASH_OBUSR

3.2.2 读操作

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

取指令和取数据都是通过相同的 AHB 总线读取访问，通过设置 FLASH 访问控制寄存器（FLASH_ACR），能够控制读操作按照所指定的方式执行。

延迟：等待的周期数，以保证正常的读取。

3.2.3 取指

Cortex-M0 通过 AHB 总线取指。

3.2.4 访问延迟

为了保证对 FLASH 的正确读取，必须在 FLASH 访问控制寄存器 FLASH_ACR 的 LATENCY[2:0]中指定取指速度，这个数值等于每次访问 FLASH 后到下次访问之间所需插入的等待周期的个数。复

位后，这个值默认为零，也就是没有插入等待周期的状态。

3.2.5 FLASH 的写/擦操作

嵌入式闪存支持在电路编程（ICP）以及在应用编程（IAP）。

ICP（In-circuit programming）使用 SWD 或 Bootloader 的方法对 FLASH 内容进行在线写/擦。

IAP（In-application programming）通过使用 MCU 支持的任意通讯接口（I/Os，UART，I2C，SPI 等）下载程序或者数据。IAP 允许用户在运行程序的过程中重写应用程序，前提是一部分应用程序必须预先用 ICP 的方法烧写进去。

烧写和擦除操作在产品的工作电压范围内都支持，相关操作寄存器如下：

- FLASH 控制寄存器（FLASH_CR）
- FLASH 状态寄存器（FLASH_SR）
- FLASH 地址寄存器（FLASH_AR）
- 选项字节寄存器（FLASH_OBR）
- 选项字节用户配置寄存器（FLASH_OBUSR）

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

在对 FLASH 空间进行写/擦除操作时，内部 RC 振荡器（HSI）必须打开。

3.2.5.1 FLASH 空间解锁

复位后，FLASH 存储器默认是受保护状态的，这样可以防止意外的擦除动作。FLASH_CR 寄存器的改写具有加锁保护，只有对 FLASH_KEYR 寄存器执行一个解锁序列才能开启对 FLASH_CR 的访问权限。这个解锁序列由下面 2 个写操作构成：

- 写密钥值 1（KEY1）=0x45670123
- 写密钥值 2（KEY2）=0xCDEF89AB

任何错误的写入值会锁死 FLASH_CR 寄存器直到下次复位。

当写入错误的密钥值时，会由总线错误引发一次硬件错误中断。如果 KEY1 出错就会立即产生中断，或者如果 KEY1 正确但 KEY2 错误就会在 KEY2 错误的时候产生中断。

3.2.5.2 主闪存编程（Main Flash Memory Programming）

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

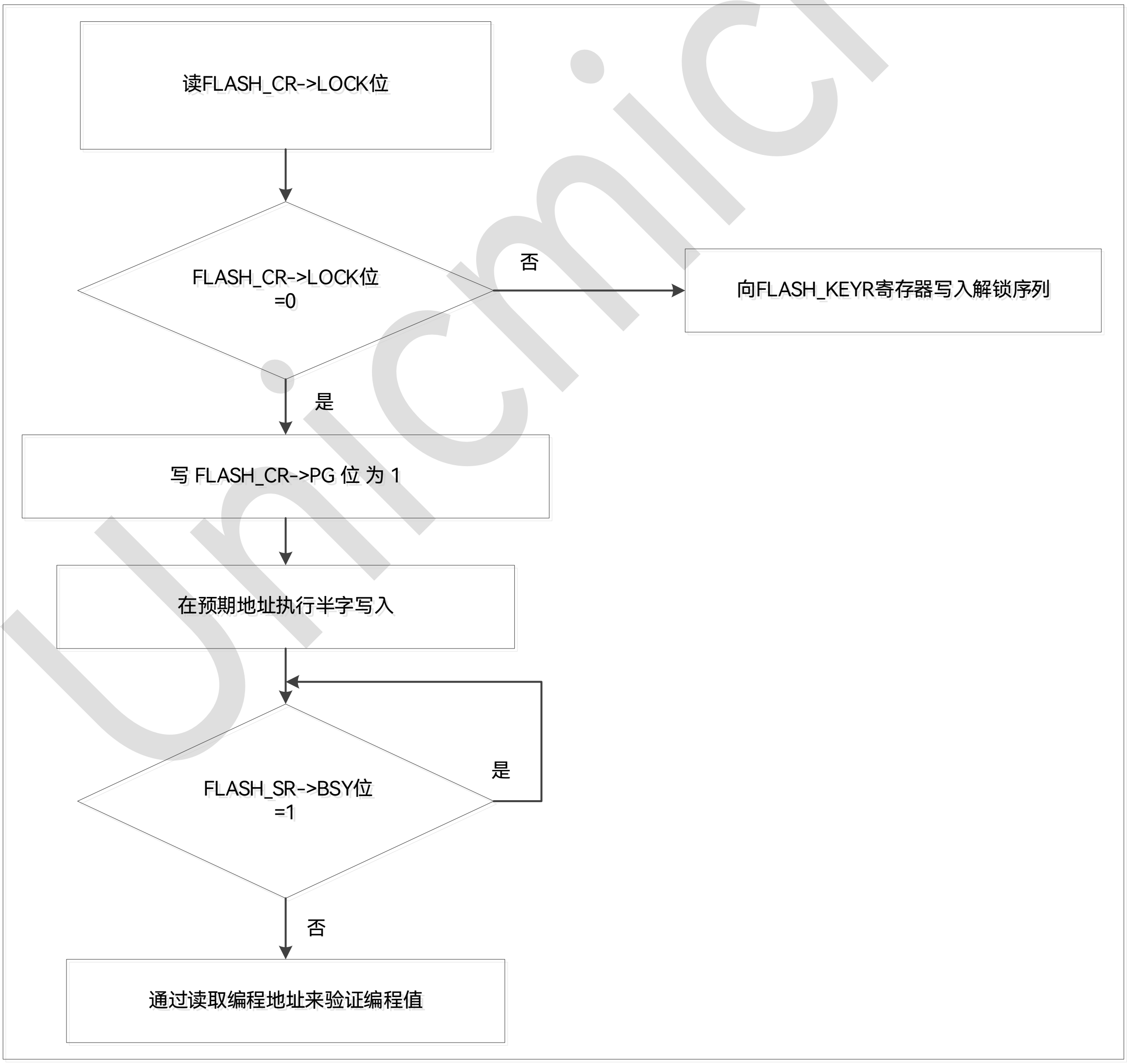


图 3-1：编程流程图

FLASH 存储器接口会预读一下待编程字节是否已被擦除（值是否为全 1），如果不是，那么编程操作会自动取消， 并且 FLASH_SR 寄存器的 PGERR 位被置起，提示编程错误告警。如果被编程的内容为全零（0x0000），则会例外，这时会正确编程并且不告警。

FLASH_SR 寄存器中的 EOP 位在编程动作结束置起，提示编程结束。

主闪存标准编程流程如下:

1. 检查 FLASH_SR 中的 BSY 位，以确认上次编程操作已经结束。
2. 置 FLASH_CR 寄存器中的 PG 位。
3. 以半字（16bits）为单位向目标地址写入数据。
4. 等待 FLASH_SR 寄存器中的 BSY 位归零。
5. 检查 FLASH_SR 中的 EOP 位，编程成功后 EOP 置起，软件清零。
6. 读数据以校验。

注：当 FLASH_SR 中的 BSY 位为 1 的时候，这些寄存器不可访问。

3.2.5.3 FLASH 存储器擦除（Flash Memory Erase）

Flash 存储器支持页擦（Page Erase）和全擦（Mass Erase）。

3.2.5.4 页擦（Page Erase）

页擦操作步骤如下:

1. 检查 FLASH_SR 中的 BSY 位，以确认上次编程操作已经结束。
2. 置 FLASH_CR 寄存器中的 PER 位为 1。
3. 写 FLASH_AR 寄存器以选择待擦除的页。
4. 置 FLASH_CR 寄存器中的 STRT 位为 1。
5. 等待 FLASH_SR 中的 BSY 位归零。
6. 检查 FLASH_SR 中的 EOP 位，编程成功后 EOP 置起，软件清零。
7. 读取已擦除页以校验。

注：设置 STRT 位后，至少等待 1 个 CPU 周期，软件开始检查 BSY 位是否等于“0”。

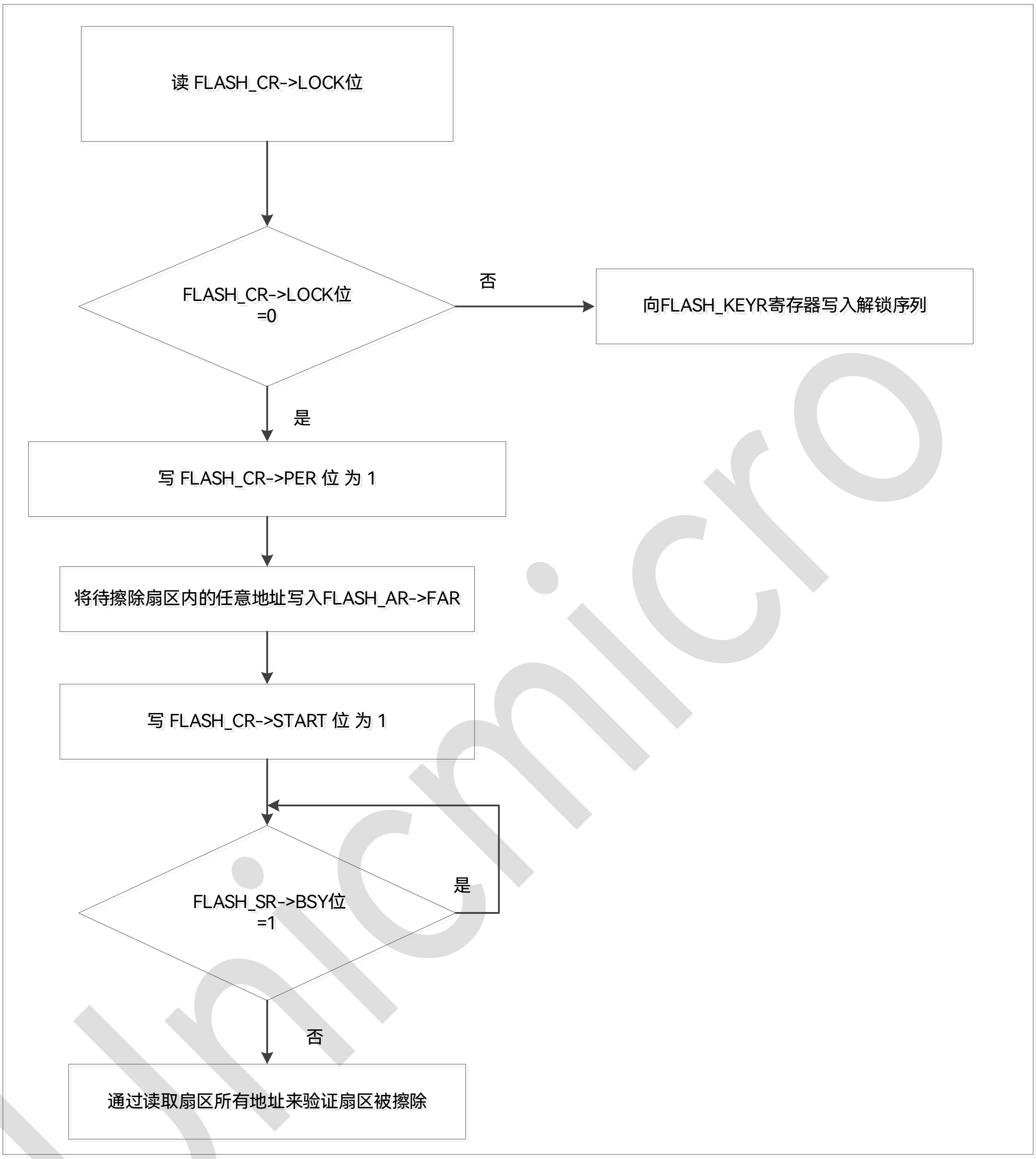


图 3-2：页擦流程图

3.2.5.5 全擦（Mass Erase）

全擦命令可以用来一次擦除主闪存区(main flash block)所有页(page)，但信息块(information block）不受这个命令影响，具体操作步骤如下：

1. 检查 FLASH_SR 中的 BSY 位，以确认上次编程操作已经结束。
2. 置 FLASH_CR 寄存器中的 MER 位为 1。

3. 置 FLASH_CR 寄存器中的 STRT 位为 1。
4. 等待 FLASH_SR 中的 BSY 位归零。
5. 检查 FLASH_SR 中的 EOP 位，编程成功后 EOP 置起，软件清零。
6. 读取全部页并校验。

   注：设置 STRT 位后，至少等待 1 个 CPU 周期，软件开始检查 BSY 位是否等于“0”。

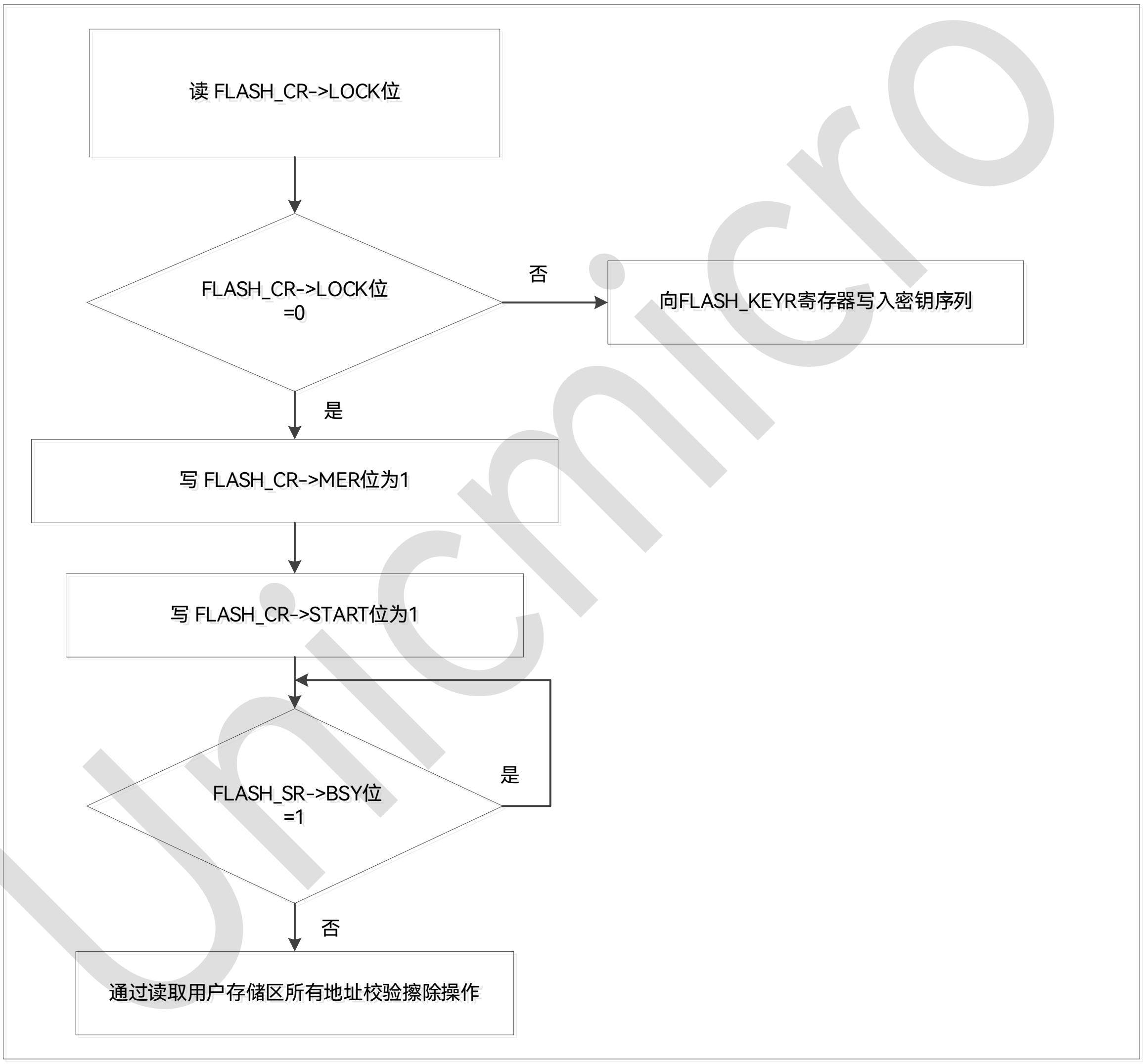


图 3-3：FLASH 全擦流程

3.2.5.6 选项字节编程

选项字节的编程与常规用户地址不同，总共就是 2 个字节（1 个读保护，1 个硬件配置）。通过

对 FLASH_OPTKEYR 写入正确的密钥序列，FLASH_CR 寄存器中的 OPTWRE 位会置 1。然后就可以先置位 FLASH_CR 中的 OPTPG 位，再按半字单位写目标地址。

选项字节编程同样会自动检查选项字节是否已经被擦除（值是否为 1)，如果不是，那么编程操作会自动取消，并且 FLASH_SR 中的 PGERR 位被置位，提示编程错误告警。FLASH_SR 寄存器中的 EOP 位在编程动作结束置起，提示编程结束。

在编程操作开始前，选项字节会自动填充到下个闪存地址，以保证选项字节及其反码总是对的。具体操作步骤如下：

1. 检查 FLASH_SR 寄存器中的 BSY 位，以确保上次编程结束。
2. 解锁使得 FLASH_CR 寄存器中的 OPTWRE 位为 1。
3. 置 FLASH_CR 寄存器中的 OPTPG 位为 1。
4. 写数据（半字）到目标地址。
5. 等待 FLASH_SR 中的 BSY 位归零。
6. 读取编程字节并校验。

当 FLASH 读保护选项字节由保护状态被改成非保护状态时，会自动引发一次整片擦除。如果用户只想改写其他的字节，则不会引发整片擦除，这个机制用于保护 FLASH 的内容。

3.2.5.7 选项字节擦除过程

选项字节的擦除过程如下：

1. 检查 FLASH_SR 寄存器中的 BSY 位，以确保上次编程结束。
2. 解锁使得 FLASH_CR 寄存器中的 OPTWRE 位为 1。
3. 置 FLASH_CR 寄存器中的 OPTER 位为 1。
4. 置 FLASH_CR 寄存器中的 STRT 位为 1。
5. 等待 FLASH_SR 中的 BSY 位归零。
6. 读取擦除字节并校验。

3.2.6 类 EEPROM 区

类 EEPROM 区一共 2K 字节，分为 4 个页，每页 512Bytes。操作方式与 FLASH 区相似，写之前需要先按页擦除，一次只能向对应地址写入 16bits。

3.2.6.1 类 EEPROM 区擦除

类 EEPROM 区的解锁操作与 FLASH 空间解锁操作一样。对类 EEPROM 区的擦除步骤如下:

1. 向 FLASH_CR 寄存器的 PER 位写 1。
2. 向 FLASH_AR 寄存器写入待擦除的页地址。
3. 向 FLASH_CR 寄存器中的 STRT 位写 1。
4. 等待 FLASH_SR 寄存器中的 BSY 位为 0。
5. 判断 FLASH_SR 寄存中的 EOP 位是否为 1，如果为 1，表示编程完成，软件清除 EOP 位。
6. 软件清除 PER 位。

3.2.6.2 类 EEPROM 区编程

编程前同样需要进行解锁操作。编程操作一次只能写入 16bits，步骤如下:

1. 对 FLASH_CR 寄存器的 PG 位写 1。
2. 向类 EEPROM 区的地址写入一个 16 位的数据（地址必须 16 位对齐）。
3. 等待 FLASH_SR 寄存器的 BSY 位为 0。
4. 判断 FLASH_SR 寄存器的 EOP 是否为 1，如果为 1，表示编程完成，软件清除 EOP 位。
5. 软件清除 FLASH_CR 寄存器的 PG 位。

3.3 存储器保护

FLASH 的用户区可以被保护，以防止被不信任代码读取。

3.3.1 读保护

设置选项字节的 RDP 字节对应位，然后系统复位，新的 RDP 被加载，读保护被激活。

注：如果读保护被置位时仍然通过 SWD 连着调试器，可以用 POR（上电复位）代替系统复位。

芯片支持 3 级读保护：LEVEL0（无保护）~LEVEL2（最大限度或禁止调试）

表 3-2：读保护级别及 RDP 对应关系

RDP 字节值	RDP 反码值	读保护级别
0xxAA	0x55	LEVEL 0
任意值 除 0xAA 和 0xCC 外	任意值（不要求互补） 除 0x55 和 0x33 外	LEVEL 1（默认）
0xCC	0x33	LEVEL 2

系统存储区不受读保护字节的影响，但该区域不允许编程和擦除操作。

3.3.1.1 LEVEL0：无保护

针对主 FLASH 区域的读写和擦除操作都被允许，选项字节也全都可以操作。

3.3.1.2 LEVEL1：读保护

这是 RDP 选项字节被擦除之后的默认保护级别。对应的 RDP 值为除 0xAA 和 0xCC 以外的任意值，包括反码不正确情况。

用户模式：在用户模式下执行的代码允许对主 FLASH 和选项字节做全部操作。

Debug、boot RAM 和 boot loader 模式：在调试模式下（with SWD）或运行在 boot RAM 和 boot loader 状态下，主 Flash 区不允许访问。在该状态下，任何简单的读访问都会引起总线错误并触发硬件错误（Hardfault）中断。主 Flash 区同时也禁止写和擦除操作，以防范恶意程序修改代码，任何尝试擦写的操作都会引起 FLASH_SR 中的 PGERR 标志置位。

当 RPD 字节的内容被修改为 0xAA，会先执行全擦除操作，然后改为 LEVEL0 的级别。

3.3.1.3 LEVEL 2：No debug

在这个级别上，包含 LEVEL1 的保护功能，除此之外，Cortex-M0 的调试功能被禁止，所以 SWD 调试口、boot RAM 和 boot loader（boot from system memory）都不再有效。

在用户执行模式下，允许对主 Flash 区做全部操作，相反，对于选项字节却只能读取和写入而并不能做擦除。

此外，RDP 字节不能再改写，因此，LEVEL2 这个保护级别永远不能清除掉，是一个不可恢复性的操作。当试图擦除或编程 RDP 字节时，FLASH_SR 寄存器中的保护错误标志 PRTERR 会被置位，如果 FLASH_CR 中的 ERRIE 位为 1，会引发一个中断。

注 1：在复位条件下，调试功能不能用。

注 2：原厂也不能对打开了 Level 2 保护功能的器件做分析处理。

表 3-3：保护状态和保护级别及运行模式对照表

区域	保护级别	用户代码执行			调试/从 RAM 启动/从系统区域启动		
		读	写	擦除	读	写	擦除
主 FLASH 区	1	Yes	Yes	Yes	No	No	No[3]
	2	Yes	Yes	Yes	N/A[1]	N/A[1]	N/A[1]
系统存储区[2]	1	Yes	No	No	Yes	No	No
	2	Yes	No	No	N/A[1]	N/A[1]	N/A[1]
选项字节	1	Yes	Yes[3]	Yes	Yes[5]	Yes[3][5]	Yes
	2	Yes	Yes[4]	No	N/A[1]	N/A[1]	N/A[1]
类 EEPROM	1	Yes	Yes	Yes	No	No	No[6]
	2	Yes	Yes	Yes	N/A[1]	N/A[1]	N/A[1]

注：

1. 当 Level 2 保护级别使能，调试口、从 RAM 启动和从系统存储区（SYSTEM ROM）启动都被禁止。
2. 系统存储区是唯一在任何情况下可读的区域。
3. 当 RDP 被修改为不保护时，主 Flash 区会被擦除。
4. 所有的选项字节中，除 RDP 字节外都能被再次编程。

5. 当读保护级别为 LEVLE1 时，系统存储区内的代码除了可以将读保护级别从 LEVEL1 改写为 LEVEL0 外，不允许对 option byte 的其他位置进行读写操作。
6. 当保护级别从 LEVE1 修改为 LEVEL0 时，EEPROM 区不会被擦除。

3.3.1.4 改变读保护级别

改变 RDP 的值到其他值（除 0xCC 以外）就可以轻松的从 LEVEL0 级迁移到 LEVEL 1 级别。将 RDP 写成 0xCC，就可以直接进入 LEVEL2 级别。相反的，绕开整片擦除动作而进入 LEVEL0（无保护）级别是不可能的。一旦 RDP 写入 0xAA，就会产生整片擦除（Mass Erase）动作。

注：为了确保保护级别生效，选项字节必须通过 FLASH 控制寄存器中的 OBL_LAUNCH 位强制重新加载。

3.3.2 选项字节写保护

选项字节默认保护开启并可读。为了对选项字节进行写/擦除操作，必须对 FLASH_OPTKEYR 寄存器写入一个密钥序列（KEY1 和 KEY2）。写入正确 FLASH_CR 中的 OPTWRE 会置位，表明解锁成功。同样，通过对该位清零，能够再度禁止对选项字节的写操作。

3.4 Flash 中断

表 3-4：Flash 中断请求

中断事件	事件标志	使能控制位
操作结束	EOP	EOPIE
编程错误	PGERR	ERRIE
保护中断	PRTERR	ERRIE

3.5 选项字节说明

本芯片支持 6 个选项字节，用户可根据需求进行配置。

3.5.1 选项字节格式

每个 32 位的选项字格式如下：

表 3-5：选项字节格式

Bit[31:24]	Bit[23:16]	Bit[15:8]	Bit[7:0]
选项字节 1 反码	选项字节 1	选项字节 0 反码	选项字节 0

注：反码需要软件写入。

3.5.2 选项字节加载

每次上电复位时，选项字节加载逻辑从 Information Block 读取选项字节内容，并将数据存储到选项字节寄存器 FLASH_OBR 和选项字节用户配置寄存器 FLASH_OBUSR 中。在选项字节加载期间，将对选项字节和相应的反码选项字节进行互补校验，若校验失败，生成选项字节错误（OPTERR），对应的 option byte 被认定为 0xFF。

注：如果选项字节和它的反码选项字节都等于 0xFF（擦除态），选项字节错误不会产生。

3.5.3 选项字节存储映射

表 3-6：选项字节地址映射表

地址	Bit[31:24]	Bit[23:16]	Bit[15:8]	Bit[7:0]
0x1FFF F800	nUSER	USER	nRDP	RDP
0x1FFF F804	nDATA1	DATA1	nDATA0	DATA0
0x1FFF F830	nUSER2	USER2	nUSER1	USER1

注：

1. 选项字节（用户 option 或读保护）配置，可以从上表中的存储器地址直接读取，也可以从选项字节寄存器（FLASH_OBR）读取获得。
2. 新写入的选项字节（用户 option，读/写保护）在系统复位后不加载。要重新加载，需要 POR 或将 OBL_LAUNCH 位设置为'1'。

3.5.4 用户及读保护选项字节

地址：0x1FFF F800

出厂值：0x8877 55AA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nUSER								nBOOT_SEL	RAM_PARITY_CHECK		nBOOT1	nBOOT0		nRST_STOP	WDG_SW
rw								rw	rw		rw	rw		rw	rw

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

位	名称	描述
31:24	nUSER	USER 反码
23	nBOOT_SEL	0：BOOT0 Pin 与 nBOOT1 一起控制芯片的启动模式 1：nBOOT0 与 nBOOT1 一起控制芯片的启动模式
22	RAM_PARITY_CHECK	Ram 奇偶校验 0：使能 1：关闭
21	-	保留位，固定为 1
20	nBOOT1	当 nBOOT_SEL 为 1 时，此位与 nBOOT0 位一起控制芯片的启动模式；当 nBOOT_SEL 为 0 时，此位与 BOOT Pin 的值一起控制芯片的启动模式。
19	nBOOT0	当 nBOOT_SEL 为 1 时，此位与 nBOOT1 位一起控制芯片的启动模式。
18	-	保留位
17	nRST_STOP	0：进入 STOP 模式产生复位 1：进入 STOP 模式不产生复位
16	WDG_SW	0：硬件看门狗 1：软件看门狗
15:8	nRDP	RDP 反码
7:0	RDP	读保护选项字节 0xAA：LEVEL0（出厂值） 0xXX：除了 0xAA 和 0xCC 其他值，LEVEL1 0xCC：LEVEL2 注：读保护状态被加载到 FLASH_OBR 寄存器。

3.5.5 用户数据选项字节

地址：0x1FFF F804

出厂值：0x00FF 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nDATA1								DATA1							
rw								rw							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
nDATA0								DATA0							
rw								rw							

位	名称	描述
31:24	nDATA1	DATA1 反码
23:16	DATA1	用户数据选项字节 1，被加载到 FLASH_OBR 寄存器
15:8	nDATA0	DATA0 反码
7:0	DATA0	用户数据选项字节 0，被加载到 FLASH_OBR 寄存器

3.5.6 用户配置选项字节

地址：0x1FFF F830

出厂值：0x00FF C03F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nUSER2								USER2							
rw								rw							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
nUSER1								MCP_PINSEL[1:0]							
rw								rw							

位	名称	描述
31:24	nUSER2	USER2（bit23~bit16）的反码
23:16	USER2	用户配置选项字节 2，保留位
15:8	nUSER1	USER1（bit7~bit0）的反码

<table>
<tr><th>位</th><th>名称</th><th>描述</th></tr>
<tr><td>7:6</td><td>MCP_PINSEL[1:0]</td><td>MCP 输出通道选择位
<table>
<tr><td></td><td>PB4</td><td>PB5</td><td>PB6</td><td>PB7</td><td>PB8</td><td>PB9</td></tr>
<tr><td>00</td><td>MCP_CH1</td><td>MCP_CH2</td><td>MCP_CH3</td><td>MCP_CH1N</td><td>MCP_CH2N</td><td>MCP_CH3N</td></tr>
<tr><td>01</td><td>MCP_CH3N</td><td>MCP_CH2N</td><td>MCP_CH1N</td><td>MCP_CH3</td><td>MCP_CH2</td><td>MCP_CH1</td></tr>
<tr><td>10</td><td>MCP_CH1</td><td>MCP_CH1N</td><td>MCP_CH2</td><td>MCP_CH2N</td><td>MCP_CH3</td><td>MCP_CH3N</td></tr>
<tr><td>11</td><td>MCP_CH3N</td><td>MCP_CH3</td><td>MCP_CH2N</td><td>MCP_CH2</td><td>MCP_CH1N</td><td>MCP_CH1</td></tr>
</table></td></tr>
<tr><td>5:0</td><td>-</td><td>保留位</td></tr>
</table>

注：用户配置选项字节在复位时会被加载到 FLASH_OBUSR 寄存器中。

3.6 相关寄存器

3.6.1 Flash 访问控制寄存器（FLASH_ACR）

偏移地址：0x000

复位值：0x0000 0000

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

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

位	名称	描述
31:3	RSV	保留位，始终读为 0
2:0	LATENCY[2:0]	延迟控制，设置 SYSCLK（系统时钟）周期和 Flash 访问时间的比率关系： 000：零等待（<24MHz） 001：1T 等待（24MHz~48MHz） 010：2T 等待（>48MHz） 其他：保留 注 1：当系统时钟选择 HSI 18MHZ 时，适合配置为“000”； 注 2：当系统时钟选择 HSI 72MHZ 时，若 AHB 预分频为 1 时，适合配置为“010”；AHB 预分频为 2 时，适合配置为“001”；其余 AHB 预分频配置，适合配置“000”。

3.6.2 FLASH 区域解锁寄存器（FLASH_KEYR）

偏移地址：0x004

复位值：0x0000 0000

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

位	名称	描述
31:0	KEY[15:0]	该位段用于写入解锁的 FLASH 的密钥值 注：读此位段固定为 0

3.6.3 选项字节区域解锁寄存器（FLASH_OPTKEYR）

偏移地址：0x008

复位值：0x0000 0000

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

位	名称	描述
31:0	OPTKEY[31:0]	该位段用于写入解锁的 option byte 的密钥值 注：读此位段固定为 0

3.6.4 Flash 状态寄存器（FLASH_SR）

偏移地址：0x00C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										EOP	PRTERR	Res.	PGERR	Res.	BSY
										rc_w1	rc_w1		rc_w1		r

位	名称	描述
31:6	RSV	保留位，始终读为 0
5	EOP	操作结束（end of operation） 当 Flash 操作（写/擦除）完成时由硬件置 1 ，软件写 1 清零 注 ：用 EOP 可以判断是否每次操作都顺利完成
4	PRTERR	保护错误标志 当芯片处于读保护 LEVEL2 级别时，任何对芯片的写操作会导致该位被硬件置 1。该位可以通过软件写 1 清零。
3	RSV	保留位，始终读为 0
2	PGERR	写入错误标志（programming error） 当被编程区域的状态不为 '0xFFFF' 的情况下，执行写入操作时被硬件置 1 ，软件写 1 清零
1	RSV	保留位，始终读为 0
0	BSY	忙标志（busy） 该位标明 Flash 操作处于进行中。当开始 Flash 操作的时候被硬件置位，当操作结束时或发生错误时被硬件清零。

3.6.5 Flash 控制寄存器（FLASH_CR）

偏移地址：0x010

复位值：0x0000 0080

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		OBL_LAUNCH	EOPIE	Res.	ERRIE	OPTWRE	Res.	LOCK	STRT	OPTER	OPTPG	Res.	MER	PER	PG
		rs	rw		rw	rc_w0		rs	rs	rw	rw		rw	rw	rw

位	名称	描述
31:14	RSV	保留位，始终读为 0
13	OBL_LAUNCH	选项字节强制更新，当被写为 1 时，该位强制选项字节的重加载，该操作会引起系统复位。 0：无效 1：有效
12	EOPIE	操作结束中断使能 该位使能操作结束中断，使得 FLASH_SR 中的 EOP 位变成 1 的时候产生中断请求 0：中断禁止 1：中断使能
11	RSV	保留位，始终读为 0
10	ERRIE	错误中断使能 该位使能操作错误中断，使得 FLASH_SR 中的 PGERR 位或 PRTERR 位变成 1 的时候产生中断请求 0：中断禁止 1：中断使能
9	OPTWRE	选项字节写使能 该位为 1 时，选项字节即允许改写。对 FLASH_OPTKEYR 寄存器写入正确的密钥序列就可以将它置 1。 该位可通过软件写 0 清零。
8	RSV	保留位，始终读为 0
7	LOCK	FLASH 锁定位 软件只能将此位由 0 写到 1。当此位为 1 时，表示 FLASH 处于锁定状态。 当软件对 FLASH_KEY 寄存器执行解锁序列成功时，此位由硬件设置为 0。如果解锁失败，此位保持为 1
6	STRT	启动位 该位会触发一个擦除操作，仅由软件置 1，仅会在

位	名称	描述
		BSY 被清零时清零。
5	OPTER	选择选项字节擦除
4	OPTPG	选择选项字节编程
3	RSV	保留位，始终读为 0
2	MER	全擦除（Mass Erase） 选择全擦除（所有用户 pages 擦除）
1	PER	页擦除（Page Erase） 选择页擦除 注：此位用于 main flash 和 EEPROM 区的页擦除
0	PG	编程 选择 FLASH 或 EEPROM 编程（写入）

3.6.6 地址寄存器（FLASH_AR）

偏移地址：0x014

复位值：0x0000 0000

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

位	名称	描述
31:0	FAR[31:0]	FLASH 地址 当 PG 位被选中时，选择待写入的地址，或当 PER 位被选中时，选择待擦除的页地址。 注 1：本寄存器由硬件根据当前和上次操作的地址更新。对于页擦除操作，该寄存器该由软件来更新以便匹配要擦除的页。 注 2：当 FLASH_SR 中的 BSY 位为 1 时，对这个寄存器的写访问将被阻止。

3.6.7 选项字节寄存器（FLASH_OBR）

偏移地址：0x01C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA1[7:0]								DATA0[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nBOOT_SEL	RAM_PARITY_CHECK	Res.	nBOOT1	nBOOT0	Res.	nRST_STOP	WDG_SW	Res.					RDPRT[1:0]		OPTERR
r	r		r	r		r	r						r		r

位	名称	描述
31:24	DATA1[7:0]	DATA1
23:16	DATA0[7:0]	DATA0
15	nBOOT_SEL	用户选项字节，启动配置选择项
14	RAM_PARITY_CHECK	用户选项字节，Ram 奇偶校验
13	RSV	保留位，始终读为 0
12	nBOOT1	用户选项字节，nBOOT1
11	nBOOT0	用户选项字节，nBOOT0
10	RSV	保留位，始终读为 0
9	nRST_STOP	用户选项字节，nRST_STOP
8	WDG_SW	用户选项字节，WDG_SW
7:3	RSV	保留位，始终读为 0
2:1	RDPRT[1:0]	读保护级别状态 00：读保护 LEVEL0（出厂默认配置） 01：读保护 LEVEL1 1x：读保护 LEVEL2
0	OPTERR	选项字节错误 此位置 1，表示加载的选项字节和反码字节不匹配； FLASH_OBR 寄存器相对应的字节被设置为 0xFF。

3.6.8 选项字节用户配置寄存器（FLASH_OBUSR）

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.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USER2[7:0]								USER1[7:0]							
r								r							

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:8	USER2[7:0]	用户选项字节 USER2 的值
7:0	USER1[7:0]	用户选项字节 USER1 的值

4 电源控制（PWR）

4.1 特性

- VDD 为 I/O 引脚和内部调压器供电。
- VSSA，VDDA：为 ADC、复位模块、振荡器的模拟部分提供供电。VDDA 和 VSSA 分别连接到 VDD 和 VSS。

本产品内部集成了上电复位（POR）/掉电复位（PDR）电路，该电路始终处于工作状态，保证系统在供电超过 2.0V 时工作；当 VDD 低于设定的阀值（VPOR/VPDR）时，置器件于复位状态，而不必使用外部复位电路。

器件中还有一个可编程电压监测器（PVD），它监视 VDD/VDDA 供电并与阀值 VPVD 比较，当 VDD 低于或高于阀值 VPVD 时产生中断，中断处理程序可以发出警告信息或将微控制器转入安全模式。PVD 功能需要通过程序开启。

4.2 电源框图

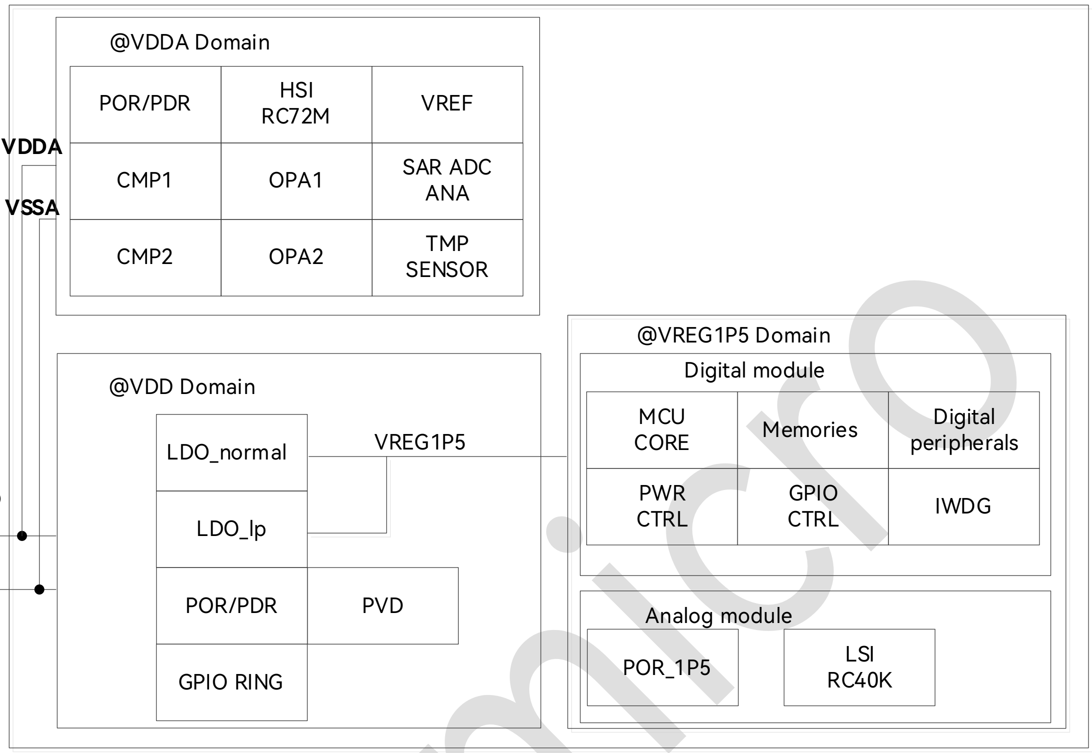


图 4-1：电源框图

4.2.1 ADC 供电和参考电压

ADC 使用独立的电源供电（VDDA 和 VSSA），VDDA 和 VSSA 分别连接到 VDD 和 VSS。

4.2.2 电压调节器（Voltage Regulator）

芯片包含 2 个电压调节器：VDD 电压域下的 LDO_normal 和 LDO_lp，输出连接在一起（VREG1P5），复位后电压调节器 LDO_normal 和 LDO_lp 保持打开，LDO_lp 在任何模式下都不会关闭，LDO_normal 根据应用可配置为 2 种不同的工作模式：

- 运行模式和睡眠模式：调节器 LDO_normal 开启，提供全功耗模式为 1.5V 电压域（内核，内存和数字外设）提供电源。
- 停止模式：调节器 LDO_normal 可以选择开启或关闭，LDO_normal 关闭时，LDO_lp 以低功耗模式为 1.5V 电压域提供电源，为 VREG1P5 电压域下寄存器及 SRAM 保持数据。

4.3 电源管理器

4.3.1 POR/PDR

芯片内置 POR（power on reset）和 PDR（power down reset）电路，一直保持有效，保证最低工作电压以上正常工作。

当被监测电源的电压低于限定门限时，器件维持复位状态，而无需外部复位电路。有关上电与掉电复位的电压门限请参见数据手册的电气特性章节。

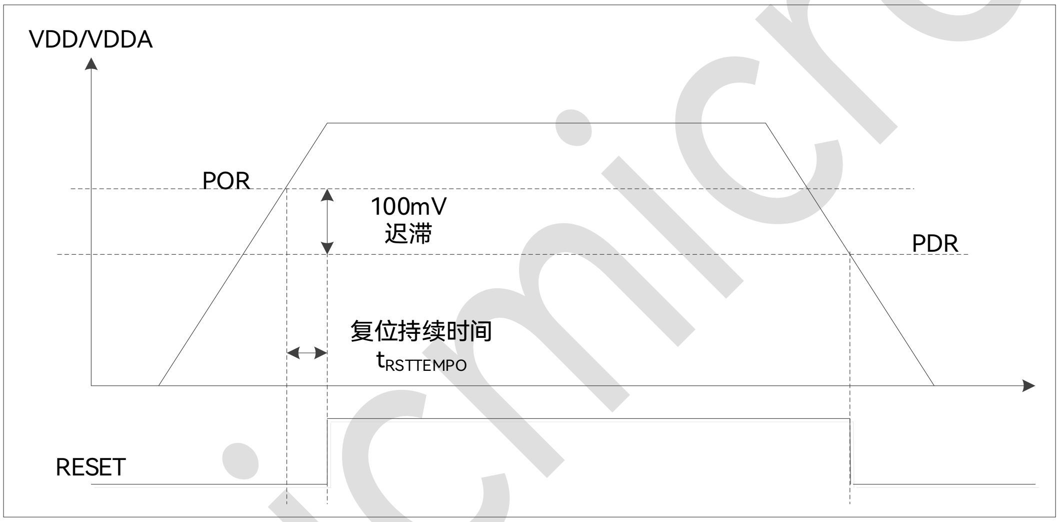


图 4-2：POR/PDR 波形图

4.3.2 PVD (Programmable Voltage Detector)

用户可以通过设置 PVD 的电压阀值去监视 VDD 电源电压。

设置控制寄存器 PWR_CR 中 PLS[2:0]位，选择不同的电压阈值；通过设置 PVDE 位来使能 PVD。

电源控制/状态寄存器 PWR_CSR 中的 PVDO 标志用来表明 VDD 是高于还是低于 PVD 的电压阀值。该事件在内部连接到外部中断 EXTI 的 Line16，如果该中断被使能，该事件就会产生中断。根据 Line16 沿触发设置（上升/下降沿）就会触发对应的 PVD 中断。

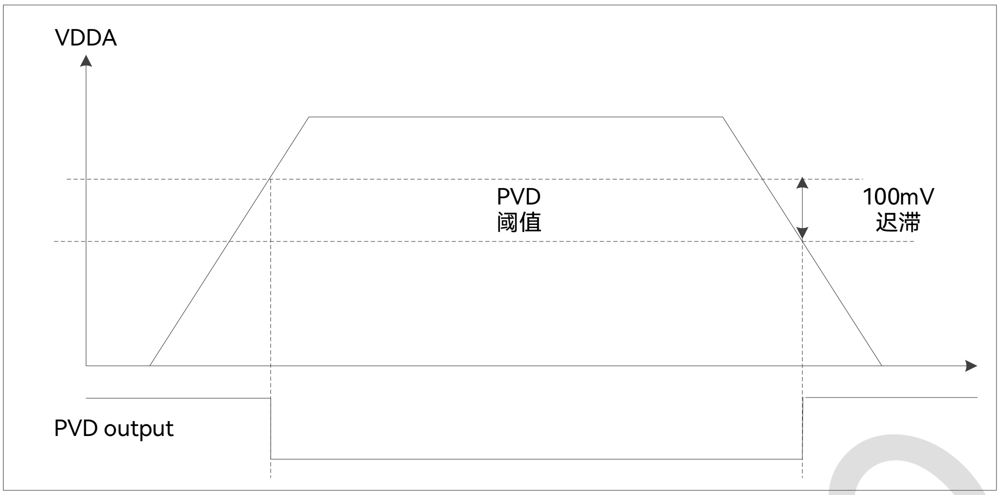


图 4-3：PVD 波形图

4.4 低功耗模式

本产品支持低功耗模式，用户可根据最低功耗、最短启动时间和可用的唤醒事件等条件，选择最佳的低功耗模式，进而达到最佳的平衡。

本产品支持 2 种低功耗模式：SLEEP 模式和 STOP 模式。

此外，运行模式（RUN mode）下，可以通过以下方式降低功耗：

- 降低系统时钟频率
- 关闭未使用外设时钟

表 4-1：低功耗模式

模式	进入	唤醒	对 VREG1P5 区域时钟的影响	对 VDD/VDDA 区域时钟的影响	电压调节器
睡 眠（SLEEP）(SLEEP NOW or SLEEP ON EXIT)	WFI	任一中断	CPU 时钟关闭，对其它时钟及模拟时钟无影响	无	LDO_normal 开启 LDO_lp 开启
	WFE	唤醒事件			
停机（STOP）	LPDS 位 + SLEEPDEEP 位 + WFI 或 WFE	任一外部中断（在 EXTI 寄存器中设置） 指定通信口接收事件（UART, I2C）	大多数 VREG1P5 区域（shut-down module）时钟关闭； 部分模块（always-on module）时钟开启	HSI 振荡器关闭	LDO_normal 可选择开启或关闭；依据电源控制寄存器(PWR_CR)的设置) LDO_lp 开启

4.4.1 降低系统时钟频率

在运行模式中，系统时钟（SYSCLK，HCLK，PCLK）可通过可编程预分频寄存器降速。在进入睡眠模式前这些分频器也可用来降低外设时钟。

4.4.2 看门狗复位

在运行模式下，可随时通过停止对外设和存储器提供时钟（HCLK 和 PCLK）来减少功耗。为了在睡眠模式下减少更多功耗，可在执行 WFI 或 WFE 的指令前关闭外设时钟。

外设时钟门控寄存器包括：AHB 外设时钟使能寄存器 RCC_AHBENR、APB 外设时钟使能寄存器 RCC_APB2ENR 和 RCC_APB1ENR。

4.4.3 睡眠模式（SLEEP MODE）

4.4.3.1 进入睡眠模式

执行 WFI（等待中断）或 WFE（等待事件）指令可让芯片进入睡眠模式。配置 Cortex-M0 系统控制寄存器中 SLEEPONEXIT 位，有两种选项可用于选择睡眠模式的进入机制：

- SLEEP NOW：当 SLEEPONEXIT 位清 0 时，只要执行 WFI 或 WFE 指令 MCU 就立即进入睡眠模式。
- SLEEP ON EXIT：当 SLEEPONEXIT 位置 1 时，MCU 从最低优先级的中断服务程序中退出时，MCU 立即再进入睡眠模式。

4.4.3.2 退出睡眠模式

如果是执行 WFI 指令进入了睡眠模式，任意一个 NVIC 中断都可唤醒 MCU，退出睡眠模式。

如果是执行 WFE 指令进入了睡眠模式，当任一事件发生时，MCU 退出睡眠模式。

唤醒事件可通过以下方式产生：

- 在外设控制寄存器中使能一个中断，但不在相应 NVIC 中使能，并且在 Cortex-M0 系统控制寄

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

- 配置一个外部或内部 EXTI 线作为事件模式。当 CPU 从 WFE 唤醒后，因为与事件 Line 对应的挂起位未被设置，不需要清除外设的中断挂起位或外设的 NVIC IRQ 通道挂起位。

因没有在中断的进入或退出上浪费时间，所以 WFE 模式唤醒所需时间最短。

表 4-2：SLEEP NOW 模式

SLEEP NOW 模式	说明
进入	在以下条件下执行 WFI（等待中断）或 WFE（等待事件）指令： • SLEEPDEEP = 0 • SLEEPONEXIT = 0 参考 Cortex-M0 系统控制寄存器。
退出	当执行 WFI 指令进入睡眠模式： • 中断唤醒：参考中断向量表 当执行 WFE 指令进入睡眠模式： • 事件唤醒：参考事件管理章节
唤醒延时	无

表 4-3：SLEEP ON EXIT 模式

SLEEP ON EXIT 模式	说明
进入	在以下条件下执行 WFI 指令： • SLEEPDEEP = 0 • SLEEPONEXIT = 1 参考 Cortex-M0 系统控制寄存器。
退出	中断唤醒：参考中断向量表
唤醒延时	无

4.4.3.3 睡眠模式中的 I/O 状态

在睡眠模式下，所有的 I/O 口状态都保持与运行模式一致。

4.4.4 停机模式（STOP MODE）

停机模式是在 Cortex-M0 的深度睡眠模式基础上结合了外设时钟控制的一种低功耗模式。在停止模式下电压调节器可运行在正常（normal，LDO_normal 开启）或低功耗（low power，LDO_normal 关闭）模式。此时在 VREG1P5 供电区域的的绝大多数时钟都被停止，HSI 振荡器被

禁止，VREG1P5 电压域下的 SRAM 和寄存器数据被保持。

4.4.4.1 进入停止模式

关于如何进入停止模式，参见下表。

在停止模式下，通过设置电源控制寄存器（PWR_CR）的 LPDS 位使调节器（VREG1P5）进入低功耗模式，能够降低更多的功耗。

如果正在进行闪存编程，须等到对存储器的访问完成，系统才进入停止模式。

如果正在进行对 APB 的访问，须等到对 APB 访问完成，系统才进入停止模式。

在停机模式中，可通过对独立的控制位进行编程来选择如下功能：

- 独立看门狗（IWDG）：独立看门狗可通过写看门狗密钥寄存器或硬件选项配置来启动。一旦启动了独立看门狗，看门狗会一直开启除非进行系统复位。详见独立看门狗（IWWDG)章节。
- 内部低速 RC 振荡器（LSI）：通过控制/状态寄存器（RCC_CSR）的 LSION 位来设置。

在停机模式下，如果在进入该模式前 ADC 没有被关闭，那么仍然消耗功耗。可通过设置寄存器 ADC_CR 的 ADON 位关闭此外设。

4.4.4.2 退出停止模式

有关如何退出停机模式，详见下表。

当中断或事件唤醒 MCU 退出停机模式时，HSI 振荡器被选为系统时钟。

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

表 4-4：STOP 模式进入和退出

STOP 模式	说明
进入	在以下条件下执行 WFI 或 WFE 指令： • SLEEPDEEP = 1，参考 Cortex-M0 系统控制寄存器 通过设置 LPDS（PWR_CR）位选择电压调节器模式 注：为了进入停机模式，所有的外部中断请求挂起位（在挂起寄存器(EXTI_ PR)中）、所有外设中断挂起位标志必须清除。否则，系统会忽略 WFI 或 WFE 指令程序继续运行。

STOP 模式	说明
退出	如果是执行了 WFI 批令进入了停止模式： • 任一外部中断线配置为中断模式（在 NVIC 中必须使能相应的 EXTI 中断向量）。 • 一些特定的通信外设（UART, I2C）中断，设置为唤醒模式（该外设必须配置为唤醒模式且在 NVI 中相应的中断向量必须使能）。 参见中断向量表 如果是执行了 WFE 指令进入了停止模式： • 任一外部中断线配置为事件模式。 参考事件管理章节
唤醒延时	HSI 唤醒时间 + 调压器从低功耗模式唤醒的时间

4.4.4.3 停机模式中的 I/O 状态

在停机模式下，所有的 I/O 口状态都保持与运行模式一致。

注：

1. UART1 和 I2C1 模块开启 Stop 唤醒功能并选择 HSI18M，则 Stop 模式 HSI 不会强制关闭。
2. STOP 模式下，LDO_normal 可配置开启/关闭模式。

4.4.5 调试模式

默认情况下，当使用调试功能时，如果应用程序将 MCU 置于停机模式，则调试连接将丢失。这是由于 Cortex-M0 内核已无时钟。

但是，通过设置 DBGMCU_CR 寄存器中的某些配置位，即使低功耗模式下也可以调试软件。

4.5 相关寄存器

4.5.1 电源控制寄存器（PWR_CR）

偏移地址：0x000

复位值：0x0000 0000

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

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

位	名称	描述
31:8	RSV	保留位，始终读为 0
7:5	PLS[2:0]	PVD 电压阈值选择 当 PVD_LOCK（SYSCFG 配置寄存器 SYSCFG_CFGR2）被使能，则 PLS[2:0]不能再被改写。 000：2.4V 001：2.4V 010：2.4V 011：2.5V 100：2.6V 101：2.7V 110：2.8V 111：2.9V 注：详细参见数据手册电气特性章节
4	PVDE	PVD 使能，软件清零和置位 当 PVD_LOCK（SYSCFG 配置寄存器 SYSCFG_CFGR2）被使能，则 PVDE 不能再被改写。 0：PVD 禁止 1：PVD 使能
3:1	RSV	保留位，始终读为 0
0	LPDS	深度睡眠下低功耗（low-power deepsleep） 0：调压器 VREG1P5 工作在正常模式 1：调压器 VREG1P5 工作在低功耗模式

4.5.2 电源控制/状态寄存器（PWR_CSR）

偏移地址：0x004

复位值：0x0000 0000

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

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

位	名称	描述
31:3	RSV	保留位，始终读为 0
2	PVDO	PVD 输出状态 此位只能硬件清零和置位，仅 PVD 使能（PVDE=1）情况下有效。 0：VDD 电压高于 PVD 设置阈值 1：VDD 电压低于 PVD 设置阈值 注：若 PVD 使能并配置，PVDO 可用于产生中断（连接到 EXTI_Line16）
1:0	RSV	保留位，始终读为 0

5 复位与时钟控制（RCC）

5.1 复位

本芯片支持 2 类复位：系统复位和电源复位。

5.1.1 系统复位

系统复位不能复位时钟控制/状态寄存器 RCC_CSR 的复位标志，其他寄存器都可以被系统复位。

当以下事件之一发生时，产生一个系统复位：

1. NRST 引脚上的低电平（外部复位）
2. 独立看门狗事件（IWDG 复位）
3. 软件复位（SW 复位）
4. 低功耗管理复位
5. 选项字节加载复位

访问控制/状态寄存器 RCC_CSR 中的复位状态标志位可识别复位事件来源。

芯片复位后，NRST/PB0 引脚固定用作复位功能。软件可以通过将 SYSCFG_CFGR2 中的 RSTCFG 位设置为 1，使得 NRST/PB0 引脚可以当作普通 GPIO 引脚使用。当 PB0 需要从普通 GPIO 功能切换到复位引脚时，硬件会读取 PB0 的电平，如果电平为低，软件无法切换回复位功能。

复位入口向量被固定在 0x0000_0004 地址处。

如果 NRST/PB0 引脚被当作复位引脚使用，当 NRST/PB0 引脚被拉低会产生外部复位。

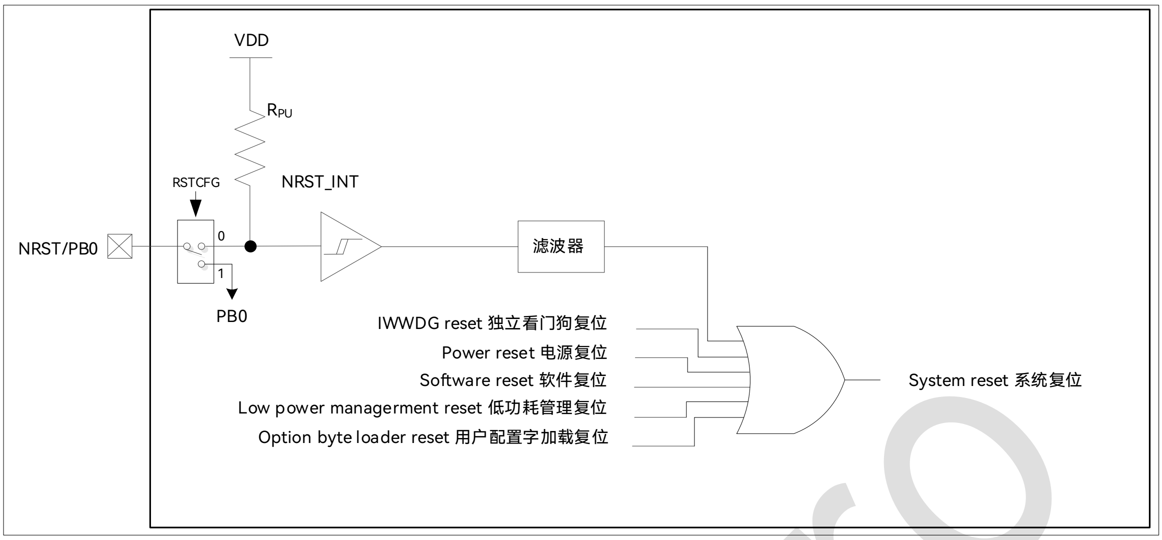


图 5-1：复位电路

5.1.1.1 软件复位

设置 SYSRESETREQ 为 1 可以产生一次软件复位，SYSRESETREQ 位在 Cortex-M0 的应用中断和复位控制寄存器（Application Interrupt and Reset Control Register）中。

详细请参见 Cortex™-M0 technical reference manual。

5.1.1.2 低功耗管理复位

在以下情况下可产生低功耗管理复位：

在进入停机（STOP）模式时产生低功耗管理复位，将用户选项字节中的 nRST_STOP 位置 1 将使能该复位。这时，若 MCU 进入停机模式，系统将被复位而不是进入停机模式。

5.1.1.3 选项字节装载器复位

设置 OBL_LAUNCH 位（FLASH_CR 寄存器中）为 1，将发生选项字节装载器复位，此位用于软件重加载选项字节。

5.1.2 电源复位（POR/PDR）

当以下事件发生时，产生电源复位：

上电/掉电复位（POR/PDR）

电源复位将复位所有寄存器。

5.2 时钟

本芯片包含以下时钟源：

- HSI72（72M RC 振荡器时钟）
- HSI（18MHz，由 HSI72 经过 4 分频后得到）
- LSI（40kHz 内部低速 RC 振荡器，为 IWDG 提供时钟）

每个时钟源都可以独立打开/关闭，当不使用时，可以关闭以降低功耗。

其中 HSI 和 HSI72 都可作为系统时钟（SYSCLK）。

AHB 和 APB 时钟域内建独立的可配置的分频器，AHB/APB 域最大时钟频率为 72MHz。

Cortex 系统定时器（SYSTICK）由 AHB 时钟驱动，其可由 AHB/8 或 AHB 时钟频率直接驱动（通过 Cortex Systick 控制/状态寄存器来配置）。

FCLK 是 Cortex-M0 的自由运行时钟，详见 ARM Cortex™-M0 technical reference manual (TRM)。

除以下几个外设，其他所有的外设时钟由其所在的总线时钟（HCLK 或 PCLK）驱动：

- 闪存编程接口时钟（FLITFCLK）固定 HSI 时钟驱动。
- 选项字节装载器时钟固定 HSI 时钟驱动。
- ADC 时钟由下列之一时钟得到（由软件选择）：
  - HSI 时钟，运行在最大的采样率。
  - APB (PCLK）时钟除 2 或除 4。
- UART1 的时钟为下列的时钟源之一（由软件选择）：

- 系统时钟 2 分频
- HSI 时钟
- APB 时钟（PCLK）

- I2C1 的时钟为下列的时钟源之一（由软件选择）:
  - 系统时钟
  - HSI 时钟
- IWDG 时钟固定为 LSI 时钟。

Cortex-M0 系统定时器（SysTick）的外部参考时钟由 AHB 时钟 8 分频后提供。SysTick 既可由 AHB/8 的时钟驱动，也可以直接由内核时钟（AHB clock）驱动，这取决于内核中 SysTick 控制状态寄存器（SysTick_CTRL）的配置。

5.2.1 时钟框图

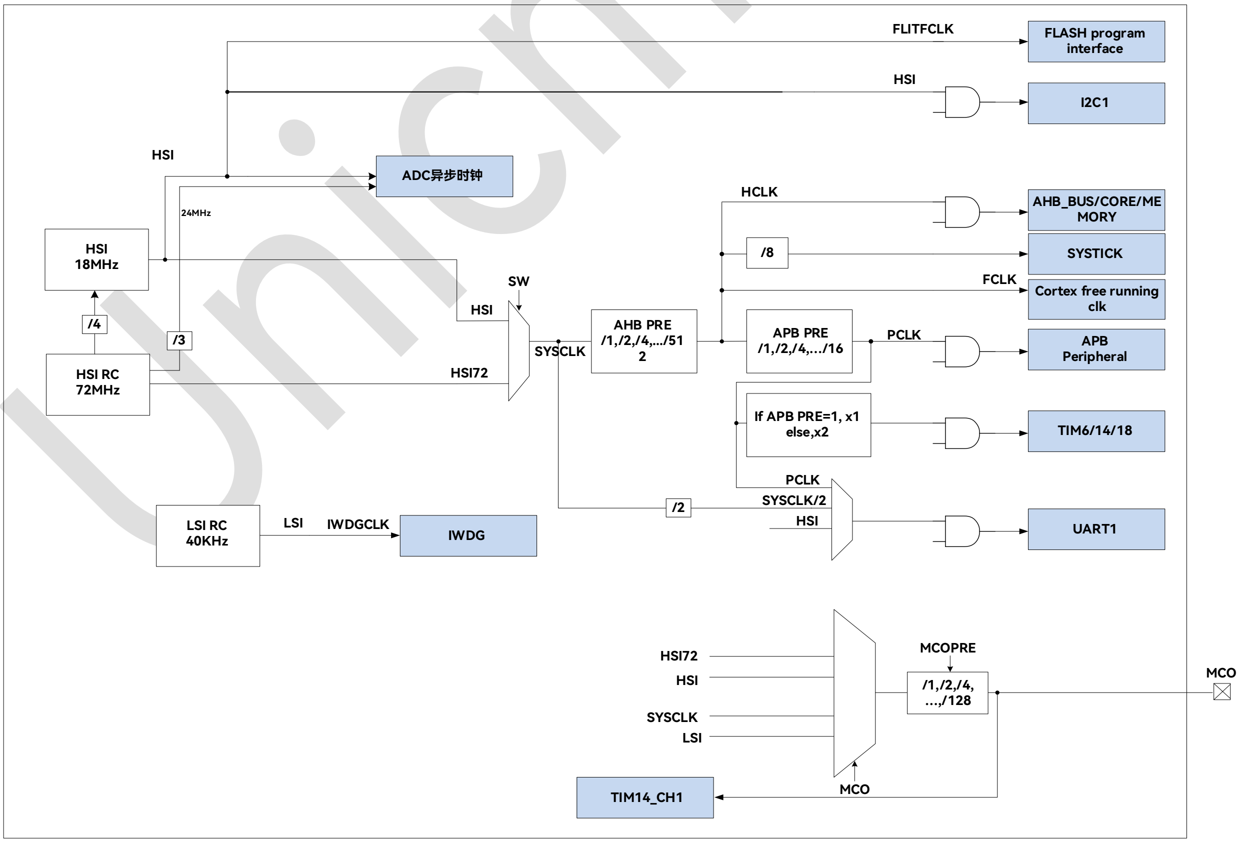


图 5-2：系统时钟框图

5.2.2 HSI 时钟

HSI 时钟信号，可直接作为系统时钟。

HSI 振荡器提供低成本时钟源（不需要外部器件）。支持出厂校准。

制造工艺决定了不同的芯片的 RC 振荡器频率有离散性，因此每颗芯片出厂都会经过精度校准（1%@25℃）。

复位后，出厂校准值会被加载至时钟控制寄存器（RCC_CR）中的 HSICAL[7:0]位。

实际应用中，不同的工作电压和环境温度，会导致 RC 振荡器频率漂移。用户可以通过修改时钟控制寄存器（RCC_CR）里的 HSITRIM[4:0]位来调整 HSI 频率。

关于 HSI 频率测量可以参见 TIM14 章节。

时钟控制寄存器（RCC_CR）中的 HSIRDY 位用来指示 HSI RC 振荡器是否稳定。在时钟启动过程中，直到这一位被硬件置 1，HSI RC 输出时钟才可以被使用。

HSI RC 可由时钟控制寄存器（RCC_CR）中的 HSION 位来启动和关闭。

此外，HSI 时钟可以通过 MCO 多路复用器输出。HSI 时钟也可以供给至 TIM14，用户进行振荡器校准和测量。

5.2.3 HSI72M 时钟

HSI72M 时钟信号，可直接作为系统时钟。

HSI72M 振荡器支持出厂校准。

时钟控制寄存器（RCC_CR）中的 HSI72MRDY 位用来指示 HSI72M RC 振荡器是否稳定。在时钟启动过程中，直到这一位被硬件置 1，HSI72M RC 输出时钟才可以被使用。

HSI72M RC 可由时钟控制寄存器（RCC_CR）中的 HSI72MON 位来启动和关闭。

此外，HSI72M 时钟可以通过 MCO 多路复用器输出。

复位后，出厂校准值会被加载至时钟控制寄存器（RCC_CR）中的 HSI72MCAL[8:0]位。

实际应用中，不同的工作电压和环境温度，会导致 RC 振荡器频率漂移。用户可以通过修改时钟控制寄存器（RCC_CR）里的 HSI72MTRIM[4:0]位来调整 HSI72M 的频率。

5.2.4 LSI 时钟

LSI 为内部低功耗时钟源，可以运行在停机和待机模式下，为 IWDG 模块提供时钟。

LSI 时钟频率约 40kHz，具体详情请参见数据手册。

LSI RC 可由时钟控制/状态寄存器（RCC_CSR）中的 LSION 位来启动和关闭。

时钟控制/状态寄存器（RCC_CSR）中的 LSIRDY 位指示 LSI 振荡器是否稳定。在该位被硬件置为 1 之前，LSI 的时钟信号将不会释放。如果使能时钟中断寄存器（RCC_CIR）里相应位，会产生一个中断。

5.2.5 系统时钟（SYSCLK）

HSI 和 HSI72M 时钟源可作为系统时钟（SYSCLK）。系统复位后，默认 HSI 振荡器为系统时钟。

当时钟源被直接作为系统时钟时，将不能被停止。

时钟的切换只有在目标时钟源可用的情况下才能进行（即时钟经过启动延时已稳定）。假如系统时钟选择了未准备好的时钟源作为当前系统时钟，那么只有在目标时钟源准备好之后才真正执行切换时钟源的操作。时钟控制寄存器（RCC_CR)中包含各个时钟的 ready（是否就绪）标志位及当前系统时钟指示位。

5.2.6 ADC 时钟

设置 ADC_CFGR2 寄存器（ADC 配置寄存器 2），可以选择不同的 ADC 时钟。ADC 时钟包括：HSI RC 振荡器（异步时钟输入）、24M RC 振荡器（异步时钟输入）和 PCLK/2（或/4）。当 APB 时钟被选为 ADC 模块时钟时，HSI RC 振荡器不能被 ADC 接口控制打开。24M RC 振荡器的使能和禁止由硬件控制，无需软件参与。

5.2.7 独立看门狗 IWDG 时钟

一旦独立看门狗（IWDG）被硬件或软件启动，LSI 振荡器将强制在开启状态，且不能被关闭。

LSI 振荡器稳定后，时钟供应给 IWDG。

5.2.8 时钟输出（MCO）

本芯片支持输出时钟信号到外部 MCO 引脚。对应 MCO 的 GPIO 须配置对应复用功能。

以下时钟信号之一可选为 MCO 时钟输出:

- HSI72M
- SYSCLK
- HSI
- LSI

MCO 时钟的选择由时钟配置寄存器 RCC_CFGR 的 MCO[3:0]位决定。

时钟配置寄存器 RCC_CFGR 的 MCOPRE[2:0]位，控制输入到 MCO 的时钟经过二进制除法器后输出，可配置/1/2/4.../128 除频因子。

5.2.9 内外时钟测量（TIM14）

通过 TIM14 通道 1 输入捕获，可以间接测量所有板载时钟源的频率，如下图所示:

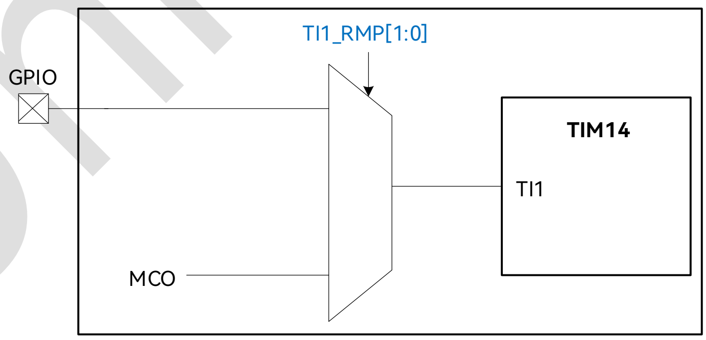


图 5-3：TIM14 测频电路（捕获模式）

通过配置 TIM14_OR 寄存器中的 TI1_RMP[1:0]位，可以选择不同的时钟源（GPIO 或内部时钟）接入 TIM14 的输入捕获通道。可选时钟如下:

- GPIO
- MCO

5.2.9.1 LSI 校准

LSI 校准与 HSI 方式相近，但需要调整参考时钟。将 LSI 时钟连接至 TIM14 的输入捕获通道，设置 HSI 作为系统时钟源。对 LSI 边沿捕获（HSI 时钟计数）进而测量 LSI 频率。

基本原理就是提供一种相对测量方法（例如，HSI/LSI 比率），比率越高测量效果越好。

5.3 低功耗模式

APB 外设时钟可以软件关闭。

睡眠模式（SLEEP mode），CPU 时钟停止，存储器接口时钟（FLASH 和 RAM 接口）可以软件停止。当连接到 APB 所有外设的时钟关闭后，进入睡眠模式期间 AHB 到 APB 桥时钟被硬件关闭。

停机模式（STOP mode），VREG1P5 电压域绝大多数时钟关闭，HSI 振荡器的时钟被关闭。

若 HSI 被选为 UART1 和 I2C1 时钟，即使在 MCU 进入停止模式，UART1 和 I2C1 仍可以打开 HSI 振荡器。当系统处于 STOP 模式且电压调节器工作在 LP 模式（低功耗模式，LDO_normal 关闭），来自这两个外设中任一时钟请求，电压调节器将切换至正常模式（MR mode，LDO_normal 开启），以便为核心逻辑提供适当的驱动电流。一旦请求被撤销，调节器回到 LP 模式，且不会唤醒 MCU。

设置 DBGMCU_CR 寄存器中的 DBG_STOP 位，CPU 的深度睡眠模式被覆盖并具有调试功能。

当 MCU 从 STOP 模式中被唤醒时，HSI 振荡器默认为系统时钟。

如果正在进行闪存编程，须等到对存储器的访问完成，系统才进入深度睡眠模式（深度睡眠延后）。如果正在进行对 APB 的访问，须等到对 APB 访问完成，系统才进入深度睡眠模式。

5.4 相关寄存器

5.4.1 时钟控制寄存器（RCC_CR）

偏移地址：0x000

复位值：0xxx40 0003，X 为未定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HSI72CAL[7:0]								Res.	HSI72TRIM[[4:0]					HSI72RDY	HSI72ON
r									rw					r	rw

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

位	名称	描述
31:24	HSI72CAL[7:0]	HSI72 时钟校准 在启动时，这些位被自动初始化为出厂校准值。
23	RSV	保留位，始终读为 0
22:18	HSI72TRIM[[4:0]	HSI72 时钟调整位 在 HSI72CAL[7:0]基础上，这些位提供了附加的用户可编程调校值，加到 HSI72CAL[7:0]位上，进而微调电压和温度带来的 HSI72 频率漂移。 默认值是 16，加到 HSI72CAL 值后，HSI72 为 72MHz ±1%。
17	HSI72RDY	HSI72 时钟就绪标志，硬件置 1，表明 HSI72 时钟是否稳定 0：HSI72 未就绪 1：HSI72 已就绪
16	HSI72ON	HSI 使能，软件置位和清零 0：HSI72 关闭 1：HSI72 使能 注：当 HSI72 作为系统时钟时，软件无法关闭。
15:2	RSV	保留位，始终读为 0
1	HSIRDY	HSI 时钟就绪标志，硬件置 1，表明 HSI 时钟是否稳定 0：HSI 未就绪 1：HSI 已就绪
0	HSION	HSI 使能，软件置位和清零 0：HSI 关闭 1：HSI 使能 注：当 HSI 作为系统时钟时，软件无法关闭。

5.4.2 时钟配置寄存器（RCC_CFGR）

偏移地址：0x004

复位值：0x0000 0000

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

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

位	名称	描述
31	RSV	保留位，始终读为 0
30:28	MCOPRE[2:0]	MCO 输出预分频，软件置位和清零 为避免产生毛刺，强烈建议在关闭 MCO 时修改此预分频器。 000：MCO/1 001：MCO/2 010：MCO/4 … 111：MCO/128
27:24	MCO[3:0]	MCO 输出时钟源选择，软件置位和清零 0000：MCO 输出关闭，MCO 引脚无时钟输出 0001：选择 HSI72（72MHz） 0010：选择 LSI 0011：保留位 0100：选择系统时钟（SYSCLK） 0101：选择 HSI（18MHz） 其他：保留
23:11	RSV	保留位，始终读为 0
10:8	PPRE[2:0]	PCLK 预分频，软件置位和清零 0xx：HCLK 不分频 100：HCLK/2 101：HCLK/4 110：HCLK/8 111：HCLK/16
7:4	HPRE[3:0]	HCLK 预分频，软件置位和清零 0xxx：SYSCLK 不分频 1000：SYSCLK/2 1001：SYSCLK/4 1010：SYSCLK/8 1011：SYSCLK/16

位	名称	描述
		1100：SYSCLK/64 1101：SYSCLK/128 1110：SYSCLK/256 1111：SYSCLK/512
3:2	SWS[1:0]	系统时钟切换状态位，硬件置位或清零来指示系统时钟源。 00：HSI 作为系统时钟 01：保留 10：保留 11：HSI72 作为系统时钟
1:0	SW[1:0]	系统时钟切换，软件置位和清零 00：HSI 作为系统时钟 01：保留 10：保留 11：HSI72 作为系统时钟 注：如果软件试图写入保留值，将无法写入成功，寄存器会保持当前值。

5.4.3 时钟中断使能寄存器（RCC_CIR）

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.												HSI72MRDYC	HSIRDYC	Res.	LSIRDYC
												rc_w1	rc_w1		rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				HSI72RDYIE	HSIRDYIE	Res.	LSIRDYIE	Res.				HSI72RDYF	HSIRDYF	Res.	LSIRDYF
				rw	rw		rw					r	r		r

位	名称	描述
31:20	RSV	保留位，始终读为 0
19	HSI72MRDYC	HSI72MRDYC 中断清除，软件置位 0：无效

位	名称	描述
		1：清 HSI72MRDYF 标志
18	HSIRDYC	HSIRDY 中断清除，软件置位 0：无效 1：清 HSIRDYF 标志
17	RSV	保留位，始终读为 0
16	LSIRDYC	LSIRDY 中断清除，软件置位 0：无效 1：清 LSIRDYF 标志
15:12	RSV	保留位，始终读为 0
11	HSI72RDYIE	HSI72RDY 中断使能，软件置位和清零 0：中断关闭 1：中断使能
10	HSIRDYIE	HSIRDY 中断使能，软件置位和清零 0：中断关闭 1：中断使能
9	RSV	保留位，始终读为 0
8	LSIRDYIE	LSIRDY 中断使能，软件置位和清零 0：中断关闭 1：中断使能
7:4	RSV	保留位，始终读为 0
3	HSI72RDYF	HSI72RDY 中断标志，软件对 HSI72RDYC 写 1 清零该位 当 HSI72 振荡器时钟稳定，HSI72RDYIE=1 且 HSI72ON =1 (RCC_CR)，硬件置位该位； 0：未发生中断 1：发生中断
2	HSIRDYF	HSIRDY 中断标志，软件对 HSIRDYC 写 1 清零该位 当 HSI 振荡器时钟稳定，HSIRDYDIE=1 且 HSION =1 (RCC_CFGR2)，硬件置位该位；当 HSION=0，但 HSI 被其他外设触发启动，此位不会置位不会产生中断 0：未发生中断 1：发生中断
1	RSV	保留位，始终读为 0
0	LSIRDYF	LSIRDY 中断标志，软件对 LSIRDYC 写 1 清零该位 0：未发生中断 1：发生中断

5.4.4 APB 外设复位寄存器 2（RCC_APB2RSTR）

偏移地址：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.									DBGMCU_RST	Res.		TIM18RST	Res.		
									rw			rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART1 RST	Res.	SPI1 RST	Res.	MCP RST	ADC RST	CMP RST	OPAMP RST	Res.						SYSCFG RST
	rw		rw		rw	rw	rw	rw							rw

位	名称	描述
31:23	RSV	保留位，始终读为 0
22	DBGMCU_RST	调试 MCU 复位，软件置位和清零 0：无效 1：复位 debug MCU
21:20	RSV	保留位，始终读为 0
19	TIM18RST	TIM18 复位，软件置位和清零 0：无效 1：复位 TIM18
18:15	RSV	保留位，始终读为 0
14	USART1RST	USART1 复位，软件置位和清零 0：无效 1：复位 USART1
13	RSV	保留位，始终读为 0
12	SPI1RST	SPI1 复位，软件置位和清零 0：无效 1：复位 SPI1
11	RSV	保留位，始终读为 0
10	MCPRST	MCP 复位，软件置位和清零 0：无效 1：复位 MCP
9	ADCRST	ADC 接口复位，软件置位和清零 0：无效

位	名称	描述
		1：复位 ADC
8	CMPRST	CMP 复位，软件置位和清零 0：无效 1：复位 CMP
7	OPAMPRST	OPAMP 复位，软件置位和清零 0：无效 1：复位 OPAMP
6:1	RSV	保留位，始终读为 0
0	SYSCFGRST	SYSCFG 复位，软件置位和清零 0：无效 1：复位 SYSCFG

5.4.5 APB 外设复位寄存器 1（RCC_APB1RSTR）

偏移地址：0x010

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							TIM14RST	Res.			TIM6RST	Res.			
							rw				rw				

位	名称	描述
31:29	RSV	保留位，始终读为 0
28	PWRRST	PWR 接口复位，软件置位和清零 0：无效 1：复位 PWR
27:22	RSV	保留位，始终读为 0
21	I2C1RST	I2C1 复位，软件置位和清零 0：无效 1：复位 I2C1
20:9	RSV	保留位，始终读为 0
8	TIM14RST	TIM14 复位，软件置位和清零 0：无效

位	名称	描述
		1：复位 TIM14
7:5	RSV	保留位，始终读为 0
4	TIM6RST	TIM6 复位，软件置位和清零 0：无效 1：复位 TIM6
3:0	RSV	保留位，始终读为 0

5.4.6 AHB 外设时钟使能寄存器（RCC_AHBENR）

偏移地址：0x014

复位值：0x0000 0014

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

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

位	名称	描述
31:19	RSV	保留位，始终读为 0
18	IOPBEN	GPIOB 时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
17	IOPAEN	GPIOA 时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
16:8	RSV	保留位，始终读为 0
7	COPROCEN	COPROCEN 模块时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
6	CRCEN	CRC 时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启

位	名称	描述
5	RSV	保留位，始终读为 0
4	FLITFEN	FLITF（flash 编程接口）时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
3	RSV	保留位，始终读为 0
2	SRAMEN	SRAM 接口时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
1:0	RSV	保留位，始终读为 0

5.4.7 APB 外设时钟使能寄存器 2（RCC_APB2ENR）

偏移地址：0x018

复位值：0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="9">Res.</td><td>DBGMCUEN</td><td colspan="2">Res.</td><td>TIM18EN</td><td colspan="3">Res.</td></tr>
<tr><td colspan="9"></td><td>rw</td><td colspan="2"></td><td>rw</td><td colspan="3"></td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Res.</td><td>USART1EN</td><td>Res.</td><td>SPI1EN</td><td>Res.</td><td>MCPEN</td><td>ADCEN</td><td>CMPEN</td><td>OPAMPEN</td><td colspan="6">Res.</td><td>SYSCFGEN</td></tr>
<tr><td></td><td>rw</td><td></td><td>rw</td><td></td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td colspan="6"></td><td>rw</td></tr>
</table>

位	名称	描述
31:23	RSV	保留位，始终读为 0
22	DBGMCUEN	MCU 调试模块时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
21:20	RSV	保留位，始终读为 0
19	TIM18EN	TIM18 时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
18:15	RSV	保留位，始终读为 0
14	USART1EN	USART1 时钟使能，软件置位和清零 0：时钟关闭

位	名称	描述
		1：时钟开启
13	RSV	保留位，始终读为 0
12	SPI1EN	SPI1 时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
11	RSV	保留位，始终读为 0
10	MCPEN	MCP 时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
9	ADCEN	ADC 接口时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
8	CMPEN	CMP 时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
7	OPAMPEN	OPAMP 时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
6:1	RSV	保留位，始终读为 0
0	SYSCFGEN	SYSCFG 时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启

5.4.8 APB 外设时钟使能寄存器 1（RCC_APB1ENR）

偏移地址：0x01C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							TIM14EN	Res.			TIM6EN	Res.			
							rw				rw				

位	名称	描述
31:29	RSV	保留位，始终读为 0
28	PWREN	PWR 接口时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
27:22	RSV	保留位，始终读为 0
21	I2C1EN	I2C1 时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
20:9	RSV	保留位，始终读为 0
8	TIM14EN	TIM14 时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
7:5	RSV	保留位，始终读为 0
4	TIM6EN	TIM6 时钟使能，软件置位和清零 0：时钟关闭 1：时钟开启
3:0	RSV	保留位，始终读为 0

5.4.9 RESET 状态寄存器（RCC_CSR）

偏移地址：0x024

复位值：0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>LPWRRSTF</td><td>Res.</td><td>IWDGRSTF</td><td>SFTRSTF</td><td>PORRSTF</td><td>PINRSTF</td><td>OBLRSTF</td><td>RMVF</td><td>V15PWRRSTF</td><td colspan="7">Res.</td></tr>
<tr><td>r</td><td></td><td>r</td><td>r</td><td>r</td><td>r</td><td>r</td><td>rc_w1</td><td>r</td><td colspan="7"></td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">Res.</td><td>LSIRDY</td><td>LSION</td></tr>
<tr><td colspan="14"></td><td>r</td><td>rw</td></tr>
</table>

位	名称	描述
31	LPWRRSTF	低功耗复位标志 低功耗管理复位发生时由硬件置 1，软件通过写 RMVF 位清除该位。

位	名称	描述
		0：无低功耗管理复位发生 1：发生低功耗管理复位
30	RSV	保留位，始终读为 0
29	IWDGRSTF	独立看门狗复位标志 独立看门狗复位发生时由硬件置 1，软件通过写 RMVF 位清除该位。 0：无独立看门狗复位发生 1：发生独立看门狗复位
28	SFTRSTF	软件复位标志 软件复位发生时由硬件置 1，软件通过写 RMVF 位清除该位。 0：无软件复位发生 1：发生软件复位
27	PORRSTF	上电/掉电复位标志 上电/掉电复位发生时由硬件置 1，软件通过写 RMVF 位清除该位。 0：无上电/掉电复位发生 1：发生上电/掉电复位
26	PINRSTF	NRST 引脚复位标志 NRST 引脚复位发生时由硬件置 1，软件通过写 RMVF 位清除该位。 0：无 NRST 引脚复位发生 1：发生 NRST 引脚复位
25	OBLRSTF	选项字节加载复位标志 选项字节加载复位发生时由硬件置 1，软件通过写 RMVF 位清除该位。 0：无选项字节加载复位发生 1：发生选项字节加载复位
24	RMVF	清除复位标志 软件写 1 来清除所有复位标志，包括 RMVF。 0：无效 1：清除复位标志
23	V15PWRRSTF	VREG1P5 电源域复位标志 VREG1P5 域上电/掉电复位发生时由硬件置 1，软件通过写 RMVF 位清除该位。 0：无 VREG1P5 域上电/掉电复位发生 1：发生 VREG1P5 域上电/掉电复位
22:2	RSV	保留位，始终读为 0
1	LSIRDY	LSI 振荡器就绪，硬件置位和清零，指示 LSI 振荡器是否稳定。 当 LSION 清零后，该位需要 3 个 LSI clk 才会清零。 0：LSI 未就绪 1：LSI 已就绪

位	名称	描述
0	LSION	LSI 使能，软件置位和清零 0：LSI 关闭 1：LSI 使能

5.4.10 AHB 外设复位寄存器（RCC_AHBRSTR）

偏移地址：0x028

复位值：0x0000 0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="13">Res.</td><td>IOPBRST</td><td>IOPARST</td><td>Res.</td></tr>
<tr><td colspan="13"></td><td>rw</td><td>rw</td><td></td></tr>
</table>

<table>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">Res.</td><td>COPROCRST</td><td colspan="7">Res.</td></tr>
<tr><td colspan="8"></td><td>rw</td><td colspan="7"></td></tr>
</table>

位	名称	描述
31:19	RSV	保留位，始终读为 0
18	IOPBRST	GPIOB 复位，软件置位和清零 0：无效 1：复位 GPIOB
17	IOPARST	GPIOA 复位，软件置位和清零 0：无效 1：复位 GPIOA
16:8	RSV	保留位，始终读为 0
7	COPROCRST	COPROC 复位，软件置位和清零 0：无效 1：复位 COPROC
6:0	RSV	保留位，始终读为 0

5.4.11 时钟配置寄存器 3（RCC_CFGR3）

偏移地址：0x030

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														UART1SW[1:0]	
														rw	

位	名称	描述
31:2	RSV	保留位，固定读为 0
1:0	UART1SW[1:0]	USART1 时钟源选择，软件置位和清零 00：PCLK 时钟被选为 UART1 时钟源（默认） 01：系统时钟（SYSCLK）2 分频被选为 UART1 的时钟源 10：保留 11：HSI 时钟被选为 UART1 时钟源

6 通用 I/O（GPIO）

6.1 概述

每组 GPIO 都有 4 个 32 位配置寄存器 (GPIOx_MODER, GPIOx_OTYPER, GPIOx_ OSPEEDR 和 GPIOx_PUPDR) , 2 个 16 位数据寄存器 (GPIOx_IDR 和 GPIOx_ODR) 和 2 个原子位操作寄存器 (32 位 GPIOx_BSRR，16 位 GPIOx_BRR)，还包含 2 个 32 位复用功能寄存器（GPIOx_AFRH 和 GPIOx_AFRL）。

6.2 特性

- 输出：支持推挽输出和开漏输出（带上拉/下拉控制）
- 输入：支持浮空、上拉/下拉、模拟输入
- 每个 IO 速度可选
- 支持位操作（置位/复位寄存器 GPIOx_BSRR，复位寄存器 GPIOx_BRR）
- 支持复用功能选择和模拟功能
- 支持 GPIO 快速翻转（2 clocks）

6.3 功能描述

每个 GPIO 口支持以下多种模式：

- 浮空输入
- 上拉输入
- 下拉输入
- 模拟输入
- 具有上拉或下拉能力的开漏输出

- 具有上拉或下拉能力的推挽输出
- 复用功能且具有上拉或下拉能力的推挽输出
- 复用功能且具有上拉或下拉能力的开漏输出

GPIOx_ BSRR 和 GPIOx_BRR 寄存器作用是实现对 GPIOx_ODR 寄存器读-修改-写的原子操作，这样可以避免在读和修改操作之间因为发生中断而带来的风险。

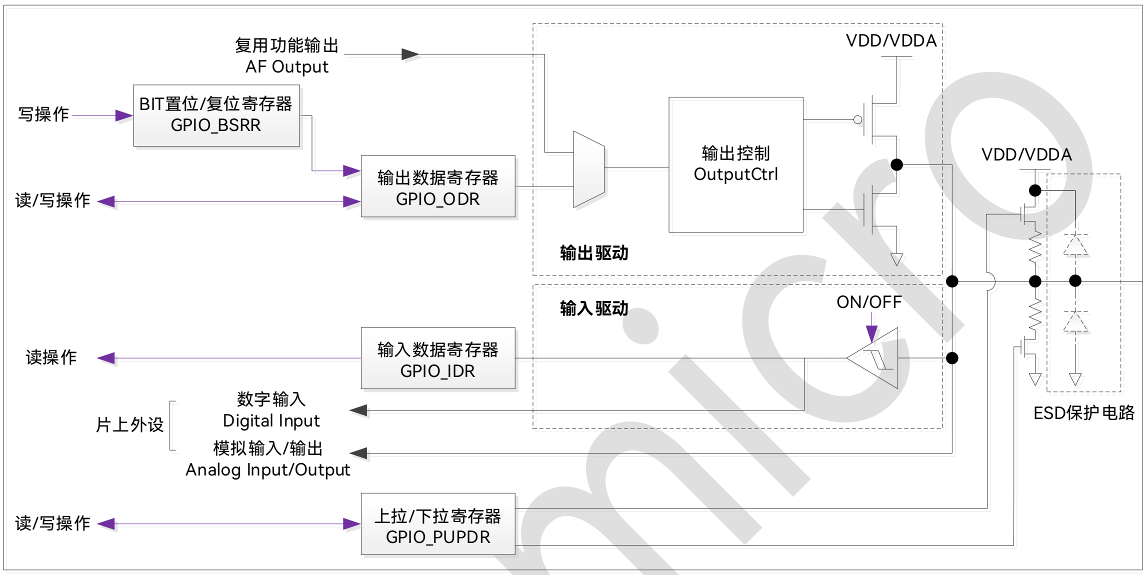


图 6-1：标准 I/O 端口基本结构框图

表 6-1：端口位配置汇总表

MODER[1:0] 模式控制	OTYPER 开漏控制	OSPEEDR[1:0] IO 速度	PUPDR[1:0] 上拉/下拉		I/O 配置	
01	0	OSPEEDR[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	OSPEEDR[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

MODER[1:0] 模式控制	OTYPER 开漏控制	OSPEEDR[1:0] IO 速度		PUPDR[1:0] 上拉/下拉		I/O 配置	
	1			0	1	AF	OD + PU
	1			1	0	AF	OD + PD
	1			1	1	保留	
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		

注：GP-General Purpose，PP-Push Pull，PU-Pull Up，PD-Pull Down，OD-Open Drain，AF-Alternate Function

6.3.1 通用 IO（GPIO）

复位期间和刚复位后，复用功能不开启，大多数 I/O 端口被配置为输入浮空模式。

复位后，调试引脚被置为复用功能+上拉/下拉模式：

- PA13：SWCLK +下拉模式
- PA14：SWDIO +上拉模式

当配置作为输出时，写到输出数据寄存器 GPIOx_ODR 的值输出到相应的 I/O 引脚上。支持推挽模式或开漏模式（仅输出低被驱动，输出高为高阻态）。

输入数据寄存器 GPIOx_IDR 在每个 AHB 时钟周期捕捉 I/O 引脚上的数据。

所有 GPIO 引脚都有一个内部弱上拉和弱下拉电阻，通过 GPIOx_PUPDR 寄存器可以配置为开启或关闭。

6.3.2 I/O 复用功能映射

内置外设/模块功能通过多路复用器连接 I/O 口。同一时刻仅允许一个外设的复用功能（AF）连

接到一个 I/O 口上，以防止同一 I/O 口上可用的外设功能冲突。

每个 I/O 引脚支持最多 16 个复用功能输入（从 AF0 到 AF15）的多路复用器，其可通过配置 GPIOx_AFRL 寄存器（pin 0~7）和 GPIOx_AFRH 寄存器（pin 8~15）来实现。

- 复位后，复用器选择为 AF0 功能，通过 GPIOx_MODER 寄存器设置为复用功能（AF）模式。
- 有关每个引脚的具体复用功能分配，参见芯片数据手册。

除了这种灵活的 I/O 复用结构，部分外设的复用功能还可以映射到不同的 I/O 引脚上，用于小封装器件上优化更多的可用外设。

不同的 IO 使用，按如下原则配置：

- 调试功能引脚：器件复位后，这些引脚立即配置为复用功能，调试器可用。
- GPIO：在 GPIOx_MODER 寄存器中配置所需的 I/O 口为输出、输入或模拟功能。
- 外设的复用功能：
  - 设置连接至 I/O 所需的 AFx ，AFx 定义在 GPIOx_AFRL 或 GPIOx_AFRH 寄存器中。
  - 配置 GPIOx_OTYPER、GPIOx_PUPDR 和 GPIOx_OSPEEDER 寄存器，来选择相应引脚的开漏、上拉/下拉和输出速度。
  - 在 GPIOx_MODER 寄存器中配置所需的 I/O 口为复用功能模式。
- 附加功能：
  - 对于 ADC，无论配置 GPIO 模式如何，ADC 连接都可以在 ADC 寄存器启用。建议使用 ADC 时，在 GPIOx_MODER 寄存器中配置所需的 I/O 口为模拟模式。

有关详细的 I/O 口复用功能映射，请参见芯片数据手册。

6.3.3 I/O 端口控制寄存器

每组 GPIO 都有 4 个 32 位配置寄存器（GPIOx_MODER, GPIOx_OTYPER, GPIOx_ OSPEEDR 和 GPIOx_PUPDR）。GPIOx_MODER 用来配置 IO 模式（输入、输出、复用或模拟），GPIOx_OTYPER 用来选择输出类型（开漏或推挽），GPIOx_ OSPEEDR 用来控制输出速度，GPIOx_PUPDR 用来选择上/下拉方式。

6.3.4 I/O 端口数据寄存器

每组 GPIO 都有 2 个 16 位数据寄存器（GPIOx_IDR 和 GPIOx_ODR）。GPIOx_IDR 用于存储输入数据，该寄存器为只读寄存器；GPIOx_ODR 用于存储输出数据，可进行读/写访问。

6.3.5 I/O 数据位处理

每组GPIO都有1个32位的置位/复位寄存器GPIOx_BSRR和1个16位复位寄存器GPIOx_BRR。

GPIOx_BSRR 用于对输出数据寄存器 GPIOx_ODR 每个位进行置位/复位操作，其有效位宽为 GPIOx_ODR 的两倍。

对于 GPIOx_ODR 中的每一位，在 GPIOx_BSRR 中有 2 位与之对应：BS(i)和 BR(i)。对 BS(i)写 1，置位对应的 ODR(i)位；对 BR(i)写 1，复位对应的 ODR(i)位。

对 GPIOx_BSRR 的任意位写 0 无效，不影响 GPIOx_ODR 的值。若对 BS(i)和 BR(i)同时写 1，置位优先（即执行对应位的置位操作）。

GPIOx_BRR，仅用于对输出数据寄存器 GPIOx_ODR 每个位进行复位操作，功能同 GPIOx_BSRR 的 BR(i)。

通过 GPIOx_BSRR 寄存器或 GPIOx_BRR 寄存器对 GPIOx_ODR 的各个位操作，为“单次”行为，不会锁存 GPIOx_ODR 对应位。GPIOx_ODR 的位仍可以被直接访问。GPIOx_BSRR 寄存器和 GPIOx_BRR 寄存器提供一种执行原子位操作方法。

通过 GPIOx_BSRR 或 GPIOx_BRR 寄存器对 GPIOx_ODR 的位操作，不需要关闭中断，因为单 AHB 周期写访问可以实现一个或多个位修改。

6.3.6 I/O 复用功能输入/输出

每组 GPIO 有 2 个 32 位复用功能寄存器（GPIOx_AFRH 和 GPIOx_AFRL）。

用户通过设置这两个寄存器，根据应用需求选择每个 GPIO 的复用功能。

详细复用功能映射，请参见产品数据手册。

6.3.7 外部中断/唤醒线

所有 I/O 口都可以配置为外部中断口，若要 I/O 用作外部中断 Line，对应的端口不能配置为模拟模式，这样输入触发器就会保持开启状态。

6.3.8 输入配置

当 I/O 配置为输入模式：

- 输出缓冲器关闭
- 输入施密特触发器激活
- 由 GPIOx_PUPDR 寄存器激活上拉/下拉电阻控制
- 每个 AHB 时钟周期，I/O 引脚上的数据被采样进入输入数据寄存器
- 通过读取输入数据寄存器获取 I/O 状态

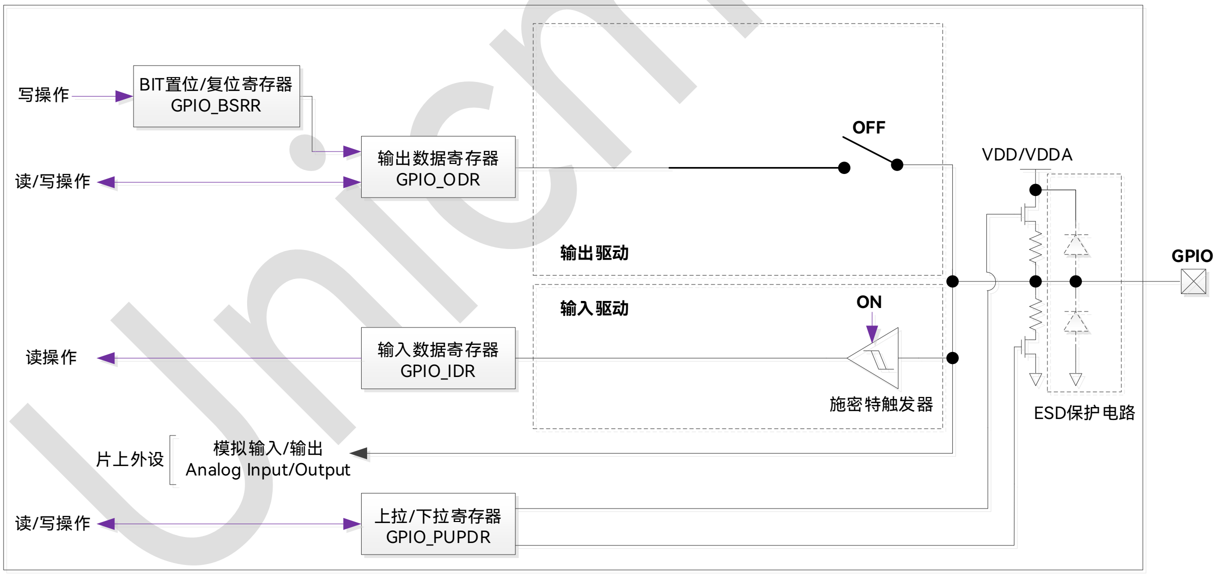


图 6-2：输入浮空/上拉/下拉配置

6.3.9 输出配置

当 I/O 配置为输出模式：

- 输出缓冲器开启

- 开漏模式：输出数据寄存器写 0 激活输出 N-MOS（输出低），写 1 端口处于高阻态（P-MOS 不被激活）。
- 推挽模式：输出数据寄存器写 0 激活输出 N-MOS（输出低），写 1 激活输出 P-MOS（输出高）。

- 输入施密特触发器激活
- 由 GPIOx_PUPDR 寄存器激活上拉/下拉电阻控制
- 每个 AHB 时钟周期，I/O 引脚上的数据被采样进入输入数据寄存器
- 通过读取输入数据寄存器获取 I/O 状态
- 通过读取输出数据寄存器获取最后写入该寄存器的值

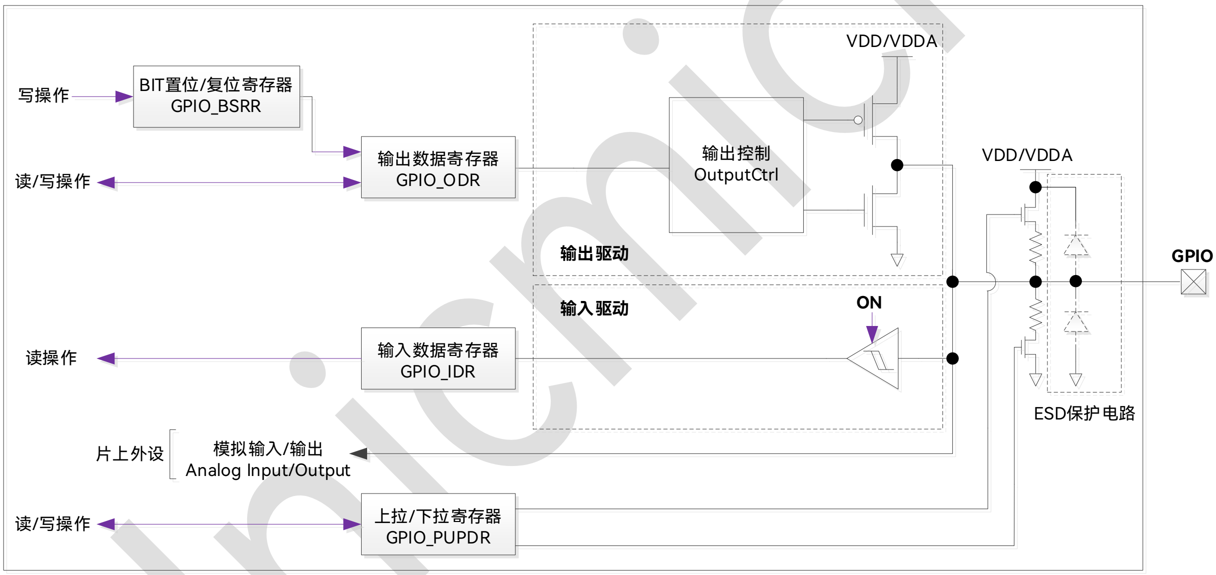


图 6-3：输出配置

6.3.10 复用功能配置

当 I/O 配置为复用功能模式：

- 输出缓冲器可被配置为开漏或推挽
- 外设信号（发送使能或数据）控制输出缓冲器
- 输入施密特触发器激活
- 由 GPIOx_PUPDR 寄存器激活上拉/下拉电阻控制

- 每个 AHB 时钟周期，I/O 引脚上的数据被采样进入输入数据寄存器
- 通过读取输入数据寄存器获取 I/O 状态

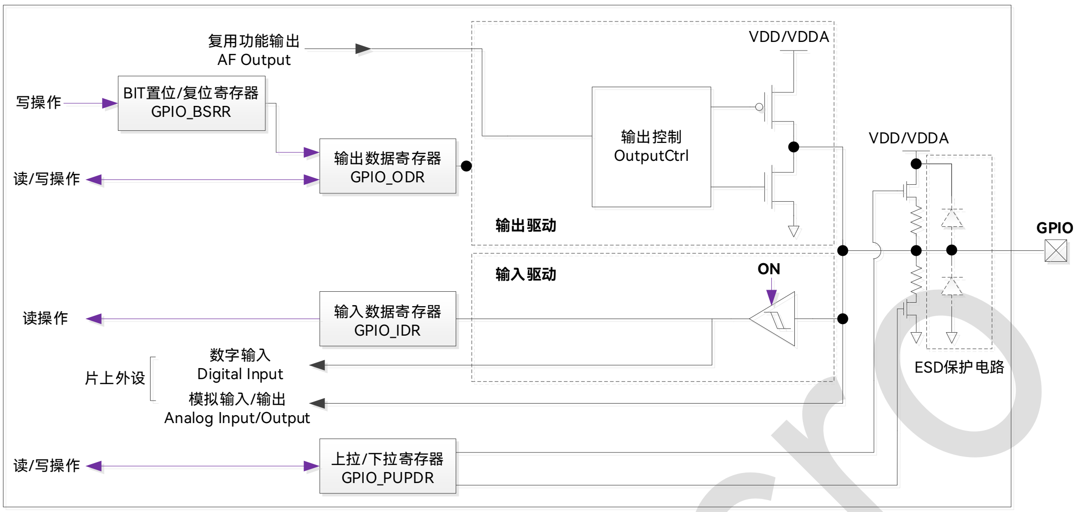


图 6-4：复用功能配置

6.3.11 模拟功能配置

当 I/O 配置为模拟功能模式：

- 输出缓冲器关闭
- 输入施密特触发器禁止输入，减少模拟端口功耗，施密特输出（数字输入）固定为 0
- 弱上拉/下拉电阻硬件关闭
- 每个 AHB 时钟周期，I/O 引脚上的数据被采样进入输入数据寄存器
- 读取输入数据寄存器固定为 0

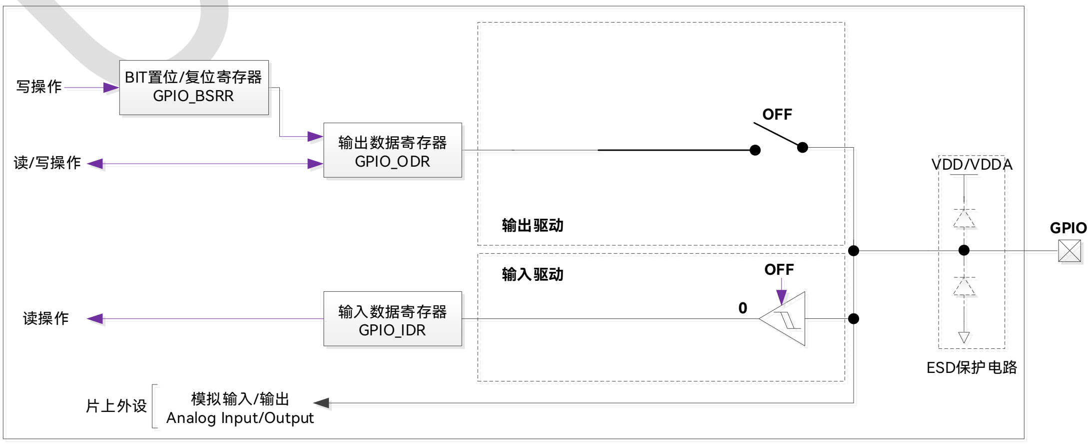


图 6-5：高阻/模拟配置

6.4 相关寄存器

6.4.1 端口模式寄存器（GPIO_MODER）

偏移地址：0x000

复位值：0x2800 0000（GPIOA）

0x0000 0000（GPIOB）

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">MODER15[1:0]</td><td colspan="2">MODER14[1:0]</td><td colspan="2">MODER13[1:0]</td><td colspan="2">MODER12[1:0]</td><td colspan="2">MODER11[1:0]</td><td colspan="2">MODER10[1:0]</td><td colspan="2">MODER9[1:0]</td><td colspan="2">MODER8[1:0]</td></tr>
<tr><td colspan="2">rw</td><td colspan="2">rw</td><td colspan="2">rw</td><td colspan="2">rw</td><td colspan="2">rw</td><td colspan="2">rw</td><td colspan="2">rw</td><td colspan="2">rw</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">MODER7[1:0]</td><td colspan="2">MODER6[1:0]</td><td colspan="2">MODER5[1:0]</td><td colspan="2">MODER4[1:0]</td><td colspan="2">MODER3[1:0]</td><td colspan="2">MODER2[1:0]</td><td colspan="2">MODER1[1:0]</td><td colspan="2">MODER0[1:0]</td></tr>
<tr><td colspan="2">rw</td><td colspan="2">rw</td><td colspan="2">rw</td><td colspan="2">rw</td><td colspan="2">rw</td><td colspan="2">rw</td><td colspan="2">rw</td><td colspan="2">rw</td></tr>
</table>

位	名称	描述
31:0	MODERy[1:0]	GPIOx 的模式配置位（x=A,B），软件置位和清零 (y=0~15) 00：输入模式 01：通用输出模式 10：复用功能模式 11：模拟模式

6.4.2 端口输出类型寄存器（GPIO_OTYPER）

偏移地址：0x004

复位值：0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Res.</td></tr>
<tr><td colspan="16"></td></tr>
</table>

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	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	OTy	GPIOx 的输出类型配置位，软件置位和清零（y=0~15） 0：推挽输出（复位状态） 1：开漏输出

6.4.3 端口输出速度寄存器（GPIO_OSPEEDR）

偏移地址：0x008

复位值：0x0C00 0000（GPIOA）

0x0000 0000（GPIOB）

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
OSPEEDR15[1:0]	OSPEEDR14[1:0]	OSPEEDR13[1:0]	OSPEEDR12[1:0]	OSPEEDR11[1:0]	OSPEEDR10[1:0]	OSPEEDR9[1:0]	OSPEEDR8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

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

位	名称	描述
31:0	OSPEEDRy[1:0]	GPIOx 的输出速度配置位（x=A,B），软件置位和清零（y=0~15） x0：低速 01：中速 11：高速 注：每个速度档位对应频率、电源及负载条件，请参见产品数据手册。

6.4.4 端口上拉/下拉寄存器（GPIO_PUPDR）

偏移地址：0x00C

复位值：0x2400 0000（GPIOA）

0x0000 0000（GPIOB）

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
PUPDR15 [1:0]	PUPDR14 [1:0]	PUPDR13 [1:0]	PUPDR12 [1:0]	PUPDR11 [1:0]	PUPDR10 [1:0]	PUPDR9 [1:0]	PUPDR8 [1:0]
rw	rw	rw	rw	rw	rw	rw	rw

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

位	名称	描述
31:0	PUPDRy[1:0]	GPIOx 的上拉/下拉配置位（x=A,B），软件置位和清零（y=0~15） 00：无上拉和下拉 01：上拉 10：下拉 11：保留

6.4.5 端口输入数据寄存器（GPIO_IDR）

偏移地址：0x010

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	IDRy	端口输入数据位，只读（y=0~15） 相应 I/O 口的输入值。

6.4.6 端口输出数据寄存器（GPIO_ODR）

偏移地址：0x014

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	ODRy	GPIOx 输出数据，软件置位和清零（y=0~15） 注：对于原子置位/复位，可通过对 GPIOx_BSRR 和 GPIOx_BRR 寄存器操作来实现。

6.4.7 端口位置位/复位寄存器（GPIO_BSRR）

偏移地址：0x018

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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	名称	描述
31:16	BRy	GPIOx 复位位操作，只写，读这些位返回 0x0000（y=0~15） 0：对相应的 ODRx 位无影响 1：复位相应的 ODRx 位 注：若 BSx 和 BRx 同时置位，BSx（置位）有优先权。
15:0	BSy	GPIOx 置位位操作，只写，读这些位返回 0x0000（y=0~15） 0：对相应的 ODRx 位无影响 1：置位相应的 ODRx 位

6.4.8 复用功能低位寄存器（GPIO_AFRL）

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR7[3:0]				AFR6[3:0]				AFR5[3:0]				AFR4[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR3[3:0]				AFR2[3:0]				AFR1[3:0]				AFR0[3:0]			
rw				rw				rw				rw			

位	名称	描述
31:0	AFRy[3:0]	GPIOx pin y 的复用功能选择（y= 0~7），软件置位和清零 0000：AF0 0001：AF1 0010：AF2 0011：AF3 0100：AF4 0101：AF5 0110：AF6 0111：AF7 其他：保留

6.4.9 复用功能高位寄存器（GPIO_AFRH）

偏移地址：0x024

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR(8+y)[3:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR(8+y)[3:0]															
rw															

位	名称	描述
31:0	AFR(8+y)[3:0]	GPIOx pin (8+y)的复用功能选择（y=0~7），软件置位和清零 0000：AF0 0001：AF1 0010：AF2 0011：AF3 0100：AF4 0101：AF5 0110：AF6 0111：AF7 其他：保留

6.4.10 端口复位寄存器（GPIO_BRR）

偏移地址：0x028

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	BRy	GPIOx 复位操作，只写，读这些位返回 0x0000（y=0~7） 0: 对相应的 ODRx 位无影响 1: 复位相应的 ODRx 位 注：与 GPIOx_BSRR->BR[15:0]位相同。

7 系统配置控制（SYSCFG）

7.1 概述

系统配置寄存器主要功能如下：

- 重映射代码启动区的存储器。
- 管理连接到 GPIO 口的外部中断 Line。
- 管理系统的可靠性特性。

7.2 相关寄存器

7.2.1 配置寄存器 1（SYSCFG_CFGR1）

偏移地址：0x000

复位值：0x0000 0000

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

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

位	名称	描述
31:2	RSV	保留位，始终读为 0
1:0	MEM_MODE[1:0]	存储映射选择位，软件置位和清零 当复位后，此位值由实际 boot 模式配置值决定。 详细请参考启动配置章节。 x0：主闪存存储器映射到 0x0000 0000 01：系统存储器映射到 0x0000 0000 11：嵌入式 SRAM 映射到 0x0000 0000

7.2.2 外部中断配置寄存器 1（SYSCFG_EXTICR1）

偏移地址：0x008

复位值：0x0000 0000

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

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

位	名称	描述
31:13	RSV	保留位，始终读为 0
12	EXTI3	EXTI 3 配置位，软件置位和清零，选择 EXTI3 的外部中断源。 0: PA3 引脚 1: PB3 引脚
11:9	RSV	保留位，始终读为 0
8	EXTI2	EXTI2 配置位，软件置位和清零，选择 EXTI2 的外部中断源。 0: PA2 引脚 1: PB2 引脚
7:5	RSV	保留位，始终读为 0
4	EXTI1	EXTI1 配置位，软件置位和清零，选择 EXTI1 的外部中断源。 0: PA1 引脚 1: PB1 引脚
3:1	RSV	保留位，始终读为 0
0	EXTI0	EXTI0 配置位，软件置位和清零，选择 EXTI0 的外部中断源。 0: PA0 引脚 1: PB0 引脚

7.2.3 外部中断配置寄存器 2（SYSCFG_EXTICR2）

偏移地址：0x00C

复位值：0x0000 0000

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

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

位	名称	描述
31:13	RSV	保留位，始终读为 0
12	EXTI7	EXTI 7 配置位，软件置位和清零，选择 EXTI7 的外部中断源。 0: PA7 引脚 1: PB7 引脚
11:9	RSV	保留位，始终读为 0
8	EXTI6	EXTI6 配置位，软件置位和清零，选择 EXTI6 的外部中断源。 0: PA6 引脚 1: PB6 引脚
7:5	RSV	保留位，始终读为 0
4	EXTI5	EXTI5 配置位，软件置位和清零，选择 EXTI5 的外部中断源。 0: PA5 引脚 1: PB5 引脚
3:1	RSV	保留位，始终读为 0
0	EXTI4	EXTI4 配置位，软件置位和清零，选择 EXTI4 的外部中断源。 0: PA4 引脚 1: PB4 引脚

7.2.4 外部中断配置寄存器 3（SYSCFG_EXTICR3）

偏移地址：0x010

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			EXTI11	Res.			EXTI10	Res.			EXTI9	Res.			EXTI8
			rw				rw				rw				rw

<table>
<tr><th>位</th><th>名称</th><th>描述</th></tr>
<tr><td>31:13</td><td>RSV</td><td>保留位，始终读为 0</td></tr>
<tr><td>12</td><td>EXTI11</td><td>EXTI 11 配置位，软件置位和清零，选择 EXTI11 的外部中断源。
0: PA11 引脚
1: PB11 引脚
注：如果使用的产品型号中没有 PB11 引脚，即使软件对此位写 1 也没有相应的功能。</td></tr>
<tr><td>11:9</td><td>RSV</td><td>保留位，始终读为 0</td></tr>
<tr><td>8</td><td>EXTI10</td><td>EXTI 10 配置位，软件置位和清零，选择 EXTI10 的外部中断源。
0: PA10 引脚
1: PB10 引脚
注：如果使用的产品型号中没有 PB10 引脚，即使软件对此位写 1 也没有相应的功能。</td></tr>
<tr><td>7:5</td><td>RSV</td><td>保留位，始终读为 0</td></tr>
<tr><td>4</td><td>EXTI9</td><td>EXTI 9 配置位，软件置位和清零，选择 EXTI9 的外部中断源。
0: PA9 引脚
1: PB9 引脚
注：如果使用的产品型号中没有 PB9 引脚，即使软件对此位写 1 也没有相应的功能。</td></tr>
<tr><td>3:1</td><td>RSV</td><td>保留位，始终读为 0</td></tr>
<tr><td>0</td><td>EXTI8</td><td>EXTI 8 配置位，软件置位和清零，选择 EXTI8 的外部中断源。
0: PA8 引脚
1: PB8 引脚</td></tr>
</table>

7.2.5 外部中断配置寄存器 4（SYSCFG_EXTICR4）

偏移地址：0x014

复位值：0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Res.</td></tr>
<tr><td colspan="16"></td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">Res.</td><td>EXTI15</td><td colspan="3">Res.</td><td>EXTI14</td><td colspan="3">Res.</td><td>EXTI13</td><td colspan="3">Res.</td><td>EXTI12</td></tr>
<tr><td colspan="3"></td><td>rw</td><td colspan="3"></td><td>rw</td><td colspan="3"></td><td>rw</td><td colspan="3"></td><td>rw</td></tr>
</table>

位	名称	描述
31:13	RSV	保留位，始终读为 0
12	EXTI15	EXTI 15 配置位，软件置位和清零，选择 EXTI15 的外部中断源。 0: PA15 引脚 1: PB15 引脚 注：如果使用的产品型号中没有 PB15 引脚，即使软件对此位写 1 也没有相应的功能。
11:9	RSV	保留位，始终读为 0
8	EXTI14	EXTI 14 配置位，软件置位和清零，选择 EXTI14 的外部中断源。 0: PA14 引脚 1: PB14 引脚 注：如果使用的产品型号中没有 PB14 引脚，即使软件对此位写 1 也没有相应的功能。
7:5	RSV	保留位，始终读为 0
4	EXTI13	EXTI 13 配置位，软件置位和清零，选择 EXTI13 的外部中断源。 0: PA13 引脚 1: PB13 引脚 注：如果使用的产品型号中没有 PB13 引脚，即使软件对此位写 1 也没有相应的功能。
3:1	RSV	保留位，始终读为 0
0	EXTI12	EXTI 12 配置位，软件置位和清零，选择 EXTI12 的外部中断源。 0: PA12 引脚 1: PB12 引脚 注：如果使用的产品型号中没有 PB12 引脚，即使软件对此位写 1 也没有相应的功能。

7.2.6 配置寄存器 2（SYSCFG_CFGR2）

偏移地址：0x018

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							SRAM_PEF	Res.				RSTCFG	PVD_LOCK	SRAM_PARITY_LOCK	LOCKUP_LOCK
							rc_w1					rw	rs	rw	rw

位	名称	描述
31:9	RSV	保留位，始终读为 0
8	SRAM_PEF	SRAM 校验错误标志 当 SRAM 校验错误被检测到时，硬件置位。对该位写 1 清零。 0：无 SRAM 校验错误 1：检测到 SRAM 校验错误
7:4	RSV	保留位，始终读为 0
3	RSTCFG	复位引脚功能设置 0：NRST/PB0 引脚被当作复位引脚使用 1：NRST/PB0 引脚被当作普通 IO 使用
2	PVD_LOCK	PVD 锁定使能位，软件置位，系统复位清零 可用于使能并锁定 PVD 连接到 MCP 的刹车（Break）输入 ，锁定 PVDE 和 PLS[2:0]（PWR_CR 寄存器） 0: PVD 中断信号断开与 MCP 刹车（Break）输入的连接。PVDE 和 PLS[2:0]位可被应用编程。 1: PVD 中断信号连接到 MCP 刹车（Break）输入，PVDE 和 PLS[2:0]位为只读。
1	SRAM_PARITY_LOCK	SRAM 校验锁定位，软件置位，系统复位清零 可用于锁定 SRAM 校验错误信号连接到 MCP 的刹车（Break）输入。 0: SRAM 校验错信号断开与 MCP 刹车（Break）输入的连接 1: SRAM 校验错信号连接到 MCP 刹车（Break）输入
0	LOCKUP_LOCK	Cortex-M0 LOCKUP 位使能位，软件置位，系统复位清零 可用于使能并锁定连接 Cortex-M0 LOCKUP（硬件故障）输出到 MCP 的刹车（Break）输入。 0: Cortex-M0 LOCKUP 输出断开与 MCP 刹车（Break）输入的连接 1: Cortex-M0 LOCKUP 输出连接到 MCP 刹车（Break）输入

8　中断和事件（NVIC/Systick/EXTI）

8.1　嵌套向量中断控制（NVIC）

本产品内置嵌套向量中断控制器，能够处理多达 16 个可屏蔽中断通道（不包括 16 个 Cortex-M0 的中断线）和 4 个可编程优先级。

- 紧耦合的 NVIC 能够达到低延迟的中断响应处理
- 中断向量入口地址直接进入内核
- 紧耦合的 NVIC 接口
- 允许中断的早期处理
- 处理晚到的较高优先级中断
- 支持中断末尾连锁功能
- 自动保存处理器状态
- 中断返回时自动恢复现场，无需额外指令开销

该模块以最小的中断延迟提供灵活的中断管理功能。更多异常和 NVIC 编程的说明请参考编程手册。

表 8-1：中断异常向量表

位置	优先级	优先级类型	名称	说明	地址
-	-	-	-	保留（Reserved）	0x0000 0000
-	-3	固定	Reset	复位（Reset）	0x0000 0004
-	-2	固定	NMI	不可屏蔽中断。	0x0000 0008
-	-1	固定	HardFault	所有类型的错误（Fault）	0x0000 000C
-	-	可设置	-	保留	0x0000 0010 ~0x0000 002B
-	3	可设置	SVCall	通过 SWI 指令调用的系统服务	0x0000 002C
-	-	可设置	-	保留	0x0000 0030 ~0x0000 0034
-	5	可设置	PendSV	可挂起的系统服务	0x0000 0038
-	6	可设置	SysTick	系统嘀嗒定时器	0x0000 003C
0	7	可设置	-	保留	0x0000 0040

位置	优先级	优先级类型	名称	说明	地址
1	8	可设置	PVD	连接到 EXTI Line16 的可编程电压检测（PVD）中断	0x0000 0044
2	9	可设置	-	保留	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 Line[1:0]中断	0x0000 0054
6	13	可设置	EXTI2_3	EXTI Line[3:2]中断	0x0000 0058
7	14	可设置	EXTI4_15	EXTI Line[15:4]中断	0x0000 005C
8	15	可设置	-	保留	0x0000 0060
9	16	可设置	-	保留	0x0000 0064
10	17	可设置	-	保留	0x0000 0068
11	18	可设置	-	保留	0x0000 006C
12	19	可设置	ADC	ADC 和 CMP1/2 中断 CMP1 连接至 EXTI Line21 CMP2 连接至 EXTI Line22	0x0000 0070
13	20	可设置	-	保留	0x0000 0074
14	21	可设置	-	保留	0x0000 0078
15	22	可设置	MCP	MCP 更新/占空比中断	0x0000 007C
16	23	可设置	MCP_FAULT	MCP 故障检测/短路保护中断	0x0000 0080
17	24	可设置	-	保留	0x0000 0084
18	25	可设置	TIM18_CC	TIM18 捕获/比较中断	0x0000 0088
19	26	可设置	TIM14	TIM14 全局中断	0x0000 008C
20	27	可设置	TIM6	TIM6 更新中断	0x0000 0090
21	28	可设置	TIM18	TIM18 全局中断	0x0000 0094
22	29	可设置	-	保留	0x0000 0098
23	30	可设置	I2C1	I2C1 全局中断（与 EXTI Line23 共用）	0x0000 009C
24	31	可设置	-	保留	0x0000 00A0
25	32	可设置	SPI1	SPI1 全局中断	0x0000 00A4
26	33	可设置	-	保留	0x0000 00A8
27	34	可设置	UART1	UART1 全局中断（与 EXTI Line25 共用）	0x0000 00AC
28	-	可设置	-	保留	0x0000 00B0
29	-	可设置	-	保留	0x0000 00B4
30	-	可设置	-	保留	0x0000 00B8
31	-	可设置	-	保留	0x0000 00BC

8.2 Systick Timer

这个定时器是专用于实时操作系统，也可当成一个标准的递减计数器。它具有下述特性:

- 24 位的递减计数器
- 自动重加载功能
- 当计数器为 0 时能产生一个可屏蔽系统中断
- 可编程时钟源

8.3 外部中断与事件（EXTI）

外部中断/事件控制器 EXTI 管理外部和内部异步中断/事件，产生事件请求到 MCU 和中断控制器，产生唤醒请求到电源管理器。

外部中断/事件控制器包含多个边沿检测器（外部/内部事件 Line），用于产生中断/事件请求。每个外部中断 Line 都可以独立地配置它的触发事件（上升沿或下降沿或双边沿），而内部中断 Line 有效沿固定为上升沿。

一个中断可以一直保持挂起：如果发生外部中断，一个状态寄存器被实例化并表明中断来源；一个事件通常是一个简单的脉冲，用于触发内核的唤醒（如 Cortex-M0 的 RXEV 引脚）。对于内部中断，挂起状态由生成的 IP 保证，所以不需要额外再定义特定的标志位。

每个输入 Line 都可以独立屏蔽其中断和事件的产生，另外，内部 Line 仅在 STOP 模式下采样。中断控制器可以通过写特定的寄存器，软件模拟外部事件（only）（与对应的硬件事件 Line 复用）。

所有通用 I/O 口连接到 16 个外部中断线。

8.3.1 EXTI 特性

- 支持产生多个事件/中断请求
- 作为外部或内部事件请求的每一线都可独立配置
- 每个事件/中断线都有独立的屏蔽

- 当系统不处于停机（STOP）模式时自动禁止内部各线
- 独立触发外部事件/中断线
- 每个外部中断线都有专用的状态位
- 仿真所有的外部事件请求

8.3.2 EXTI 框图

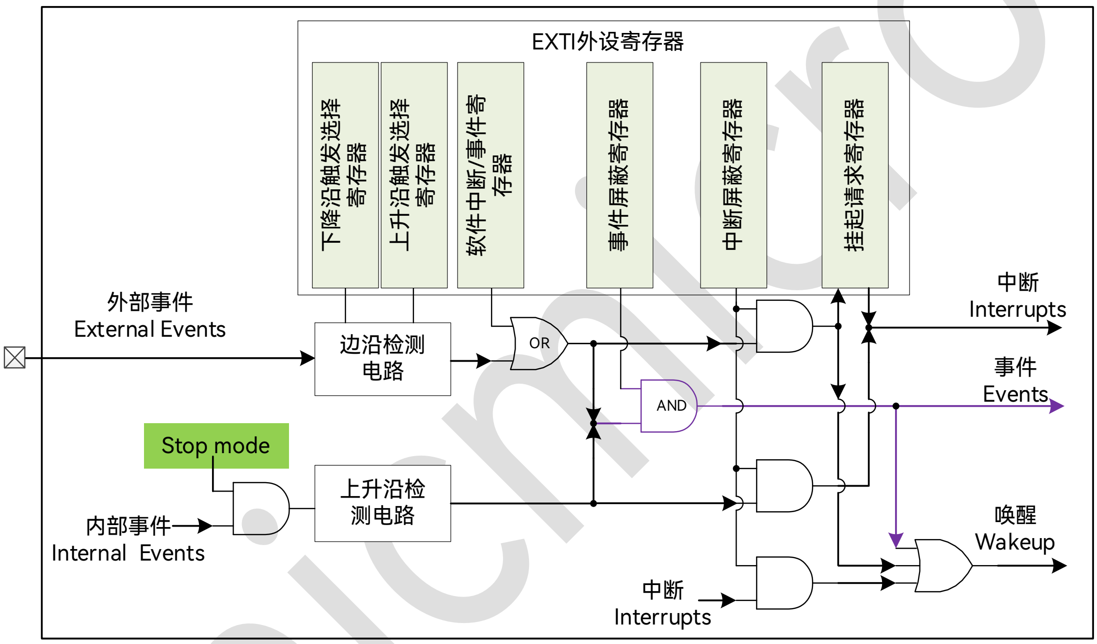


图 8-1：外部中断/事件框图

注:

1. 内部仅在 stop mode 有效。
2. 软件中断/事件控制器仅对外部事件有效。

8.3.3 事件管理

芯片支持外部或内部事件唤醒内核（WFE）。唤醒事件可以通过以下方式产生:

- 外设控制寄存器中使能中断，NVIC 中不开启，并在 Cortex-M0 System control 寄存器中启用 SEVONPEND 位；当 MCU 从 WFE 恢复时，EXTI 外设中断挂起位和外设 NVIC IRQ 通道挂起位

（在 NVIC 中断清除挂起寄存器中）必须被清除。

- 或者，在事件模式下配置外部或内部 EXTI 线。当 CPU 从 WFE 恢复时，不需要清除外设中断挂起位或者 NVIC IRQ 通道挂起位，因为事件线对应的挂起位没有置起。

8.3.4 功能描述

对于外部中断线，要产成中断，中断线需进行配置并启用。根据预期的边沿检测，编程两个触发寄存器 EXTI_RTSR 和 FTSR，对中断掩码寄存器 EXTI_IMR 的相应位写‘1’来启用中断请求。当所选边沿出现在外部中断线上时，就会产生一个中断请求。对应于中断线的挂起位也被置位。通过在挂起寄存器中写入‘1’来复位该请求。

对于内部中断线，有效沿保持为上升沿，在中断掩码寄存器中中断默认是启用的，并且在挂起寄存器中没有对应的挂起位。

要生成事件，事件线需进行配置并启用。根据预期的边沿检测，编程两个触发寄存器 EXTI_RTSR 和 EXTI_FTSR，对事件掩码寄存器 EXTI_EMR 的相应位写‘1’来启用事件请求。当所选边沿出现在事件线上时，将生成一个事件脉冲。事件行对应的挂起位不会置位。

对于外部线，中断/事件请求也可以通过软件产生，在软件中断/事件寄存器 EXTI_SWIER 中写入‘1’来实现。

注：只有当系统处于 STOP 模式时，才能触发与内部线相关的中断或事件。如果系统仍在运行，则不会产生任何中断/事件。

8.3.4.1 硬件中断选择

要将 1 个线配置为中断源，请使用以下步骤:

- 在 EXTI_IMR 寄存器中配置相应的掩码位。
- 配置中断线的触发选择位（EXTI_RTSR 和 EXTI_FTSR）。
- 配置 NVIC 使能位，控制 NVIC IRQ 通道映射到 EXTI，以便来自 EXTI 线之一的中断可以被正确地响应。

8.3.4.2 硬件事件选择

要将 1 个线配置为事件源，请使用以下步骤:

- 在 EXTI_EMR 寄存器中配置相应的掩码位。
- 配置事件线的触发选择位（EXTI_RTSR 和 EXTI_FTSR）。

8.3.4.3 软件中断/事件选择

任何外部线都可以配置为软件中断/事件线。下面是生成软件中断的步骤:

- 配置相应的掩码位（EXTI_IMR 或 EXTI_EMR）。
- 置位软件中断寄存器 EXTI_SWIER 中所需的位。

8.3.5 外部和内部中断/事件线映射

所有的 GPIO 以下面方式连接到 16 个外部中断/事件线:

表 8-2: 外部中断/事件线

GPIOs 及外设	对应的 EXTI line	控制位@寄存器
PA0/PB0	EXTI0	EXTI0[3:0]@ SYSCFG_EXTICR1
PA1/PB1	EXTI1	EXTI1[3:0]@ SYSCFG_EXTICR1
……	……	……
PA15	EXTI15	EXTI15[3:0]@ SYSCFG_EXTICR4
PVD 输出	EXTI16	-
保留，内部固定 low	EXTI17	-
保留，内部固定 low	EXTI18	-
保留，内部固定 low	EXTI19	-
保留，内部固定 low	EXTI20	-
比较器 1 输出	EXTI21	-
比较器 2 输出	EXTI22	-
内部 I2C1 唤醒事件	EXTI23	-
保留，内部固定 low	EXTI24	-
内部 UART1 唤醒事件	EXTI25	-
保留，内部固定 low	EXTI26~31	-

注：在某些产品上保留或不使用的 EXTI 线被视为内部的。

8.4 EXTI 寄存器

8.4.1 中断屏蔽寄存器（EXTI_IMR）

偏移地址：0x000

复位值：0x0280 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						IM25	Res.	IM23	IM22	IM21	Res.				IM16
						rw		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	名称	描述
31:26	RSV	保留位，固定读为 0
25	IM25	Line 25 中断屏蔽 0：屏蔽来自 Line 25 的中断请求 1：开启来自 Line 25 的中断请求
24	RSV	保留位，固定读为 0
23	IM23	Line 23 中断屏蔽 0：屏蔽来自 Line 23 的中断请求 1：开启来自 Line 23 的中断请求
22	IM22	Line 22 中断屏蔽 0：屏蔽来自 Line 22 的中断请求 1：开启来自 Line 22 的中断请求
21	IM21	Line 21 中断屏蔽 0：屏蔽来自 Line 21 的中断请求 1：开启来自 Line 21 的中断请求
20:17	RSV	保留位，固定读为 0
16	IM16	Line 16 中断屏蔽 0：屏蔽来自 Line 16 的中断请求 1：开启来自 Line 16 的中断请求
15~0	IMx	Line x（x=0~15）中断屏蔽 0：屏蔽来自 Line x 的中断请求 1：开启来自 Line x 的中断请求

8.4.2 事件屏蔽寄存器（EXTI_EMR）

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						EM25	Res.	EM23	EM22	EM21	Res.				EM16
						rw		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	名称	描述
31:26	RSV	保留位，固定读为 0
25	EM25	Line 25 事件屏蔽位 0：屏蔽来自 Line 25 的事件请求 1：开启来自 Line 25 的事件请求
24	RSV	保留位，固定读为 0
23	EM23	Line 23 事件屏蔽位 0：屏蔽来自 Line 23 的事件请求 1：开启来自 Line 23 的事件请求
22	EM22	Line 22 事件屏蔽位 0：屏蔽来自 Line 22 的事件请求 1：开启来自 Line 22 的事件请求
21	EM21	Line 21 事件屏蔽位 0：屏蔽来自 Line 21 的事件请求 1：开启来自 Line 21 的事件请求
20:17	RSV	保留位，固定读为 0
16	EM16	Line 16 事件屏蔽位 0：屏蔽来自 Line 16 的事件请求 1：开启来自 Line 16 的事件请求
15:0	EMx	Line x（x=0~15）事件屏蔽位 0：屏蔽来自 Line x 的事件请求 1：开启来自 Line x 的事件请求

8.4.3 上升沿/高电平触发模式选择寄存器（EXTI_RTSR）

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.									RT22	RT21	Res.				RT16
									rw	rw					rw

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	名称	描述
31:23	RSV	保留位，固定读为 0
22	RT22	Line 22 上升沿触发事件配置位 0：禁止上升沿触发（中断和事件） 1：允许上升沿触发（中断和事件）
21	RT21	Line 21 上升沿触发事件配置位 0：禁止上升沿触发（中断和事件） 1：允许上升沿触发（中断和事件）
20:17	RSV	保留位，固定读为 0
16	RT16	Line 16 上升沿触发事件配置位 0：禁止上升沿触发（中断和事件） 1：允许上升沿触发（中断和事件）
15:0	RTx	Line x 上升沿触发事件配置位（x = 0~17） 0：禁止上升沿触发（中断和事件） 1：允许上升沿触发（中断和事件）

8.4.4 下降沿/低电平触发模式选择寄存器（EXTI_FTSR）

偏移地址：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.									FT22	FT21	Res.				FT16
									rw	rw					rw

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	名称	描述
31:23	RSV	保留位，固定读为 0
22	FT22	Line 22 下降沿触发事件配置位 0：禁止下降沿触发（中断和事件） 1：允许下降沿触发（中断和事件）
21	FT21	Line 21 下降沿触发事件配置位 0：禁止下降沿触发（中断和事件） 1：允许下降沿触发（中断和事件）
20:17	RSV	保留位，固定读为 0
16	FT16	Line 16 下降沿触发事件配置位 0：禁止下降沿触发（中断和事件） 1：允许下降沿触发（中断和事件）
15:0	FTx	Line x 下降沿触发事件配置位（x = 0~15） 0：禁止下降沿触发（中断和事件） 1：允许下降沿触发（中断和事件）

8.4.5 软件中断事件寄存器（EXTI_SWIER）

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.									SWI22	SWI21	Res.				SWI16
									rw	rw					rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWI15	SWI14	SWI13	SWI12	SWI11	SWI10	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	名称	描述
31:23	RSV	保留位，固定读为 0
22	SWI22	Line 22 的软中断位 当该位为‘0’时，写‘1’将置位 EXTI_PR 中相应的挂起位。如果在 EXTI_IMR 允许此 Line 产生该中断，则此时将产生一个中断请求。 通过对 EXTI_PR 的对应位写入‘1’，可以清除该位为‘0’。
21	SWI21	Line 21 的软中断位 当该位为‘0’时，写‘1’将置位 EXTI_PR 中相应的挂起位。如果在 EXTI_IMR 允许此 Line 产生该中断，则此时将产生一个中断请求。 通过对 EXTI_PR 的对应位写入‘1’，可以清除该位为‘0’。
20:17	RSV	保留位，固定读为 0
16	SWI16	Line 16 的软中断位 当该位为‘0’时，写‘1’将置位 EXTI_PR 中相应的挂起位。如果在 EXTI_IMR 允许此 Line 产生该中断，则此时将产生一个中断请求。 通过对 EXTI_PR 的对应位写入‘1’，可以清除该位为‘0’。
15:0	SWIx	Line x 的软中断位（x = 0~15） 当该位为‘0’时，写‘1’将置位 EXTI_PR 中相应的挂起位。如果在 EXTI_IMR 允许此 Line 产生该中断，则此时将产生一个中断请求。 通过对 EXTI_PR 的对应位写入‘1’，可以清除该位为‘0’。

8.4.6 挂起寄存器（EXTI_PR）

偏移地址：0x014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.									PIF22	PIF21	Res.				PIF16
									rc_w1	rc_w1					rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIF15	PIF14	PIF13	PIF12	PIF11	PIF10	PIF9	PIF8	PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0
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	名称	描述
31:23	RSV	保留位，固定读为 0
22	PIF22	Line 22 挂起位 0：没有发生触发请求 1：发生了选择的触发请求 当在外部中断线上发生了选择的边沿事件，该位被置‘1’。在该位中写入‘1’可以清除它。
21	PIF21	Line 21 挂起位 0：没有发生触发请求 1：发生了选择的触发请求 当在外部中断线上发生了选择的边沿事件，该位被置‘1’。在该位中写入‘1’可以清除它。
20:17	RSV	保留位，固定读为 0
16	PIF16	Line 16 挂起位 0：没有发生触发请求 1：发生了选择的触发请求 当在外部中断线上发生了选择的边沿事件，该位被置‘1’。在该位中写入‘1’可以清除它。
15:0	PIFx	Line x 挂起位（x =0~15） 0：没有发生触发请求 1：发生了选择的触发请求 当在外部中断线上发生了选择的边沿事件，该位被置‘1’。在该位中写入‘1’可以清除它。

9 循环冗余校验计算单元（CRC）

9.1 概述

CRC（循环冗余校验）计算单元，根据特定的多项式，从一个 8 位、16 位或 32 位的数据字中产生 CRC 码。

CRC 在应用中主要来验证数据传输或存储的完整性。从功能安全标准考量，也提供了 FLASH 存储可靠性验证方法。

CRC 计算单元可以用于在运行期间计算软件签名，并与链接时生成的参考签名比对。

9.2 特性

- 支持 CRC-32 多项式：0x4C11DB7（以太网）
- 支持 CRC-16-CCITT 多项式：0x1021
- 支持 CRC-16 多项式：0x8005
- 支持 8 位、16 位和 32 位数据
- CRC 初值可编程
- 输入/输出 32 位数据寄存器为同一个寄存器
- 输入缓存可以避免计算过程总线停顿
- 32 位数据计算周期：4 AHB clock（HCLK）
- 输入/输出数据位序颠倒可选

9.3 功能描述

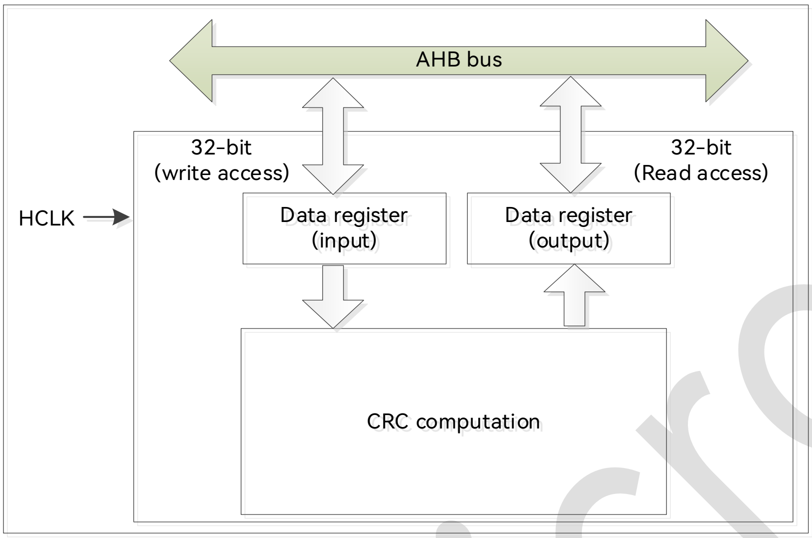


图 9-1：CRC 模块框图

9.3.1 CRC 操作

CRC 计算单元中包含一个 32 位可读/写的数据寄存器（CRC_DR）。它用来输入新的计算数据（写操作），也用来输出上次计算的结果（读操作）。

每次对该寄存器的写操作都会使得 CRC 计算单元将所写入的数据和上次计算得到的数据进行综合计算，并得到一个新的计算结果。

CRC 计算单元根据写入数据的格式不同来决定是整字计算还是一个字节一个字节计算。

通过 CRC_CR 寄存器的 MOD[1:0]位段可以选择 CRC 是工作在 32 位还是 16 位模式下，当工作在 32 位模式下，计算结果在 CRC_DR 寄存器中的有效位为 DR[31:0]。当工作在 16 位模式下，计算结果在 CRC_DR 寄存器中的有效位为 DR[15:0]。

CRC_DR 寄存器可以按字（word）访问，也可以按半字（右对齐）或者字节（右对齐）的方式访问。其他寄存器仅支持 32-bit 位宽访问。

不同数据宽度计算持续时间：

- 32 位需要 4 个 AHB 时钟周期
- 16 位需要 2 个 AHB 时钟周期

- 8 位需要 1 个 AHB 时钟周期

内建一个输入缓冲区，可以实现第二个数据立即写入，而无需等待当前计算持续时间结束。

数据位宽可以动态调整，以最大限度地减少给定数据的写访问次数。例如，一个 5 个字节的 CRC 运算，可以通过一个字（word）写入和一个字节（byte）写入的组合来完成计算。

输入数据高低位序可以颠倒，以适应各种不同的大小端体系。数据位序颠倒操作可以按 8 位、16 位和 32 位进行，这个功能由 CRC_CR 寄存器中的 REV_IN[1:0]位来选择设置。

数据位序按 32 位进行颠倒，就是在一个 32 位数据里，依次对 bit0 与 bit31 交换位置，bit1 与 bit30 交换位置，直到 bit15 与 bit16 交换位置，然后组成一个新的 32 位数据。

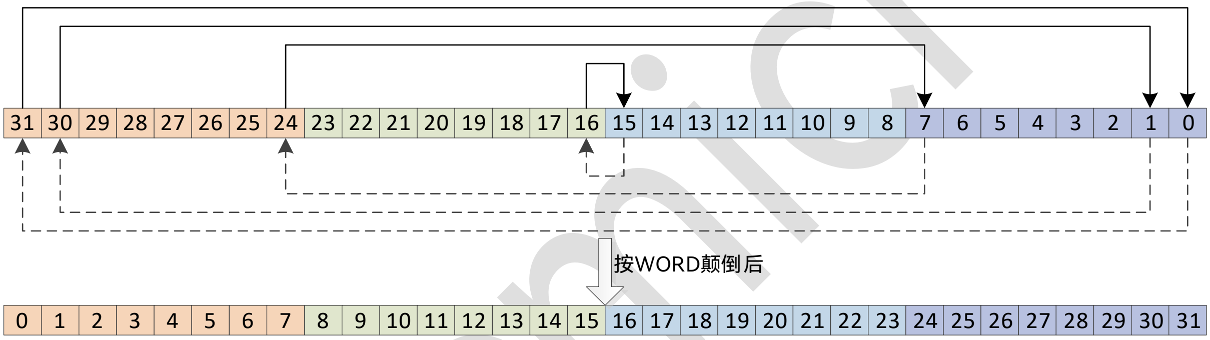


图 9-2：数据位序按 32 位颠倒

数据位序按 16 位颠倒，就是一个 16 位数据里，依次对 bit0 与 bit15 交换位置，bit1 与 bit14 交换位置，直到 bit7 与 bit8 交换位置，交换后组成一个新的 16 位数据。如果是一个 32 位的数据，则高 16 位与低 16 位分别按上述规则进行颠倒，最后组成一个新的 32 位数据。

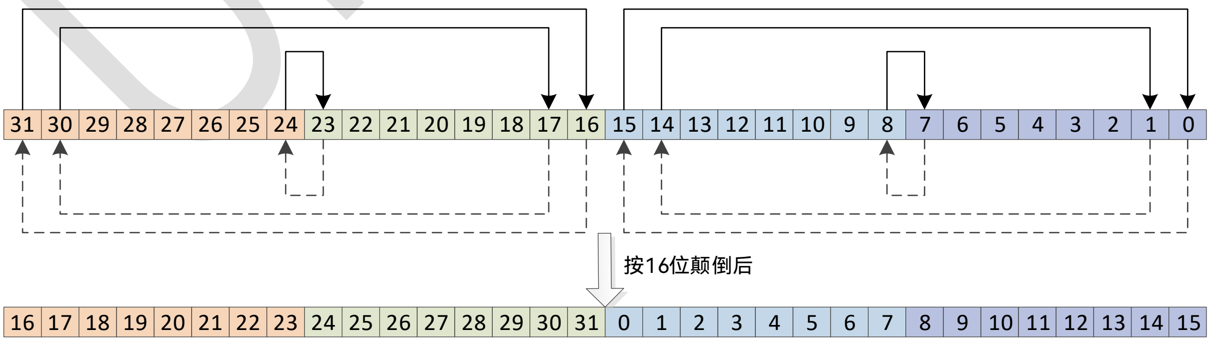


图 9-3：数据位序按 16 位颠倒

数据位序按 8 位颠倒，就是一个 8 位数据，依次对 bit0 与 bit7 交换位置，bit1 与 bit6 交换位置，直到 bit3 与 bit4 交换位置，然后组成一个新的 8 位数据。如果是一个 16 位或 32 位数据，则

里面的每个字节按上述规则进行交换位置，然后重新组成一个新的 16 位或 32 位数据。

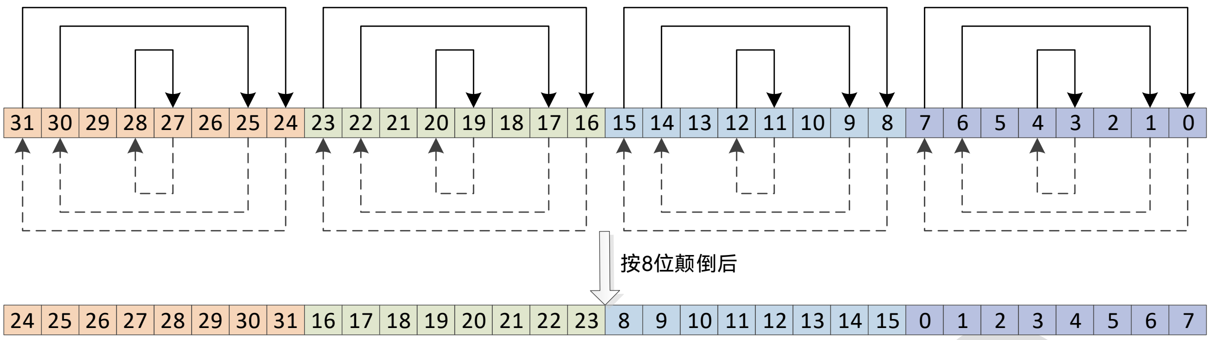


图 9-4：数据位序按 8 位颠倒

举例 1： 输入数据 0x1A2B3C4D 用于 CRC 计算 ：

- 按字节颠倒就是：0x58D43CB2
- 按半字颠倒就是：0xD458B23C
- 按整字颠倒就是：0xB23CD458

注：当输入的数据长度低于位序颠倒所选择的长度时，实际参与运算的是输入的数据长度。比如 CRC_DR 寄存器按 8 位数据访问，输入 0x1A，按半字进行位序颠倒，颠倒后依然是 0x58 参与 CRC 计算，其他位忽略。

输出数据也可以通过设置 CRC_CR 寄存器中的 REV_OUT 位来进行位序颠倒。当 CRC 工作在 32 位模式下，该操作以 32 位为有效位按位进行颠倒的，当 CRC 工作在 16 位模式下，该操作是以低 16 位为有效位按位进行颠倒，高 16 位不参与位序颠倒。

举例 2:

输出数据 0x11223344 转换过来就是 0x22CC4488。

软件对 CRC_CR 寄存器中的 RESET 位写 1 可以将 CRC 进行复位，同时会将 CRC_INIT 的值更新到 CRC_DR 寄存器中。RESET 位只能通过软件写 1，由硬件自动清零。

初始化 CRC_DR 寄存器

CRC 的初始值可以通过 CRC_INIT 寄存器来指定。当软件对 CRC_INIT 寄存器写入新值时，CRC_INIT 寄存器的值自动更新到 CRC_DR 寄存器中，并停止 CRC 计算。新写入的值作为下一次校验的初始值，等到下一次对 CRC_DR 进行写操作时，开启 CRC 计算。

9.3.2 CRC 多项式

本芯片支持:

- CRC-32 多项式：0x4C11DB7（以太网）
  - $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- CRC-16-CCITT 多项式：0x1021
  - $X^{16} + X^{12} + X^5 + 1$
- CRC-16 多项式：0x8005
  - $X^{16} + X^{15} + X^2 + 1$

通过设置 CRC_CR 寄存器中的 MOD[1:0] 位选择不同的计算多项式。

选择 2 种 CRC-16 多项式模式时，CRC_DR 和 CRC_INIT 寄存器仅低 16bit 有效，高 16bit 无效。

9.4 相关寄存器

9.4.1 CRC 数据寄存器（CRC_DR）

偏移地址：0x000

复位值：0xFFFF FFFF

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[31:0]															
rw															

位	名称	描述
31:0	DR[31:0]	数据寄存器位 该寄存器用于写入待计算的新数据，直接将其写入即可。 读取该寄存器得到的则是上次 CRC 计算的结果。 如果数据不足 32 位，右对齐读/写正确数据。

9.4.2 CRC 控制寄存器（CRC_CR）

偏移地址：0x008

复位值：0x0000 0000

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

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

位	名称	描述
31:8	RSV	保留位，始终读为 0
7	REV_OUT	输出数据位序颠倒，该位控制输出数据的位序颠倒 0：不颠倒 1：颠倒
6:5	REV_IN[1:0]	输入数据位序颠倒，该位控制输入数据的位序颠倒 00：不颠倒 01：按字节为单位颠倒 10：按半字为单位颠倒 11：按字为单位颠倒
4:3	RSV	保留位，始终读为 0
2:1	MOD[1:0]	CRC 计算模式选择： 00: CRC-32 多项式：0x4C11DB7 01: CRC-16-CCITT 多项式：0x1021 1x: CRC-16 多项式：0x8005
0	RESET	复位控制，软件置位，硬件自动清零 此位用来复位整个 CRC 计算单元，并将 CRC_INIT 寄存器中的值更新到数据寄存器中。

9.4.3 初值寄存器（CRC_INIT）

偏移地址：0x010

复位值：0xFFFF FFFF

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC_INIT[31:0]															
rw															

位	名称	描述
31:0	CRC_INIT[31:0]	CRC 初值 该寄存器用来设置 CRC 的初值。

10 协处理器（COPROC）

10.1 概述

COPROC 模块包括除法模块和开方模块。整个模块挂载在 AHB 总线下。

10.2 除法模块特性

- 支持 32 位宽无符号/有符号数的除法运算
- 商和余数均为 32 位宽度
- 18 个周期完成除法运算
- 写除数寄存器启动除法运算

除法模块主要由被除数、除数、商和余数寄存器组成，均为 32 位寄存器。使用时先向被除数寄存器写入数据，再向除数寄存器写入数据时，硬件会自动启动计算。如果除数为 0，COPROC_CSR 寄存器中的 DIVOVF 位会置 1，此时可以软件向此位写 1 清零。

硬件开始计算后，COPROC_CSR 寄存器中的 DIVSTART 位会自动置 1，完成计算后会由硬件清零。软件可以通过读取此位判断计算是否完成。当在计算过程中，软件向此位写 1，会强制停止硬件计算，此时除法模块的内部数据和状态标志由硬件清除，为下一次计算做好准备。

当硬件在计算过程中，软件读取 COPROC_DIVQTR 或 COPROC_DIVRMDR 寄存器，自动等待计算完成并返回结果。

10.3 开方模块特性

- 支持 32 位宽无符号数的开方运算
- 平方根为位 32 位宽度
- 17 个周期完成开方运算

- 写被开方数寄存器启动开方运算

开方模块主要由被开方数和平方根寄存器组成，均为 32 位寄存器。使用时向被开方数寄存器 COPROC_RADR 中写入数据，硬件会自动启动计算。

硬件开始计算后，COPROC_CSR 寄存器中的 SQRTSTART 位会自动置 1，完成计算后由硬件清零。软件可以通过读取此位判断计算是否完成。当计算过程中，软件向此位写 1，会强制停止硬件计算，此时开方模块的内部数据和状态由硬件清除，为下一次计算做好准备。

当硬件在计算过程中，软件读取开方根寄存器 COPROC_SQRT，自动等待计算完成并返回结果。

10.4 相关寄存器

10.4.1 控制状态寄存器（COPROC_CSR）

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.											DIVOVF	Res.			
											rc_w1				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											DIVSTART	Res.			SQRTSTART
											rc_w1				rc_w1

位	名称	描述
31:21	RSV	保留位，始终读为 0
20	DIVOVF	除数为 0 溢出状态标志位 此位由硬件置 1，软件写 1 清零 0: 除数不为 0 1: 当前写入的除数为 0
19:5	RSV	保留位，始终读为 0
4	DIVSTART	除法运算状态控制位 此位只能由硬件开始运算时自动置 1，软件可以写 1 清零，硬件停止运算。

位	名称	描述
		0：未运算/运算完成 1：正在运算中 注：当运算过程中软件写 1 将此位清零，硬件需要清除内部的数据和 FLAG。
3:1	RSV	保留位，始终读为 0
0	SQRTSTART	开方运算状态控制位 此位只能由硬件开始运算时自动置 1，软件可以写 1 清零，硬件停止运算。 0：未运算/运算完成 1：正在运算中 注：当运算过程中软件写 1 将此位清零，硬件需要清除内部的数据和 FLAG。

10.4.2 被开方数寄存器（COPROC_RADR）

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
RAD[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAD[31:0]															
rw															

位	名称	描述
31:0	RAD[31:0]	开始计算前由软件写入被开方数 Radicand，写入后硬件自动开始运算。 在运算过程中，软件对此寄存器的写入操作无效。

10.4.3 均方根寄存器（COPROC_SQRTR）

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
SQRT[31:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQRT[31:0]															
r															

位	名称	描述
31:0	SQRT[31:0]	计算结束后保存平方根 Square root 在计算过程中软件对该寄存器的读操作，会自动等待运算结束。

10.4.4 被除数寄存器（COPROC_DIVDR）

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
DIVIDEND[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVIDEND[31:0]															
rw															

位	名称	描述
31:0	DIVIDEND[31:0]	计算前写入被除数 Dividend 在计算过程中软件对该寄存器的写操作无效。

10.4.5 除数寄存器（COPROC_DIVSR）

偏移地址：0x14

复位值：0x0000 0001

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVISOR[31:0]															
rw															

位	名称	描述
31:0	DIVISOR[31:0]	计算前写入除数 Divisor，写入后硬件自动开始计算，在计算过程中软件对该寄存器的写操作无效。

10.4.6 商寄存器（COPROC_DIVQTR）

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
QUOTIENT[31:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUOTIENT[31:0]															
r															

位	名称	描述
31:0	QUOTIENT[31:0]	计算结束后保存商 Quotient 在计算过程中软件对该寄存器的读操作，会自动等待运算结束。

10.4.7 余数寄存器（COPROC_DIVRMDR）

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
RMD[31:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RMD[31:0]															
r															

位	名称	描述
31:0	RMD[31:0]	计算结束后保存商余数 Remainder 在计算过程中软件对该寄存器的读操作，会自动等待运算结束。

11 模数转换器（Analog to Digital Converter，ADC）

11.1 概述

本产品内嵌 1 个 12 位的逐次逼近型模拟/数字转换器（SAR ADC），多达 16 个外部通道和 4 个内部通道（温度传感器、电压参考和 2 路运算放大器输出），可以实现单次、连续模式转换。单个通道或多个通道可以编入序列中自动转换，数据结果可选择左对齐或右对齐存储于对应的 16 位结果寄存器 ADC_DRx 中。

模拟看门狗功能允许非常精准地监视一路选中的通道，当被监视的信号超出预置的阀值时，将产生中断。

11.2 主要特性

- 高性能
  - 12-bit/10-bit 分辨率可选
  - ADC 转换时间：1.0 μs @12-bit 分辨率（1MHz），0.93 μs @10-bit 分辨率
  - 自校准
  - 采样时间可编程
  - 数据对齐
- 低功耗
- 应用程序可以降低 PCLK 频率，以实现低功耗，同时保持最佳 ADC 性能。（举例，无论 PCLK 的频率如何，保持 1.0 μs 的转换时间，此模式 ADC 转换时钟须设置为 24MHz 的异步时钟）
- 模拟输入通道
  - 16 个外部模拟输入通道

  - 1 个内部温度传感器通道（VSENSE）
  - 1 个内部电压参考通道（VREFINT）
  - 2 个运算放大器输出测量通道
- 多种启动方式
  - 软件启动
  - 硬件触发（MCP，TIM18）
- 转换模式
  - 单次模式，每次触发转换选定所有通道
  - 连续模式，连续转换选定的所有通道
- 中断产生：段转换完成、序列转换完成、模拟看门狗、溢出事件
- 模拟看门狗
- ADC 供电：2.4V~5.5V
- ADC 输入范围：$V_{SSA} \leq V_{IN} \leq V_{DDA}$

11.3 功能描述

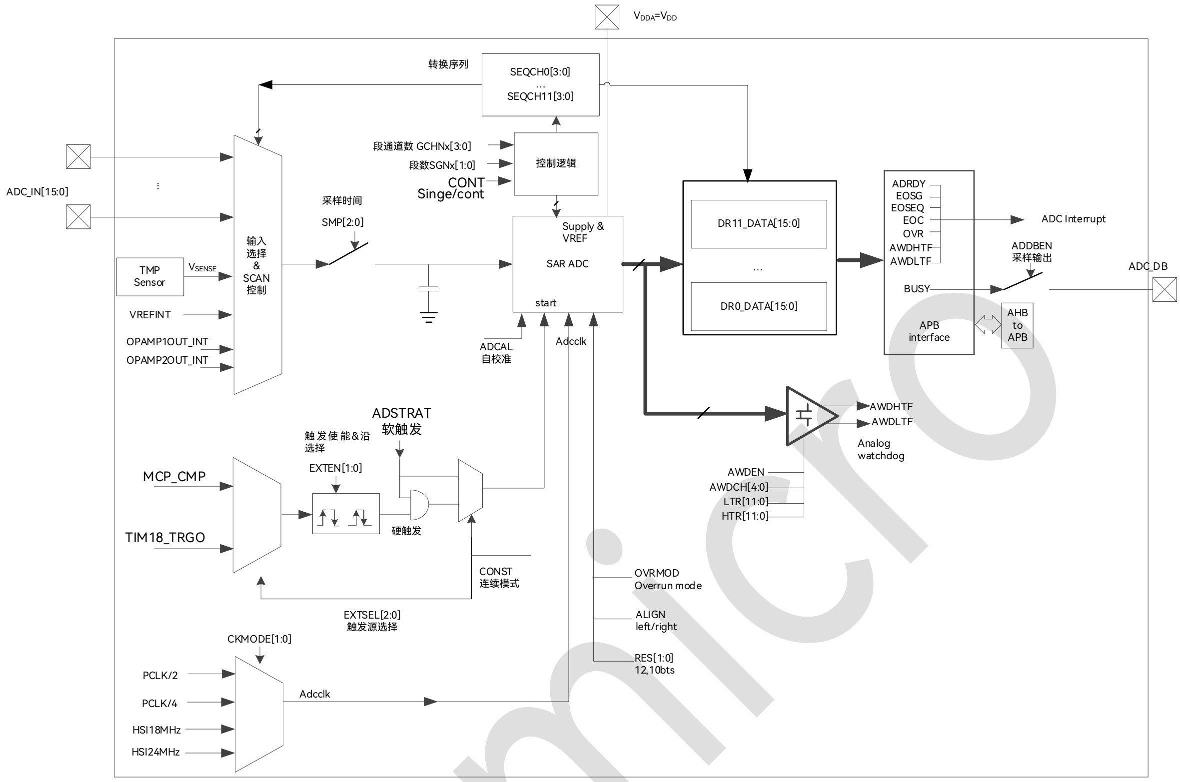


图 11-1：ADC 框图

11.3.1 定义

通道：共有两种涵义，其一是芯片的外部输入模拟通道，即 AN0~AN15，OP，TS，VREF。其二是指段和序列中的采样通道数。后文如无特殊说明，通道即指段和序列中的通道，而外部通道则用模拟通道进行描述。

段：一次触发采样的最小单位，可以由一个模拟通道或多个模拟通道组成，一段最多由 12 个模拟通道组成，也可以是同一个模拟通道在一个段中采样 12 次。对一段进行采样，即是对段中的每个模拟通道依次进行转换。

采样序列：由一段或几段组成一个采样序列，每一段最多由 12 个模拟通道组成，一个采样序列最多由 4 个段组成。

结果序列：最多保存 12 个采样结果数据，分别与 ADC_DR0~ADC_DR11 对应。

11.3.2 ADC 校准

ADC 支持校准功能，在此过程，ADC 计算一个校准因子，该因子内部作用于 ADC，在 ADC 断电后失效。在 ADC 校准期间直到校准完成，ADC 禁止使用。

在 A/D 转换前应执行校准操作，用于消除各芯片 AD 转换的偏移误差。

设置 ADCAL=1，启动校准，且只能在 ADC 禁用（ADEN=0）时启动校准。校准期间 ADCAL 保持为 1，当完成校准后，ADCAL 位由硬件清零。此时，可以从 ADC_DR0 读取校准因子（bit6~0）。

当 ADC 禁用（ADEN=0）时，内部模拟校准因子保留。当 ADC 工作条件发生变化时（VDD 变化为 ADC offset 误差的主要原因，温度变化次之），建议重新运行一个校准循环。

校准软件操作流程：

- 确保 ADEN=0
- 设置 ADCAL=1
- 等待直到 ADCAL=0
- 校准因子可从 ADC_DR0 寄存器读取（bit6~0）

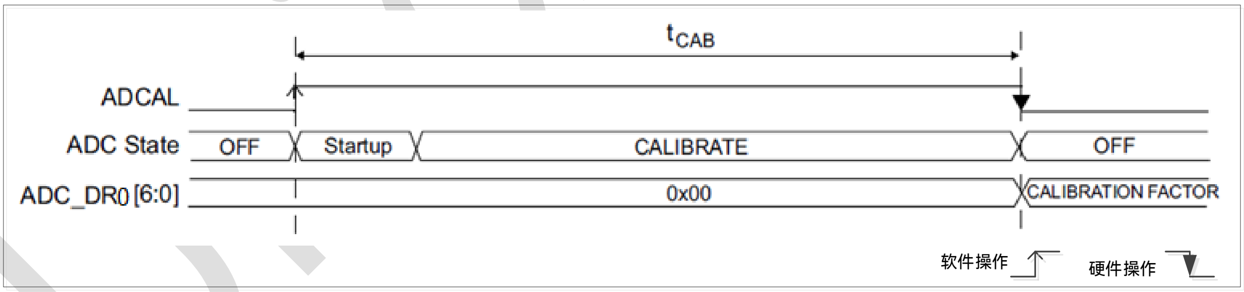


图 11-2：ADC 校准

11.3.3 ADC 开关控制（ADEN，ADDIS，ADRDY）

在 MCU 上电时，ADC 被禁用并处于掉电模式（ADEN=0）。

如下图所示，ADC 开始精确转换前，需要一个稳定时间 t_{STAB}。

两个控制位用于开启或关闭 ADC：

- 设置 ADEN=1 开启 ADC。当 ADC 模块准备好时，ADRDY 标志置 1。
- 设置 ADDIS=1 来关闭 ADC，并使 ADC 处于断电模式。当 ADC 模块全关断后，硬件自动清除 ADEN 和 ADDIS 位。

ADC 转换既可有软件设置 ADSTART=1 启动，也可以由外部触发事件启动（触发启动开启）。

使能 ADC 流程如下:

- ADC_CR 寄存器中设置 ADEN=1。
- 等待 ADRDY=1（ADC_ISR 寄存器）（ADRDY 在 ADC 启动时间之后被置位）。如果置位 ADRDYIE 位（ADC_IER 寄存器）启用中断，这可以通过中断来处理。

关闭 ADC 流程如下:

- 检查 ADSTART=0（ADC_CR 寄存器）以确保 ADC 不在转换中。若需要，可对 ADSTP（ADC_CR 寄存器）写‘1’，停止正在进行 ADC 转换，并等待 ADSTP 被清 0。
- 设置 ADC_CR 寄存器中的 ADDIS=1。
- 若应用需要，等待直至 ADCEN=0（ADC_CR 寄存器），表示 ADC 模块已完全关闭（一旦 ADEN=0，ADDIS 自动清 0）。

注：当 ADCAL=1 时，ADEN 不能被置位。

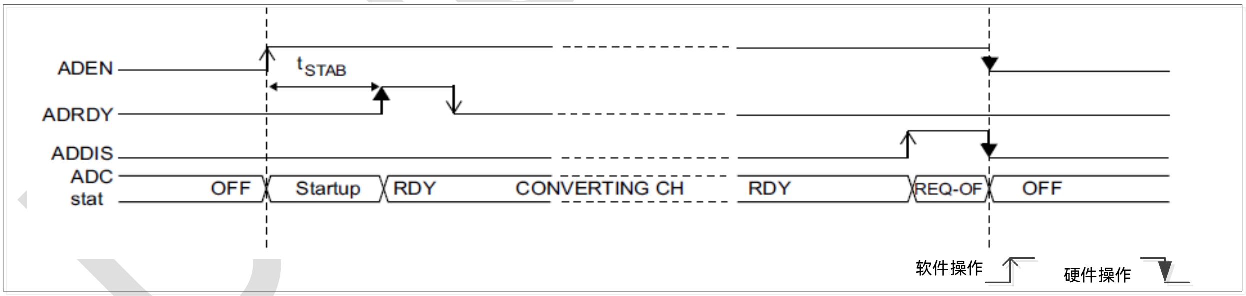


图 11-3：开启/关闭 ADC

11.3.4 ADC 时钟（CKMODE）

ADC 具有双时钟架构，ADC 时钟可以选择异步时钟，独立于 APB 时钟（PCLK）。

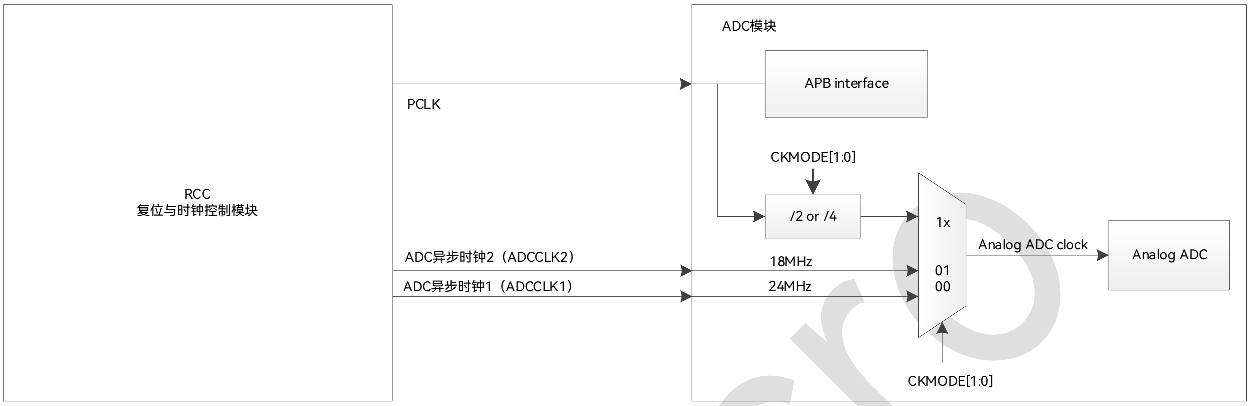


图 11-4：ADC 时钟框图

模拟 ADC 可以选择 2 种不同的时钟源（时钟使能参见章节：5 复位与时钟控制（RCC））：

a) ADC 异步时钟（24MHz 或 18MHz），在 ADC_CFGR2 寄存器中，设置 CKMODE[1:0]=00b，选择 24MHz 的 ADC 异步时钟 1（ADCCLK1），设置 CKMODE[1:0]=01b，选择 18MHz 的 ADC 异步时钟 2（ADCCLK2）。

b) APB 时钟/2 或/4，由 ADC 总线接口派生得到，设置 ADC_CFGR2 寄存器中的 CKMODE[1:0] 为 10b 或 11b。

选项 a）优点在于，无论 APB 时钟如何选择，ADC 都可以达到最大 ADC 时钟频率。

选项 b）优点在于，避免时钟域的重同步，当 ADC 由定时器触发，且应用要求精确触发 ADC 时，对避免不确定性是非常有用的（否则，两个时钟域间的重同步将增加触发瞬间的不确定性）。

表 11-1：触发和开始转换之间的延迟

ADC 时钟源	CKMODE[1:0]	触发事件和开始转换之间的延迟
24MHz	00	延迟的不确定性（抖动）
18MHz HSI RC 时钟	01	延迟的不确定性（抖动）
PCLK / 2	10	2~3 个 ADC 时钟周期
PCLK / 4	11	2~3 个 ADC 时钟周期

11.3.5 ADC 配置

ADC_CR 寄存器中的 ADCAL 和 ADEN 位，软件必须在 ADC 禁止（ADEN 必须为 0）的情况下改写。

ADC_CR 寄存器中的 ADSTART 和 ADDIS 位，软件必须在 ADC 开启且无关闭请求挂起（即 ADEN=1 且 ADDIS=0）的情况下改写。

对于其他的控制位（在 ADC_IER、ADC_CFGR1/2、ADC_SMPR、ADC_HTR、ADC_LTR、ADC_SEQCHSELR1、ADC_SEQCHSELR2 和 ADC_CCR 寄存器中），软件必须在 ADC 开启（ADEN=1）且无进行的转换（ADSTART = 0）的情况下，才能进行改写。

ADC_CR 寄存器中的 ADSTP 位，软件必须在 ADC 开启且无关闭请求挂起（ADSTART=1 和 ADDIS=0）的情况下改写。

注：没有硬件保护机制，以防止上述规则禁止的写操作。若发生了禁止的写操作，ADC 将进入不确定状态。要恢复至正确操作，ADC 必须被关闭（ADEN=0，且 ADC_CR 寄存器清 0）。

11.3.6 可编程采样时间

在启动数模转换之前，ADC 需要在被测电压源和内嵌采样电容间建立直接连接。则必须有足够长的采样时间，以便输入电压源对采样保持电容充电并达到输入电压的水平。

根据输入电压源的输入阻抗，编程采样时间进而调整转换速度。

输入电压采样的时钟周期个数，通过 ADC_SMPR 寄存器中的 SMP1[2:0] 和 SMP2[2:0]位来进行修改。此两组采样时钟选择可以用于所有的 ADC 通道，通过 ADC_SMPR 寄存器中的 SMPSELx 来选择相应通道是使用 SMP1[2:0]还是 SMP2[2:0]中所设置的采样时间。比如当 SMPSEL8=0 时，表示 ADC_IN8 使用 SMP1[2:0]中设置的采样时间，当 SMPSEL8=1 时，表示 ADC_IN8 使用 SMP2[2:0]中设置的采样时间。

可编程采样时间对所对应的通道生效。如有应用需求，则可在每次转换之间，软件改变和调整采样时间。总的转换时间计算如下:

$$t_{CONV} = \text{采样时间} + 14.5 \times \text{ADC 时钟周期}$$

举例:

当 ADC_CLK=18 MHz 且采样时间为 3.5 ADC 时钟周期:

$$t_{CONV} = 3.5+14.5 = 18 \text{ ADC clk} = 1.0\ \mu s$$

11.3.7 单次转换

当 ADC_CFGR1 寄存器中的 CONT 位为 0 时，ADC 工作在单次转换模式，软件使能 ADSTART 或硬件触发一次转换一段采样。在此工作模式下，一个采样序列可以由一段或多段组成，每段的数量可以通过 ADC_CR 寄存器中的 SGCHNx[3:0]位设置，每段的数量最大不超过 12。

如果一个采样序列由多个段组成，每次软件设置 ADSTART 或硬件触发时，硬件依次启动一个段的转换，结束后 ADC_ISR 寄存器中的 EOSG 位会置起，如果 ADC_IER 寄存器中的 EOSGIE 位为 1，会触发一个中断。如果整个采样序列中所有的段转换结束，ADC_ISR 寄存器中的 EOSEQ 位置起，如果 ADC_IER 寄存器中的 EOSEQIE 位为 1，此时会触发一个中断。

假如有 8 路外部模拟通道需要被转换，分别为 ADC_IN3，ADC_IN4，ADC_IN5，ADC_IN7，ADC_IN8，ADC_IN10，ADC_IN13，ADC_IN2。这 8 个外部通道共分为 3 段，每段的数量依次为 3，2，3。设置 ADC_CR 寄存器中的 SGN[1:0]=10b，SGCHN1[3:0]=2，SGCHN2[3:0]=1，SGCHN3[3:0]=2。设置 ADC_SEQCHSELR1 寄存器中的 SEQCH0=3，SEQCH1=4，SEQCH2=5，SEQCH3=7，SEQCH4=8，SEQCH5=10，设置 ADC_SEQCHSELR2 寄存器中的 SEQCH6=13，SEQCH7=2。这些外部通道的转换结果将依次保存在 ADC_DR0~ADC_DR7 中。当第一次启动转换时，SEQCH0~SEQCH2 被转换，结果存放在 ADC_DR0~ADC_DR2 中。第二次启动转换时，SEQCH3、SEQCH4 被转换，结果存放在 ADC_DR3、ADC_DR4 中。第三次启动转换时，SEQCH5~SEQCH7 被转换，结果存放在 ADC_DR5~ADC_DR7 中。此时，三段转换完成，即一个采样序列转换完成。下一次硬件启动转换时，将重新从第一段开始转换，即依次转换 SEQCH0~SEQCH2 通道。

如果一个采样序列中所有段的通道数量总和超过 12，那么当硬件转换完 SEQCH11 时会自动从 SEQCH0 开始转换，直到把所有段中的通道全部转换完成为止。假如一个采样序列共分为 3 段，每段的数量分别为 7，4，6。待转换的模拟通道依次为 ADC_IN3，ADC_IN4，ADC_IN6，ADC_IN7，

ADC_IN9，ADC_IN10，ADC_IN15，ADC_IN0，ADC_IN2，ADC_IN4，ADC_IN17，ADC_IN16。设置 ADC_CR 寄存器中的 SGN[1:0]=10b，SGCHN1[3:0]=6，SGCHN2[3:0]=3，SGCHN3[3:0]=5。设置 ADC_SEQCHSELR1 寄存器中的 SEQCH0=3，SEQCH1=4，SEQCH2=6，SEQCH3=7，SEQCH4=9，SEQCH5=10，设置 ADC_SEQCHSELR2 寄存器中的 SEQCH6=15，SEQCH7=0，SEQCH8=2，SEQCH9=4，SEQCH10=17，SEQCH11=16。这些外部通道的转换结果将依次保存在 ADC_DR0~ADC_DR11 中。当第一次启动转换时，SEQCH0~SEQCH6 被转换，结果存放在 ADC_DR0~ADC_DR6 中。第二次启动转换时，SEQCH7~SEQCH10 被转换，结果存放在 ADC_DR7~ADC_DR10 中。第三次启动转换时，SEQCH11，SEQCH0~SEQCH4 被转换，结果存放在 ADC_DR11，ADC_DR0~ADC_DR4 中。此时，三段转换完成，即一个采样序列转换完成。下一次硬件启动转换时，将重新从第一段开始转换，即依次转换 SEQCH0~SEQCH6 通道。

11.3.8 连续转换

当 ADC_CFGR1 寄存器中的 CONT 位为 1 时，ADC 工作在连续转换模式下，软件置位 ADSTART 或者接收到硬件触发，ADC 开始进行转换，转换完一个序列便进行此序列的下一次转换，一直循环此序列转换。连续转换模式下一个采样序列只能有一个段，即一个采样序列最多由 12 个模拟通道组成，序列的通道数由 ADC_CR 寄存器中的 SGCHN1[3:0]位段决定，而 ADC_CR 中的 SGCHN2[3:0]、SGCHN3[3:0]和 SGCHN4[3:0]在此模式下无效，ADC_CR 中的 SGN[1:0]位也始终为“00”。

假如有 4 个外部模拟通道需要组成一个序列进行转换，依次分别为 ADC_IN1，ADC_IN0，ADC_IN5，ADC_IN7。因此需要设置 ADC_CR 中的 SGCHN1[3:0]=3，ADC_SEQCHSELR1 寄存器中的 SEQCH0=1，SEQCH1=0，SEQCH2=5，SEQCH3=7，转换开始后，ADC_IN1、ADC_IN0、ADC_IN5、ADC_IN7 的转换结果将依次存放在 ADC_DR0、ADC_DR1、ADC_DR2、ADC_DR3 中。该示例中，各关键信号的时序关系如图所示：

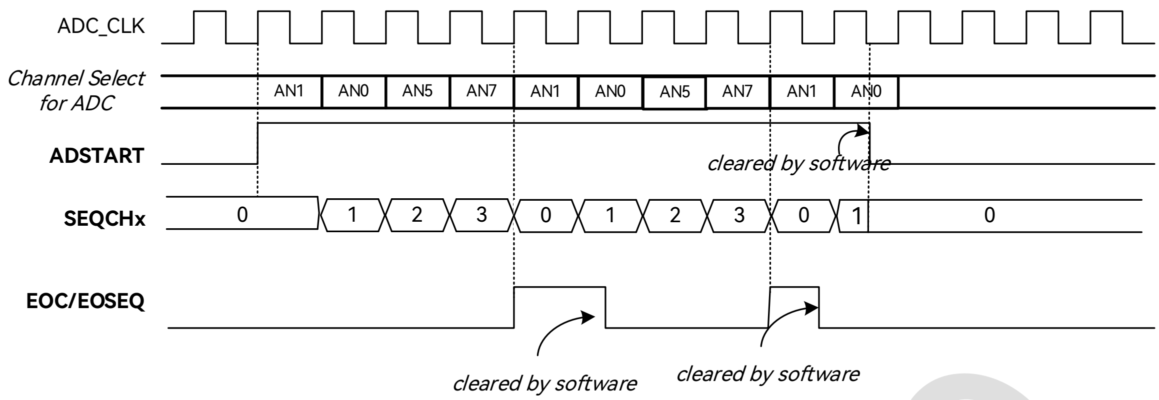


图 11-5：连续转换方式波形示例图

11.3.9 启动转换（ADSTART）

ADSTART 位由软件进行置 1，表示 ADC 开始转换或等待转换信号（硬件触发信号）。

当 EXTEN[1:0]=0x0 时，表示 ADC 由软件触发，此时对 ADSTART 位写 1，ADC 立即开始转换。当 EXTEN[1:0] ≠0x0 时，表示 ADC 由硬件触发，此时需要先对 ADSTART 位写 1，然后在 EXTSEL[1:0] 位选择的边沿发生硬件触发时，ADC 开始转换。

ADSTART 位也用于表明目前 ADC 转换操作是否正在进行。当 ADSTART=0，表明 ADC 处于空闲时，可重新配置 ADC。

ADSTART 位由硬件清零。当 ADC 在单次转换模式并且是由软件触发时，段转换完成后（EOSG=1），硬件会清除 ADSTART 位。任何转换模式和触发条件下，当软件对 ADSTP 位写 1 时，硬件会中止当前正在进行的转换并对 ADSTART 清零，同时也会将 ADSTP 位清零。

注 1：在连续转换模式（CONT=1）下，ADSTART 位不能由 EOSG 引发的硬件清除，其原因是自动重新开始序列转换。

注 2：当 ADC 在单次转换模式并且是硬件触发时，当 EOSG 标志置 1 后，ADSTART 不会被硬件清零。这是为了避免软件重新设置 ADSTART 位而导致错过硬件触发事件。

11.3.10 停止当前转换（ADSTP）

软件设置 ADC_CR 寄存器中的 ADSTP=1 可以停止当前正在进行的转换。

这会复位 ADC 的操作并让 ADC 进入空闲状态，为下次转换作好准备。

当 ADSTP 由软件置 1，任何当前进行的转换中止且转换结果丢弃（当前的转换不更新到 ADC_DRx 寄存器)。

一旦结束该过程，ADSTP 和 ADSTART 位都由硬件清 0。当开启下一次转换，会按配置好的序列从头开始转换。

11.3.11 数据寄存器与数据对齐

在每个模拟通道转换结束后，转换的结果数据被存放到相应的 16 位 ADC_DRx 数据寄存器中。

ADC_DRx 数据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 位用于选择数据存储的对齐方式，数据可选为右对齐 (ALIGN=0) 或左对齐 (ALIGN=1)。不同的对齐方式与分辨率下数据在 ADC_DRx 寄存器 DATA[15:0] 中的保存方式如下表所示。

表 11-2：数据对齐与分辨率

<table>
<tr><th>ALIGN</th><th>RES</th><th>B 15</th><th>B 14</th><th>B 13</th><th>B 12</th><th>B 11</th><th>B 10</th><th>B9</th><th>B8</th><th>B7</th><th>B6</th><th>B2</th><th>B4</th><th>B3</th><th>B2</th><th>B1</th><th>B0</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="4">0x00</td><td colspan="12">DATA[11:0]</td></tr>
<tr><td>1</td><td colspan="6">0x00</td><td colspan="10">DATA[9:0]</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td colspan="12">DATA[11:0]</td><td colspan="4">0x00</td></tr>
<tr><td>1</td><td colspan="10">DATA[9:0]</td><td colspan="6">0x00</td></tr>
</table>

11.3.12 数据溢出

ADC 溢出标志（OVR）表示一个数据溢出事件，当转换好的数据未被 CPU 及时读取，新的转换数据已经有效时，会发生 ADC 溢出事件。

每个结果寄存器 ADC_DRx 都对应一个内部的溢出标志位，当转换的数据保存到 ADC_DRx 时，对应的内部溢出标志位置 1，软件读取此结果寄存器时，标志位清零。

当一个新的转换完成时，对应的结果寄存器 ADC_DRx 上一次的数据未被读走，此时对应的内部溢出标志位为 1，则在 ADC_ISR 寄存器中置位 OVR 标志。如果在 ADC_IER 中设置了 OVRIE 位，就会产生一个 OVR 中断。

当发生溢出时，ADC 继续工作，并继续转换，除非软件停止转换（通过设置 ADC_CR 寄存器中的 ADSTP 为 1）。

OVR 标志软件写 1 清零。

通过对 ADC_CFGR1 寄存器中的 OVRMOD 位编程，可配置当发生溢出事件时的数据是被保留还是被覆盖：

- OVRMOD=0
  - 溢出事件发生，数据寄存器(ADC_DRx)的值被保留不被覆盖：老数据被保持，新的转换数据丢弃。若 OVR 保持为 1，则后续的转换会被执行，但结果需要根据内部的溢出标志位来决定是丢弃还是保存到 ADC_DRx。如果当前 ADC_DRx 的内部溢出标志位为 1，那么当次转换结果丢弃，如果当前 ADC_DRx 的内部溢出标志位为 0，那么当次转换结果保存到 ADC_DRx 中。假如 ADC_DR5 对应的内部溢出标志位为 1，ADC_DR6 对应的内部溢出标志位为 0，当下一次转换准备存储到 ADC_DR5 时，OVR 标志位会置 1 且当次转换结果不会存储到 ADC_DR5。如果 OVR 保持为 1，下一个转换结果准备存储到 ADC_DR6，当次转换结果会存储到 ADC_DR6 中。
- OVRMOD=1
  - 数据寄存器的值被最后的转换结果覆盖，之前未读取的数据将丢失。若 OVR 保持为 1，则后续的转换会被执行且数据寄存器（ADC_DRx）一直保持最后转换结果。

11.3.13 外部触发转换和触发极性（EXTSEL，EXTEN）

一次转换或一个序列的转换可由软件或外部事件（例如：定时器捕获）触发。若 EXTEN[1:0] ≠00b，则外部事件根据所选择极性可以用于触发转换。当软件设置 ADSTART=1 时，触发选择生效。

当正在进行 ADC 转换时，任何硬件触发都会被忽略。

表 11-3：配置触发极性

EXTEN[1:0]	源
00	触发检测禁止
01	在上升沿时检测
10	在下降沿时检测
11	在上升沿及下降沿都检测

注：只有 ADC 不在转换时（ADSTART=0），才允许修改外部触发极性。

软件触发事件可由设置 ADC_CR 寄存器中的 ADSTART 位来产生。

表 11-4：外部触发源

EXTSEL[2:0]	名称	源
000	TRG0	MCP_CMP
001	TRG1	
010	TRG2	
011	TRG3	
100	TRG4	
101	TRG5	
110	TRG6	TIM18_TRGO
111	TRG7	

注：只有 ADC 不在转换时（ADSTART=0），才允许修改外部触发源选择。

11.3.14 调试信号输出

设置 ADC_CR 寄存器中的 ADDBEN 位为 1，可以将 ADC_ISR 寄存器中的 BUSY 位通过 ADC_DB 引脚输出。当 ADC 处于转换中，BUSY 位为 1，ADC_DB 引脚输出高电平，当 ADC 转换完成或 ADC 模块未启动，BUSY 位为 0，ADC_DB 引脚输出低电平。

此功能主要用于调试时使用，方便与通过示波器或其他工具准确的了解 ADC 的工作状态。

11.3.15 模拟窗口看门狗

AWD 模拟看门狗的功能由 ADC_CFGR1 寄存器中的 AWDEN 位置位来开启。它可用于监测所选的模拟通道保持在配置电压范围（window）内，如下图所示。

如果模拟看门狗被使能，则在 ADC_CFGR1 寄存器的 AWDCH[4:0]位中指定的模拟通道转换完

成时将转换结果与 ADC_HTR 和 ADC_LTR 的值比较，ADC_ISR 寄存器中的 AWDHTF 位和 AWDLTF 位指示比较的结果。若 ADC 转换的模拟电压低于阈值下限或高于阈值上限，模拟看门狗的上限状态标志 AWDHTF 或下限标志位 AWDLTF 被置位。高低阈值在 ADC_HTR 和 ADC_LTR 寄存器中设置。若置位 ADC_IER 寄存器中的 AWDIE 位，无论是 AWDHTF 还是 AWDLTF 置位都将触发中断。

AWDHTF 和 AWDLTF 标志位，软件可以写 1 清零。

若转换分辨率低于 12 位（RES 位选择），则阈值寄存器中的低位必须保持为 0，因为内部数据比较都是按照全 12 位（左对齐）的方式进行比较的。

表 11-5：模拟看门狗比较操作

分辨率 RES	模拟看门狗比较值		说明
	原始转换数据，左对齐	阈值	
00：12 位	DATA[11:0]	LT[11:0]，HT[11:0]	-
01：10 位	DATA[11:2]+00B	LT[11:0]，HT[11:0]	用户必须配置 LT[1:0]=00B, HT[1:0]=00B

注：看门狗比较，发生在对齐处理前，对原始转换数据进行比较。

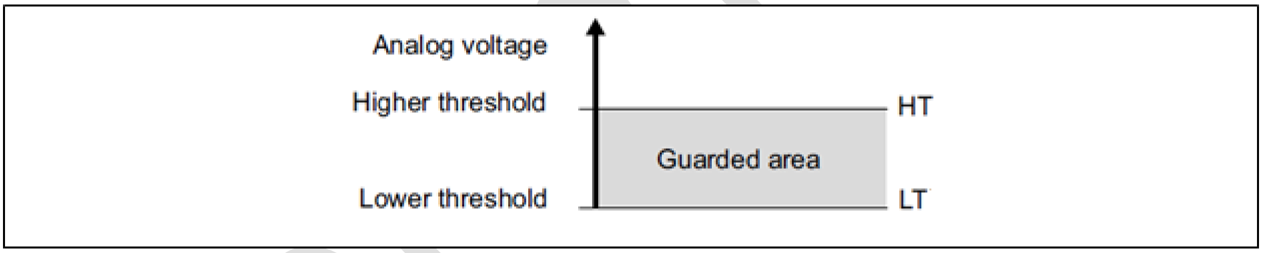


图 11-6：模拟看门狗保护区

11.3.15.1 看门狗通道选择

由 ADC_CFGR1 寄存器中的 AWDCH[4:0]位选择模拟看门狗监控哪一个外部模拟通道。

11.3.16 温度传感器和内部参考电压

温度传感器可以用来测量器件的结温（T_J）。

温度传感器内部连接 ADC_IN16 输入通道，用于将传感器的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表中指定的最小 T_{S_temp} 值。不使用时可将传感器置于下电模式。

温度传感器输出电压随温度呈线性变化，但由于工艺的不同，其特性在不同芯片之间可能会有

很大的变化。为了提高温度传感器的精度（特别是绝对温度测量），在出厂测试时，会测量每颗芯片的校准值，并存储在系统存储区（system memory）。有关其他信息，请参阅特定芯片数据手册。

内部电压参考（VREFINT）提供一个稳定的（bandgap）电压输出，提供给 ADC 和模拟比较器。VREFINT 内部连接到 ADC_IN17 输入通道。VREFINT 的精确电压，在出厂测试时，为每个芯片单独测量，并存储在系统存储区（system memory）。

下图展示了温度传感器、内部电压基准和 ADC 之间的连接框图。

TSEN 位置位，ADC_IN16（温度传感器）的转换使能，VREFEN 位置位，ADC_IN17（VREFINT）的转换使能。

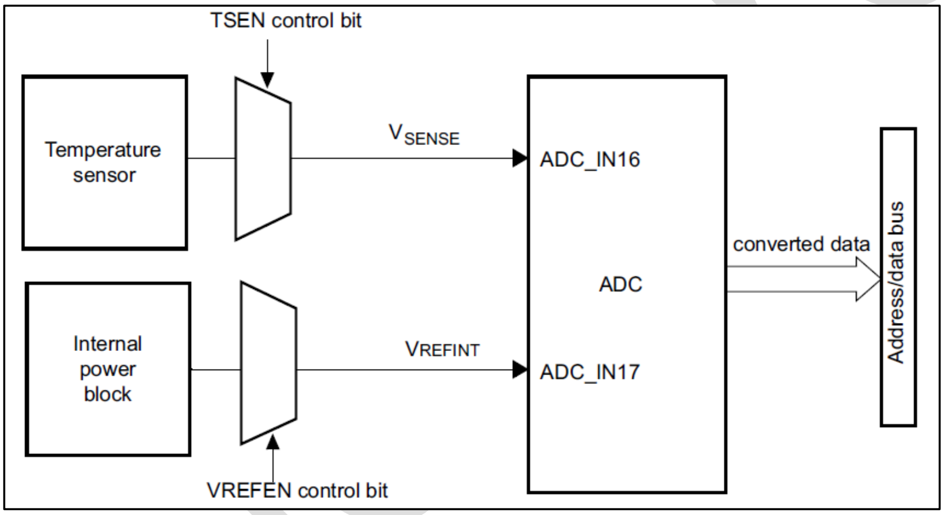


图 11-7：温度传感器和 VREFINT 通道框图

11.3.16.1 读取温度流程

1. 选择 ADC1_IN16 输入通道

   选择适当的采样时间，参考芯片数据手册 T_{S_temp} 参数。

   ADC_CCR 寄存器中设置 TSEN 位用来唤醒从断电模式下的温度传感器，并等待其稳定时间（t_{START}）。

2. 置位 ADC_CR 寄存器中的 ADSTART 位（也可用外部触发）来启动 ADC 转换。

3. 从 ADC_DRx 寄存器中读取转换数据 ADC_Data。

   计算温度：T_{ADC_Data}(℃) = （VDDA / FULL_SCALE x ADC_Data − b）/ k

4. 从 sysrom 出厂区读出 V_{ADC_Temp1}、Sens_Temp1 为出厂校准值；详细参考数据手册描述

   计算校准点温度（芯片）：$T_{ADC\_Temp1}(℃) = (V_{ADC\_Temp1} - b) / k$

   计算校准点温度（环境）：$T_{Sens\_Temp1}(℃) = 0.0625 \times$ Sens_Temp1

   计算温度 offset：$T_{offset} = T_{Sens\_Temp1} - T_{ADC\_Temp1}$

5. 修正实际温度：

   实际温度 T（℃）$= T_{ADC\_DATA} + T_{offset}$

   其中，

   - VDDA_3v3 为出厂校准时模拟电源，电压= 3.3 V
   - VDDA 为实际应用中模拟电源，由实际应用确定
   - FULL_SCALE = 4095
   - ADC_Data，实际 ADC 转换值
   - V_{ADC_Data} = VDDA / FULL_SCALE x ADC_DATA
   - V_{ADC_Temp1} 为出厂测试的温度传感器电压输出，单位 mV，参见数据手册描述
   - Sens_Temp1 为出厂测试值，为测试 V_{ADC_Temp1} 时的常温温度值，参见数据手册描述
   - k 为温度传感器电压平均斜率 Avg_Slope = 0.0043v/℃，以实际数据手册为准
   - b 为温度传感器电压 vs 温度一次曲线常数项（截距），b=1.063 V

   若实际应用 VDDA=VDDA_3v3 为 3.3V，公式可以简化为：

$$\text{实际温度 T（℃）} = ((V_{ADC\_Data}-V_{ADC\_Temp1})/Avg\_slope) + T_{Sens\_Temp1}$$

注：传感器从断电模式下唤醒时到能正确输出 VSENSE 的值要有一个启动时间，另 ADC 从上电后启动也有一个启动时间，若要最小化这个延时，则需要在同一时间置位 ADEN 和 TSEN 位。

11.3.16.2 使用内部参考电压计算实际 VDDA 电压

MCU 的 VDDA 电源电压可能会发生变化或不能精确知道。内部基准电压的出厂测试数据是在常温条件下通过 ADC 生产测试获得，此数据可以用于评估实际 VDDA 电压值变化。

实际 VDDA 电压计算如下：

$$VDDA = FULL\_SCALE / V_{REFINT\_DATA} \times V_{REFINT}$$

其中：

- V_{REFINT} 为 V_{REFINT} 出厂测试值
- V_{REFINT_DATA} 为实际 $V_{REFTINT}$ 通道 ADC 转换值。
- FULL_SCALE 为 ADC 转换的最大数字量化值。举例，12-bit 分辨率为$2^{12}-1=4095$。

11.3.16.3 通过相对电源电压的 ADC 测量计算绝对电压

ADC 的转换参考源为 VDDA 电源，对一个通道进行 ADC 转换，得到的数字量化值反映通道电压和 VDDA 电压的线性关系。

应用中，若 VDDA 电压已知，ADC 转换值右对齐，可以用下面公式计算绝对电压：

$$V_{CHANNEL} = VDDA / FULL\_SCALE \times ADC\_DATA$$

应用中，若 VDDA 电压未知，可以通过章节<使用内部参考电压计算实际 VDDA 电压>中 VDDA 和 VREFINT 关系式替换，结果如下：

$$V_{CHANNEL} = (V_{REFINT}/ V_{REFINT\_DATA}) \times ADC\_DATA$$

其中：

- V_{REFINT} 为 V_{REFINT} 出厂测试值。
- V_{REFINT_DATA} 为实际 $V_{REFTINT}$ 通道 ADC 转换值。
- ADC_DATA 为通道 ADC 测量值，右对齐。
- FULL_SCALE 为 ADC 转换的最大数字量化值。举例，12bit 分辨率为 212-1=4095，8bit 分辨率为 28-1=255。

注：如果 ADC 测量使用的不是 12 位右对齐的输出格式，那么在计算之前，所有参数必须首先转换为兼容格式。

11.3.17 ADC 中断

ADC 中断可由以下任一事件产生：

- ADC 上电，ADC 准备就绪（ADRDY 标志）

- 任何一个外部模块通道转换完成（EOC 标志）
- 任何一段转换结束（EOSG 标志）
- 采样序列转换结束（EOSEQ 标志）
- 当模拟看门狗检测发生（AWD 标志）
- 当数据溢出发生（OVR 标志）

独立的中断使能位用于灵活设置 ADC 中断。

表 11-6：ADC 中断

中断事件	事件标志	使能控制位
ADC 准备就绪（ready）	ADRDY	ADRDYIE
单个模块通道转换完成	EOC	EOCIE
段转换结束	EOSG	EOSGIE
采样序列转换结束	EOSEQ	EOSEQIE
模拟看门狗状态置位	AWDHTF 或 AWDLTF	AWDIE
溢出（overrun）	OVR	OVRIE

11.4 相关寄存器

11.4.1 ADC 中断标志和清除寄存器（ADC_ISR）

偏移地址：0x00

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								AWDHTF	AWDLTF	BUSY	OVR	EOSEQ	EOC	EOSG	ADRDY
								rc_w1	rc_w1	r	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位	名称	描述
31:8	RSV	保留位，固定读为 0
7	AWDHTF	模拟看门狗上限比较中断标志位 该位可由软件写 1 清零 0：无上限中断发生，AWDCH[4:0]位段所指定的模拟通道转换结果的最

位	名称	描述
		近一次更新值小于 ADC_HTR 中的值 1：发生上限中断，AWDCH[4:0]位段所指定的模拟通道转换结果的最近一次更新值大于 ADC_HTR 中的值
6	AWDLTF	模拟看门狗下限比较中断标志位 该位可由软件写 1 清零 0：无下限中断发生，AWDCH[4:0]位段所指定的模拟通道转换结果的最近一次更新值大于 ADC_LTR 中的值 1：发生下限中断，AWDCH[4:0]位段所指定的模拟通道转换结果的最近一次更新值小于 ADC_LTR 中的值
5	BUSY	ADC 忙信号标志 该位只能由硬件置 1 或清零，软件只读。当 ADC 启动转换时，此位由硬件置 1，当 ADC 转换完成，此位由硬件清零。 0：ADC 不在转换状态或转换已完成 1：ADC 正在转换
4	OVR	ADC 溢出 当溢出产生时该位由硬件置位，即新的转换结束且内部有数据未被读取。该位软件写 1 清零。 0：无溢出事件产生（或该事件标志已被软件确认并清零） 1：溢出发生
3	EOSEQ	序列转换结束标志 序列转换结束后，该位由硬件置位。软件对该位写 1 清零。 0: 序列转换未完成（或该事件标志已被软件确认并清零） 1: 序列转换完成
2	EOC	通道转换结束标志 单个通道转换结束后，该位由硬件置位。软件对该位写 1 清零。 0: 通道转换未完成（或该事件标志已被软件确认并清零） 1: 通道转换完成
1	EOSG	段转换结束标志 段转换结束后，该位由硬件置位。软件对该位写 1 清零。 0: 段转换未完成（或该事件标志已被软件确认并清零） 1: 段转换完成
0	ADRDY	ADC 就绪 在 ADC 使能后（ADEN=1）且 ADC 达到准备接受转换请求的状态时，该位由硬件置位。该位软件写 1 清零。 0: ADC 未准备好（或该事件标志已被软件确认并清零） 1: ADC 已准备好开始转换

11.4.2 ADC 中断使能寄存器（ADC_IER）

偏移地址：0x04

复位值：0x0000 0000

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

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

位	名称	描述
31:8	RSV	保留位，固定读为 0
7	AWDIE	模拟看门狗中断使能，软件置位和清零 0：模拟看门狗中断禁止 1：模拟看门狗中断使能 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写该位。
6:5	RSV	保留位，固定读为 0
4	OVRIE	溢出中断使能，软件置位和清零 0：溢出中断关闭 1：溢出中断开启。当 OVR 置位时产生中断。 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写该位。
3	EOSEQIE	序列转换结束中断使能，软件置位和清零 0: EOSEQ 中断关闭 1: EOSEQ 中断开启。当 EOSEQ 置位时产生中断。 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写该位。
2	EOCIE	通道转换结束中断使能，软件置位和清零 0: EOC 中断关闭 1: EOC 中断开启。当 EOC 置位时产生中断。 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写该位。
1	EOSGIE	段转换结束中断使能，软件置位和清零 0: EOC 中断关闭 1: EOC 中断开启。当 EOC 置位时产生中断。 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写该位。
0	ADRDYIE	ADC 就绪中断使能，软件置位和清零

位	名称	描述
		0: ADRDY 中断关闭 1: ADRDY 中断开启。当 ADRDY 置位时产生中断。 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写该位。

11.4.3 ADC 控制寄存器（ADC_CR）

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAL	Res.			SGCHN4[3:0]				SGCHN3[3:0]				SGCHN2[3:0]			
rs				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SGCHN1[3:0]				Res.		SGN[1:0]		ADDBEN	Res.		ADSTP	Res.	ADSTART	ADDIS	ADEN
rw						rw		rw			rs		rs	rs	rs

位	名称	描述
31	ADCAL	ADC 校准 该位由软件置位来启动 ADC 校准。当校准完成后，由硬件清零。 0：校准完成。 1：写 1 开始校准 ADC，读为 1 时意味着校准进行中。 注：只有在 ADC 关闭情况下（ADCAL=0，ADSTART=0，ADSTP=0，ADDIS=0 和 ADEN=0），才允许软件置位 ADCAL。
30:28	RSV	保留位，固定读为 0
27:24	SGCHN4[3:0]	第四段的通道数量（segment channel number) 采样序列中第四段的通道数量(SGCHN4[3:0]+1)，此位值最大不超过 11
23:20	SGCHN3[3:0]	第三段的通道数量（segment channel number) 采样序列中第三段的通道数量(SGCHN3[3:0]+1)，此位值最大不超过 11
19:16	SGCHN2[3:0]	第二段的通道数量（segment channel number) 采样序列中第二段的通道数量(SGCHN2[3:0]+1)，此位值最大不超过 11
15:12	SGCHN1[3:0]	第一段的通道数量（segment channel number) 采样序列中第一段的通道数量(SGCHN1[3:0]+1)，此位值最大不超过 11
11:10	RSV	保留位，固定读为 0
9:8	SGN[1:0]	组成采样序列的段数量（segment number） 00：一段 01：两段 10：三段

位	名称	描述
		11：四段 注：此位只在单次模式下有效
7	ADDBEN	采样输出位: 0：禁止通过 ADC_DB 引脚输出 ADC_SR 中的 BUSY 状态 1：使能通过 ADC_DB 引脚输出 ADC_SR 中的 BUSY 状态
6:5	RSV	保留位，固定读为 0
4	ADSTP	ADC 停止转换命令 该位由软件置位来停止和放弃正在进行中的转换。 当转换停止生效时，该位由硬件清零，同时 ADC 准备好接受新的转换命令。 0: 没有正在进行的 ADC 停止转换命令。 1: 写 1 停止 ADC 转换，读为 1 时表明 ADSTP 命令正在执行中。 注：只有当 ADSTART=1 和 ADDIS=0（ADC 开启且可能正在转换，且无禁止 ADC 的挂起请求 ），软件才能对该位进行置位。
3	RSV	保留位，固定读为 0
2	ADSTART	ADC 通道序列启动 AD 转换请求位 该位由软件置 1，硬件清零。 - 当选择为软件触发时，软件置 1 后立即开始转换，转换完成由硬件清零。 - 当选择为硬件触发时，软件置 1 后，需要等待硬件触发信号才开始转换，转换完成后硬件不对该位清零。 - 正在转换时，软件对 ADSTP 位写 1，硬件会清除 ADSTART 位，同时 ADSTP 位也由硬件清零。 0：未进行转换或转换已完成。 1：正在进行转换或等待转换开始信号（硬件触发信号） 注：只有当 ADEN=1 和 ADDIS=0（ADC 开启且可能正在转换，但无禁止 ADC 挂起的请求）时才允许软件设置 ADSTART 位。
1	ADDIS	ADC 关闭命令 该位由软件置位来禁止 ADC（ADDIS 命令）并让 ADC 处于掉电状态（关断状态）。一旦 ADC 关闭生效（ADEN 同时被硬件清零）后由硬件清零该位。 0: 无 ADDIS 命令进行中 1: 写 1 关闭 ADC。读为 1 时表明 ADDIS 命令正在执行中。 注：只有当 ADEN=1 和 ADSTART=0（确保无进行中的转换）时，才允许软件置位 ADDIS 位。
0	ADEN	ADC 使能命令 该位由软件置位来使能 ADC。一旦 ADRDY 标志置为 1 时，表明 ADC 可供使用。执行 ADDIS 命令后，ADC 关断，并由硬件清零该位。

位	名称	描述
		0：ADC 关闭（关断状态） 1：写 1 来使能 ADC 注：只有在 ADC_CR 寄存器所有位为 0 （ADCAL=0，ADSTP=0，ADSTART=0，ADDIS=0 和 ADEN=0）的情况下，软件才能置位 ADEN 位。

11.4.4 ADC 配置寄存器 1（ADC_CFGR1）

偏移地址：0x0C

复位值：0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>Res.</td><td colspan="5">AWDCH[4:0]</td><td colspan="2">Res.</td><td>AWDEN</td><td colspan="7">Res.</td></tr>
<tr><td></td><td colspan="5">rw</td><td colspan="2"></td><td>rw</td><td colspan="7"></td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">Res.</td><td>CONT</td><td>OVRMOD</td><td colspan="2">EXTEN[1:0]</td><td>Res.</td><td colspan="3">EXTSEL[2:0]</td><td>ALIGN</td><td>Res.</td><td>RES</td><td colspan="3">Res.</td></tr>
<tr><td colspan="2"></td><td>rw</td><td>rw</td><td colspan="2">rw</td><td></td><td colspan="3">rw</td><td>rw</td><td></td><td>rw</td><td colspan="3"></td></tr>
</table>

位	名称	描述
31	RSV	保留位，固定读为 0
30:26	AWDCH[4:0]	模拟看门狗通道选择，软件置位和清零 设置模拟看门狗监视的 ADC 的模拟通道。 00000：模拟通道 ADC_IN0 00001：模拟通道 ADC_IN1 … 10010：模拟通道 ADC_IN18 10011：模拟通道 ADC_IN19 其它值：保留，不可使用 注 1：只有当 ADSTART=0 时（确定无进行中的转换）才允许改写这些位。
25:24	RSV	保留位，固定读为 0
23	AWDEN	模拟看门狗使能位，软件置位和清零 0：模拟看门狗关闭 1：模拟看门狗开启 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写该位。
22:14	RSV	保留位，固定读为 0

位	名称	描述
13	CONT	单次/连续转换模式，软件置位和清零 若该位置位，转换为连续模式直到该位被清零。 0：单次转换模式 1：连续转换模式 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写这些位。
12	OVRMOD	溢出管理模式，软件置位和清零，来配置溢出管理 0: 当检测到溢出事件时，ADC_DRx 寄存器保持为老数据 1: 当检测到溢出事件时，ADC_DRx 寄存器用最后一次的转换数据覆盖 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写这些位。
11:10	EXTEN[1:0]	外部触发使能和极性选择，软件置位和清零 00：硬件触发检测关闭（可由软件启动转换） 01：在上升沿进行硬件触发检测 10：在下降沿进行硬件触发检测 11：在上升和下降沿都进行硬件触发检测 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写这些位。
9	RSV	保留位，固定读为 0
8:6	EXTSEL[2:0]	外部触发选择 这些位用于选择触发 ADC 转换的外部事件：参考表：外部触发源 000: TRG0（MCP_CMP） 001: TRG1（保留） 010: TRG2（保留） 011: TRG3（保留） 100: TRG4（保留） 101: TRG5（保留） 110: TRG6（TIM18_TRGO） 111: TRG7（保留） 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写这些位。
5	ALIGN	数据对齐，软件置位和清零，用来选择数据的左对齐或右对齐。 0：右对齐

位	名称	描述
		1：左对齐 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写这些位。
4	RSV	保留位，固定读为 0
3	RES	数据分辨率，软件置位和清零，用于选择 ADC 转换的数据分辨率。 0: 12 位 1: 10 位 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写这些位。
2:0	RSV	保留位，固定读为 0

11.4.5 ADC 配置寄存器 2（ADC_CFGR2）

偏移地址：0x10

复位值：0x0000 0000

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

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

位	名称	描述
31:30	CKMODE[1:0]	ADC 时钟模式，软件置位和清零，定义模拟 ADC 时钟 00：ADCCLK1（异步时钟模式） 01：ADCCLK2（异步时钟模式） 10：PCLK/2（同步时钟模式） 11：PCLK/4（同步时钟模式） 在所有的同步时钟模式中，从定时器触发器到开始转换的延迟没有抖动。 注：只有在 ADC 关闭情况下（ADCAL=0, ADSTART=0, ADSTP=0, ADDIS=0 和 ADEN=0)，才允许软件改写此位。
29:0	RSV	保留位，固定读为 0

11.4.6 采样时间寄存器（ADC_SMPR）

偏移地址：0x014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				SMPSEL19	SMPSEL18	SMPSEL17	SMPSEL16	SMPSEL15	SMPSEL14	SMPSEL13	SMPSEL12	SMPSEL11	SMPSEL10	SMPSEL9	SMPSEL8
				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		

位	名称	描述
31:28	RSV	保留位，固定读为 0
27:8	SMPSELx	通道 x（x=0~19）采样时间选择位 此位由软件置位或清零用于选择相应通道的采样时间 0：通道 x 采样时间使用 SMP1[2:0]定义的时间 1：通道 x 采样时间使用 SMP2[2:0]定义的时间 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写此位。
7	RSV	保留位，固定读为 0
6:4	SMP2[2:0]	采样时间选择 2，软件改写。 000: 9.5 ADC 时钟周期 001: 13.5 ADC 时钟周期 010: 25.5 ADC 时钟周期 011: 41.5 ADC 时钟周期 100: 55.5 ADC 时钟周期 101: 71.5 ADC 时钟周期 110: 98.5 ADC 时钟周期 111: 325.5 ADC 时钟周期 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写这些位。
3	RSV	保留位，固定读为 0

位	名称	描述
2:0	SMP1[2:0]	采样时间选择 1，软件改写。 000: 9.5 ADC 时钟周期 001: 13.5 ADC 时钟周期 010: 25.5 ADC 时钟周期 011: 41.5 ADC 时钟周期 100: 55.5 ADC 时钟周期 101: 71.5 ADC 时钟周期 110: 98.5 ADC 时钟周期 111: 325.5 ADC 时钟周期 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写这些位。

11.4.7 ADC 上限阈值寄存器（ADC_HTR）

偏移地址：0x20

复位值：0x0000 0FFF

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

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

位	名称	描述
31:12	RSV	保留位，固定读为 0
11:0	HT[11:0]	ADC 模拟看门狗上限比较阈值 当 ADC_CFGR1 寄存器中 AWDCH[4:0]位段指定的模拟通道转换完成时，转换结果立刻与 ADC_HTR 中的值比较，若大于 ADC_HTR 中的值则令 ADC_ISR 中的 AWDHTF 置 1，AWDHTF 位将保持为 1 直到软件将其清除

11.4.8 ADC 下限阈值寄存器（ADC_LTR）

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.															

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

位	名称	描述
31:12	RSV	保留位，固定读为 0
11:0	LT[11:0]	ADC 模拟看门狗下限比较阈值 当 ADC_CFGR1 寄存器中 AWDCH[4:0]位段指定的模拟通道转换完成时，转换结果立刻与 ADC_LTR 中的值比较，若小于 ADC_LTR 中的值则令 ADC_ISR 中的 AWDLTF 置 1，AWDLTF 位将保持为 1 直到软件将其清除

11.4.9 ADC 通道选择寄存器 1（ADC_SEQCHSELR1）

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.		SEQCH5[4:0]					SEQCH4[4:0]					SEQCH3[4:0]			
		rw					rw					rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEQCH3[4:0]	SEQCH2[4:0]					SEQCH1[4:0]					SEQCH0[4:0]				
rw	rw					rw					rw				

位	名称	描述
31:30	RSV	保留位，固定读为 0
29:0	SEQCHx[4:0]	序列中第 x（x=0~5）转换通道选择 通道选择，软件改写，用来定义所要转换序列的通道。

位	名称	描述
		00000：输入通道 0 00001：输入通道 1 … 10010：输入通道 18 10011：输入通道 19 其它值：保留，不可使用 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写这些位。

11.4.10 ADC 通道选择寄存器 2（ADC_SEQCHSELR2）

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.		SEQCH(6+x)[4:0]													
		rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEQCH(6+x)[4:0]															
rw															

位	名称	描述
31:30	RSV	保留位，固定读为 0
29:0	SEQCH(6+x)[4:0]	序列中第 x（x=0~5）转换通道选择 通道选择，软件改写，用来定义所要转换序列的通道。 00000：输入通道 0 00001：输入通道 1 … 10010：输入通道 18 10011：输入通道 19 其它值：保留，不可使用 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写这些位。

11.4.11 ADC 结果寄存器 0（ADC_DR0）

偏移地址：0x40

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	DATA[15:0]	存放结果序列中通道 0 的采样数据

11.4.12 ADC 结果寄存器 1（ADC_DR1）

偏移地址：0x44

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	DATA[15:0]	存放结果序列中通道 1 的采样数据

11.4.13 ADC 结果寄存器 2（ADC_DR2）

偏移地址：0x48

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	DATA[15:0]	存放结果序列中通道 2 的采样数据

11.4.14 ADC 结果寄存器 3（ADC_DR3）

偏移地址：0x4C

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	DATA[15:0]	存放结果序列中通道 3 的采样数据

11.4.15 ADC 结果寄存器 4（ADC_DR4）

偏移地址：0x50

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	DATA[15:0]	存放结果序列中通道 4 的采样数据

11.4.16 ADC 结果寄存器 5（ADC_DR5）

偏移地址：0x54

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	DATA[15:0]	存放结果序列中通道 5 的采样数据

11.4.17 ADC 结果寄存器 6（ADC_DR6）

偏移地址：0x58

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	DATA[15:0]	存放结果序列中通道 6 的采样数据

11.4.18 ADC 结果寄存器 7（ADC_DR7）

偏移地址：0x5C

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	DATA[15:0]	存放结果序列中通道 7 的采样数据

11.4.19 ADC 结果寄存器 8（ADC_DR8）

偏移地址：0x60

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	DATA[15:0]	存放结果序列中通道 8 的采样数据

11.4.20 ADC 结果寄存器 9（ADC_DR9）

偏移地址：0x64

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	DATA[15:0]	存放结果序列中通道 9 的采样数据

11.4.21 ADC 结果寄存器 10（ADC_DR10）

偏移地址：0x68

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	DATA[15:0]	存放结果序列中通道 10 的采样数据

11.4.22 ADC 结果寄存器 11（ADC_DR11）

偏移地址：0x6C

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	DATA[15:0]	存放结果序列中通道 11 的采样数据

11.4.23 通用配置寄存器（ADC_CCR）

偏移地址：0x308

复位值：0x0000 0000

31:24	23	22	21:16
Res.	TSEN	VREFEN	Res.
	rw	rw	

15:0
Res.

位	名称	描述
31:24	RSV	保留位，固定读为 0
23	TSEN	温度传感器使能，软件置位和清零，打开/关闭温度传感通道。 0：温度传感通道关闭 1：温度传感通道开启 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写该位。
22	VREFEN	VREFINT 使能，软件置位和清零，打开/关闭 VREFINT 通道。 0: VREFINT 通道关闭 1: VREFINT 通道开启 注：只有当 ADSTART=0 时（确定无进行中的转换)才允许改写该位。
21:0	RSV	保留位，固定读为 0

12 电机控制（Motor Control PWM, MCP）

12.1 概述

电机控制 PWM 模可以输出 3 路互补或 6 路独立的 PWM，支持死区功能。

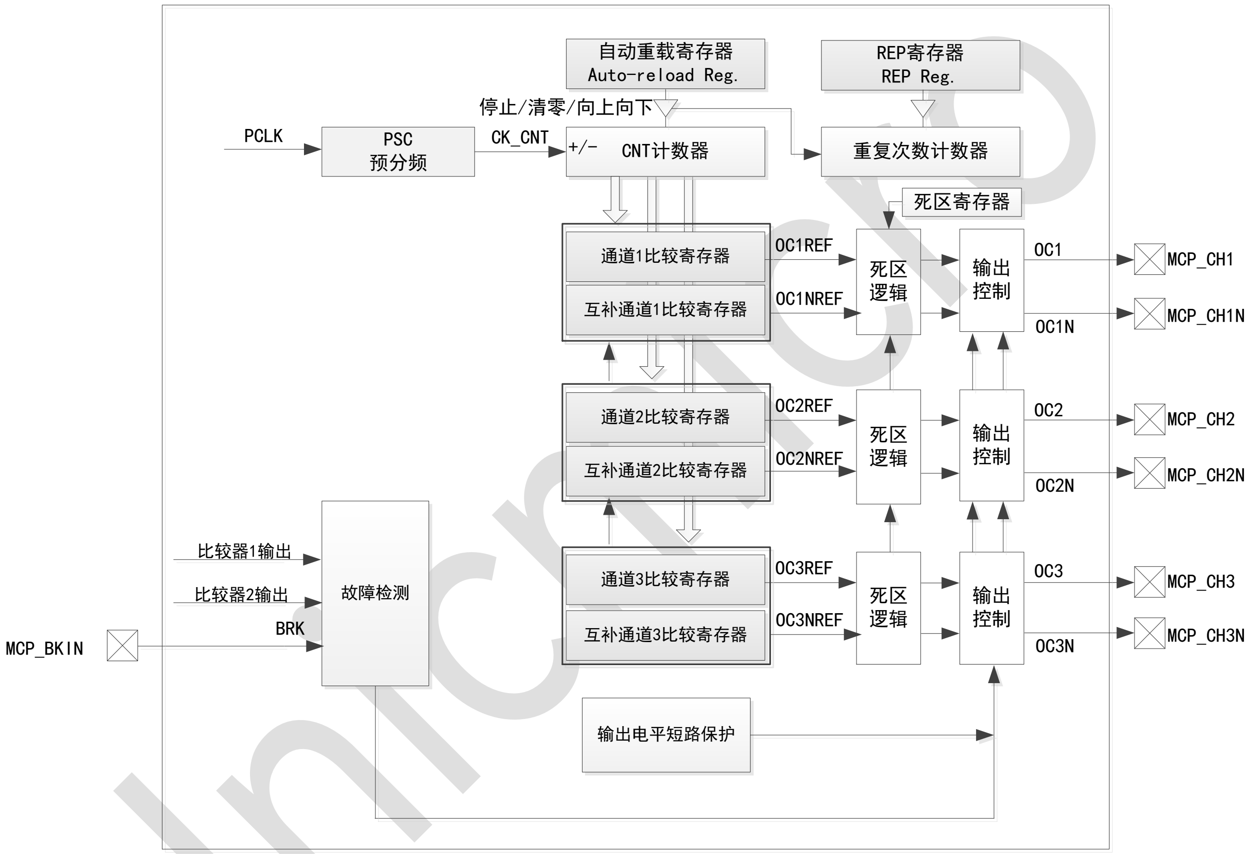


图 12-1：MCP 框图

12.2 主要特性

- 16 位时基计数器
- 计数器有边沿对齐计数、中心对齐计数和单次计数模式
- 输出波形分为互补模式与独立模式
- 互补模式下支持死区插入，并支持对称和非对称两种模式
- 提供 PWM 周期匹配、归零匹配和占空比比较匹配中断、故障保护中断

- 3 路互补 PWM 输出或 6 路独立 PWM 输出，输出电平可配置
- 支持故障检测，发生故障时可硬件关闭 PWM 输出，输出电平可配置
- 一个 PWM 周期内任意四个时刻触发 ADC

12.3 功能描述

12.3.1 时基单元

时基单元主要由一个 16 位的计数器 MCP_CNT、预分频器 MCP_PSC 和自动装载寄存器 MCP_ARR 组成。计数器有向上计数的边沿计数模式和向上、向下计数的中央对齐计数模式。计数器时钟由预分频器分频后得到。计数器寄存器 MCP_CNT、自动装载寄存器 MCP_ARR，预分频寄存器 MCP_PSC 可以由软件读写，即使计数器在运行时软件读写仍然有效。

计数器由预分频器的时钟输出 CK_CNT 驱动，只有设置了 MCP_CR1 寄存器中的 CEN 位后 CK_CNT 才有效。

自动重装载寄存器 MCP_ARR 的更新可以选择立即更新和固定时刻更新。当 MCP_CR1 寄存器中的 ARPE 位为 0 时，写入 MCP_ARR 寄存器的值会立即更新。当 MCP_CR1 寄存器中的 ARPE 位为 1 时，写入 MCP_ARR 寄存器的值会在计数器产生更新事件（中央对齐模式下则为下溢更新事件）时更新。

预分频寄存器 MCP_PSC 只能在计数器产生更新事件（中央对齐模式下则为下溢更新事件）时更新。

注:

1. 如果为边沿计数模式，写入 MCP_CNT 的初值大于等于 MCP_ARR 的值，则 MCP_CNT 先从 MCP_ARR 到 0，之后开始递增计数。
2. 如果为中心对齐模式，写入 MCP_CNT 的初值大于 MCP_ARR 的值，则 MCP_CNT 从 MCP_ARR 开始递减计数。
3. MCP_CNT 在模块停止时被清零一次，之后可以写。

12.3.2 计数器模式

时基单元中 16 位的计数器有 3 种工作方式，由 MCP_CR1 寄存器中的 CMS[1:0]位选择边沿对齐模式、中央对齐模式和单次计数模式。计数模式的修改需要在 MCP 模块使能之前。

12.3.2.1 边沿对齐计数模式

设置 MCP_CR1 寄存器中的 CMS[1:0]位段为 00，计数器将工作在边沿对齐计数模式下。此模式下，软件令 CEN 位置 1，PWM 计数器将从 MCP_CNT 值开始向上计数，直到与周期寄存器 MCP_ARR 匹配，然后计数器复位为 0 并继续向上计数，如此往复。计数器复位为 0 的时会产生一个更新事件。软件令 CEN 位清 0，计数器将停止计数并在下一个 PCLK 边沿复位为 0 并停止计数。如设置 MCP_ARR = 5，初始 MCP_CNT=0，则计数器在此模式下的工作如下图所示。

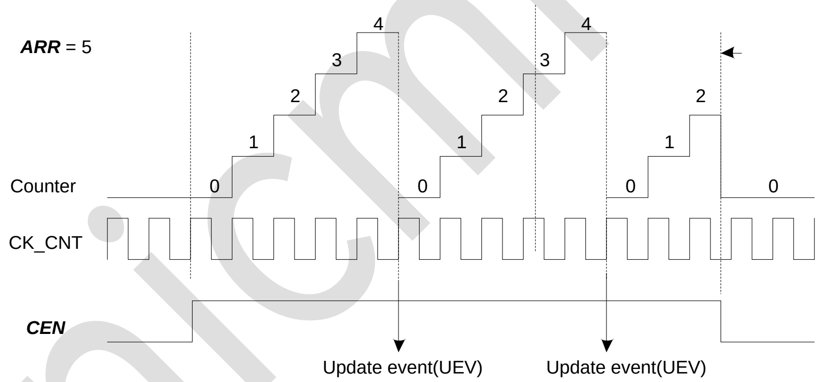


图 12-2：边沿对齐计数模式

12.3.2.2 中心对齐计数模式

设置 MCP_CR1 中 CMS[1:0]位段为 01，计数器将工作在中心对齐计数模式下。此模式下，软件令 CEN 位置 1，计数器将从 MCP_CNT 值开始向上计数，直到与周期寄存器 MCP_ARR 匹配，然后计数器开始向下计数直至归 0，如此往复。计数器向下计数至零时会产生下溢更新事件，计数器与周期寄存器匹配时会产生上溢更新事件。软件令 CEN 位清 0，计数器将停止计数并在下一个 PCLK 边沿复位为 0 并停止计数。如设置 MCP_ARR = 5，初始 MCP_CNT=0，则计数器在此模式下的工作如下图所示。

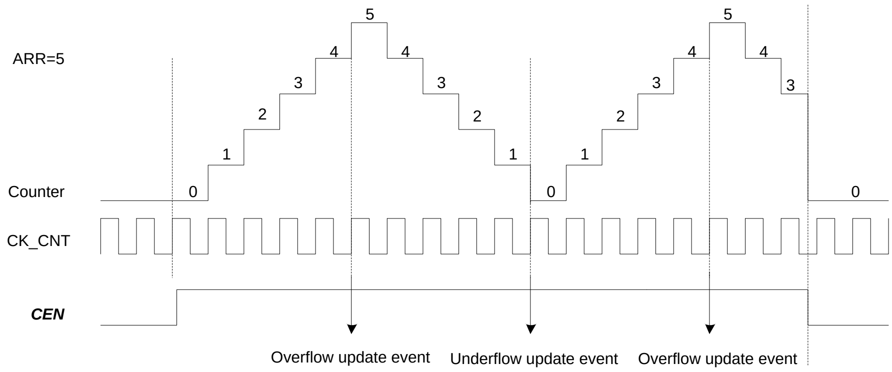


图 12-3：中心对齐计数模式

12.3.2.3 单次计数模式

设置 MCP_CR1 中 CMS[1:0]位段为 10，计数器将工作在单次计数模式下。此模式下，软件令 CEN 位置 1，计数器将从 MCP_CNT 值开始向上计数，直到与周期寄存器 MCP_ARR 匹配时，计数器复位为 0，同时 CEN 位将由硬件清 0，PWM 输出根据 MCP_CR1 寄存器中的 OPMS[1:0]位可以选择高阻状态、保持原来的输出电平、全为低电平或全为高电平。计数器复位为 0 时产生更新事件。

如果重复计数寄存器 MCP_RCR 值不为 0，则单次计数模式的 CEN 会在更新事件（中央对齐计数模式下的下溢更新事件）标志位 UIF 置起时被硬件清零。PWM 输出根据 MCP_CR1 寄存器中的 OPMS[1:0]位可以选择高阻状态、保持原来的输出电平、全为低电平或全为高电平。

计数器向上计数过程中若软件令 CEN 位清 0，则计数器将在下一个 PCLK 边沿复位为 0 并停止计数。如设置 MCP_ARR = 5，初始 MCP_CNT=0，则计数器在此模式下的工作如下图所示。

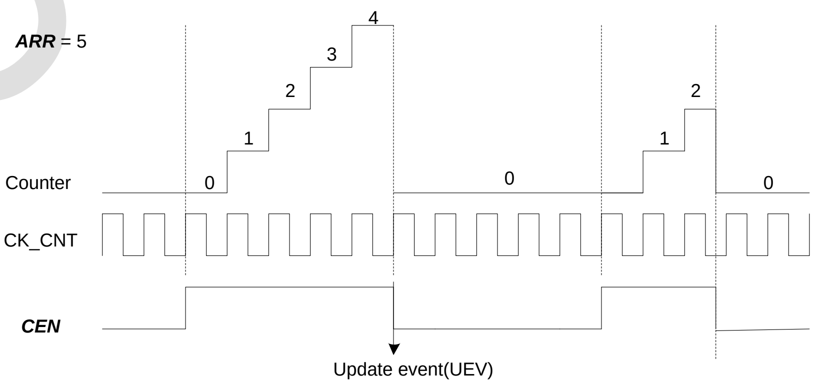


图 12-4：单次计数模式

12.3.3 重复计数

重复计数寄存器 MCP_RCR 是用来控制更新事件（包括上溢更新事件和下溢更新事件）、事件比较寄存器触发 ADC 的发生频率。当 MCP_RCR 寄存器的值为 0 时，更新事件和事件比较器触发 ADC 的频率不分频，即每个周期都会产生。当 MCP_RCR 寄存器的值不为 0 时，更新事件和事件比较器触发 ADC 的频率根据 MCP_RCR 的值进行分频产生，分频系数为 MCP_RCR+1。

假如计数器工作在边沿对齐计数模式下，MCP_RCR 寄存器的值为 0，在某段时间内共产生 100 次更新事件。那么相同条件下，如果 MCP_RCR 寄存器的值为 9，则相同时间内只会产生 10 次更新事件。

在不产生更新事件的计数器计数周期内，MCP_SR 寄存器中的 UIF 位和 OUIF 位也不会被置 1。

重复计数寄存器 MCP_RCR 只对更新事件和事件比较寄存器触发 ADC 有效，对占空比匹配不产生影响，即占空比匹配在计数器的每个周期都会发生，相应的标志位在匹配成功时也会由硬件置 1。

重复计数寄存器 MCP_RCR 只能在计数器产生更新事件（中央对齐模式下则为下溢更新事件）时更新。

12.3.4 PWM 输出模式

MCP 模块中包含有 3 个输出比较模块，对应于 3 对 PWM 输出：OCxREF/OCxNREF(x=1~3)。对于每对 PWM 输出可以分别配置为互补模式或独立模式。

12.3.4.1 互补输出模式

当 MCP_CR1 寄存器中的 OUTMODE 位为 0，输出比较模块工作在互补输出模式。在互补模式下，当 MCP_CR1 寄存器中的 SYM 位为 0 时，输出对称 PWM 波形，当 SYN 位为 1 时，输出非对称 PWM 波形（仅在中央对齐模式下有效）。

当工作在对称 PWM 输出模式下，16 位计数器与占空比寄存器 MCP_OCDRx 比较以产生互补的输出波形，因此 MCP_CHx 和 MCP_CHxN 引脚上最终输出的 PWM 波形使用的是同一占空比寄存器 MCP_OCDRx，并可由寄存器 MCP_DTRRx 和 MCP_DTFRx 设置死区。

当工作在非对称 PWM 输出模式下（仅在中央对齐技术模式下有效），16 位计数器在向上计数过程中与占空比寄存器 MCP_OCDRx 比较以产生 OCxREF 和 OCxNREF 波形，在向下计数过程中与占空比寄存器 MCP_OCNDRx 比较以产生 OCxREF 和 OCxNREF 波形，并可由寄存器 MCP_DTRRx 和 MCP_DTFRx 设置死区。

12.3.4.2 独立输出模式

当 MCP_CR1 寄存器中的 OUTMODE 位为 1，输出比较模块工作在独立输出模式。在此模式下，计数器与占空比寄存器 MCP_OCDRx 比较产生 OCxREF 波形，计数器与占空比寄存器 MCP_OCNDRx 的值比较产生 OCxNREF 波形，但设置 MCP_DTRRx 和 MCP_DTFRx 对波形不产生影响，即独立输出模式下的 PWM 无死区。

OCxREF 和 OCxNREF 再经过死区逻辑、故障检测及输出控制模块，最终由 MCP_CHx 和 MCP_CHxN 输出。

12.3.5 PWM 输出极性

一个 PWM 周期中的占空比有效区域由 MCP_CR1 寄存器中的 DUTY 位定义。

当 DUTY=0 时，计数值小于占空比寄存器定义为有效区域（active area），计数器值大于或等于占空比寄存器值为无效区域（inactive area）。

当 DUTY=1 时，计数值大于或等于占空比寄存器定义为有效区域（active area），计数器值小于占空比寄存器值为无效区域（inactive area）。

MCP_CR1 寄存器中的 CxP 位决定输出通道 MCP_CHx 的电平。当 CxP 为 0 时，MCP_CHx 在占空比有效区域为高电平，其余时间为低电平，当 CxP 为 1 时，MCP_CHx 在占空比有效区域为低电平，其余时间为高电平。

MCP_CR1 寄存器中的 CxNP 位决定输出通道 MCP_CHxN 的电平。当 CxNP 为 0 时，MCP_CHxN 在占空比有效区域为低电平，其余时间为高电平，当 CxNP 为 1 时，MCP_CHxN 在占空比有效区域为高电平，其余时间为低电平。

12.3.6 边沿对齐计数模式下 PWM 输出

当 MCP_CR1 寄存器中的中 CMS[1:0]为“00”时，计数器工作在边沿对齐计数模式。在互补输出模式下，PWM 输出波形 MCP_CHx 和 MCP_CHxN。设置 MCP_CR1 寄存器中的 DUTY 位为 0，当设置 MCP_CR1 寄存器中的 CEN 位为 1 后，计数器值小于 MCP_OCDRx 时处于 DUTY 区，因此 MCP_CHx 输出高电平，MCP_CHxN 输出低电平。当计数器与占空比 MCP_OCDRx 寄存器匹配后，MCP_CHx 输出低电平，MCP_CHxN 输出高电平。

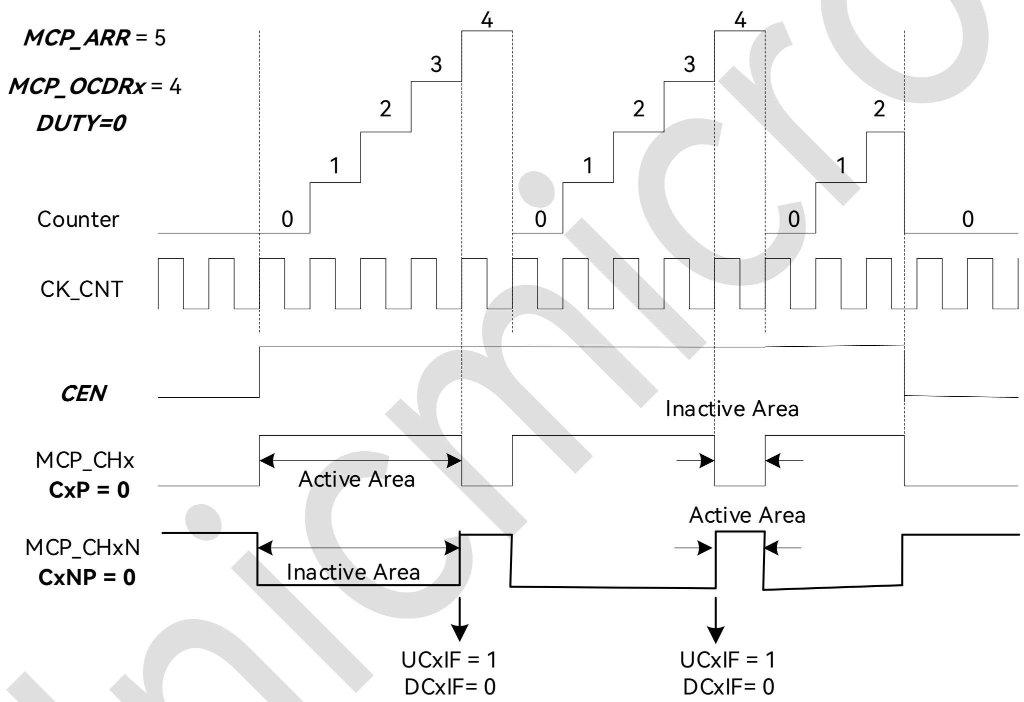


图 12-5：边沿对齐计数、互补输出模式下的 PWM 波形

在独立输出模式下，CEN 置 1 后，MCP_CHx 和 MCP_CHxN 波形变高，计数器与占空比寄存器 MCP_OCDRx 匹配后 MCP_CHx 波形将变低直至计数器归零；PWM 计数器与占空比寄存器 MCP_OCNDRx 匹配后 MCP_CHxN 波形将变低。

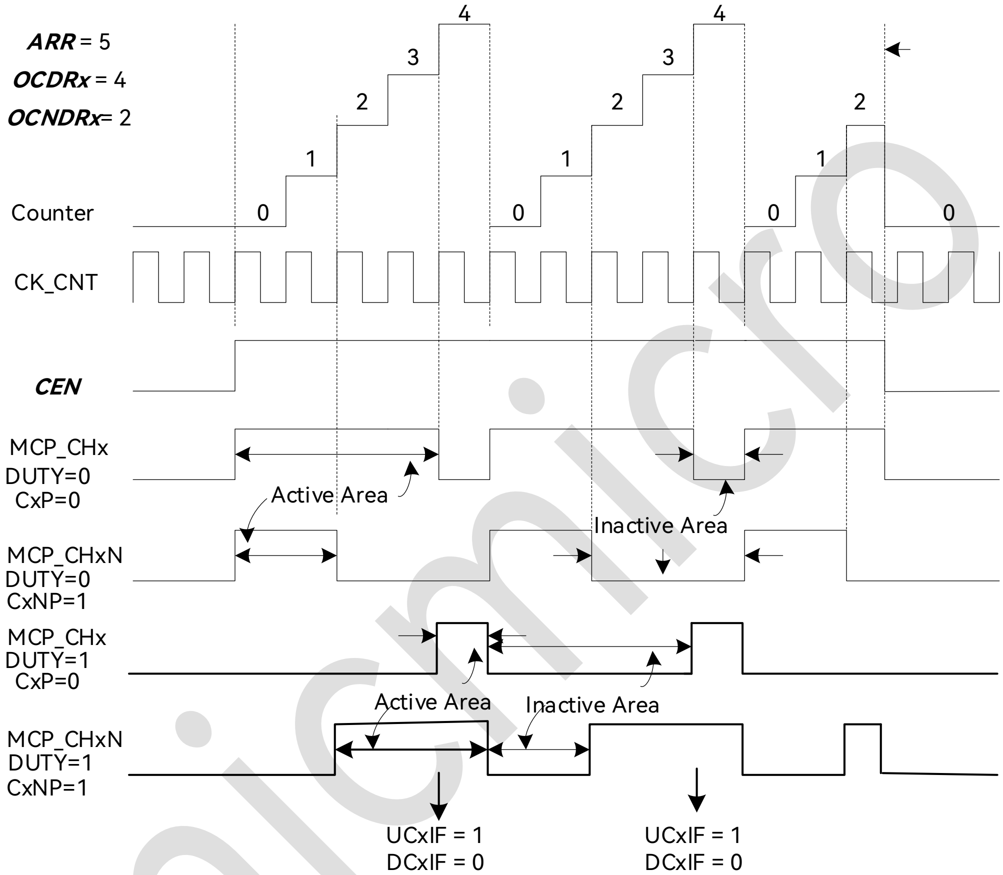


图 12-6：边沿对齐计数、独立输出模式下的 PWM 波形

12.3.7 中心对齐计数模式下 PWM 输出

当 MCP_CR1 寄存器中的 CMS[1:0]位为“01”时，计数器工作在中心对齐计数模式。

在互补输出模式下，当 MCP_CR1 寄存器中的 SYM 位为 0 时，输出比较模块工作在对称波形输出模式，PWM 输出波形 MCP_CHx 和 MCP_CHxN 如图所示。CEN 置 1 后，MCP_CHx 和 MCP_CHxN 波形保持为高，计数器向上计数时与 MCP_OCDRx 匹配后 MCP_CHx 和 MCP_CHxN 波形将变低直至计数器减计数时再次与 MCP_OCDRx 匹配，如此往复。

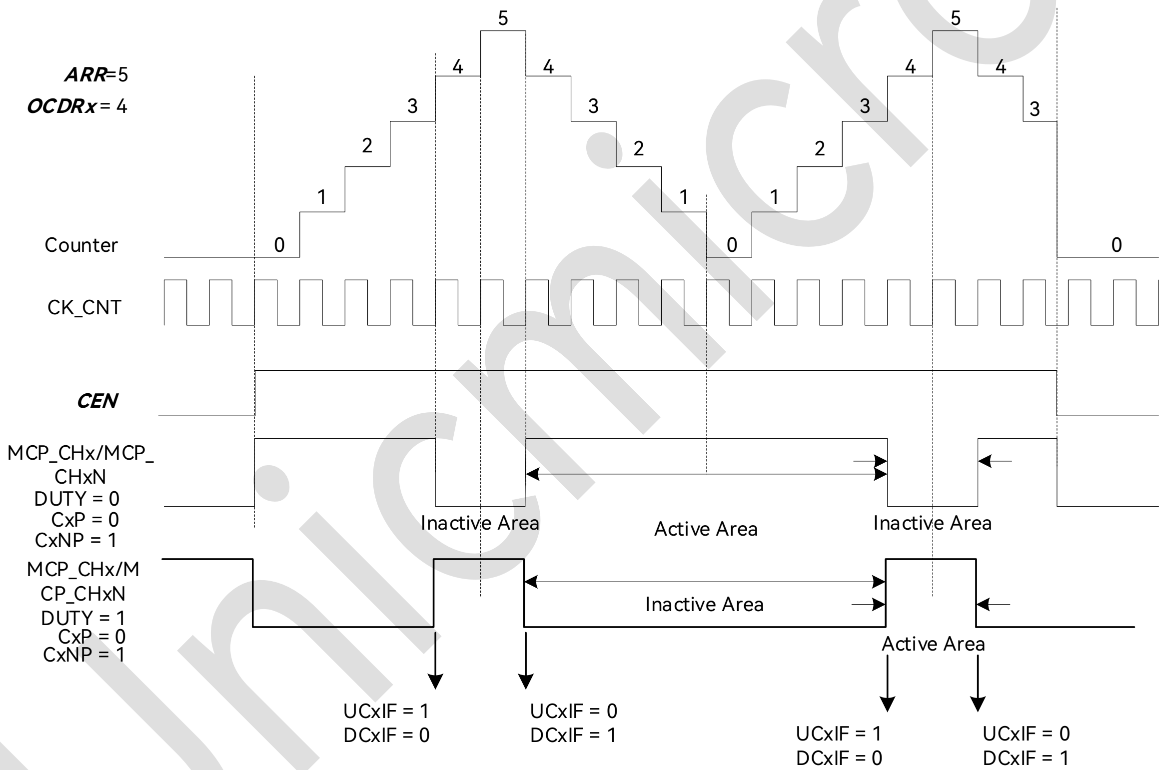


图 12-7：中心对齐计数、互补输出模式下的 PWM 波形（对称波形）

在互补输出模式下，当 MCP_CR1 寄存器中的 SYM 位为 1 时，输出比较模块工作在非对称波形输出模式，PWM 输出波形 MCP_CHx 和 MCP_CHxN 如图所示。设置 MCP_CR1 寄存器中的 CEN 位为 1 后，MCP_CHx 和 MCP_CHxN 波形保持为高，计数器向上计数时与 MCP_OCDRx 匹配后 MCP_CHx 和 MCP_CHxN 波形将变低直至计数器向下计数时与 MCP_OCNDRx 匹配。

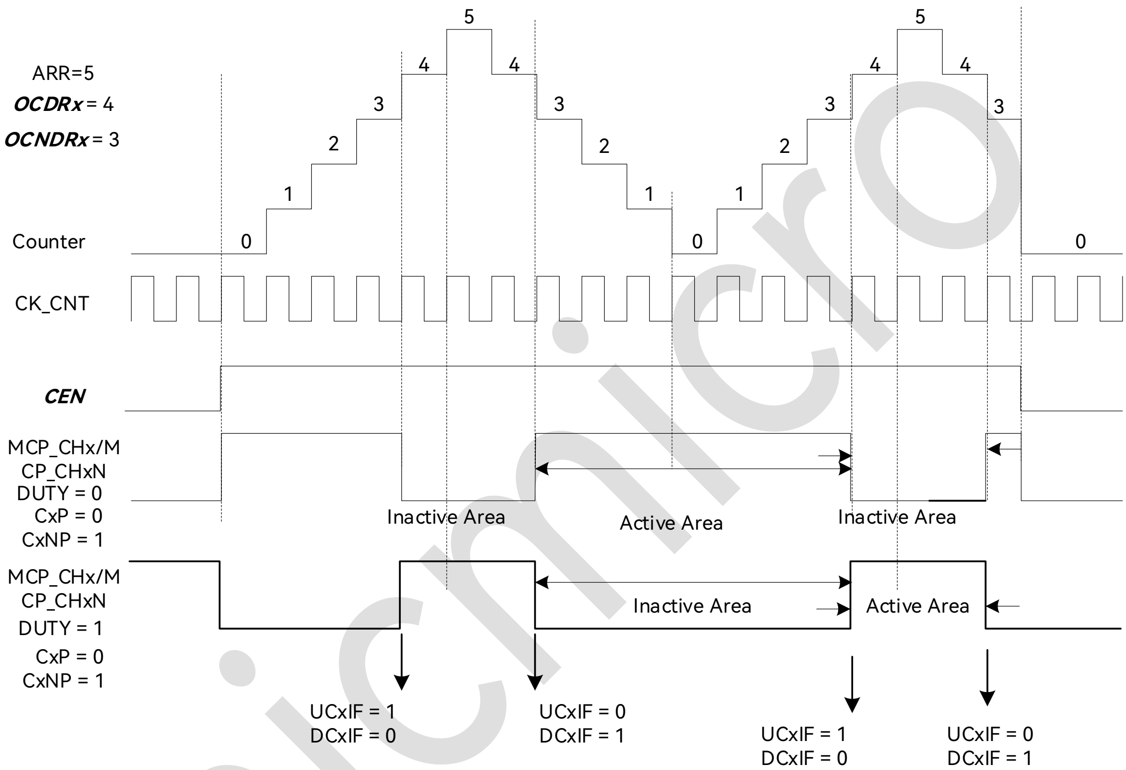


图 12-8：中心对齐计数、互补输出模式下的 PWM 原始波形（非对称波形）

独立输出模式下，PWM 输出波形 MCP_CHx 和 MCP_CHxN 如图所示。CEN 置 1 后，MCP_CHx 和 MCP_CHxN 波形变高，计数器向上计数时与 MCP_OCDRx 匹配后 MCP_CHx 波形将变低直至计数器向下计数时再次与 MCP_OCDRx 匹配；计数器向上计数时与 MCP_OCNDRx 匹配后 MCP_CHxN 波形将变低直至计数器向下计数时再次与 MCP_OCNDRx 匹配。

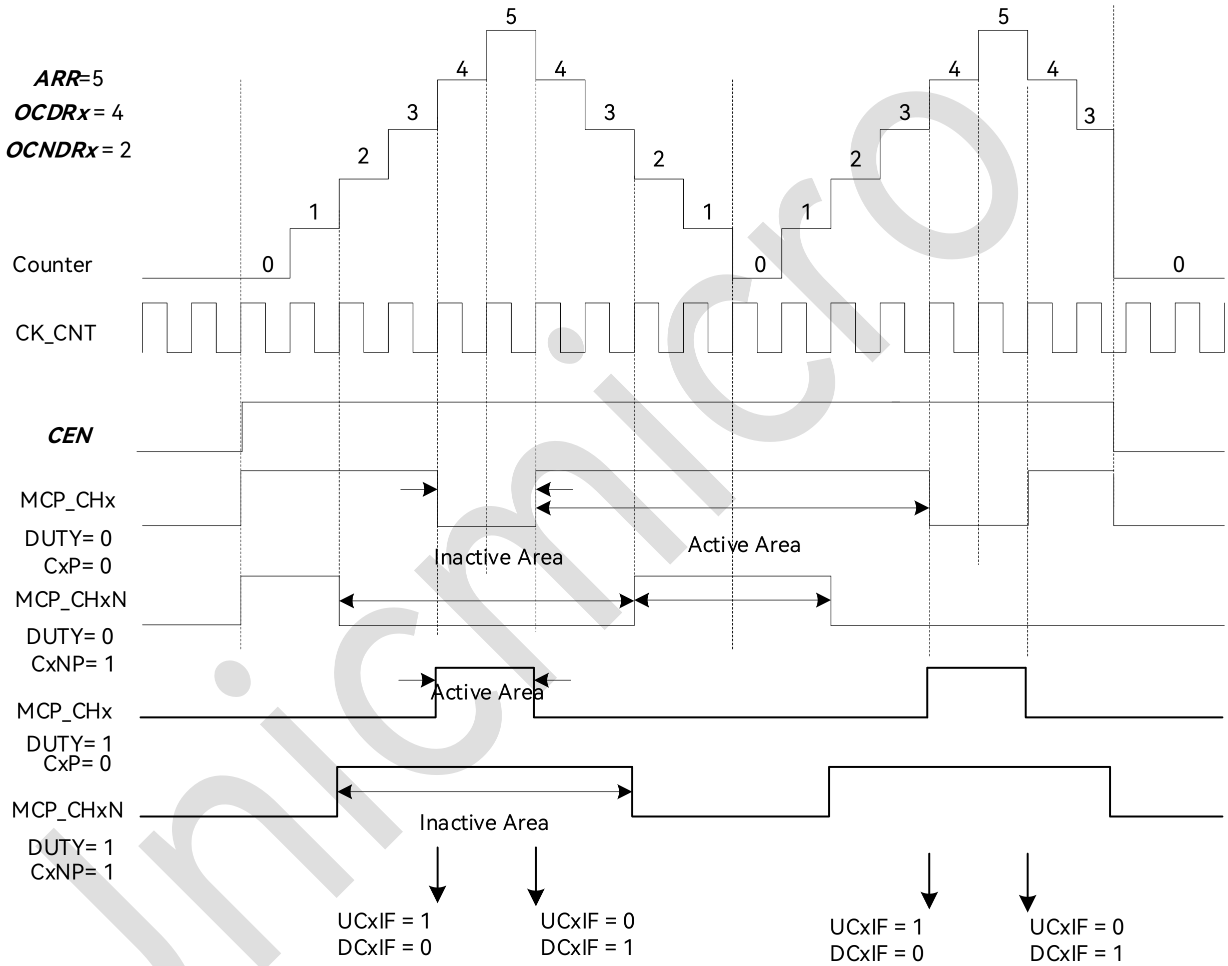


图 12-9：中心对齐计数、独立输出模式下的 PWM 波形

12.3.8 单次计数模式下 PWM 输出

当 MCP_CR1 寄存器中的 CMS[1:0]位为“10”时，计数器工作在单次计数模式。

在互补输出模式下，PWM 输出波形 MCP_CHx 和 MCP_CHxN 如图所示。CEN 置 1 后，MCP_CHx 和 MCP_CHxN 波形变高，计数器向上计数时与 MCP_OCDRx 匹配后 MCP_CHx 和 MCP_CHxN 波形将变低。

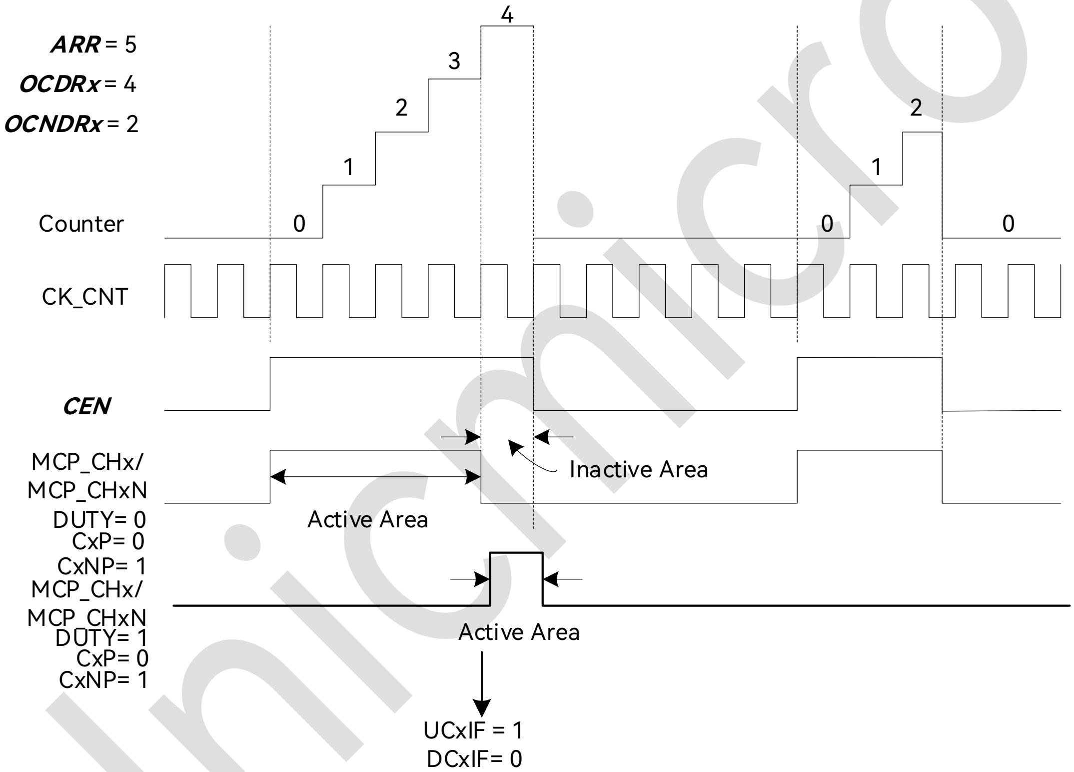


图 12-10：单次计数、互补输出模式下的 PWM 波形

在独立输出模式下，CEN 置 1 后，MCP_CHx 和 MCP_CHxN 波形变高（DUTY=0），计数器向上计数时与 MCP_OCDRx 匹配后 MCP_CHx 波形将变低；PWM 计数器向上计数时与 MCP_OCNDRx 匹配后 MCP_CHxN 波形将变低。

12.3.9 占空比寄存器生效时刻

当计数器工作在边沿对齐模式或单次计数模式下，修改占空比寄存器的生效时刻由 MCP_CR2 寄存器中的 DPE 位决定是立即生效还是在计数器发生更新事件（UEV）时生效。当 DPE 为 0 时，修改占空比寄存器会立即生效。当 DPE 为 1 时，同时需要设置 MCP_CR2 寄存器中的 UDE 为 1，修改占空比寄存器会在计数器发生更新事件时生效。

当计数器工作在中心对齐模式下，修改占空比寄存器的生效时刻由 MCP_CR2 寄存器中的 DEP 位决定是立即生效还是在计数器发生更新事件（包括上溢更新事件）时生效。当 DPE 为 0 时，修改占空比寄存器会立即生效。当 DPE 为 1 时，如果设置 MCP_CR2 寄存器中的 UDE 为 1，修改占空比寄存器会在计数器发生（下溢）更新事件时生效，如果设置 MCP_CR2 寄存器中的 OUDE 为 1，修改占空比寄存器会在计数器发生上溢更新事件时生效。

12.3.10死区插入

死区逻辑通过将输出比较模块的输出波形 OCxREF 的上升和 OCxNREF 的下降沿延迟一段时间而实现。MCP_CR1 寄存器中 OUTMODE 位为 1 时，OCxREF 和 OCxNREF 不会经过死区逻辑，即输出为独立模式时，没有死区插入功能；MCP_CR1 寄存器中 OUTMODE 位为 0 时，OCxREF 和 OCxNREF 经过死区逻辑，即输出为互补模式时，将引入死区。死区时间由寄存器 DTRRx 和 DTFRx 决定：

上升沿延迟时间（Rising Edge Delay）= DTRRx * TCK_CNT

下降沿延迟时间（Falling Edge Delay）= DTFRx * TCK_CNT

注：DTRRx = 0 时，输出波形 OCxREF 的上升沿不会触发延时；DTFRx = 0 时，输出波形 OCxNREF 的下降沿也不会触发延时。

死区插入在不同输出极性下的波形，如下图所示：

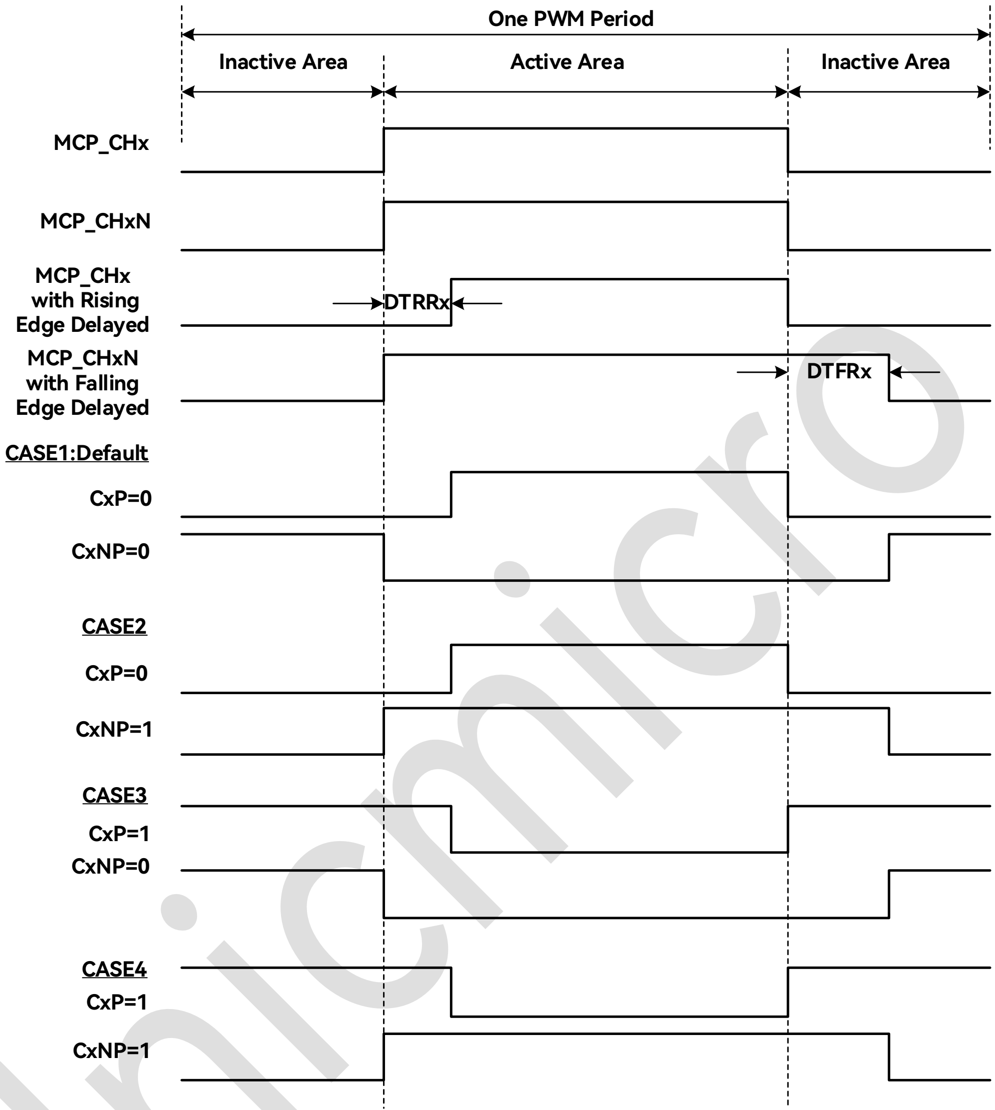


图 12-11：不同极性下的死区波形

死区寄存器修改后不会立即生效，需要计数器发生更新事件（包括上溢更新事件）时才会根据设置决定是否生效。当 MCP_CR2 寄存器中的 UDE 位为 1 时，计数器发生更新事件（或中央对齐模式下的下溢更新）时更新修改后的死区寄存器 MCP_DTRRx 和 MCP_DTFRx。当 MCP_CR2 寄存器中的 OUDE 位为 1 时，计数器发生上溢更新事件时才会更新修改后的死区寄存器 MCP_DTRRx 和 MCP_DTFRx。

12.3.11 故障检测模式

MCP 中有一个故障检测模块，当 FDCFGR 寄存器中 FD0EN、FD1EN、FD2EN 位为 1 时，使能该模块功能，若都为 0，则禁止该模块功能。故障检测主要目的是：当故障发生时（比如过流）可以切断 PWM 的输出，进入无效驱动状态（可设为高电平、低电平和高阻态），进而达到保护外部功率器件的目的。

12.3.11.1 故障信号输入源的选择

可以选择比较器 1、比较器 2 的输出或 MCP_BKIN 引脚输入作为故障检测输入信号。无论故障信号来自比较器 1 输出、比较器 2 输出还是 MCP_BKIN 引脚输入，都可以设置高低电平来保护 PWM 输出。

如下图所示，若 FDxEN = 1，则一旦对应信号变为 FDxP（x = 0 – 2）所指定的信号，并保持滤波时间（只有 BKIN 引脚才有滤波时间，比较器输出的滤波时间在比较器外设中设置），则六路 PWM 引脚 MCP_CHx/MCP_CHxN 立即进入无效驱动状态。

当使用 PWM 模块来控制如下图所示的逆变桥，六个管子为高电平驱动的功率管：设置 CxP=0/CxNP = 1（x = 0 – 2），则 MCP_CHx 的输出在占空比时区期间为高电平，MCP_CHxN 的输出在占空比时区期间为低电平，设置 MCP_CR1 寄存器中 OUTMODE 位为 0，即互补输出模式。使能故障检测后，若发生故障，则六路 PWM 进入无效驱动状态（可设为高电平、低电平和高阻态）。

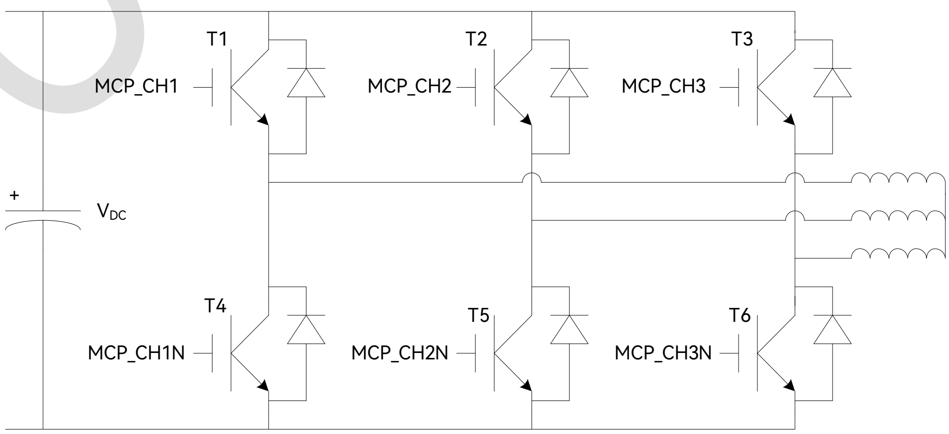


图 12-12：PWM 控制逆变桥示例图

12.3.11.2 故障检测输入信号滤波

当 MCP_FDCFGR 寄存器中的 FD2EN 位为 1 时，MCP_BKIN 引脚的输入信号会作为故障检测输入信号使用，此时可以通过设置 MCP_FDCFGR 寄存器中的 FD2F[3:0]位段来调整对此信号的滤波时间。FD2F[3:0] = 0 时，无滤波作用。当 MCP_FDCFGR 寄存器中的 FD2P 位为 0 时，故障检测输入信号的电平由低变高时立即触发保护；FD2F[3:0] = 0 – 15 时，可以将滤波时间设置为 0~32μs 共 16 档，故障检测输入信号由低变高时，高电平需要至少保持 FD2F[3:0]定义的时间累加长度，故障检测模块才认为故障检测输入信号的电平变高了, 从而触发保护; 故障检测输入信号的电平由高变低时，低电平需要至少保持 FD2F[3:0]定义的时间累加长度，故障检测模块才认为 MCP_BKIN 引脚上的有效故障电平消失了，六路 PWM 的输出根据保护模式发生动作。当 FD2P = 1 时，故障检测信号变为低电平有效，情况与 FD2P = 0 时相反。根据功率管的特性，恰当的设置滤波时间可以滤除故障检测输入信号上的噪声。

12.3.11.3 滤波时间设置说明

1. 计数器上限设置为滤波常数，下限为 0，超出上下限，不执行加减动作
2. 滤波计数器初值设定:若 MCP_BKIN 引脚设为高电平有效，计数器初始值为 0；若 MCP_BKIN 引脚设为低电平有效，则滤波计数器初值设为滤波常数。
3. 滤波器输出稳定需要一个建立时间，大致为滤波常数设置的时间，如滤波常数设置为 256，则在 256 个系统时钟后，滤波器输出稳定，在这之前，滤波器输出不确定。

当选比较器 1、比较器 2 的输出信号作为故障检测输入信号使用，此时比较器 1、比较器 2 的输出滤波时间和算法分别在较器外设的寄存器中设置。

12.3.11.4 保护模式

有两种故障检测模式：锁存模式与逐次模式。当发生故障时，立即进入无效电平状态，无效电平可以通过寄存器设置为高阻、低电平或高电平。

设置 MCP_FDCR 寄存器中 FDxM 位为 0 将使能锁存模式。当故障检测模块探测到有效故障信号，六路 PWM 引脚 MCP_CHx/MCP_CHxN 立即切换 MCP_FDCR 寄存器中的 FDOCP 位和 FDOCNP 位设定的电平状态。

若有效故障信号没有消失，六路 PWM 引脚 MCP_CHx/MCP_CHxN 将保持输出 FDOCP 位和 FDOCNP 位设定电平状态，软件无法清除 FDxSTAT 标志位，六路 PWM 引脚无法恢复正常输出。

若有效故障信号消失后，只有软件将 FDxSTAT 位清 0，六路 PWM 引脚 MCP_CHx/MCP_CHxN 亦不会恢复正常输出，只有软件将 FDCR 寄存器中的 FDxSTAT 位清 0 后，MCP_CHx/MCP_CHxN 才会在 FDxSTAT 被清零之后的最近一次计数器产生更新事件（上溢更新事件或下下溢更新事件）时恢复正常输出。

设置 MCP_FDCR 寄存器中 FDxM 位为 1 将使能逐次模式。在逐次模式下，MCP_CHx/MCP_CHxN 引脚是否输出 PWM 波形直接由比较器滤波后的输出信号控制。FDxSTAT 位也都由滤波之后的故障信号直接控制。

当故障检测模块探测到有效故障信号，六路 PWM 引脚 MCP_CHx/ MCP_CHxN 立即输出无效电平状态（x = 0 – 2），FDxSTAT 位亦会被硬件置 1。

若有效故障信号没有消失，六路 PWM 引脚 MCP_CHx/ MCP_CHxN 将保持输出无效电平状态（x = 0 – 2），软件无法使得六路 PWM 引脚恢复正常输出。

当有效故障信号消失后，六路 PWM 引脚 MCP_CHx/ MCP_CHxN 会在有效故障信号消失后的最近一次计数器发生更新事件（上溢更新事件或下溢更新事件）时自动恢复正常输出，FDxSTAT 位也将归 0。

注：MCP 使能前，PWM 输出固定为高阻态。

12.3.12 PWM 异常保护

MCP 模块可以对输出电平进行保护，当上下桥同时输出设定的有效电平时，PWM 进入异常保护。当发生异常保护时，PWM 输出状态可以设定为高电平、低电平和高阻态。

当 MCP_CHx 和 MCP_CHxN 同时输出 MCP_OSCR 寄存器中 OSCHx/OSCHxN 设定的有效电平，置起 MCP_SR 寄存器中的 OSIF 异常标志位，如果 MCP_IER 寄存器中的 OSIE 位为 1，则可以触发一个中断。同时可以选择发生保护后 PWM 输出状态，通过 MCP_FDCR 寄存器中的 FDOCP 位和 FDOCNP 位可选择输出为高电平、低电平或高阻态。如下图，设定有效电平都为低电平，输出保护后输出高阻态。

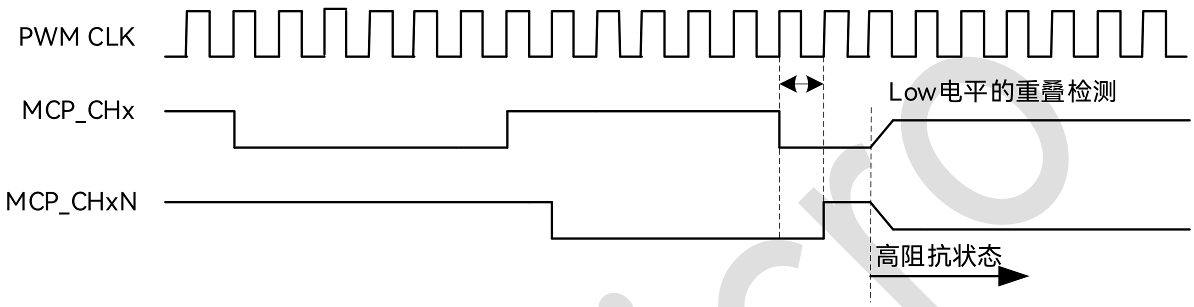


图 12-13：PWM 输出电平比较

注：

1. 假如只设定 3 个 MCP_CHx 输出或者 3 个 MCP_CHxN 输出，永远不会出现输出电平异常保护。
2. 电平异常保护时，当处于异常电平时，即使给异常标志位写 0，也不能清除标志位。
3. 异常保护在 MCP 模块时钟关闭时也可以正常工作。

12.3.13 PWM 输出控制模块

当 MCP 模块处于使能状态时，若 MOCR1 寄存器中的 MOCHxEN/MOCHxNEN 位为 1，则 MCP_CHx/MCP_CHxN 引脚将输出 MOCR2 寄存器中 MOCHxP/MOCHxNP 位的值; 若 MOCR1 寄存器中的 MOCHxEN/MOCHxNEN 位为 0，则 MCP_CHx/MCP_CHxN 引脚将输出 PWM 波形。在手动设置 PWM 输出模式下，同样受到故障检测输入信号控制。

更改 MOCR1 和 MOCR2 寄存器时，MCP_CHx/MCP_CHxN 引脚输出的改变与 PCLK 同步，输出状态改变立即生效。

12.3.14 事件触发功能

若寄存器 MCP_CR2 中的 CMPx =1（x=1-4），当计数器值与 MCP_CMPDRx（x=1-4）设定值相等时可以自动触发一次 ADC 转换；若 CMPx = 2 （x=1-4），只有在计数器向上计数过程中，当计数器值与 CMPDRx（x=1-4）设定值相等时可以自动触发一次 ADC 转换；若 CMPx = 3（x=1-4），只有在计数器向下计数过程中,当计数器值与 CMPDRx（x=1-4)设定值相等时可以自动触发一次 ADC 转换。PWM 比较寄存器触发 ADC 信号也受重复计数寄存器 MCP_RCR 控制。

在边沿对齐计数模式和单次计数模式下，事件比较值寄存器 MCP_CMPDRx 可以选择是否用更新事件重载。如果 MCP_CR2 寄存器中的 UCE 位为 1，则硬件会在计数器产生更新事件时将修改后的比较值更新到 MCP_CMPDRx 寄存器。

在中心对齐模式下，比较值寄存器除了可以用下溢更新事件重载外，还可以用上溢更新事件重载。如果 MCP_CR2 中 OUCE 位为 1，则硬件会在计数器产生上溢更新事件时将修改后的比较值更新到 MCP_CMPDRx 寄存器。

因此在边沿计数模式和单次计数模式下，由于时基模块只有归零匹配信号，故 PWM 计数器每次复位为 0 时都将发生一次比较值寄存器的重载；中心对齐计数模式下，时基模块能产生归零匹配以及周期匹配信号，故 PWM 计数器每次复位为 0 以及与周期寄存器匹配时都可以重载比较值寄存器。

比较器事件触发设置位 MCP_CR2 的 CMPx (x=1-4) 的重载与事件比较值寄存器 MCP_CMPDRx 一样。

12.3.15 中断

当计数器工作在边沿对齐计数模式和单次计数模式时，计数器值与 MCP_ARR 寄存器值匹配会产生一个更新事件（Update Event，UEV），MCP_SR 寄存器中的 UIF 位被置 1。如果 MCP_IER 寄存器中的 UIE 位为 1，则会产生一个中断。

当计数器工作在中央对齐计数模式时，计数器向上计数时，当计数器值与 MCP_ARR 寄存器值匹配时会产生一个上溢更新事件（Overflow Update Event），MCP_SR 寄存器中的 OUIF 位被置 1，

如果 MCP_IER 寄存器中的 OUIE 位为 1，则会产生一个中断。计数器向下计数时，当计数器值与 0 匹配时会产生一个下溢更新事件（Underflow Update Event），MCP_SR 寄存器中的 UIF 位被置 1，如果 MCP_IER 寄存器中的 UIE 位为 1，则会产生一个中断。

不论计数器工作在何种模式，当计数器向上计数时，计数值与占空比寄存器 MCP_OCDRx 匹配时 MCP_SR 寄存器中的 UCxIF 位会置 1，此时如果 MCP_IER 寄存器中的 UCxIE 位为 1，则能够触发占空比中断。在中心对齐计数模式中，当计数器向下计数时，计数值与 MCP_OCDRx 匹配时 MCP_SR 寄存器中的 DCxIF 位会置 1，此时如果 MCP_IER 寄存器中的 DCxIE 位为 1，则能够触发占空比中断。

注：

1. 互补对称输出模式和独立输出模式，占空比中断只是在计数器与 MCP_OCDRx 比较时产生，与 OCNDRx 比较匹配时不会改变占空比中断标志。
2. 互补非对称输出模式，占空比中断在向上计数时计数器与 MCP_OCDRx 比较时产生，在向下计数时计数器与 MCP_OCNDRx 比较时产生。
3. 在中央对齐计数模式下，当占空比设为 0 时，只有 DCxIF 位在比较匹配时会被置 1。当占空比设为周期值时，只有 UCxIF 位在比较匹配时会被置 1。

当发生故障保护时，MCP_SR 寄存器中的 FDxIF 位会被置 1，如果 MCP_IER 寄存器中的 FDxIE 位为 1，则会触发一个中断。

当发生输出短路保护时，MCP_SR 寄存器中的 OSIF 位会被置 1，如果 MCP_IER 寄存器中的 OSIE 位为 1，则会触发一个中断。

MCP_SR 寄存器中的标志位，当被硬件置 1 后，可以由软件向相应的位写 0 清零。

表 12-1：MCP 模块中断表

中断事件	事件标志	使能控制位
更新事件（下溢更新事件）	UIF	UIE
上溢更新事件	OUIF	OUIE
向上计数输出比较匹配	UCxIF	UCxIE
向下计数输出比较匹配	DCxIF	DCxIE
故障检测保护事件	FDxIF	FDxIE
输出短路保护事件	OSIF	OSIE

12.3.16 注意事项

支持在线仿真，在仿真状态运行和正常状态运行没有区别，如果 DBGMCU 模块中的 DBG_MCP_STOP 位设为 1，在仿真状态停止运行或单步运行，PWM 输出会切换为高阻态，保证不会误触发外部功率管。

进入 SLEEP 模式后，PWM 输出口不受影响，比如 MCP_CHx 输出 20K 的方波，进入 SLEEP 模式后会继续输出 20K 的方波，如果设置使能了 MCP 模块的中断，会唤醒 SLEEP 模式。

进入 STOP 模式时，如果 MCP 模块输出使能，则会关闭 CEN，PWM 输出引脚会输出高阻状态，即使设置了 MCP 模块的中断，也不能唤醒 STOP 模式，必须通过其它方式唤醒。在退出 STOP 模式后需软件打开 CEN。

注：为避免在从 STOP 模式下恢复后状态不确定，建议在进入 STOP 之间软件关闭 MCP 模块，在退出 STOP 后重新配置和打开 MCP 模块。

12.4 相关寄存器

12.4.1 MCP 控制寄存器 1（MCP_CR1）

偏移地址：0x00

复位值：0x0000 0000

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

15	14 13	12 11	10	9 8	7	6	5	4	3	2	1	0
OUTMODE	OPMS[1:0]	CMS[1:0]	SYM	Res.	DUTY	C3NP	C2NP	C1NP	C3P	C2P	C1P	CEN
rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw

位	名称	描述
31:16	RSV	保留位，始终读为 0
15	OUTMODE	输出模式（独立/互补）控制位（out mode）： 0：整个模块配置为 3 通道互补输出

位	名称	描述
		1：整个模块配置为 6 路独立输出
14:13	OPMS[1:0]	单次计数模式结束时输出电平选择（one pulse mode selection): 00：高阻 01：保持输出电平 10：全为低电平 11：全为高电平
12:11	CMS[1:0]	计数器工作模式选择（counter mode selection)： 00：边沿对齐计数模式，在此模式下，计数器只会发生更新事件 01：中心对齐计数模式，在此模式下，计数器会发生上溢更新事件和下溢更新事件 10：单次计数模式，在此模式下，计数器只会发生更新事件 11：保留
10	SYM	PWM 波形对称性控制位（symmetry)： 0：互补输出对称 PWM 波形 1：互补输出非对称 PWM 波形 注：仅在中心对齐计数模式的互补输出时有效
9:8	RSV	保留位，始终读为 0
7	DUTY	占空比有效区域选择位（duty): 0：计数器值小于占空比寄存器时定义为 DUTY 区 1：计数器值大于等于占空比寄存器时定义为 DUTY 区
6	C3NP	MCP_CH3N 比较输出通道有效电平（Active Polar）选择位（compare channel 3N polar): 0：MCP_CH3N 通道在 Duty 区输出低电平，其余时间输出高电平 1：MCP_CH3N 通道在 Duty 区输出高电平，其余时间输出低电平
5	C2NP	MCP_CH2N 比较输出通道有效电平（Active Polar）选择位（compare channel 2N polar): 0：MCP_CH2N 通道在 Duty 区输出低电平，其余时间输出高电平 1：MCP_CH2N 通道在 Duty 区输出高电平，其余时间输出低电平
4	C1NP	MCP_CH1N 比较输出通道有效电平（Active Polar）选择位（compare channel 1N polar): 0：MCP_CH1N 通道在 Duty 区输出低电平，其余时间输出高电平 1：MCP_CH1N 通道在 Duty 区输出高电平，其余时间输出低电平
3	C3P	MCP_CH3 比较输出通道有效电平（Active Polar）选择位（compare channel 3 polar): 0：MCP_CH3 通道在 Duty 区输出高电平，其余时间输出低电平 1：MCP_CH3 通道在 Duty 区输出低电平，其余时间输出高电平
2	C2P	MCP_CH2 比较输出通道有效电平（Active Polar）选择位（compare

位	名称	描述
		channel 2 polar): 0：MCP_CH2 通道在 Duty 区输出高电平，其余时间输出低电平 1：MCP_CH2 通道在 Duty 区输出低电平，其余时间输出高电平
1	C1P	MCP_CH1 比较输出通道有效电平（Active Polar）选择位（compare channel 1 polar): 0：MCP_CH1 通道在 Duty 区输出高电平，其余时间输出低电平 1：MCP_CH1 通道在 Duty 区输出低电平，其余时间输出高电平
0	CEN	计数器使能位（counter enable): 0：禁止计数器 1：使能计数器

12.4.2 MCP 控制寄存器 2（MCP_CR2）

偏移地址：0x04

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARPE	OUCE	UCE	OUDE	UDE	CPE	DPE	MOSYNC	CMP4[1:0]		CMP3[1:0]		CMP2[1:0]		CMP1[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw		rw		rw		rw	

位	名称	描述
31:16	RSV	保留位，始终读为 0
15	ARPE	自动重装载预装载允许位（Auto-reload preload enable): 0：MCP_ARR 寄存器没有缓冲 1：MCP_ARR 寄存器有缓冲
14	OUCE	计数器上溢更新事件时重载事件触发参数使能位（overflow update event compare preload enable): 0：上溢更新事件时不允许重载事件触发参数 1：上溢更新事件时允许重载事件触发参数 注：重载的寄存器为事件比较值寄存器（MCP_CMPDRx，x=1,2,3,4）、触发 AD 方式寄存器（CMPx[1:0]）使能位（只在中心对齐模式下有效）
13	UCE	计数器更新事件（或下溢更新事件）时重载事件触发参数使能位（update（or underflow update）event compare preload enable:

位	名称	描述
		0：更新事件（下溢更新事件）时不允许重载事件触发参数 1：更新事件（下溢更新事件）时允许重载事件触发参数 注：重载的寄存器为事件比较值寄存器（MCP_CMPDRx，x=1,2,3,4）、触发 AD 方式寄存器(CMPx[1:0])使能位
12	OUDE	计数器上溢更新事件时重载占空比及死区寄存器使能位（overflow update event duty/dead time preload enable）: 0：上溢更新事件时不允许重载占空比寄存器和死区寄存器 1：上溢更新事件时允许重载占空比寄存器和死区寄存器 注：只在中心对齐模式下有效
11	UDE	计数器更新事件（下溢更新事件）时重载占空比及死区寄存器使能位(update event duty/dead time preload enable): 0：更新事件（下溢更新事件）时不允许重载占空比寄存器和死区寄存器 1：更新事件（下溢更新事件）时允许重载占空比寄存器和死区寄存器
10	CPE	事件触发比较值寄存器立即生效使能位（compare preload enable): 0：立即生效 1：与 PWM 周期同步后生效 注：当 CPE=1 时，根据需要还要设置 OUCE 和 UCE 才能重载事件触发比较值寄存器（MCP_CMPDRx，x=1~4）、触发 AD 方式寄存器（CMPx[1:0]）使能位
9	DPE	占空比寄存器立即生效使能位（duty preload enable）: 0：立即生效 1：与 PWM 周期同步后生效 注：当 DPE=1 时，根据需要还要设置 OUDE 和 UDE 才能重载占空比寄存器
8	MOSYNC	手动修改 PWM 输出通道同步位（manual output Synchronize): 0：手动修改后立即生效 1：手动修改后当计数器产生更新事件（或下溢更新事件）时生效 注：手动修改通过设置 MCP_MOCR1 和 MCP_MOCR2 寄存器实现，任一寄存器的修改,何时生效都由此寄存器控制
7:6	CMP4[1:0]	比较数据寄存器 4 触发 ADC 控制位（compare 4): 00：不触发 ADC 01：计数器与比较数据寄存器 4 相等时会触发 ADC 10: 计数器在向上计数过程中与比较数据寄存器 4 相等时会触发 ADC（仅在中心对齐模式下有效） 11：计数器在向下计数过程中与比较数据寄存器 4 相等时会触发 ADC（仅在中心对齐模式下有效）
5:4	CMP3[1:0]	比较数据寄存器 3 触发 ADC 控制位（compare 3): 00：不触发 ADC

位	名称	描述
		01：计数器与比较数据寄存器 3 相等时会触发 ADC 10: 计数器在向上计数过程中与比较数据寄存器 3 相等时会触发 ADC（仅在中心对齐模式下有效） 11：计数器在向下计数过程中与比较数据寄存器 3 相等时会触发 ADC（仅在中心对齐模式下有效）
3:2	CMP2[1:0]	比较数据寄存器 2 触发 ADC 控制位（compare 2): 00：不触发 ADC 01：计数器与比较数据寄存器 2 相等时会触发 ADC 10: 计数器在向上计数过程中与比较数据寄存器 2 相等时会触发 ADC（仅在中心对齐模式下有效） 11：计数器在向下计数过程中与比较数据寄存器 2 相等时会触发 ADC（仅在中心对齐模式下有效）
1:0	CMP1[1:0]	比较数据寄存器 1 触发 ADC 控制位（compare 1): 00：不触发 ADC 01：计数器与比较数据寄存器 1 相等时会触发 ADC 10: 计数器在向上计数过程中与比较数据寄存器 1 相等时会触发 ADC（仅在中心对齐模式下有效） 11：计数器在向下计数过程中与比较数据寄存器 1 相等时会触发 ADC（仅在中心对齐模式下有效）

12.4.3 MCP 中断使能控制寄存器（MCP_IER）

偏移地址：0x0C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				OSIE	FD2IE	FD1IE	FD0IE	DC3IE	UC3IE	DC2IE	UC2IE	DC1IE	UC1IE	OUIE	UIE
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	名称	描述
31:12	RSV	保留位，始终读为 0
11	OSIE	输出短路的中断允许位（output short interrupt enable）: 0：禁止中断请求 1：允许中断请求
10	FD2IE	故障检测通道 2 中断允许位（fault detection 2 interrupt enable）: 0：禁止故障检测中断 1：允许故障检测中断
9	FD1IE	故障检测通道 1 中断允许位（fault detection 1 interrupt enable): 0：禁止故障检测中断 1：允许故障检测中断
8	FD0IE	故障检测通道 0 中断允许位（fault detection 0 interrupt enable): 0：禁止故障检测中断 1：允许故障检测中断
7	DC3IE	计数器向下计数时与输出比较通道 3 匹配时的中断使能位（down-counter compare 3 interrupt enable): 0：禁止计数器向下计数时与输出比较通道 3 匹配时触发中断 1：允许计数器向下计数时与输出比较通道 3 匹配时触发中断 注：只在中心对齐模式下有效
6	UC3IE	计数器向上计数时与输出比较通道 3 匹配时的中断使能位（up-counter compare 3 interrupt enable): 0：禁止计数器向上计数时与输出比较通道 3 匹配时触发中断 1：允许计数器向上计数时与输出比较通道 3 配时触发中断
5	DC2IE	计数器向下计数时与输出比较通道 2 匹配时的中断使能位（down-counter compare 2 interrupt enable): 0：禁止计数器向下计数时与输出比较通道 2 匹配时触发中断 1：允许计数器向下计数时与输出比较通道 2 匹配时触发中断 注：只在中心对齐模式下有效
4	UC2IE	计数器向上计数时与输出比较通道 2 匹配时的中断使能位（up-counter compare 2 interrupt enable): 0：禁止计数器向上计数时与输出比较通道 2 匹配时触发中断 1：允许计数器向上计数时与输出比较通道 2 匹配时触发中断
3	DC1IE	计数器向下计数时与输出比较通道 1 匹配时的中断使能位（down-counter compare 1 interrupt enable): 0：禁止计数器向下计数时与输出比较通道 1 匹配时触发中断 1：允许计数器向下计数时与输出比较通道 1 匹配时触发中断 注：只在中心对齐模式下有效
2	UC1IE	计数器向上计数时与输出比较通道 1 匹配时的中断使能位（up-counter

位	名称	描述
		compare 1 interrupt enable)： 0：禁止计数器向上计数时与输出比较通道 1 匹配时触发中断 1：允许计数器向上计数时与输出比较通道 1 匹配时触发中断
1	OUIE	计数器上溢更新事件中断允许位（overflow update event interrupt enable)： 0：禁止计数器上溢更新事件中断 1：允许计数器上溢更新事件中断 注：只在中心对齐模式下有效
0	UIE	计数器更新事件（或下溢更新事件）中断允许位（update event interrupt enable)： 0：禁止计数器更新事件（或下溢更新事件） 1：允许计数器更新事件（或下溢更新事件）

12.4.4 MCP 中断标志和清除寄存器（MCP_SR）

偏移地址：0x10

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				OSIF	FD2IF	FD1IF	FD0IF	DC3IF	UC3IF	DC2IF	UC2IF	DC1IF	UC1IF	OUIF	UIF
				rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位	名称	描述
31:12	RSV	保留位，始终读为 0
11	OSIF	输出短路标志（output short interrupt flag): 0：不同时变为有效电平 1：同时变为有效电平（MCP 中 3 组通道相输出中，至少有 1 组同时为有效电平）
10	FD2IF	故障检测 2 中断标志位（fault detection 2 interrupt flag)： 0：无故障检测中断 1：发生故障检测中断
9	FD1IF	故障检测 1 中断标志位（fault detection 1 interrupt flag)： 0：无故障检测中断 1：发生故障检测中断
8	FD0IF	故障检测 0 中断标志位（fault detection 0 interrupt flag)： 0：无故障检测中断

位	名称	描述
		1：发生故障检测中断
7	DC3IF	计数器向下计数时与输出比较通道 3 匹配时中断标志位（down-counter compare 3 interrupt flag）: 0：未发生中断 1：发生中断 注：只在中心对齐模式下有效
6	UC3IF	计数器向上计数时与输出比较通道 3 匹配时中断标志位（up-counter compare 3 interrupt flag）: 0：未发生中断 1：发生中断
5	DC2IF	计数器向下计数时与输出比较通道 2 匹配时中断标志位（down-counter compare 2 interrupt flag): 0：未发生中断 1：发生中断 注：只在中心对齐模式下有效
4	UC2IF	计数器向上计数时与输出比较通道 2 匹配时中断标志位（up-counter compare 2 interrupt flag): 0：未发生中断 1：发生中断
3	DC1IF	计数器向下计数时与输出比较通道 1 匹配时中断标志位（down-counter compare 1 interrupt flag）: 0：未发生中断 1：发生中断 注：只在中心对齐模式下有效
2	UC1IF	计数器向上计数时与输出比较通道 1 匹配时中断标志位（up-counter compare 1 interrupt flag): 0：未发生中断 1：发生中断
1	OUIF	计数器上溢更新事件中断标志位（overflow update interrupt flag）: 0：无计数器上溢更新事件中断 1：发生计数器上溢更新事件中断 注：只在中心对齐模式下有效
0	UIF	计数器更新事件（或下溢更新事件）中断标志位（update interrupt flag）: 0：无计数器更新事件（或下溢更新事件）中断 1：发生计数器更新事件（或下溢更新事件）中断

12.4.5 MCP 计数器寄存器（MCP_CNT）

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.															

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	CNT[15:0]	计数寄存器

12.4.6 MCP 时钟预分频寄存器（MCP_PSC）

偏移地址：0x28

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	PSC[15:0]	MCP 计数器时钟为（PCLK/(PSC[15:0]+1)

12.4.7 MCP 周期寄存器（MCP_ARR）

偏移地址：0x2C

复位值：0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Res.</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">ARR[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	ARR[15:0]	MCP 周期寄存器

12.4.8 MCP 重复计数寄存器（MCP_RCR）

偏移地址：0x30

复位值：0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Res.</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="12">Res.</td><td colspan="4">REP[3:0]</td></tr>
<tr><td colspan="12"></td><td colspan="4">rw</td></tr>
</table>

位	名称	描述
31:4	RSV	保留位，始终读为 0
3:0	REP[3:0]	后分频系数选择位段: 0000：中断、重载信号与事件触发信号将 1 分频（不分频） 0001：中断、重载信号与事件触发信号将 2 分频 0010：中断、重载信号与事件触发信号将 3 分频 0111：中断、重载信号与事件触发信号将 4 分频 …. 1110：中断、重载信号与事件触发信号将 15 分频 1111：中断、重载信号与事件触发信号将 16 分频

12.4.9 MCP 输出比较通道 1 占空比寄存器（MCP_OCDR1）

偏移地址：0x34

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	OCD[15:0]	占空比数值

12.4.10 MCP 输出比较通道 2 占空比寄存器（MCP_OCDR2）

偏移地址：0x38

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	OCD[15:0]	占空比数值

12.4.11 MCP 输出比较通道 3 占空比寄存器（MCP_OCDR3）

偏移地址：0x3C

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
1:0	OCD[15:0]	占空比数值

12.4.12 MCP 输出比较互补通道 1 占空比寄存器（MCP_OCNDR1）

偏移地址：0x40

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15~:0	OCD[15:0]	占空比数值

12.4.13 MCP 输出比较互补通道 2 占空比寄存器（MCP_OCNDR2）

偏移地址：0x44

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	OCD[15:0]	占空比数值

12.4.14 MCP 输出比较互补通道 3 占空比寄存器（MCP_OCNDR3）

偏移地址：0x48

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	OCD[15:0]	占空比数值

12.4.15 MCP 事件触发比较寄存器 1（MCP_CMPDR1）

偏移地址：0x54

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	CMPD[15:0]	事件触发比较值

12.4.16 MCP 事件触发比较寄存器 2（MCP_CMPDR2）

偏移地址：0x58

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	CMPD[15:0]	事件触发比较值

12.4.17 MCP 事件触发比较寄存器 3（MCP_CMPDR3）

偏移地址：0x5C

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	CMPD[15:0]	事件触发比较值

12.4.18 MCP 事件触发比较寄存器 4（MCP_CMPDR4）

偏移地址：0x60

复位值：0x0000 0000

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

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

位	名称	描述
3:16	RSV	保留位，始终读为 0
15:0	CMPD[15:0]	事件触发比较值

12.4.19 MCP 通道 1 上升沿死区控制寄存器（MCP_DTRR1）

偏移地址：0x64

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	DT[15:0]	上升沿死区数值

12.4.20 MCP 通道 1 下降沿死区控制寄存器（MCP_DTFR1）

偏移地址：0x68

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	DT[15:0]	下降沿死区数值

12.4.21 MCP 通道 2 上升沿死区控制寄存器（MCP_DTRR2）

偏移地址：0x6C

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	DT[15:0]	上升沿死区数值

12.4.22 MCP 通道 2 下降沿死区控制寄存器（MCP_DTFR2）

偏移地址：0x70

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	DT[15:0]	下降沿死区数值

12.4.23 MCP 通道 3 上升沿死区控制寄存器（MCP_DTRR3）

偏移地址：0x74

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	DT[15:0]	上升沿死区数值

12.4.24 MCP 通道 3 下降沿死区控制寄存器（MCP_DTFR3）

偏移地址：0x78

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	DT[15:0]	下降沿死区数值

12.4.25 MCP 手动输出控制寄存器 1（MCP_MOCR1）

偏移地址：0x84

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										MOCH3NEN	MOCH2NEN	MOCH1NEN	MOCH3EN	MOCH2EN	MOCH1EN
										rw	rw	rw	rw	rw	rw

位	名称	描述
31:6	RSV	保留位，始终读为 0
5	MOCH3NEN	MCP_CH3N 输出控制位（manual output channel 3N enable）: 0：MCP_CH3N 输出 PWM 波形 1：MCP_CH3N 输出由 MCP_MOCR2 寄存器中的 MOCH3NP 位控制
4	MOCH2NEN	MCP_CH2N 输出控制位（manual output channel 2N enable）: 0：MCP_CH2N 输出 PWM 波形 1：MCP_CH2N 输出由 MCP_MOCR2 寄存器中的 MOCH2NP 位控制
3	MOCH1NEN	MCP_CH1N 输出控制位（manual output channel 1N enable）: 0：MCP_CH1N 输出 PWM 波形 1：MCP_CH1N 输出由 MCP_MOCR2 寄存器中的 MOCH1NP 位控制
2	MOCH3EN	MCP_CH3 输出控制位（manual output channel 3 enable）: 0：MCP_CH3 输出 PWM 波形 1：MCP_CH3 输出由 MCP_MOCR2 寄存器中的 MOCH3P 位控制
1	MOCH2EN	MCP_CH2 输出控制位（manual output channel 2 enable）: 0：MCP_CH2 输出 PWM 波形 1：MCP_CH2 输出由 MCP_MOCR2 寄存器中的 MOCH2P 位控制
0	MOCH1EN	MCP_CH1 输出控制位（manual output channel 1 enable）: 0：MCP_CH1 输出 PWM 波形 1：MCP_CH1 输出由 MCP_MOCR2 寄存器中的 MOCH1P 位控制

12.4.26 MCP 手动输出控制寄存器 2（MCP_MOCR2）

偏移地址：0x88

复位值：0x0000 0000

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

15 14 13 12 11 10 9 8 7 6	5	4	3	2	1	0
Res.	MOCH3NP	MOCH2NP	MOCH1NP	MOCH3P	MOCH2P	MOCH1P
	rw	rw	rw	rw	rw	rw

位	名称	描述
31:6	RSV	保留位，始终读为 0
5	MOCH3NP	MOCH3NEN = 1 时，决定 MCP_CH3N 的输出电平（manual output channel 3N polarity）： 0：输出 0 1：输出 1
4	MOCH2NP	MOCH2NEN = 1 时，决定 MCP_CH2N 的输出电平（manual output channel 2N polarity）： 0：输出 0 1：输出 1
3	MOCH1NP	MOCH1NEN = 1 时，决定 MCP_CH1N 的输出电平（manual output channel 1N polarity）： 0：输出 0 1：输出 1
2	MOCH3P	MOCH3EN = 1 时，决定 MCP_CH3 的输出电平（manual output channel 3 polarity）： 0：输出 0 1：输出 1
1	MOCH2P	MOCH2EN = 1 时，决定 MCP_CH2 的输出电平（manual output channel 2 polarity）： 0：输出 0 1：输出 1

位	名称	描述
0	MOCH1P	MOCH1EN = 1 时，决定 MCP_CH1 的输出电平（manual output channel 1 polarity）: 0：输出 0 1：输出 1 注意：首次配置 MCP_MOCR1 寄存器时，必须先配置 MCP_MOCR2 寄存器。

12.4.27 MCP 故障检测控制寄存器（MCP_FDCR）

偏移地址：0x8C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FD0EN	FD0P	FD0M	FD0STAT	FD1EN	FD1P	FD1M	FD1STAT	FD2EN	FD2P	FD2M	FD2STAT	FD2F[3:0]			
rw	rw	rw	rc_w0	rw	rw	rw	rc_w0	rw	rw	rw	rc_w0	rw			

位	名称	描述
31:24	RSV	保留位，始终读为 0
23:22	FDOCNP	故障发生后，MCP_CHxN 引脚输出状态选择位(fault detection output channel N polarity) 0x：输出高阻态 10：输出低电平 11：输出高电平
21:20	FDOCP	故障发生后，MCP_CHx 引脚引脚输出状态选择位(fault detection output channel polarity) 0x：输出高阻态 10：输出低电平 11：输出高电平
19:16	RSV	保留位，始终读为 0
15	FD0EN	故障检测 0 使能位（fault detection 0 enable） 0：故障检测 0 功能禁止

位	名称	描述
		1：故障检测 0 功能打开，输入源为比较器 1
14	FD0P	故障检测 0 有效电平选择位（fault detection 0 polarity)： 0：故障检测 0 高电平有效 1：故障检测 0 低电平有效
13	FD0M	故障检测 0 功能模式选择（fault detection 0 mode)： 0：锁存模式，当检测到故障输入有效，FD0STAT 被硬件置 1，PWM 将立即停止输出，当故障输入变为无效时状态保持不变，只有当 FD0STAT 被软件清零后，PWM 波形才会在计数器发生更新事件（或下溢更新事件）时恢复输出。（当故障输入一直有效时 FD0STAT 无法被软件清除） 1：逐次模式，PWM 输出直接由故障检测输入端来控制，如果故障输入有效，FD0STAT 被硬件置 1，立即关闭 PWM 输出。如果故障输入变为无效，FD0STAT 自动清 0，PWM 波形将在计数器产生更新事件（或下溢更新事件）时自动恢复输出
12	FD0STAT	故障检测标志位（fault detection 0 status)： 0：PWM 处于正常输出状态 1：出现过流，PWM 处于停止输出状态，如果在锁存模式，可软件清 0，恢复 PWM 输出 注：FD0STAT 与 FD0IF 的区别为逐次模式下，FD0STAT 会被清零
11	FD1EN	故障检测 1 使能位（fault detection 1 enable)： 0：故障检测 1 功能禁止 1：故障检测 1 功能打开，输入源为比较器 2
10	FD1P	故障检测 1 有效电平选择位（fault detection 1 polarity)： 0：故障检测 1 高电平有效 1：故障检测 1 低电平有效
9	FD1M	故障检测 1 功能模式选择（fault detection 1 mode)： 0：锁存模式，当检测到故障输入有效，FD1STAT 被硬件置 1，PWM 将立即停止输出，当故障输入变为无效时状态保持不变，只有当 FD1STAT 被软件清零后，PWM 波形才会在计数器发生更新事件（或下溢更新事件）时恢复输出。（当故障输入一直有效时 FD1STAT 无法被软件清除） 1：逐次模式，PWM 输出直接由故障检测输入端来控制，如果故障输入有效，FD1STAT 被硬件置 1，立即关闭 PWM 输出。如果故障输入变为无效，FD1STAT 自动清 0，PWM 波形将在计数器产生更新事件（或下溢更新事件）时自动恢复输出
8	FD1STAT	故障检测 1 标志位（fault detection 1 status)： 0：PWM 处于正常输出状态 1：出现过流，PWM 处于停止输出状态，如果在锁存模式，可软件清 0，恢复 PWM 输出

位	名称	描述
		注：FD1STAT 与 FD1IF 的区别为逐次模式下，FD1STAT 会被清零
7	FD2EN	故障检测 2 功能使能位（fault detection 2 enable): 0：故障检测 2 功能禁止 1：故障检测 2 功能打开，输入源为 BKIN 引脚
6	FD2P	故障检测 2 有效电平选择位（fault detection 2 polarity): 0：故障检测 2 高电平有效 1：故障检测 2 低电平有效
5	FD2M	故障检测 2 功能模式选择（fault detection 2 mode): 0：锁存模式，当检测到故障输入有效，FD2STAT 被硬件置 1，PWM 将立即停止输出，当故障输入变为无效时状态保持不变，只有当 FD2STAT 被软件清零后，PWM 波形才会在计数器发生更新事件（或下溢更新事件）时恢复输出。(当故障输入一直有效时 FD2STAT 无法被软件清除) 1：逐次模式，PWM 输出直接由故障检测输入端来控制，如果故障输入有效，FD2STAT 被硬件置 1，立即关闭 PWM 输出。如果故障输入变为无效，FD2STAT 自动清 0，PWM 波形将在计数器产生更新事件（或下溢更新事件）时自动恢复输出
4	FD2STAT	故障检测 2 标志位（fault detection 2 status)： 0：PWM 处于正常输出状态 1：出现过流，PWM 处于停止输出状态，如果在锁存模式，可软件清 0，恢复 PWM 输出 注：FD2STAT 与 FD2IF 的区别为逐次模式下，FD2STAT 会被清零
3:0	FD2F[3:0]	故障检测 2 输入滤波参数选择（fault detection 2 filter)： 0000：无滤波 0001：0.5µs 0010：1µs 0011：1.5µs 0100：2µs 0101：3µs 0110：4µs 0111：6µs 1000：8µs 1001：10µs 1010：12µs 1011：14µs 1100：16µs 1101：20µs 1110：24µs

位	名称	描述
		1111：32μs 注 1：上述滤波常数时间不是精确值，仅供参考。 注 2：滤波器算法说明：用 HSI 时钟采样滤波器输入。采用数字积分，如果采样结果为高电平，计数器加 1，计数器结果超出设定的常数，则滤波器输出 1 同时计数器置为滤波常数；如果采样结果为低电平，计数器减 1，计数器结果小于滤波器常数,则滤波器输出 0 同时计数器置为 0。

12.4.28 MCP 输出电平保护控制寄存器（MCP_OSCR）

偏移地址：0x90

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								OSEN	Res.	OSCH3NP	OSCH2NP	OSCH1NP	OSCH3P	OSCH2P	OSCH1P
								rw		rw	rw	rw	rw	rw	rw

位	名称	描述
31:8	RSV	保留位，始终读为 0
7	OSEN	输出短路保护的使能位（output short enable）: 0：禁止 1：使能
6	RSV	保留位，始终读为 0
5	OSCH3NP	输出短路保护中通道 MCP_CH3N 的有效电平设定位（output short channel 3N polarity） 0：低电平有效 1：高电平有效
4	OSCH2NP	输出短路保护中通道 MCP_CH2N 的有效电平设定位（output short channel 2N polarity）: 0：低电平有效 1：高电平有效

位	名称	描述
3	OSCH1NP	输出短路保护中通道 MCP_CH1N 的有效电平设定位（output short channel 1N polarity）: 0：低电平有效 1：高电平有效
2	OSCH3P	输出短路保护中通道 MCP_CH3 的有效电平设定位（output short channel 3 polarity）: 0：低电平有效 1：高电平有效
1	OSCH2P	输出短路保护中通道 MCP_CH2 的有效电平设定位（output short channel 2 polarity）: 0：低电平有效 1：高电平有效
0	OSCH1P	输出短路保护中通道 MCP_CH1 的有效电平设定位（output short channel 1 polarity）: 0：低电平有效 1：高电平有效

13 通用定时器（TIM14）

13.1 概述

该定时器基于一个 16 位自动重载递增计数器和一个 16 位预分频器。

TIM14 具有一个通道，用于输入捕获/输出比较，PWM 输出。

在调试模式下，其计数器可被冻结。

13.2 特性

- 16 位自动装载计数器
- 16 位可编程（可实时修改）预分频器，支持 1 ~ 65535 之间的任意分频
- 1 个独立通道
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿对齐）
- 下述事件触发中断：
  - 更新（Update）：计数器上溢，计数器初始化（通过软件）
  - 输入捕获（input capture）
  - 输出比较（output compare）

13.3 功能描述

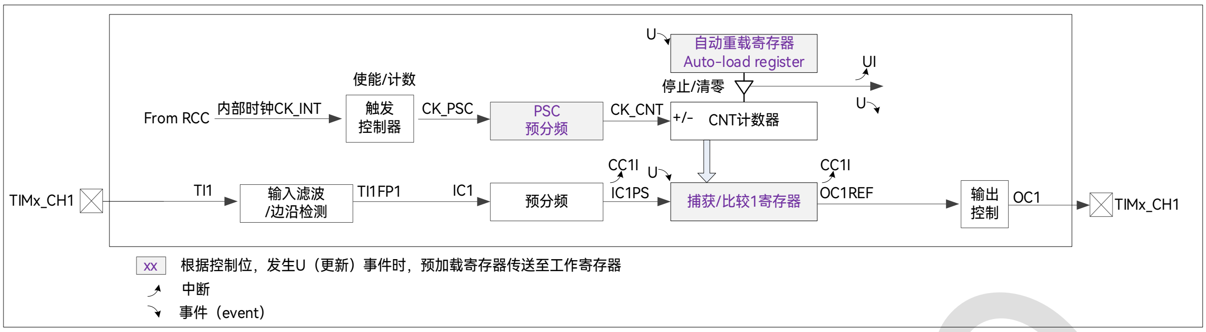


图 13-1：TIM14 模块框图

13.3.1 时基单元

通用定时器 TIM14 主要由 16 位计数器及相关自动重载寄存器构成，计数器支持向上计数。计数器时钟支持预分频。

计数器寄存器、自动重载寄存器和预分频寄存器支持软件读写，即使计数器正在运行读写仍然有效。

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，读写自动重载寄存器将访问预装载寄存器。设置 TIMx_CR1 寄存器中的自动重载预装载使能位（ARPE），选择预装载寄存器的内容永久传送至缓存寄存器或在每次更新事件（UEV）传送至缓存寄存器。当计数器达到溢出条件并当 TIMx_CR1 寄存器中的 UDIS 位等于 0 时，产生更新事件。更新事件也可由软件产生。有关更新事件的产生，针对每种配置后续章节会详细描述。

计数器时钟由预分频后输出 CK_CNT 驱动，需置位 TIMx_CR1 寄存器中的计数器使能位（CEN）时，CK_CNT 才有效。

注：设置 TIMx_CR 寄存器的 CEN 位，计数器会立即开始计数。

13.3.1.1 预分频

预分频器支持计数器时钟 1~65536 之间任意分频，基于 1 个 16 位的计数器，由 TIMx_PSC 寄存器中的 16 位寄存器控制。此寄存器内部带有缓存器，支持运行时修改；新修改的预分频值在下一次的更新事件发生时生效。

下图给出运行时更改预分频值，计数器行为：

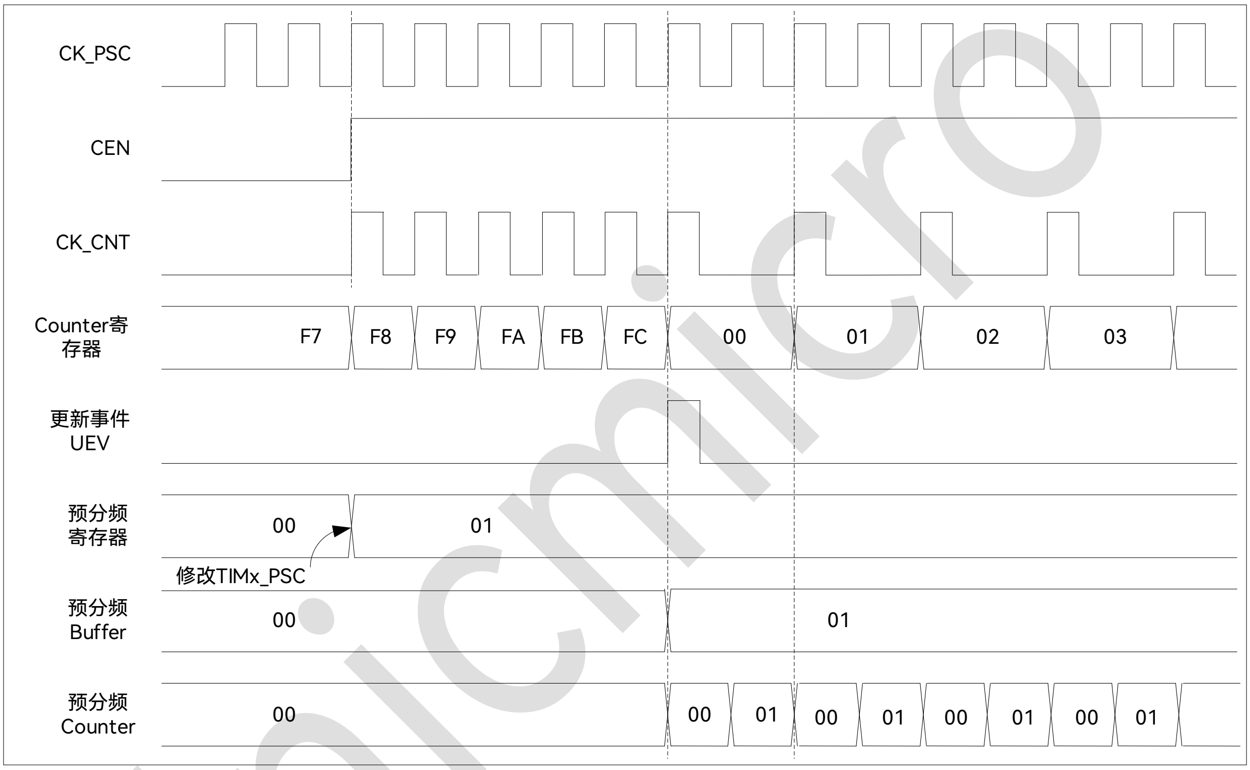


图 13-2：计数器时序图：预分频修改，从 1 到 2

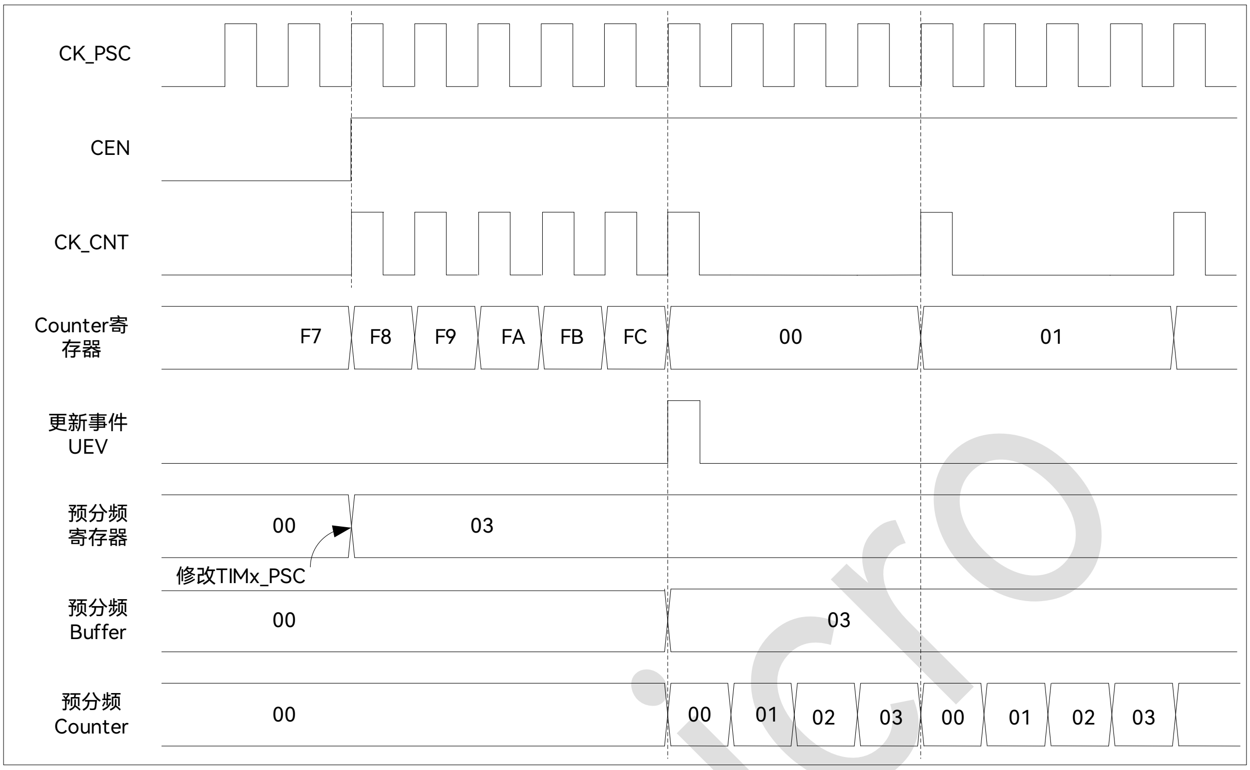


图 13-3：计数器时序图：预分频修改，从 1 到 4

13.3.2 计数器模式

计数器从 0 计数到自动重载值（TIMx_ARR 计数器的值），然后重新从 0 开始计数并且产生一个计数器溢出事件（overflow）。

每次计数器溢出时都会产生更新事件。

在 TIMx_EGR 寄存器中（通过软件方式）置位 UG 位也同样可以产生一个更新事件。

通过软件置位 TIMx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新缓存寄存器。在 UDIS 位被清'0'之前，将不产生更新事件。但是，在应该产生更新事件时，计数器会被清'0'，同时预分频器的计数器也被清 0（但预分频器的数值不变）。

此外，如果置位 TIMx_CR1 寄存器中的 URS 位（update request selection），置位 UG 位将产生一个更新事件 UEV，但硬件不置位 UIF 标志（即不产生中断）。这是为了避免在发生捕获事件（capture event）清除计数器时，同时产生更新和捕获两个中断。

当发生一个更新事件时，所有的寄存器都被更新，同时（依据 URS 位）置位更新标志位（TIMx_SR 寄存器中的 UIF 位）：

- 预分频器的缓冲区被写入预装载寄存器的值（TIMx_PSC 寄存器的值）
- 自动重载缓存寄存器更新为预装载寄存器的值（TIMx_ARR）

举例如下：TIMx_ARR=0x36 时，计数器在不同时钟频率下计数器行为。

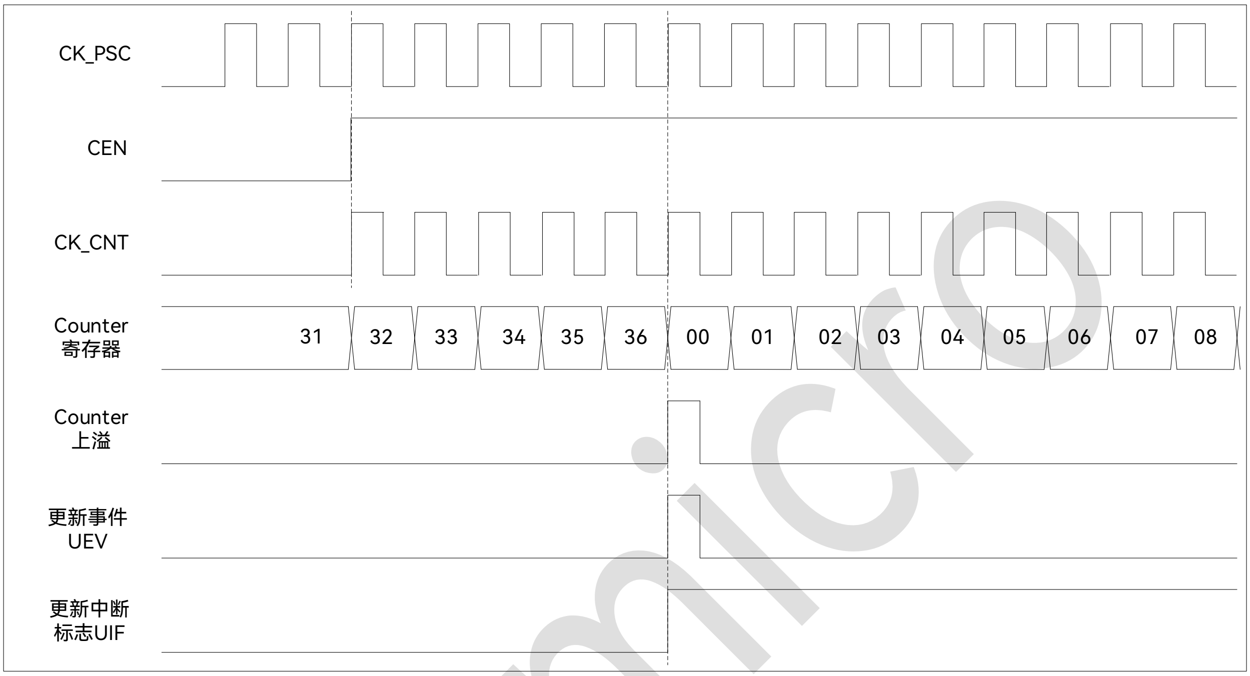


图 13-4：计数器时序图：预分频因子=1

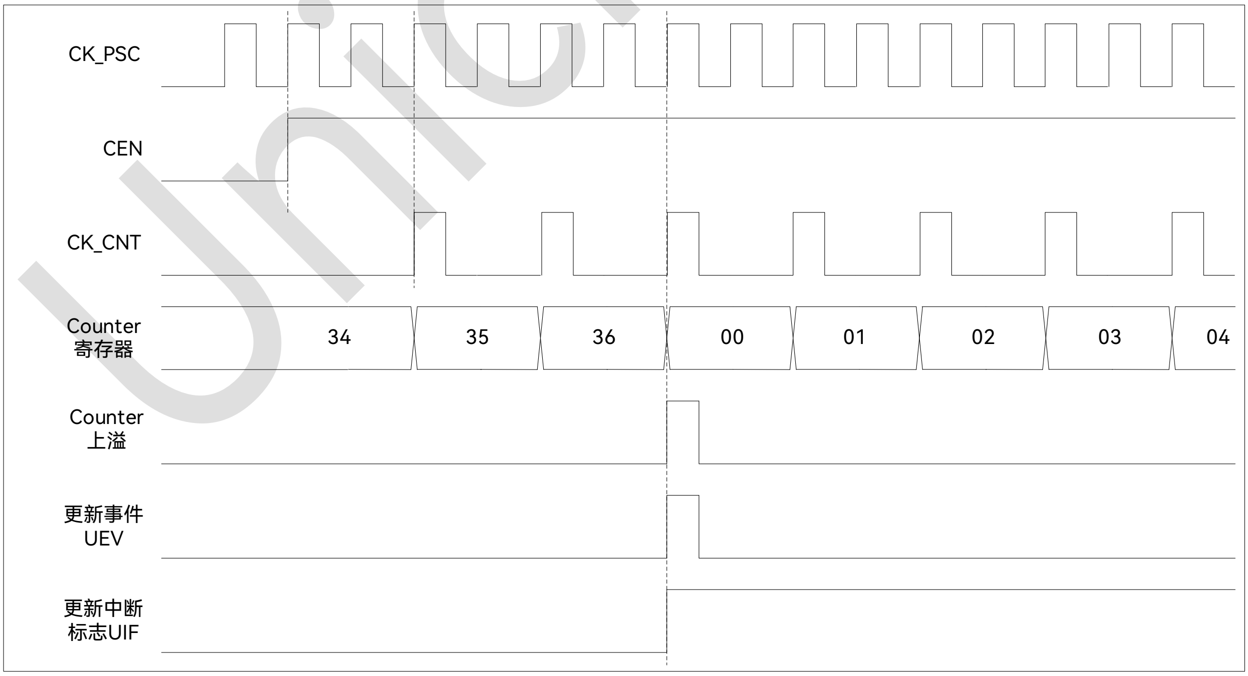


图 13-5：计数器时序图：预分频因子=2

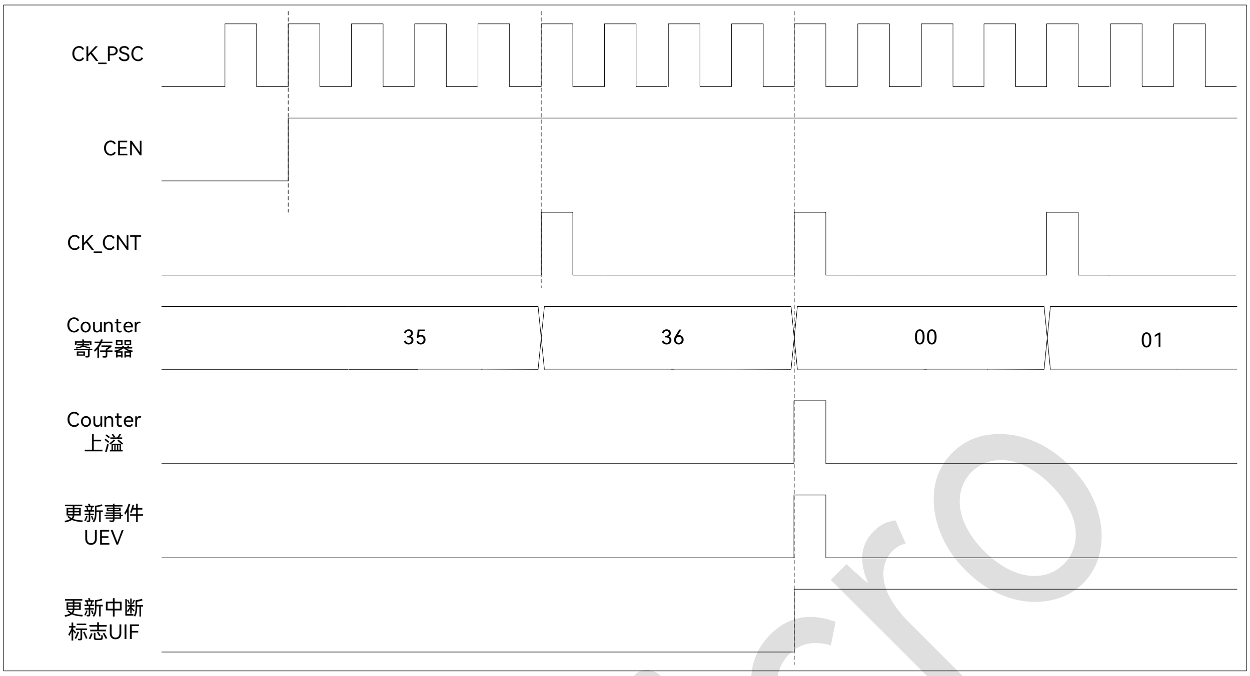


图 13-6：计数器时序图：预分频因子=4

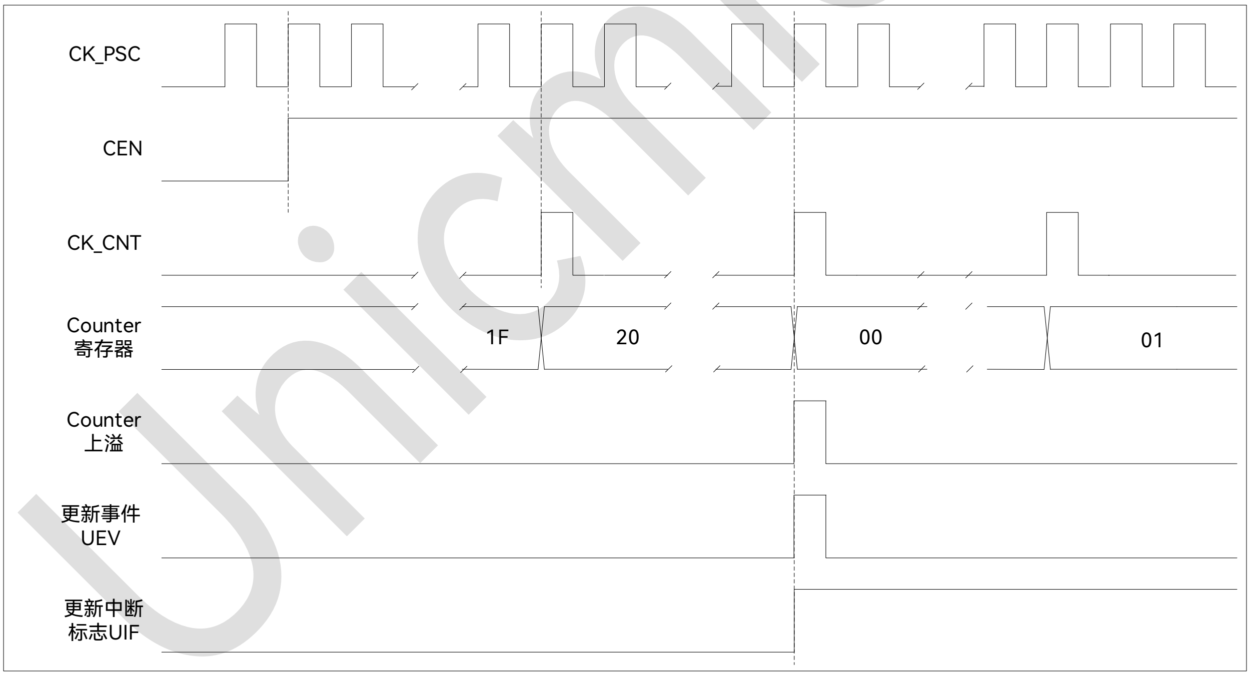


图 13-7：计数器时序图：预分频因子=N

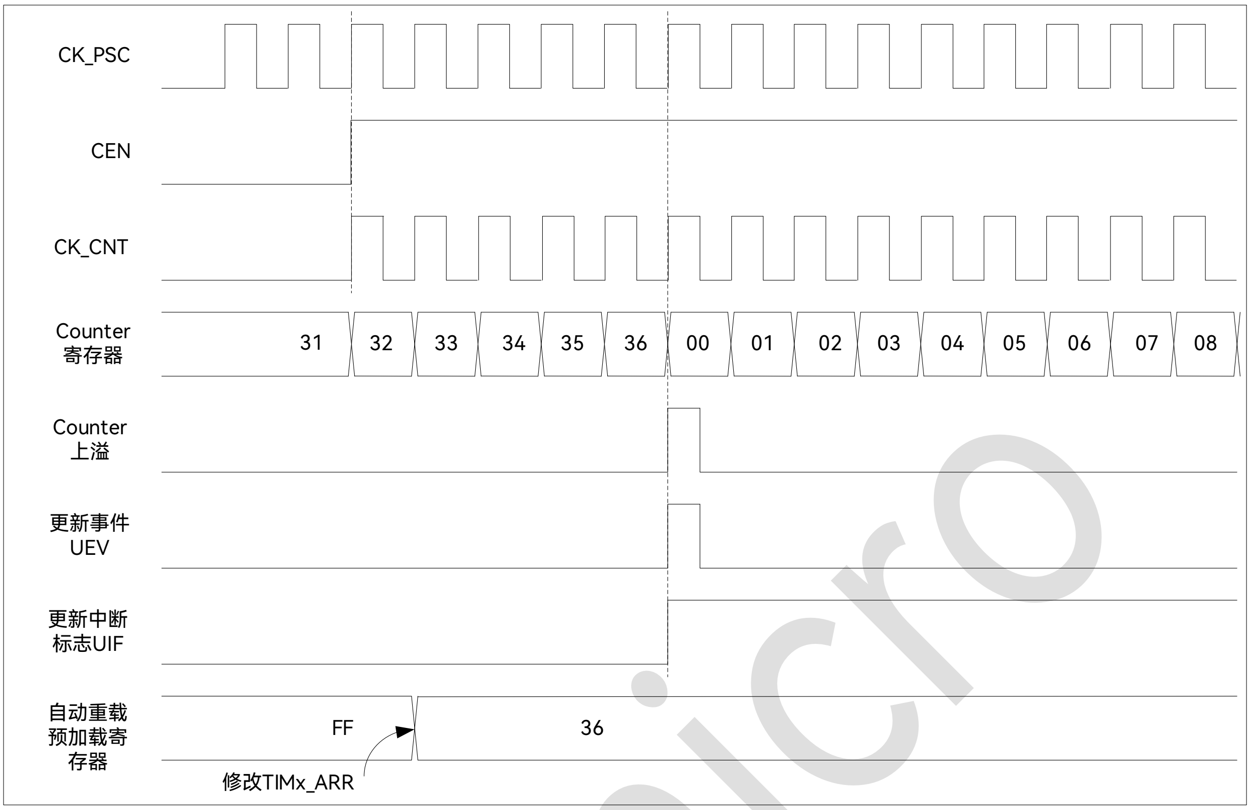


图 13-8：计数器时序图：预分频因子=1，ARPE=0（TIMx_ARR 没有预装载）

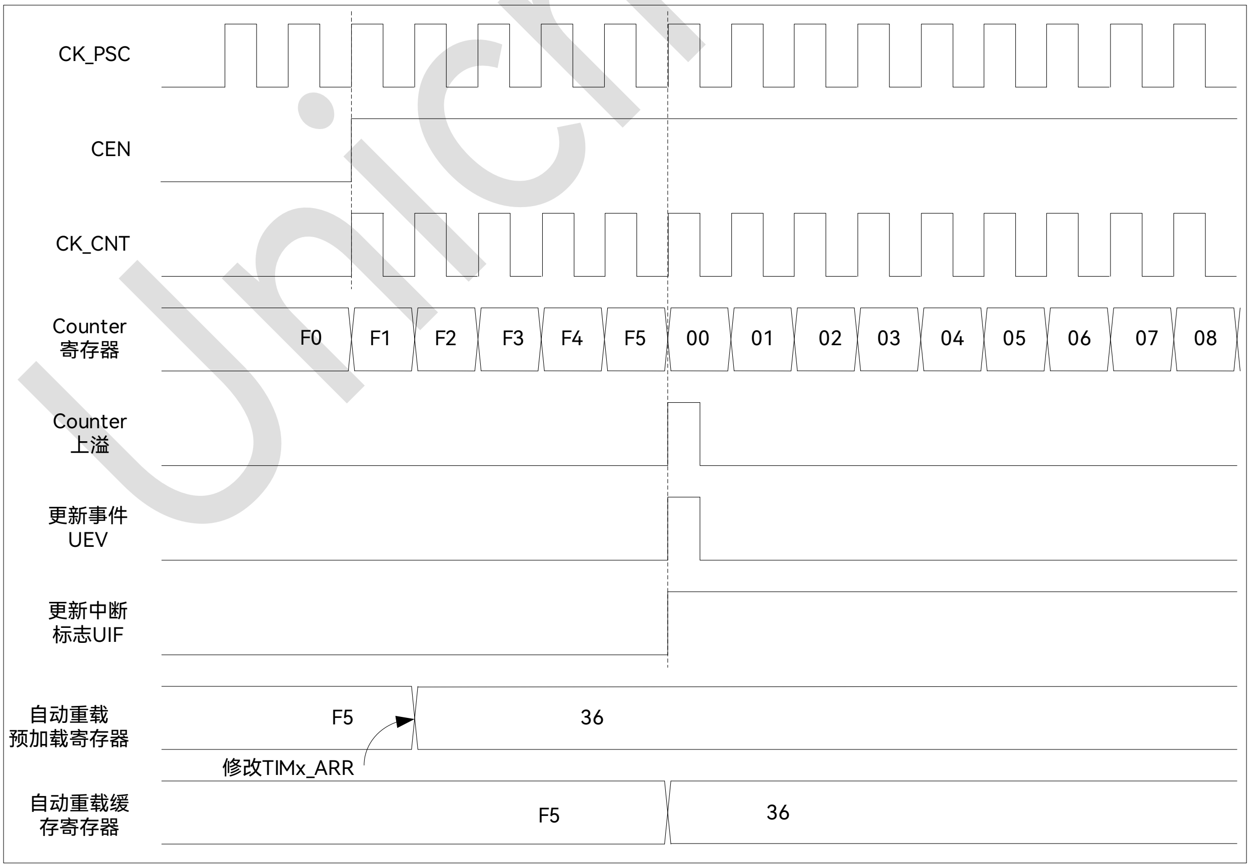


图 13-9：计数器时序图：预分频因子=1，ARPE=1（TIMx_ARR 已预装载）

13.3.3 时钟源

计数器时钟源包括：

- 内部时钟 CK_INT

13.3.3.1 内部时钟源（CK_INT）

CEN（TIMx_CR1 寄存器）和 UG 位（TIMx_ EGR 寄存器）为实际控制位，并且只能被软件修改（除了 UG 位保持自动被清除）。一旦 CEN 位被写成‘1’，预分频器的时钟就由内部时钟 CK_INT 提供。

图 13-10：计数器时序图：内部时钟，预分频因子=1，正常模式

13.3.4 捕获/比较通道

每一个捕获/比较通道都是包括：一个捕获/比较寄存器（包含缓存寄存器），一个用于捕获的输入级（带数字滤波、多路复用选择和预分频器），和一个输出级（带比较器和输出控制）。

13.3.4.1 输入级

对 TIx 的输入信号进行采样，产生一个滤波后 TIxF 信号，经过带有极性选择的边沿检测器生成 TIxFPx 信号，TIxFPx 可以作为捕获命令。该信号预分频后（ICxPS）进入捕获寄存器。

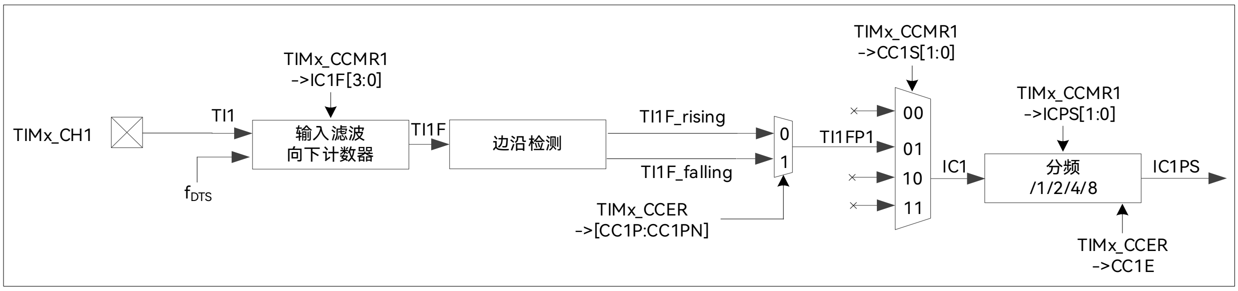


图 13-11：捕获/比较通道输入级（举例：通道 1）

13.3.4.2 输出级

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

捕获/比较模块由一个预装载寄存器（preload register）和一个缓存寄存器（shadow register）组成。读写过程仅操作预装载寄存器。

在捕获模式下，捕获发生在缓存寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到缓存寄存器中，然后缓存寄存器的内容和计数器进行比较。

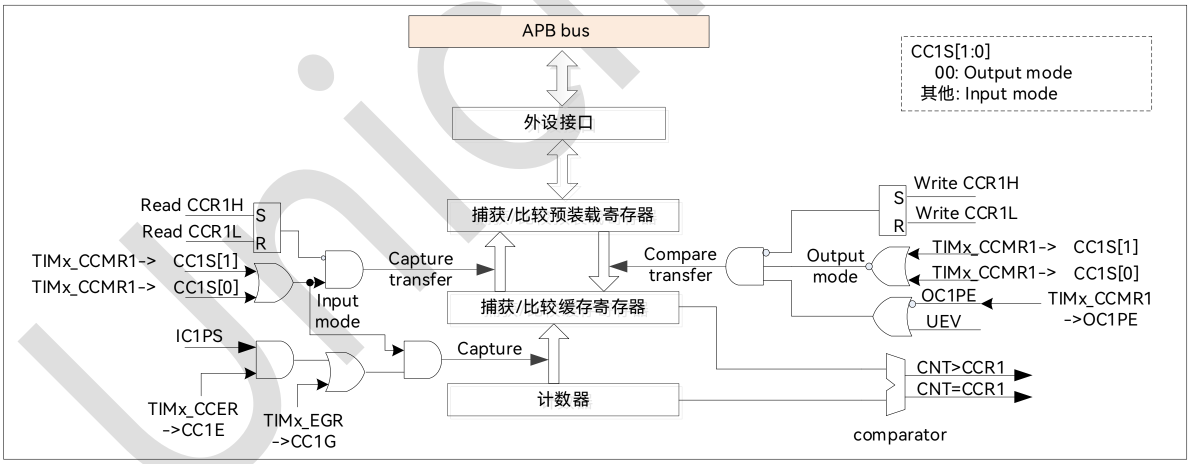


图 13-12：捕获/比较通道主电路

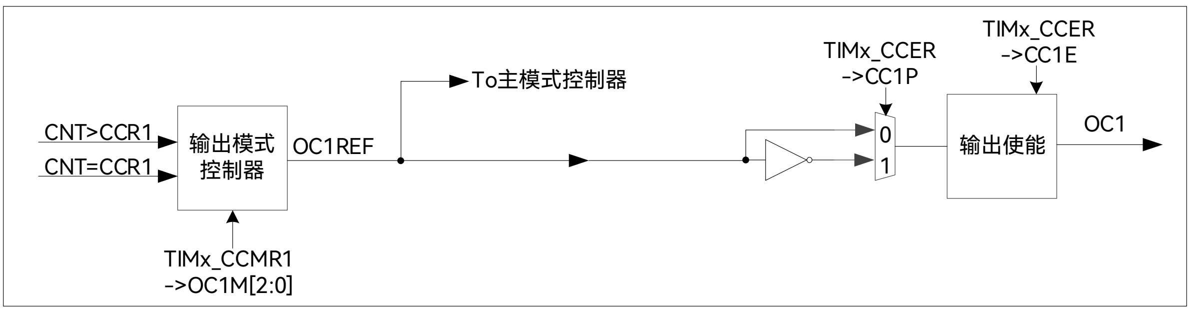


图 13-13：捕获/比较通道输出级（通道 1）

13.3.5 输入捕获模式

在输入捕获模式下，当检测 ICx 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（TIMx_CCRx）中。当发生捕获事件时，相应 CCxIF 标志（TIMx_SR 寄存器）被置 1，如果使能了中断，则产生一个中断请求。如果发生捕获事件时 CCxIF 标志已经为高，那么重复捕获标志 CCxOF（TIMx_SR 寄存器）被置 1。软件写 CCxIF=0 可清除 CCxIF，或读取存储在 TIMx_CCRx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。

举例，在 TI1 输入的上升沿时，捕获计数器的值到 TIMx_CCR1 寄存器中，操作步骤如下:

1. 选择 TIMx_CCR1 的有效输入：置 TIMx_CCMR1 寄存器的 CC1S=01（选中 TI1），只要 CC1S 不为‘00’，通道被配置为输入并且 TIMx_CCR1 寄存器变为只读。
2. 根据连接到计数器的输入信号，配置输入滤波器时间（输入为 TIx 时，输入滤波器控制位是 TIMx_CCMRx 寄存器中的 ICxF 位）。举例，当 TI1 翻转时，信号抖动最多 5 个内部时钟，则须配置滤波器时间大于 5 个时钟周期，TIMx_CCMR1 寄存器中写入 IC1F=0011 配置采样次数 8（以 f_{DTS} 频率采样），通过连续 8 次采样以确认电平变换。
3. 选择 TI1 通道有效沿。写 TIMx_CCER 寄存器中 CC1P=0 和 CC1NP=0（本例中为上升沿）。
4. 配置输入预分频器。在本例中，设置为每个有效的电平转换时发生一次捕获，因此预分频器被禁止（写 TIMx_CCMR1 寄存器的 IC1PS=00）。
5. 写 TIMx_CCER 寄存器的 CC1E=1，允许计数器的值被捕获至捕获寄存器中。
6. 根据需要，置位 TIMx_DIER 寄存器中的 CC1IE 位允许相关中断请求。

当发生一个输入捕获时:

- 有效的电平转换发生时，计数器的值被传送至 TIMx_CCR1 寄存器。
- CC1IF 标志被置位（中断标志）。当发生至少 2 个连续的捕获时，且 CC1IF 未被清除，CC1OF 也被置位。
- 若置位 CC1IE 位，则会产生一个中断。

为了处理捕获溢出，建议在读出捕获溢出标志之前读取数据，这是为了避免在读取标志之后和读取数据之前可能发生的捕获溢出。

注：通过软件设置 TIMx_EGR 寄存器中相应的 CCxG 位，也可以产生输入捕获中断请求。

13.3.6 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中 CCxS=00）下，输出比较信号（OCxREF 和相应的 OCx/OCxN）能够直接由软件强制为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

设置 TIMx_CCMRx 寄存器中相应的 OCxM=101，输出比较信号（OCxREF/OCx）强制为有效状态。这样 OCxREF 被强制高电平（OCxREF 始终为高电平有效），置位 TIMx_CCER 的 CCxP 位，OCx 可得到极性相反的信号。

举例，CCxP=0（OCx 高电平有效），则 OCx 被强制为高电平。

设置 TIMx_CCMRx 寄存器中相应的 OCxM=100，输出比较信号（OCxREF/OCx）强制为低电平。

该模式下，在 TIMx_CCRx 缓存寄存器和计数器之间的比较仍然在执行，相应的标志也会被置位。并会产生相应的中断请求。下面的输出比较模式一节中将详细介绍。

13.3.7 输出比较模式

此模式用于控制输出波形或指示一段时间到时。

当计数器与捕获/比较寄存器的内容匹配（相等）时，输出比较功能做如下操作：

- 根据输出比较模式（TIMx_CCMRx 寄存器中 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）的配置，输出至对应引脚。在比较匹配发生时，输出引脚可以保持它的电平(OCxM=000)、被设置成有效电平（OCxM=001）、被设置成无效电平（OCxM=010）或进行翻转（OCxM=011）。
- 置位中断状态寄存器中的标志位（TIMx_SR 寄存器中的 CCxIF 位）。
- 若置位相应的中断屏蔽位（TIMx_DIER 寄存器中的 CCxIE 位），则产生一个中断。

设置 TIMx_CCMRx 中的 OCxPE 位选择 TIMx_CCRx 寄存器是否使用预装载寄存器。

在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。计时分辨率为计数器的一次计数。

输出比较模式也能用来输出一个单脉冲（单脉冲模式）。

输出比较模式的配置步骤：

1. 选择计数器时钟（内部，外部，预分频器）。
2. 根据需求，写入数据至 TIMx_ARR 和 TIMx_CCRx 寄存器中。
3. 如果要产生一个中断请求，设置 CCxIE 位。
4. 选择输出模式，举例如下：
   - 写 OCxM=011，计数器与 CCRx 匹配时，翻转 OCx 的输出引脚。
   - 写 OCxPE=0，禁用预装载寄存器。
   - 写 CCxP=0，选择极性为高电平有效。
   - 写 CCxE=1，使能输出。
5. 置位 TIMx_CR1 寄存器的 CEN 位，启动计数器。

TIMx_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，前提是未启用预装载寄存器（OCxPE='0'，否则 TIMx_CCRx 的缓存寄存器只在发生下次更新事件 EUV 时更新），如下图：

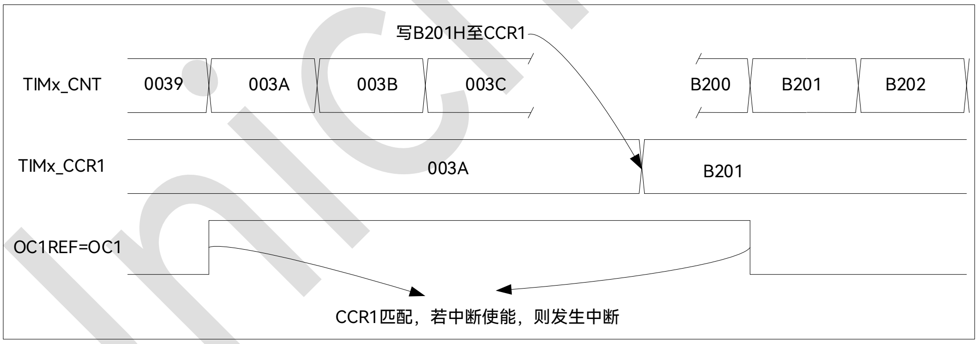


图 13-14：输出比较模式，配置为：翻转 OC1

13.3.8 PWM 模式

脉冲宽度调制模式（PWM）可以产生一个 PWM 信号，由 TIMx_ARR 寄存器确定频率、由 TIMx_CCRx 寄存器确定占空比。

在 TIMx_CCMRx 寄存器中的 OCxM 位写入'110'（PWM 模式 1）或'111'（PWM 模式 2），可以在每个通道独立地设置 PWM 模式（每个 OCx 输出一种 PWM）。必须置位 TIMx_CCMRx 寄存器

的 OCxPE 位，使能相应的预装载寄存器（preload register），最后置位 TIMx_CR1 寄存器的 ARPE 位，使能自动重载预装载寄存器（在向上计数模式中）。

只有发生一个更新事件时，预装载寄存器才能被传送到缓存寄存器，因此在计数器开始计数之前，必须通过设置 TIMx_EGR 寄存器中的 UG 位来初始化所有的寄存器。

设置 TIMx_CCER 寄存器中的 CCxP 位，配置 OCx 的极性为高电平有效或低电平有效。OCx 的输出使能通过（TIMx_CCER 寄存器中）CCxE 位控制。详细参考 TIMx_CCER 寄存器的描述。

在 PWM 模式（模式 1 或模式 2）下，TIMx_CNT 和 TIMx_CCRx 始终在进行比较，以确定是否符合 TIMx_CNT ≤ TIMx_CCRx。

由于此计数器为向上计数，仅支持边沿对齐模式的 PWM 信号。

13.3.8.1 PWM 边沿对齐模式

在 PWM 模式 1，当 TIMx_CNT<TIMx_CCRx 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR），则 OCxREF 保持为‘1’。如果比较值为 0，则 OCxREF 保持为‘0’。下图为 TIMx_ARR=8 时边沿对齐的 PWM 波形实例。

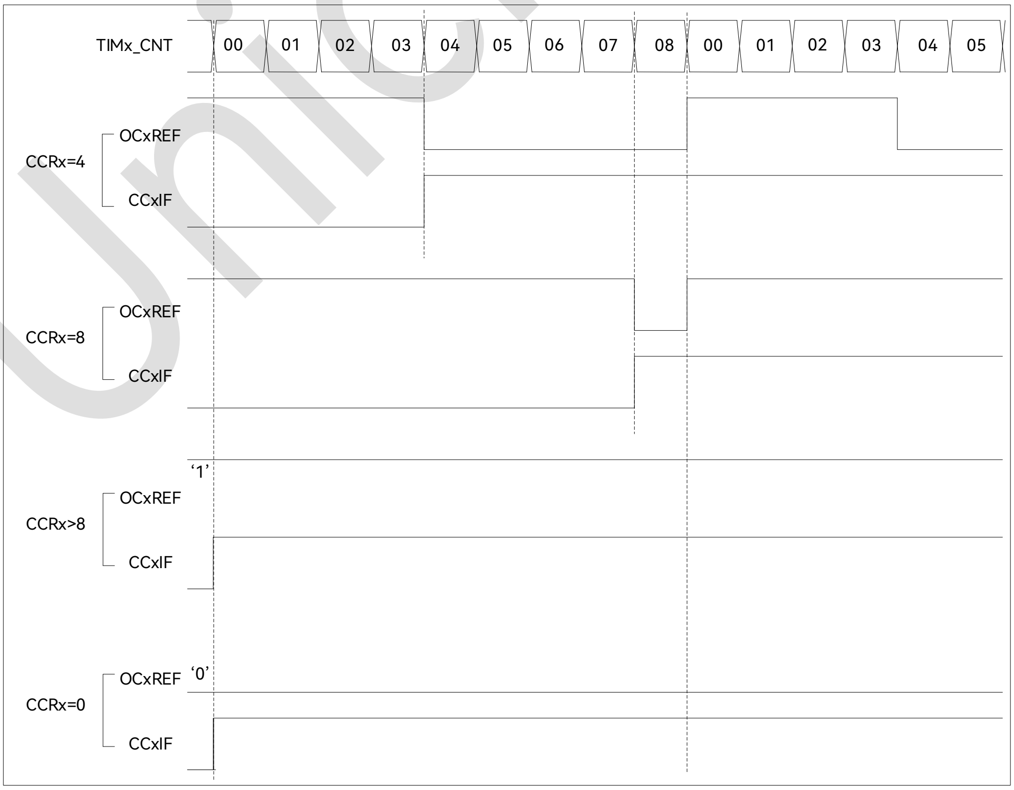


图 13-15：PWM 波形（边沿对齐，向上计数，ARR=8）

13.3.9 调试模式

当微控制器进入调试模式时（Cortex-M0 内核停止），根据 DBG 模块中 DBG_TIMx_STOP 的设置，TIMx 计数器可以或者继续正常操作，或者停止。

13.4 相关寄存器

13.4.1 TIM 控制寄存器 1（TIM14_CR1）

偏移地址：0x000

复位值：0x0000 0000

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

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

位	名称	描述
31:10	RSV	保留位，固定读为 0
9:8	CKD[1:0]	时钟分频（Clock division），通过修改此位段可以对 CK_INT 时钟进行分频，分频后的时钟作为输入滤波（Tix）采样的基准时钟。 00：$t_{DTS} = t_{CK\_INT}$ 01：$t_{DTS} = 2 * t_{CK\_INT}$ 10：$t_{DTS} = 4 * t_{CK\_INT}$ 11：保留，禁止使用
7	ARPE	自动重载预装载使能（Auto-reload preload enable）: 0：TIMx_ARR 寄存器没有缓冲 1：TIMx_ARR 寄存器有缓冲
6:3	RSV	保留位，固定读为 0
2	URS	更新请求源（Update request source）: 软件置位和清零，选择 UEV 事件的源 0：下述任一事件产生更新中断： - 计数器上溢

位	名称	描述
		- 置位 UG 位 1：仅计数器上溢产生更新中断。
1	UDIS	禁止更新（Update disable）： 软件置位和清零，通过该位允许/禁止 UEV 事件的产生 0：允许 UEV。更新（UEV）事件由下述任一事件产生: - 计数器上溢 - 置位 UG 位 具有缓存的寄存器被装入它们的预装载值。 1：禁止 UEV。不产生更新事件，影子寄存器（ARR、PSC、CCRx）保持它们的值。如果置位 UG 位，则计数器和预分频器被重新初始化。
0	CEN	使能计数器（Counter enable）： 0：禁止计数器 1：使能计数器 注：只有软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。但触发模式可以自动地通过硬件设置 CEN 位。

13.4.2 TIM DMA/中断使能寄存器（TIM14_DIER）

偏移地址：0x00C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														CC1IE	UIE
														rw	rw

位	名称	描述
31:2	RSV	保留位，固定读为 0
1	CC1IE	捕获/比较 1 中断使能: 0: CC1 中断禁止 1: CC1 中断允许
0	UIE	更新中断使能: 0：更新中断禁止 1：更新中断允许

13.4.3 TIMx 中断标志和清除寄存器（TIM14_SR）

偏移地址：0x010

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						CC1OF	Res.							CC1IF	UIF
						rc_w0								rc_w0	rc_w0

位	名称	描述
31:10	RSV	保留位，固定读为 0
9	CC1OF	捕获/比较 1 重复捕获标志（Capture/Compare 1 overcapture flag）： 仅当相应的通道被配置为输入捕获时，该标志可由硬件置 1。 软件写 0 可清除该位。 0：无重复捕获产生 1：当 CC1IF 的状态已经为‘1’，计数器的值被捕获到 TIMx_CCR1 寄存器
8:2	RSV	保留位，固定读为 0
1	CC1IF	捕获/比较 1 中断标志： **如果通道 CC1 配置为输出模式：** 当计数器值与比较值匹配时该位由硬件置 1。 它由软件清‘0’。 0：无匹配发生 1：TIMx_CNT 的值与 TIMx_CCR1 的值匹配。当 TIMx_CCR1 的内容大于 TIMx_APR 的内容时，在向上计数模式时计数器上溢条件下，CC1IF 位变高 **如果通道 CC1 配置为输入模式：** 当捕获事件发生时该位由硬件置‘1’，它由软件清‘0’或通过读 TIMx_CCR1 清‘0’。 0：无输入捕获产生 1：计数器值被捕获至 TIMx_CCR1（在 IC1 上检测到与所选极性相同的边沿）
0	UIF	更新中断标志（Update interrupt flag）： 当产生更新事件时该位由硬件置‘1’。它由软件清‘0’。 0：无更新事件产生； 1：更新中断等待响应。当寄存器被更新时该位由硬件置‘1’： - 若 TIMx_CR1 寄存器的 URS=0、UDIS=0，当设置 TIMx_EGR 寄存器的 UG=1 进而实现软件对计数器 CNT 重新初始化时。

13.4.4 TIMx 事件产生寄存器（TIM14_EGR）

偏移地址：0x014

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														CC1G	UG
														rs	rs

位	名称	描述
31:2	RSV	保留位，固定读为 0
1	CC1G	捕获/比较 1 发生（Capture/Compare 1 generation)： 该位由软件置‘1’，用于产生一个捕获/比较事件，由硬件自动清‘0’。 0：无动作 1：在通道 1 上产生一个捕获/比较事件 若通道 CC1 配置为输出： 设置 CC1IF=1，若开启对应的中断，则产生相应的中断。 若通道 CC1 配置为输入： 当前的计数器值被捕获至 TIMx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断，则产生相应的中断。若 CC1IF 已经为 1，则设置 CC1OF=1。
0	UG	产生更新事件（Update generation)： 该位由软件置‘1’，由硬件自动清‘0’。 0：无动作 1：重新初始化计数器，并产生一个（寄存器）更新事件。注意预分频器的计数器也被清‘0’（但是预分频系数不变）。

13.4.5 TIMx 捕获/比较模式寄存器 1[复用]（TIM14_CCMR1）

偏移地址：0x018

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									OC1M[2:0]			OC1PE	Res.	CC1S[1:0]	
									rw			rw		rw	

位	名称	描述
31:7	RSV	保留位，固定读为 0
6:4	OC1M[2:0]	输出比较模式 1 (Output Compare 1 mode): 该 3 位定义了输出参考信号 OC1REF 的行为，而 OC1REF 决定了 OC1、OC1N 的电平。OC1REF 是高电平有效，而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用（此模式用于产生一个时基）。 001：匹配时，设置通道 1 为有效电平。当 TIMx_CNT = TIMx_CCR1 时，强制 OC1REF 为高。 010：匹配时，设置通道 1 为无效电平。当 TIMx_CNT = TIMx_CCR1 时，强制 OC1REF 为低。 011：翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。 100：强制为无效电平。强制 OC1REF 为低。 101：强制为有效电平。强制 OC1REF 为高。 110：PWM 模式 1 - 在向上计数时，当 TIMx_CNT<TIMx_CCR1 时，通道 1 为有效电平（OC1REF=1），否则为无效电平（OC1REF=0）。 111：PWM 模式 2 - 在向上计数时，当 TIMx_CNT<TIMx_CCR1 时，通道 1 为无效电平，否则为有效电平。 注 1：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。
3	OC1PE	输出比较 1 预装载允许（Output Compare 1 preload enable): 0：禁止 TIMx_CCR1 寄存器的预装载功能，可随时写入 TIMx_CCR1 寄存器，

位	名称	描述
		并且新写入的数值立即起作用。 1：开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。
2	RSV	保留位，固定读为 0
1:0	CC1S[1:0]	捕获/比较 1 选择（Capture/Compare 1 selection）： 这 2 位定义通道的方向（输入/输出），及输入信号的选择： 00：CC1 通道被配置为输出 01：CC1 通道被配置为输入，IC1 映射在 TI1 上 10：保留 11：保留 注：CC1S 仅在通道关闭时（TIMx_CCER 寄存器的 CC1E=0）才是可写的。

13.4.6 TIMx 捕获/比较模式寄存器 1（影子寄存器）【复用】（TIM14_CCMR1_S）

偏移地址：0x018

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
								rw				rw		rw	

位	名称	描述
31:8	RSV	保留位，固定读为 0
7:4	IC1F[3:0]	输入捕获 1 滤波器（Input capture 1 filter）： 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变： 0000：无滤波器，以 f_{DTS} 采样 0001：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=2 0010：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=4 0011：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=8 0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6

位	名称	描述
		0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6 1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8 1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5 1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8 注：当 ICxF[3:0]=1、2 或 3 时，公式中的 f_{DTS} 由 CK_INT 替代。
3:2	IC1PSC[1:0]	输入捕获 1 预分频器： 这 2 位定义了 CC1 输入（IC1）的预分频系数。一旦 CC1E=0 (TIMx_CCER 寄存器中)，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获； 01：每 2 个事件触发一次捕获； 10：每 4 个事件触发一次捕获； 11：每 8 个事件触发一次捕获。
1:0	CC1S[1:0]	捕获/比较 1 选择（Capture/Compare 1 selection） 这 2 位定义通道的方向（输入/输出），及输入信号的选择： 00：CC1 通道被配置为输出 01：CC1 通道被配置为输入，IC1 映射在 TI1 上 10：保留 11：保留 注：CC1S 仅在通道关闭时（TIMx_CCER 寄存器的 CC1E=0）才是可写的。

13.4.7 TIMx 捕获/比较使能寄存器（TIM14_CCER）

偏移地址：0x020

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												CC1NP	Res.	CC1P	CC1E
												rw		rw	rw

位	名称	描述
31:4	RSV	保留位，固定读为 0
3	CC1NP	捕捉/比较 1 输出极性（Capture/Compare 1 complementary output Polarity）: 通道 CC1 配置为输出时，CC1NP 必须设置为清除，CC1NP = 0 通道 CC1 配置为输入时，CC1NP 与 CC1P 联合控制 TI1FP1 的极性 参考 CC1P 描述。
2	RSV	保留位，固定读为 0
1	CC1P	捕获/比较 1 输出极性（Capture/Compare 1 output polarity）: **CC1 通道配置为输出:** 0：OC1 高电平有效; 1：OC1 低电平有效。 **CC1 通道配置为输入:** CC1NP/CC1P 位选择在触发或捕获模式下 TI1FP1 和 TI2FP1 的有效极性。 00: 非反相/上升沿 - TIxFP1 的上升沿 - TIxFP1 非反相 01: 反相/下降沿 - TIxFP1 的下降沿 - TIxFP1 反相 10: 保留，此配置不用 11: 非反相/上升或下降沿 - TIxFP1 的上升沿和下降沿 - TIxFP1 非反相
0	CC1E	捕获/比较 1 输出使能（Capture/Compare 1 output enable）: **CC1 通道配置为输出:** 0：关闭 - OC1 未激活 1：开启 - OC1 信号输出到对应的输出引脚 **CC1 通道配置为输入:** 本位用于决定捕获的计数器值是否要装载到捕获/比较寄存器 1（TIMx_CCR1）。 0: 捕获禁止 1: 捕获允许

13.4.8 TIMx 计数器（TIM14_CNT）

偏移地址：0x024

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	CNT[15:0]	计数器值

13.4.9 TIMx 预分频器（TIM14_PSC）

偏移地址：0x028

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	PSC[15:0]	预分频值 计数器的时钟频率 CK_CNT=f_{CK_PSC}/(PSC[15:0]+1)。PSC 包含了当更新事件产生时装入当前预分频器寄存器的值。

13.4.10 TIMx 自动重装载寄存器（TIM14_ARR）

偏移地址：0x02C

复位值：0x0000 FFFF

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	ARR[15:0]	自动重装载的值（auto-reload value） ARR 是将要装载入的实际自动重载寄存器的值。详细参考<时基单元>章节：有关 ARR 的更新和动作。 当自动重装载的值为空时，计数器不工作。

13.4.11 TIMx 捕获/比较寄存器 1（TIM14_CCR1）

偏移地址：0x034

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw															

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	CCR1[15:0]	捕获/比较通道 1 的值 **若 CC1 通道配置为输出：** CCR1 是要加载到实际捕获/比较 1 寄存器的值（预装载值）。

位	名称	描述
		如果在 TIMx_CCMR1 寄存器（OC1PE 位）中未选择预装载功能，写入的数值会永久加载。否则只有当更新事件发生时，此预装载值才加载至内部活动的捕获/比较 1 寄存器中。 活动的捕获/比较寄存器与计数器 TIMx_CNT 进行比较，并在 OC1 端口上产生输出信号。 **若 CC1 通道配置为输入：** CCR1 是上一次输入捕获 1 事件（IC1）捕获的计数器值。

13.4.12 TIMx 选项寄存器（TIM14_OR）

偏移地址：0x050

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														TI1_RMP[1:0]	
														rw	

位	名称	描述
31:2	RSV	保留位，固定读为 0
1:0	TI1_RMP[1:0]	定时器输入 1 重映射（Timer Input 1 remap），软件清除和置位 00：TIM14 通道 1 连接到 GPIO：参考芯片数据手册中关于复用功能映射（Alternate function mapping） 01：保留 10：保留 11：MCO 连接到 TIM14_CH1 输入端，MCO 时钟源的选择由时钟配置寄存器（RCC_CFGR）的 MCO[3:0]选择。

14 通用定时器（TIM18）

14.1 概述

TIM18 基于一个 16 位自动重载递增计数器和一个 16 位预分频。它具有 3 个独立通道，用于输入捕获/输出比较、PWM、单脉冲模式输出。

TIM18 能处理 1 到 3 个霍尔效应传感器的数字输出。

在调试模式下，其计数器可被冻结。

14.2 特性

- 16 位向上自动装载计数器
- 16 位可编程（可实时修改）预分频器，支持 1 ~ 65536 之间的任意分频
- 多达 3 个独立通道，支持
  - 输入捕获
  - 输出比较
  - PWM 生成
- 下述事件触发中断：
  - 更新（Update）：计数器上溢，计数器初始化（通过软件或内部/外部触发）
  - 触发事件（Trigger Event）：计数器启动、停止、初始化或由内部/外部触发计数
  - 输入捕获（Input capture）
  - 输出比较（Output compare）
  - 支持霍尔传感器定位解码功能

14.3 功能描述

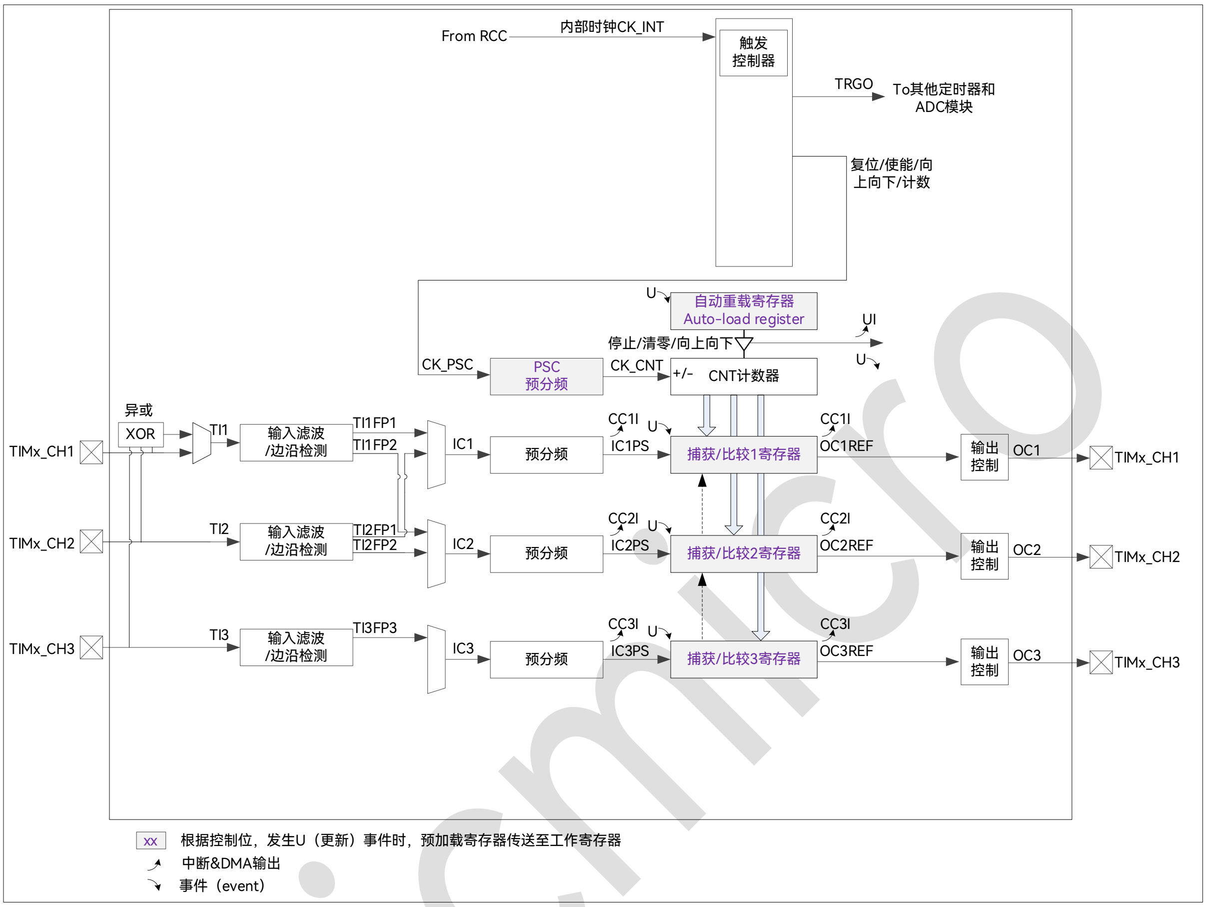


图 14-1：TIM18 模块框图

14.3.1 时基单元

通用定时器 TIM18 主要由 16 位计数器及相关自动重载寄存器构成，计数器支持向上计数。计数器时钟支持预分频。

计数器寄存器、自动重载寄存器和预分频寄存器支持软件读写，即使计数器正在运行读写仍然有效。

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，读写自动重载寄存器将访问预装载寄存器。设置 TIMx_CR1 寄存器中的自动重载预装载使能位（ARPE），选择预装载寄存器的内容永久传送至缓存寄存器或在每次更新事件（UEV）传送至缓存寄存器。当计数器达到溢出条件并当 TIMx_CR1 寄存器中的 UDIS 位等于 0 时，产生更新事件。更新事件也可由软件产生。有关更新事件的产生，针对每种配置后续章节会详细描述。

计数器时钟由预分频后输出 CK_CNT 驱动，需置位 TIMx_CR1 寄存器中的计数器使能位（CEN）时，CK_CNT 才有效。

注：设置 TIMx_CR 寄存器的 CEN 位，计数器会立即开始计数。

14.3.1.1 预分频

预分频器支持计数器时钟 1~65536 之间任意分频，基于 1 个 16 位的计数器，由 TIMx_PSC 寄存器中的 16 位寄存器控制。此寄存器内部带有缓存器，支持运行时修改；新修改的预分频值在下一次的更新事件发生时生效。

下图给出运行时更改预分频值，计数器的行为：

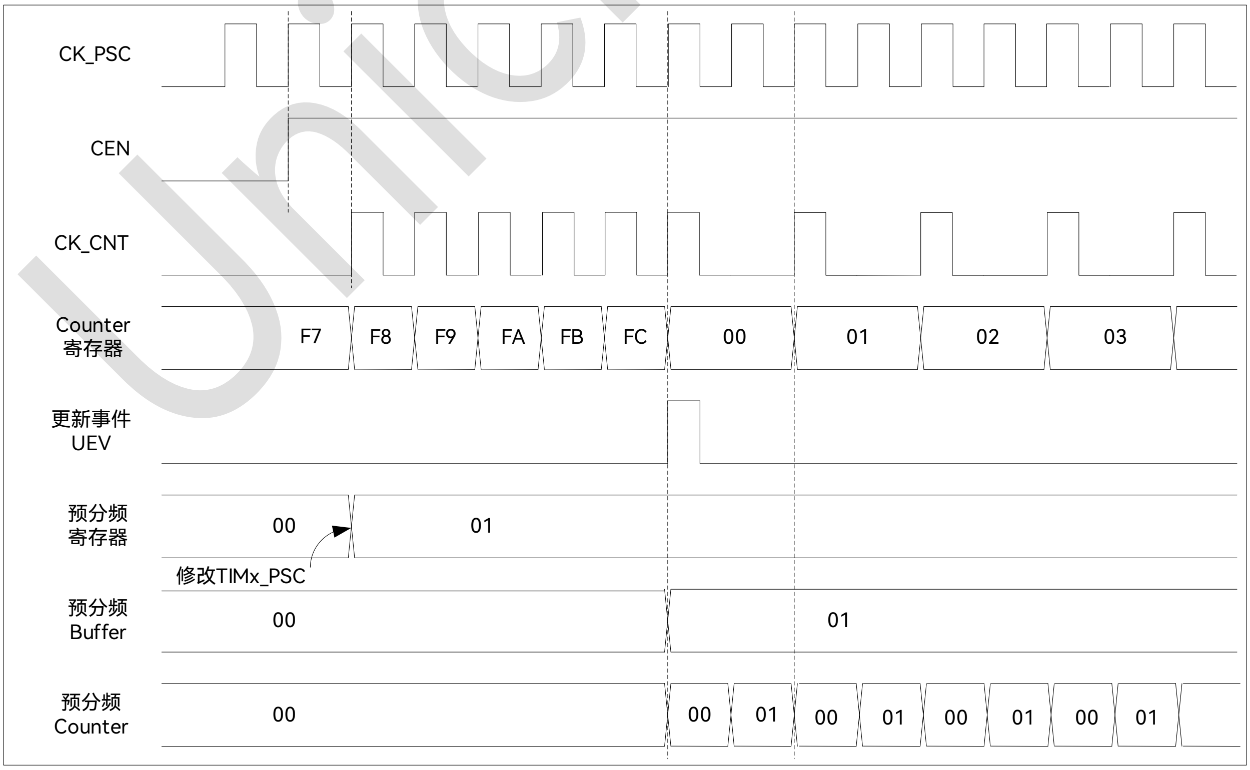


图 14-2：计数器时序图：预分频修改，从 1 到 2

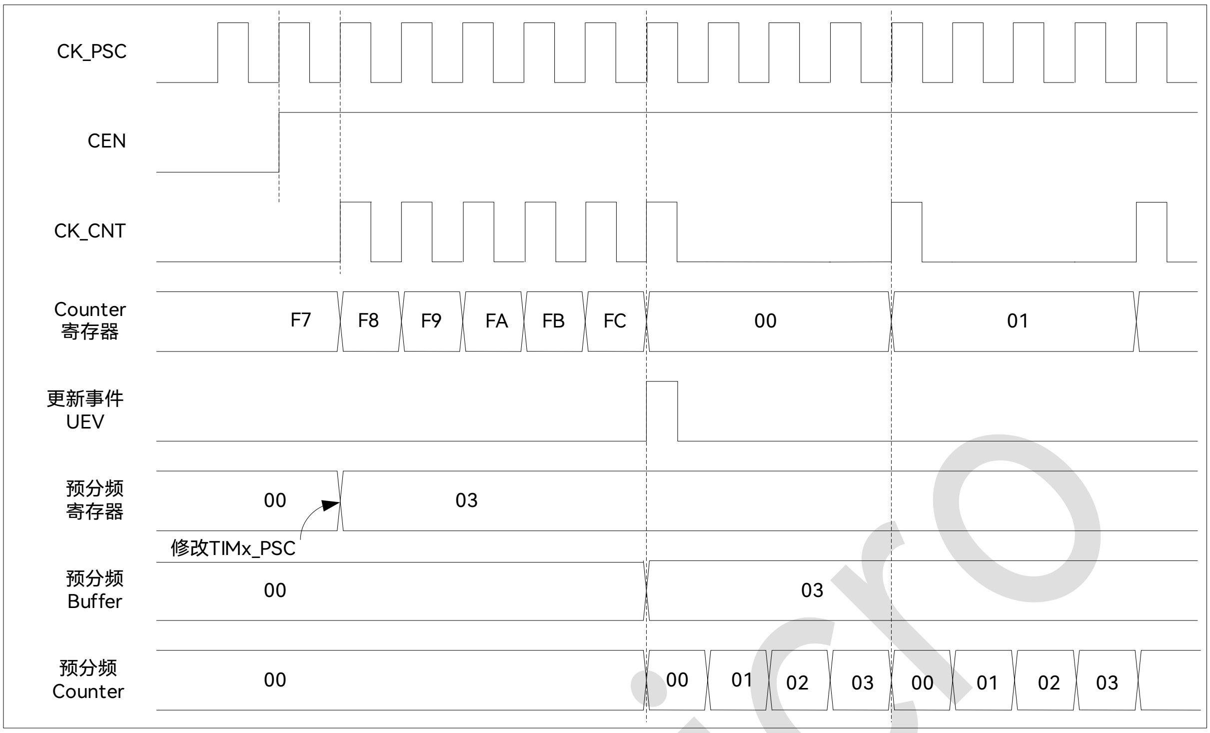


图 14-3：计数器时序图：预分频修改，从 1 到 4

14.3.2 计数器模式

计数器从 0 计数到自动重载值（TIMx_ARR 计数器的值），然后重新从 0 开始计数并且产生一个计数器溢出事件（overflow）。

每次计数器溢出时都会产生更新事件。

在 TIMx_EGR 寄存器中（通过软件方式）置位 UG 位也同样可以产生一个更新事件。

通过软件置位 TIMx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新缓存寄存器。在 UDIS 位被清‘0’之前，将不产生更新事件。但是，在应该产生更新事件时，计数器会被清‘0’，同时预分频器的计数器也被清 0（但预分频器的数值不变）。

此外，如果置位 TIMx_CR1 寄存器中的 URS 位（update request selection），置位 UG 位将产生一个更新事件 UEV，但硬件不置位 UIF 标志（即不产生中断）。这是为了避免在发生捕获事件（capture event）清除计数器时，同时产生更新和捕获两个中断。

当发生一个更新事件时，所有的寄存器都被更新，同时（依据 URS 位）置位更新标志位（TIMx_SR 寄存器中的 UIF 位）:

- 预分频器的缓冲区被写入预装载寄存器的值（TIMx_PSC 寄存器的值）
- 自动重载缓存寄存器更新为预装载寄存器的值（TIMx_ARR）

举例如下：TIMx_ARR=0x36 时，计数器在不同时钟频率下的行为。

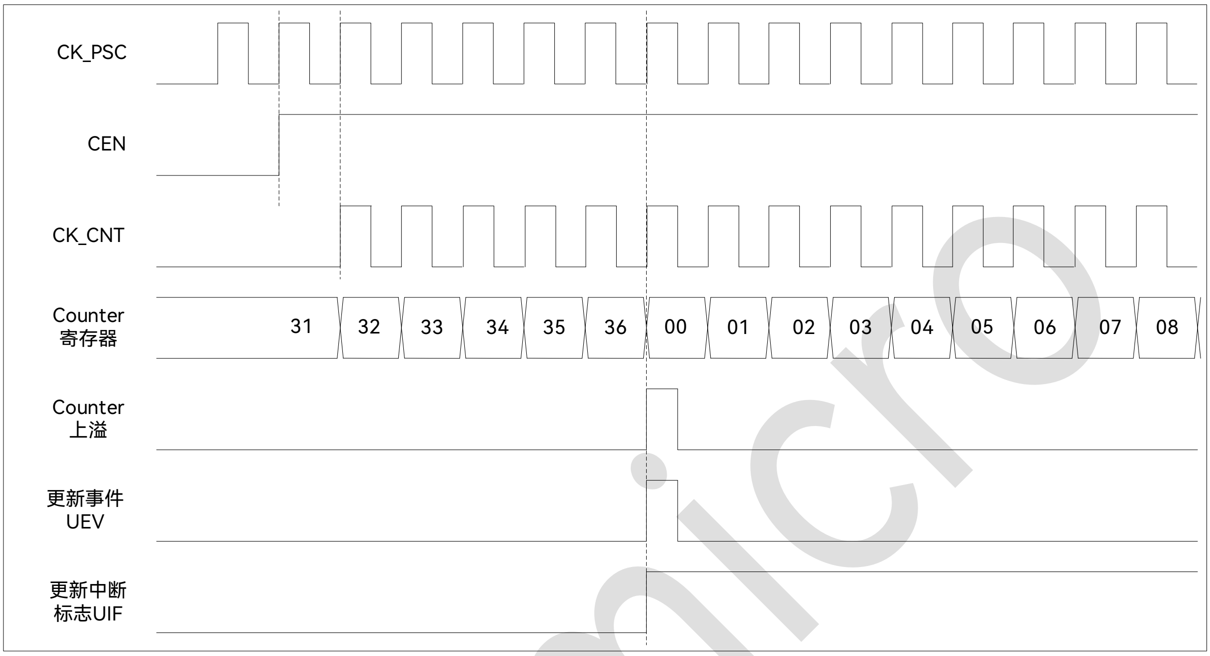


图 14-4：计数器时序图：预分频因子=1

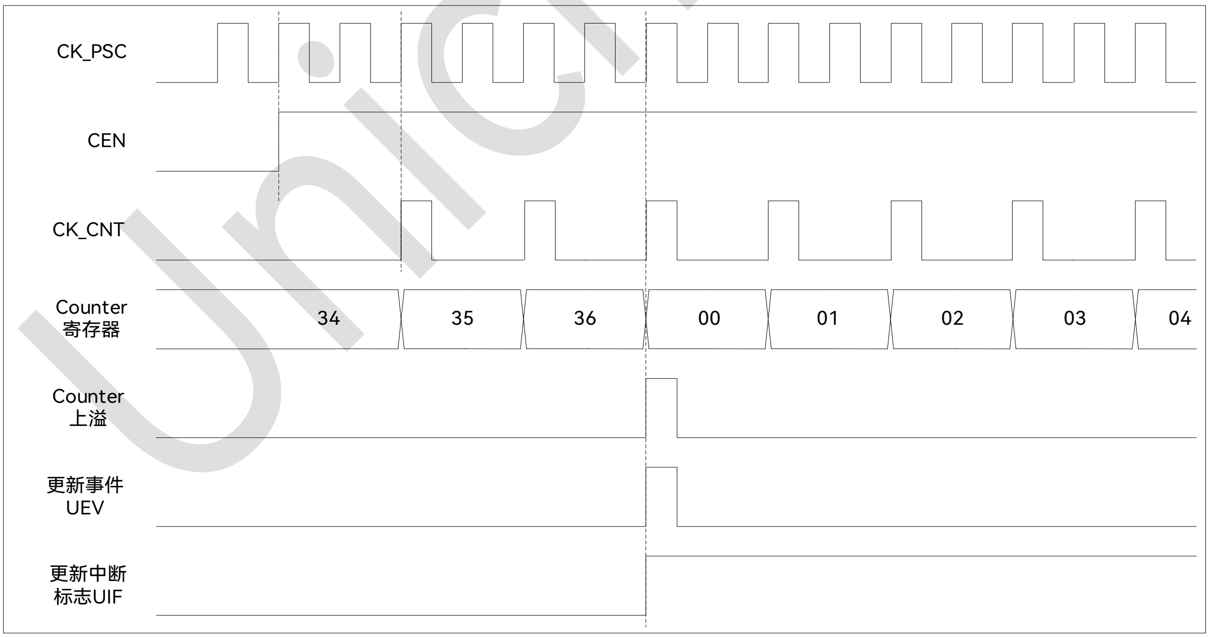


图 14-5：计数器时序图：预分频因子=2

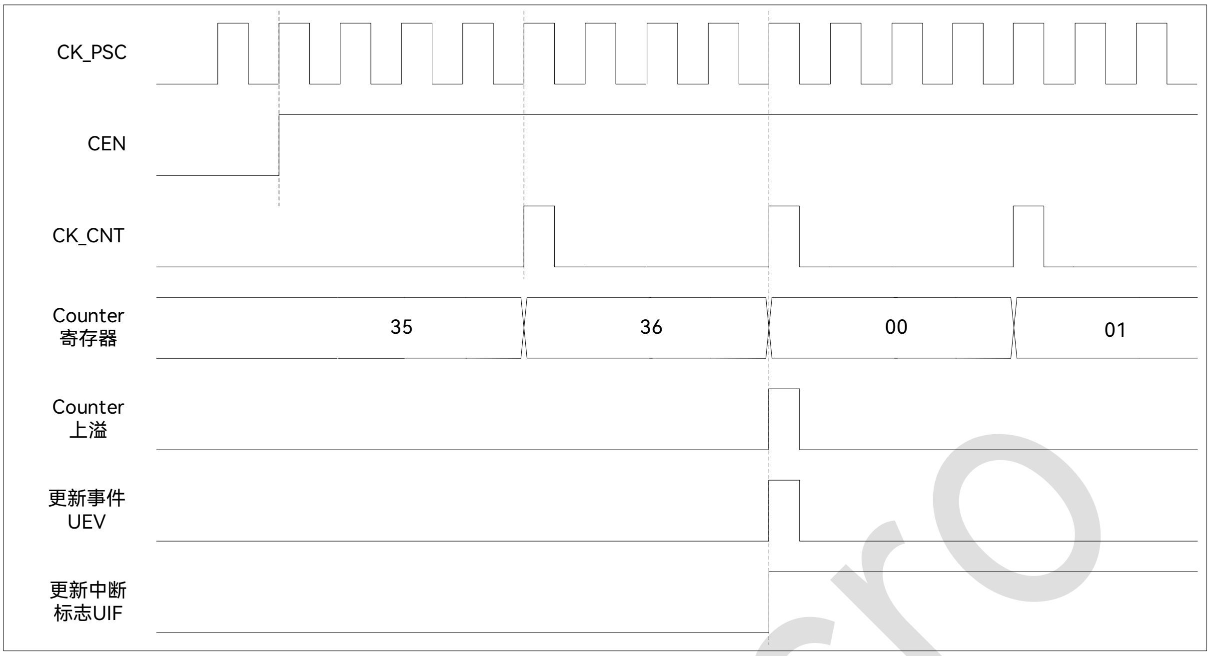


图 14-6：计数器时序图：预分频因子=4

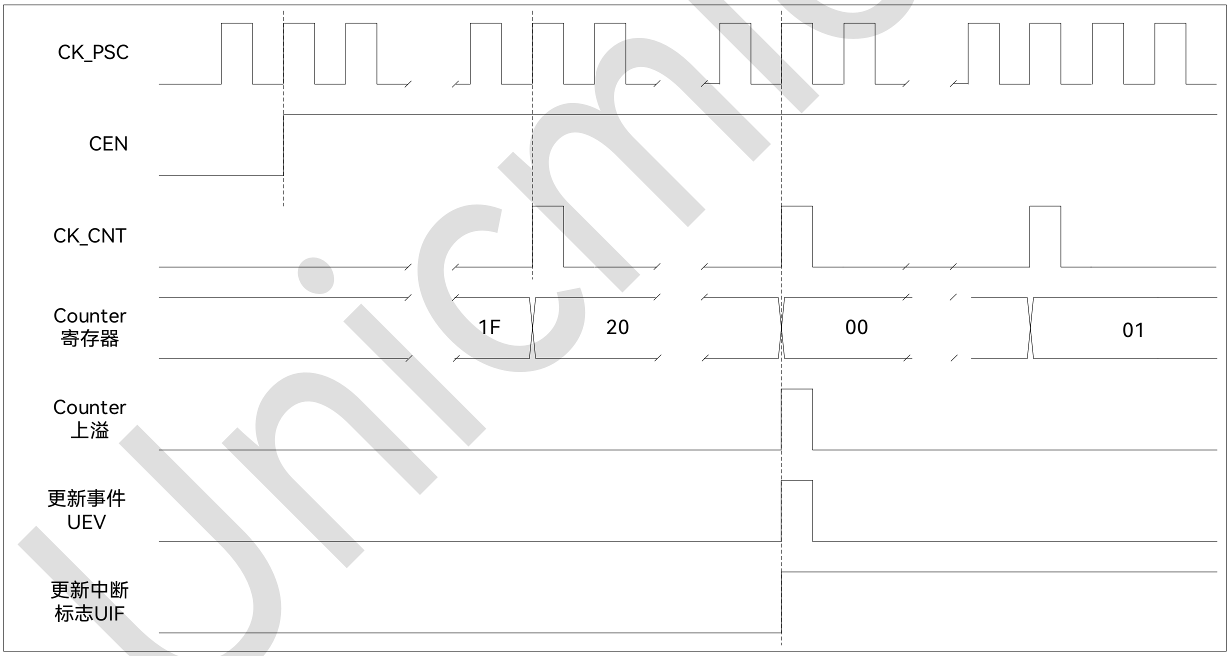


图 14-7：计数器时序图：预分频因子=N

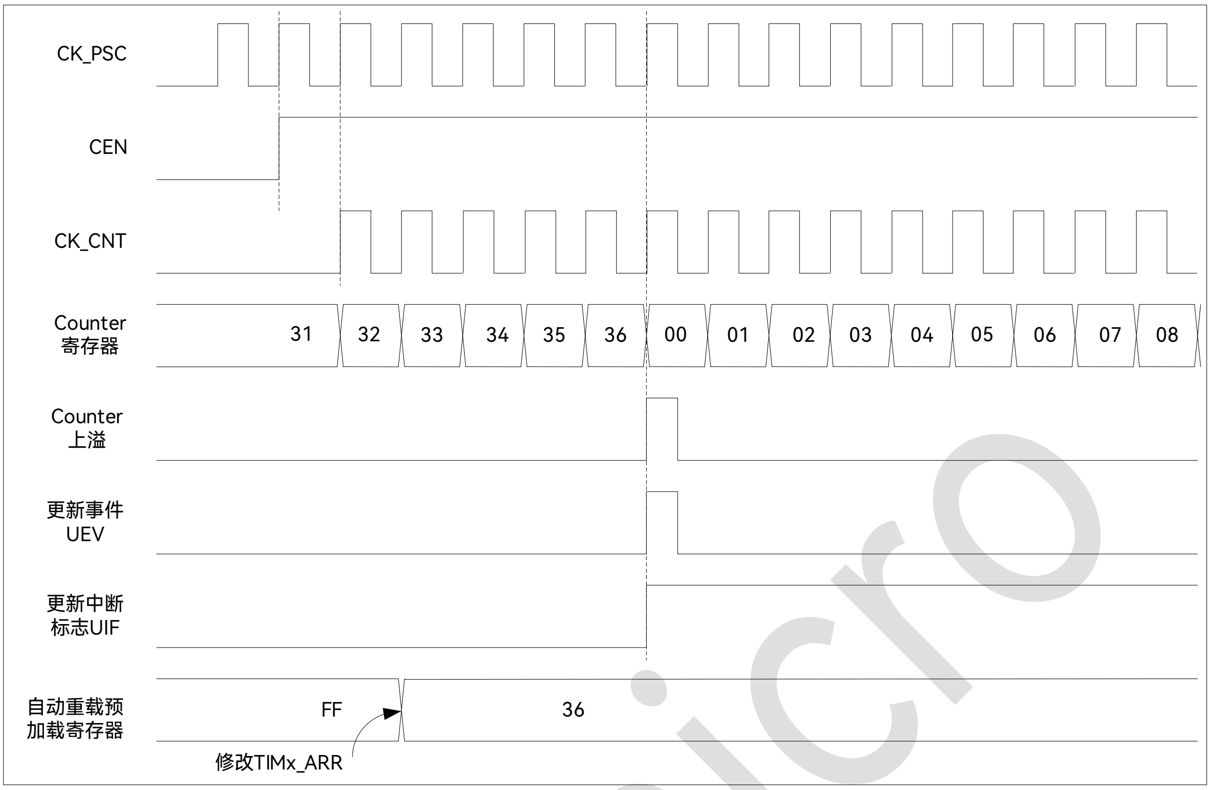


图 14-8：计数器时序图：预分频因子=1, ARPE=0（TIMx_ARR 没有预装载）

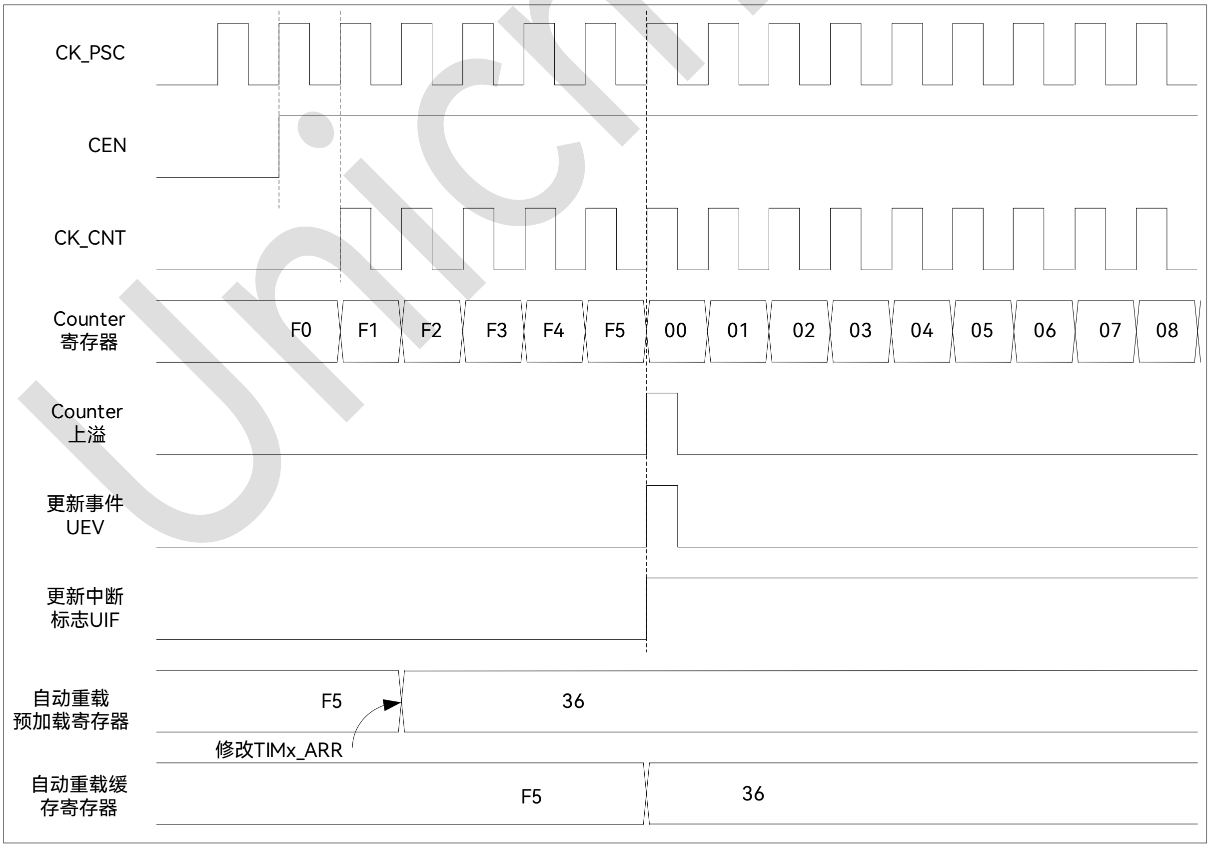


图 14-9：计数器时序图：预分频因子=1，ARPE=1（TIMx_ARR 已预装载）

14.3.3 时钟源

计数器时钟源包括：

- 内部时钟 CK_INT
- 外部时钟模式 1：外部输入引脚（TIx）

14.3.3.1 内部时钟源（CK_INT）

如果从模式控制器禁用（SMS=000），则 CEN 和 UG 位（TIMx_ EGR 寄存器）为实际控制位，并且只能被软件修改（除了 UG 位保持自动被清除）。一旦 CEN 位被写成‘1’，预分频器的时钟就由内部时钟 CK_INT 提供。

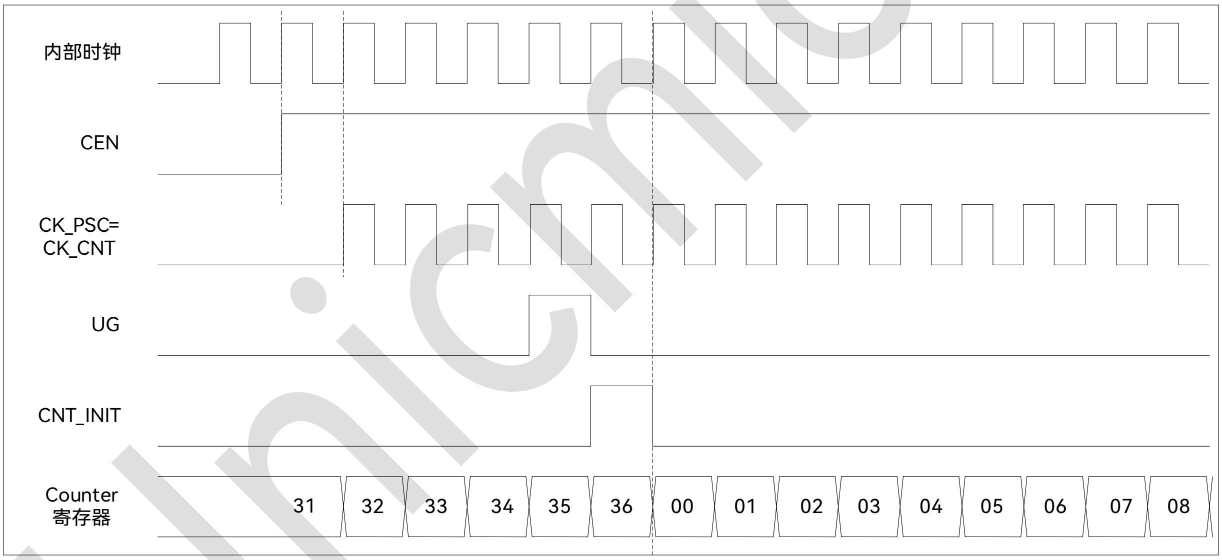


图 14-10：计数器时序图：内部时钟，预分频因子=1，正常模式

14.3.3.2 外部时钟源模式 1

设置 TIMx_SMCR 寄存器的 SMS=111，此模式被选中。计数器对选定输入端的每个上升沿或下降沿计数。

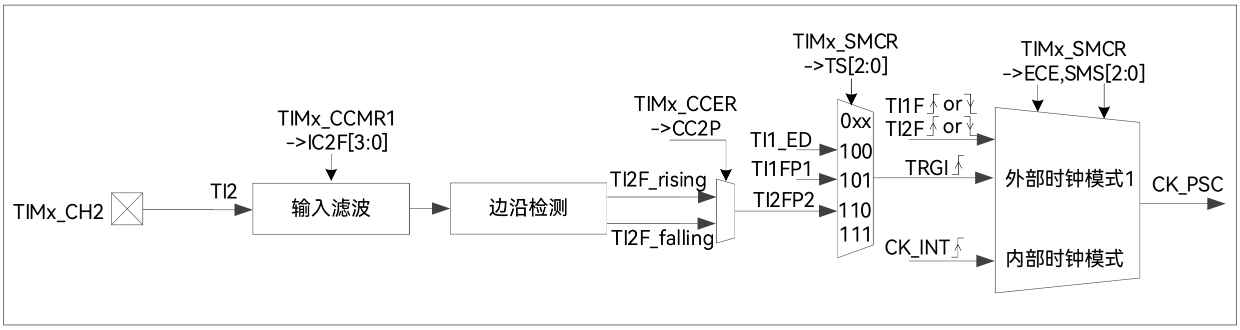


图 14-11：TI2 外部时钟源联接示例框图

举例：要配置向上计数器在 TI2 输入端的上升沿计数，步骤如下：

1. 写 TIMx_CCMR1 寄存器的 CC2S=01，配置通道 2 检测 TI2 输入的上升沿。
2. 写 TIMx_CCMR1 寄存器的 IC2F[3:0]，选择输入滤波器时间（如果不需要滤波器，保持 IC2F[3:0]=0000）。
3. 写 TIMx_CCER 寄存器的 CC2P=0，选定上升沿极性。
4. 写 TIMx_SMCR 寄存器中的 TS[2:0]=110，选定 TI2 作为触发输入源。
5. 写 TIMx_SMCR 寄存器的 SMS[2:0]=111，选择定时器外部时钟模式 1。
6. 写 TIMx_CR1 寄存器的 CEN=1，启动计数器。

注： 捕获预分频器不用作触发，所以不需要对它进行配置。

当发生一次 TI2 上升沿，计数器计数一次（定时器的预分频器的值为 0），并置位 TIF 位。

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

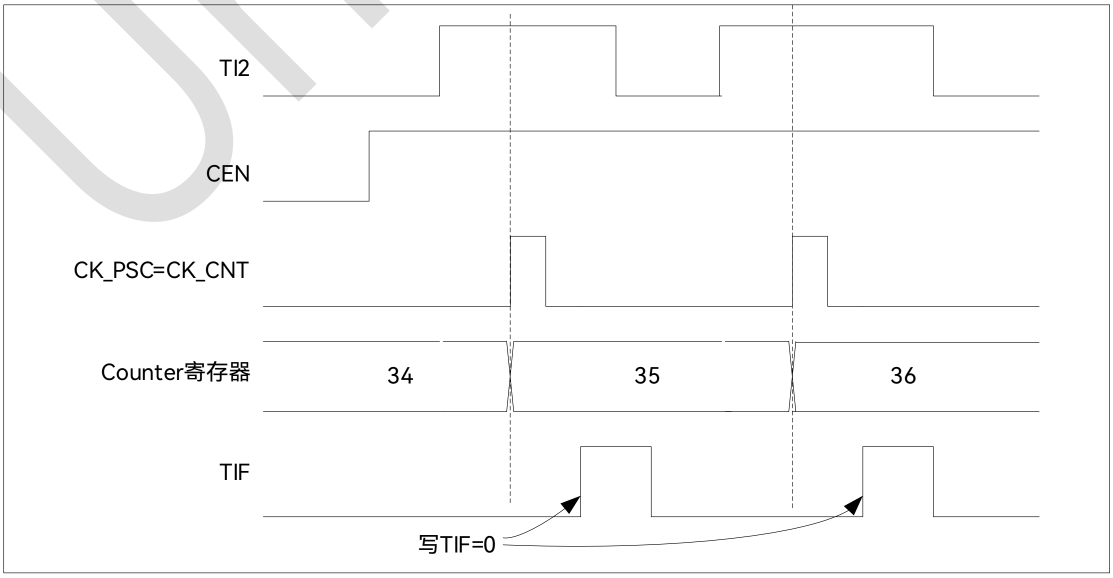


图 14-12：控制时序：外部时钟模式 1

14.3.4 主模式选择

通过设置 TIMx_CR2 寄存器的 MMS[1:0]位可以选择不同的方式触发输出信号（TRGO），TRGO 信号用于与其他外设模式联动，如 ADC。当 MMS[1:0]=00b 时，复位信号（设置 TIMx_EGR 寄存器中的 UG 位）用于触发输出 TRGO 信号。如果 ADC 模块中选择 TRGO 信号用于触发转换信号，那么当设置 UG 位时，就会启动 ADC 转换（ADC 模块需要进行相应的设置）。

14.3.5 捕获/比较通道

每一个捕获/比较通道都是包括：一个捕获/比较寄存器（包含缓存寄存器），一个用于捕获的输入级（带数字滤波、多路复用选择和预分频器），和一个输出级（带比较器和输出控制）。

14.3.5.1 输入级

对 TIx 的输入信号进行采样，产生一个滤波后 TIxF 信号，经过带有极性选择的边沿检测器生成 TIxFPx 信号，TIxFPx 可以作为从模式控制器的触发输入或者作为捕获命令。该信号预分频后（ICxPS）进入捕获寄存器。

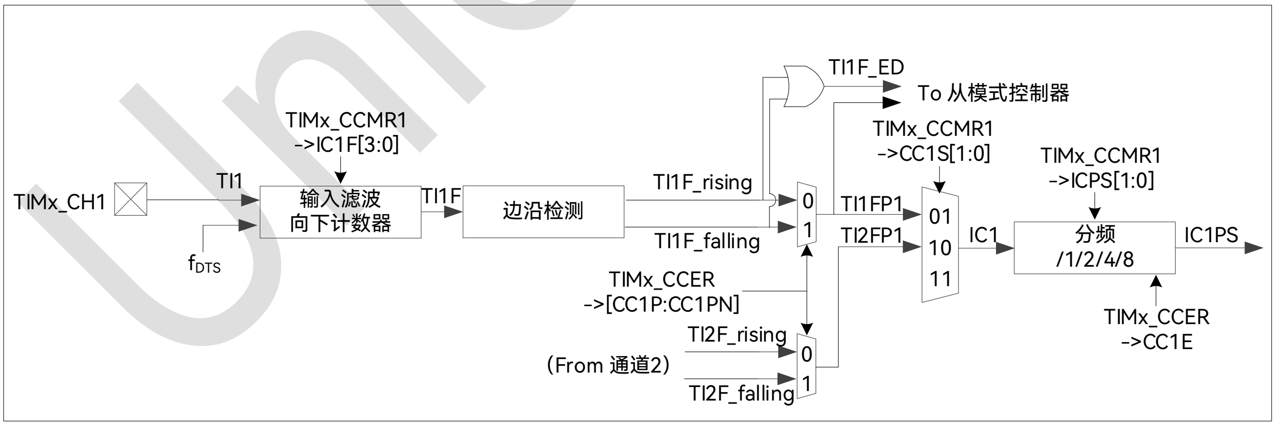


图 14-13：捕获/比较通道输入级（举例：通道 1）

14.3.5.2 输出级

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

捕获/比较模块由一个预装载寄存器（preload register）和一个缓存寄存器（shadow register）

组成。读写过程仅操作预装载寄存器。

在捕获模式下，捕获发生在缓存寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到缓存寄存器中，然后缓存寄存器的内容和计数器进行比较。

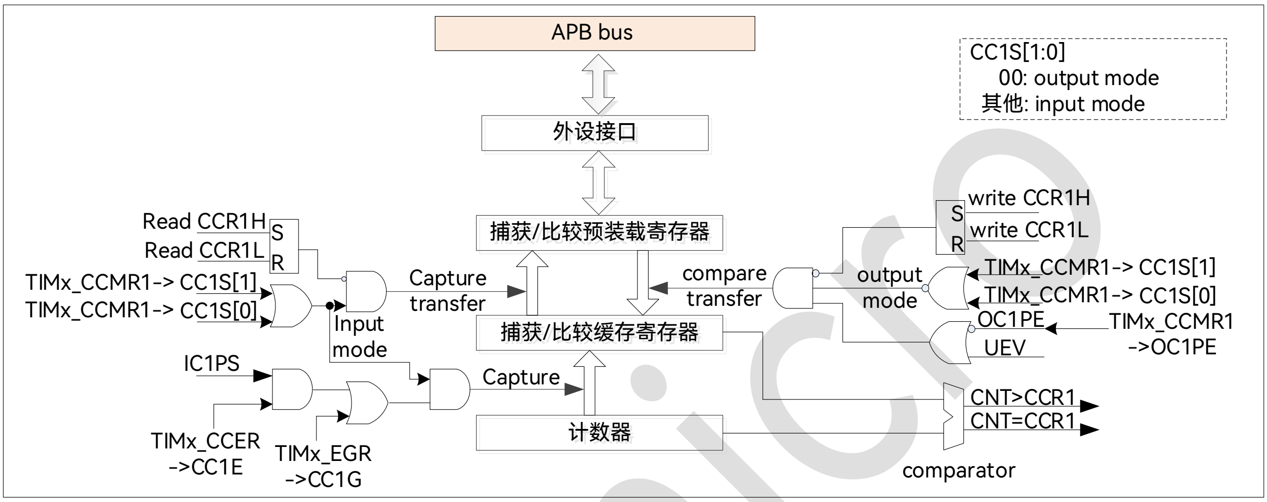


图 14-14：捕获/比较通道主电路

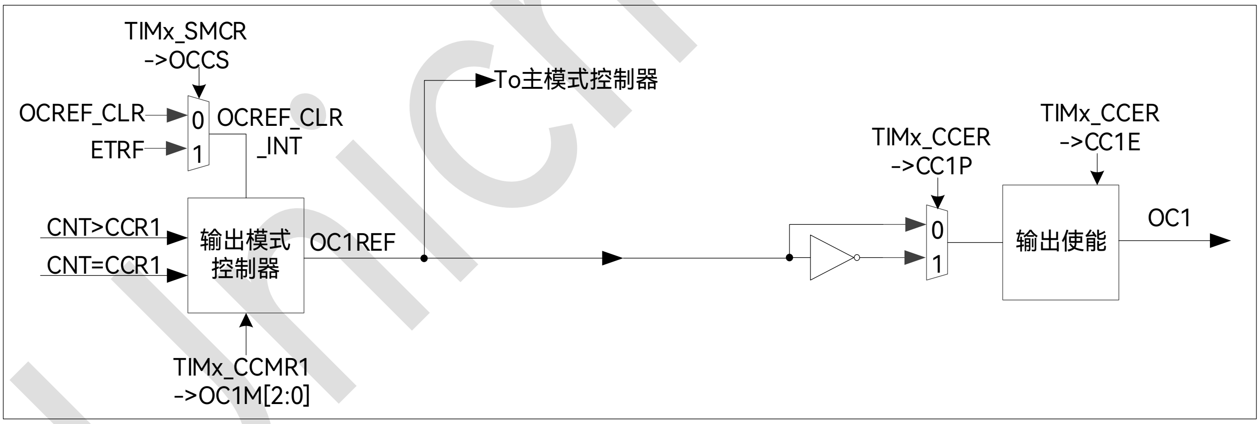


图 14-15：捕获/比较通道输出级（通道 1，通道 2/3 同通道 1）

14.3.6 输入捕获模式

在输入捕获模式下，当检测 ICx 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（TIMx_CCRx）中。当发生捕获事件时，相应 CCxIF 标志（TIMx_SR 寄存器）被置 1， 如果使能了中断，则产生一个中断。如果发生捕获事件时 CCxIF 标志已经为高，那么重复捕获标志 CCxOF（TIMx_SR 寄存器）被置 1。软件写 CCxIF=0 可清除 CCxIF，或读取存储在 TIMx_CCRx 寄存器中的

捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。

举例，在 TI1 输入的上升沿时，捕获计数器的值到 TIMx_CCR1 寄存器中，操作步骤如下：

1. 选择 TIMx_CCR1 的有效输入：置 TIMx_CCMR1 寄存器的 CC1S=01（选中 TI1），只要 CC1S 不为‘00’，通道被配置为输入并且 TIMx_CCR1 寄存器变为只读。
2. 根据连接到计数器的输入信号，配置输入滤波器时间（输入为 TIx 时，输入滤波器控制位是 TIMx_CCMRx 寄存器中的 ICxF 位）。举例，当 TI1 翻转时，信号抖动最多 5 个内部时钟，则须配置滤波器时间大于 5 个时钟周期，TIMx_CCMR1 寄存器中写入 IC1F=0011 配置采样次数 8（以 f_{DTS} 频率采样），通过连续 8 次采样以确认电平变换。
3. 选择 TI1 通道有效沿。写 TIMx_CCER 寄存器中 CC1P=0 和 CC1NP=0（本例中为上升沿）。
4. 配置输入预分频器。在本例中，设置为每个有效的电平转换时发生一次捕获，因此预分频器被禁止（写 TIMx_CCMR1 寄存器的 IC1PS=00）。
5. 写 TIMx_CCER 寄存器的 CC1E=1，允许计数器的值被捕获至捕获寄存器中。
6. 根据需要，置位 TIMx_DIER 寄存器中的 CC1IE 位允许相关中断请求。

当发生一个输入捕获时：

- 有效的电平转换发生时，计数器的值被传送至 TIMx_CCR1 寄存器。
- CC1IF 标志被置位（中断标志）。当发生至少 2 个连续的捕获时，且 CC1IF 未被清除，CC1OF 也被置位。
- 若置位 CC1IE 位，则会产生一个中断。

为了处理捕获溢出，建议在读出捕获溢出标志之前读取数据，这是为了避免在读取标志之后和读取数据之前可能发生的捕获溢出。

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

14.3.7 PWM 输入模式

该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

- 一个 TIx 输入映射至两个 ICx 信号

- 两个 ICx 信号为边沿有效，极性相反
- 两个 TIxFP 信号之一被作为触发输入信号，而从模式控制器被配置成复位模式

举例，测量输入到 TI1 上的 PWM 信号的周期（TIMx_CCR1 寄存器）和占空比（TIMx_ CCR2 寄存器），操作步骤如下（取决于 CK_INT 的频率和预分频器的值）:

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

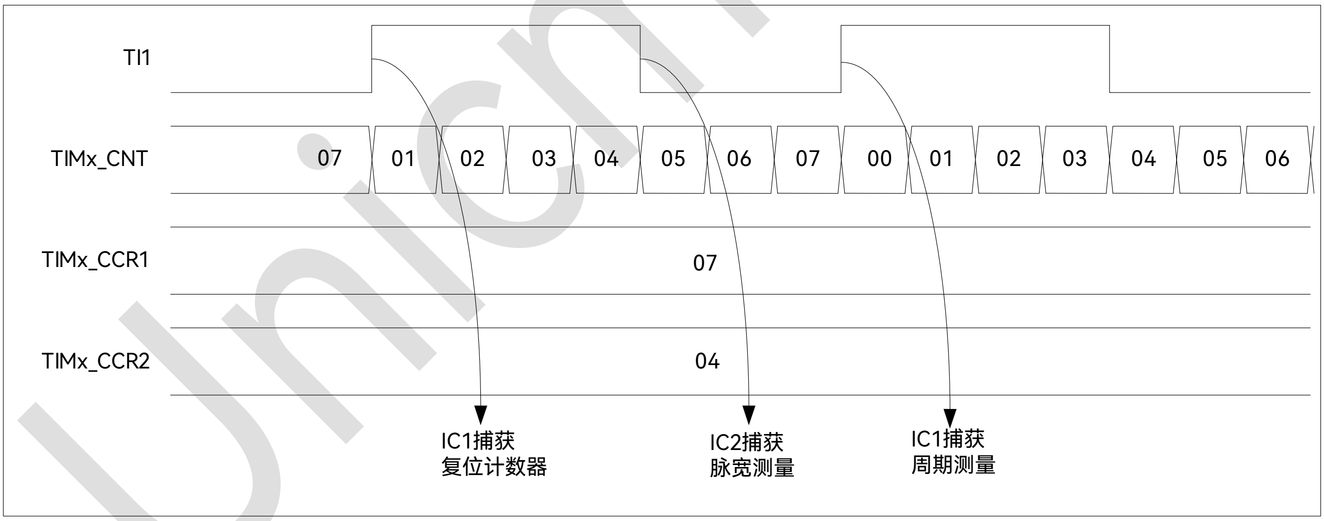


图 14-16：PWM 输入模式时序

14.3.8 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中 CCxS=00）下，输出比较信号（OCxREF 和相应的 OCx/OCxN）能够直接由软件强制为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

设置 TIMx_CCMRx 寄存器中相应的 OCxM=101，输出比较信号（OCxREF/OCx）强制为有效状

态。这样 OCxREF 被强制高电平（OCxREF 始终为高电平有效），置位 TIMx_CCER 的 CCxP 位，OCx 可得到极性相反的信号。

举例，CCxP=0（OCx 高电平有效），则 OCx 被强制为高电平。

设置 TIMx_CCMRx 寄存器中相应的 OCxM=100，输出比较信号（OCxREF/OCx）强制为低电平。

该模式下，在 TIMx_CCRx 缓存寄存器和计数器之间的比较仍然在执行，相应的标志也会被置位。并会产生相应的中断。下面的输出比较模式一节中将详细介绍。

14.3.9 输出比较模式

此模式用于控制输出波形或指示一段时间到时。

当计数器与捕获/比较寄存器的内容匹配（相等）时，输出比较功能做如下操作：

- 根据输出比较模式（TIMx_CCMRx 寄存器中 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）的配置，输出至对应引脚。在比较匹配发生时，输出引脚可以保持它的电平(OCxM=000)、被设置成有效电平（OCxM=001)、被设置成无效电平（OCxM=010）或进行翻转（OCxM=011）。
- 置位中断状态寄存器中的标志位（TIMx_SR 寄存器中的 CCxIF 位）。
- 若置位相应的中断屏蔽位（TIMx_DIER 寄存器中的 CCxIE 位），则产生一个中断。

设置 TIMx_CCMRx 中的 OCxPE 位选择 TIMx_CCRx 寄存器是否使用预装载寄存器。

在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。计时分辨率为计数器的一次计数。

输出比较模式也能用来输出一个单脉冲（单脉冲模式）。

输出比较模式的配置步骤：

1. 选择计数器时钟（内部，外部，预分频器）。
2. 根据需求，写入数据至 TIMx_ARR 和 TIMx_CCRx 寄存器中。
3. 如果要产生一个中断请求，设置 CCxIE 位。
4. 选择输出模式，举例如下：

- 写 OCxM=011，计数器与 CCRx 匹配时，翻转 OCx 的输出引脚。
- 写 OCxPE=0，禁用预装载寄存器。
- 写 CCxP=0，选择极性为高电平有效。
- 写 CCxE=1，使能输出。

5. 置位 TIMx_CR1 寄存器的 CEN 位，启动计数器。

TIMx_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，前提是未启用预装载寄存器（OCxPE=‘0’，否则 TIMx_CCRx 的缓存寄存器只在发生下次更新事件时更新），举例如下图：

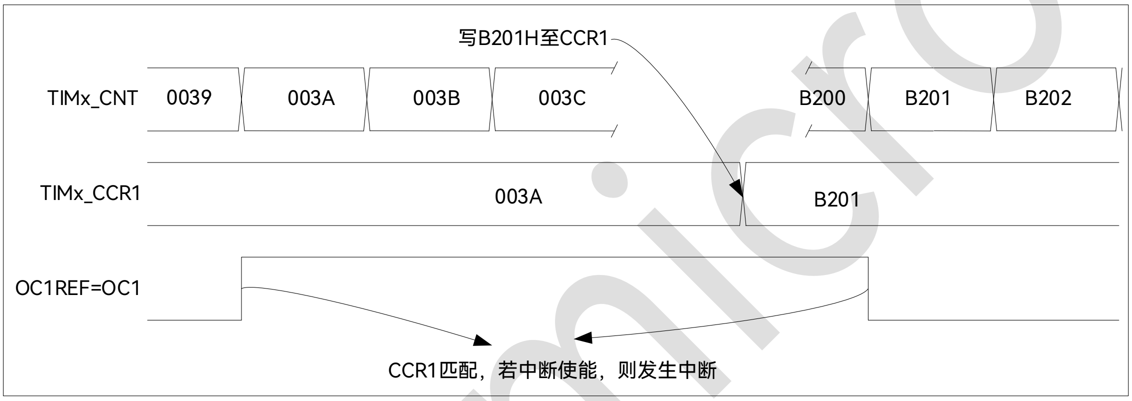


图 14-17：输出比较模式，配置为：翻转 OC1

14.3.10 PWM 模式

脉冲宽度调制模式（PWM）可以产生一个 PWM 信号，由 TIMx_ARR 寄存器确定频率、由 TIMx_CCRx 寄存器确定占空比。

在 TIMx_CCMRx 寄存器中的 OCxM 位写入‘110’（PWM 模式 1）或‘111’（PWM 模式 2），可以在每个通道独立地设置 PWM 模式（每个 OCx 输出一种 PWM）。必须置位 TIMx_CCMRx 寄存器的 OCxPE 位，使能相应的预装载寄存器（preload register），最后置位 TIMx_CR1 寄存器的 ARPE 位，使能自动重载预装载寄存器（在向上计数或中心对称模式中）。

只有发生一个更新事件时，预装载寄存器才能被传送到缓存寄存器，因此在计数器开始计数之前，必须通过设置 TIMx_EGR 寄存器中的 UG 位来初始化所有的寄存器。

设置 TIMx_CCER 寄存器中的 CCxP 位，配置 OCx 的极性为高电平有效或低电平有效。OCx 的输出使能通过（TIMx_CCER 寄存器中）CCxE 位控制。详细参考 TIMx_CCER 寄存器的描述。

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

然而，为了与 OCREF_CLR 的功能一致（在下一个 PWM 周期之前，OCREF_CLR 能够清除 OCxREF），OCxREF 信号只能在下述条件下产生:

- 当比较的结果改变
- 当输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）从“冻结”（无比较，OCxM=‘000’）切换到某个 PWM 模式（OCxM=‘110’或‘111’）。

14.3.10.1 PWM 边沿对齐模式

在 PWM 模式 1，计数器向上计数，当 TIMx_CNT<TIMx_CCRx 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR），则 OCxREF 保持为‘1’。如果比较值为 0，则 OCxREF 保持为‘0’。下图为 TIMx_ARR=8 时边沿对齐的 PWM 波形实例。

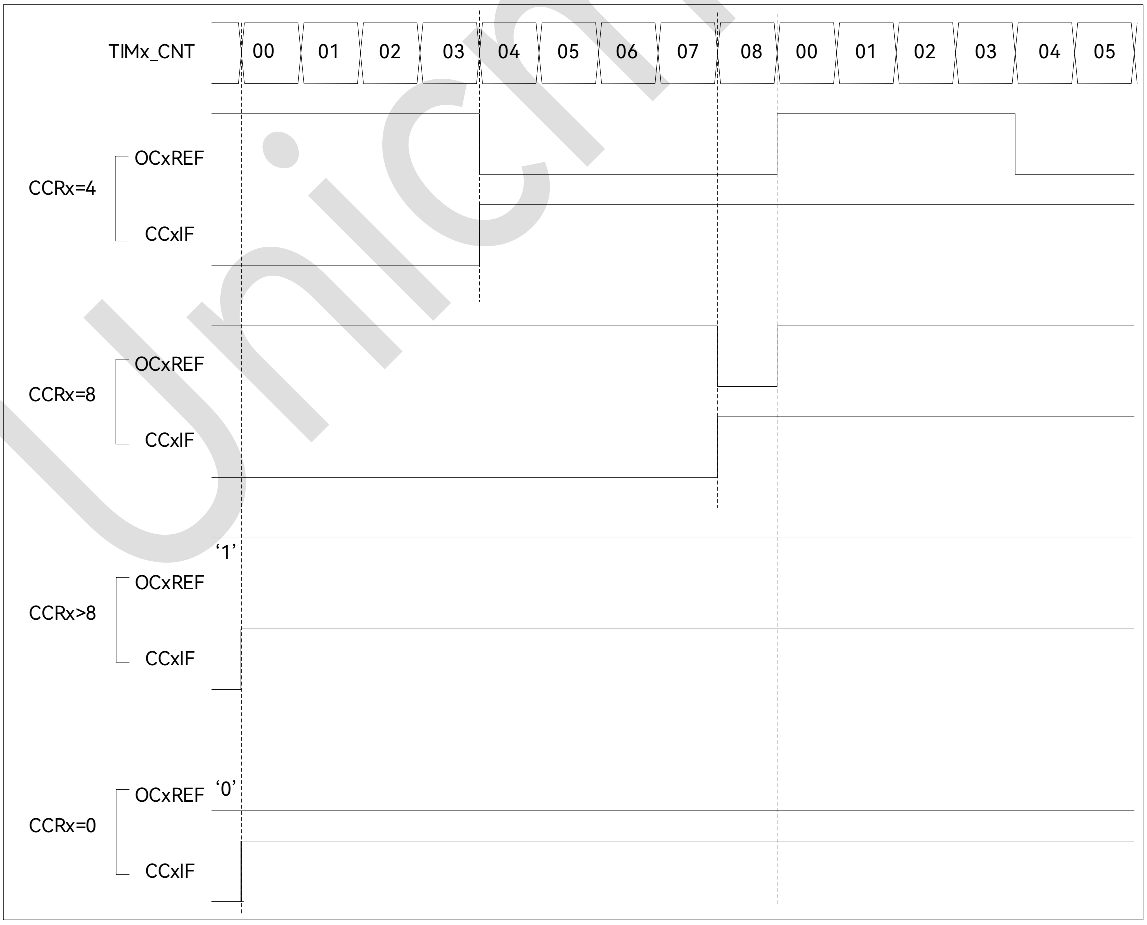


图 14-18：PWM 波形（边沿对齐，向上计数，ARR=8）

14.3.11 外部事件时清零 OCxREF 信号

设置 TIMx_CCMRx 寄存器中对应的 OCxCE 位为‘1’，指定通道的外部事件触发 OCxREF 信号清零功能开启，即当 OCxREF_CLR_INPUT 发生高电平触发 OCxREF 信号拉低，OCxREF 信号将保持为低直到发生下一次的更新事件 UEV。

该功能只能用于输出比较和 PWM 模式，而不能用于强制模式。

OCREF_CLR 可以来自于比较器 1 或比较器 2 的输出。当比较器输出为高电平时，同时相应通道的 OCxCE 位为 1，则 OCxREF 信号将被拉低。

下图显示了当 OCREF_CLR 输入变为高时，对应不同 OCxCE 的值，OCxREF 信号的行为。在此例中，定时器 TIMx 被置于 PWM 模式。

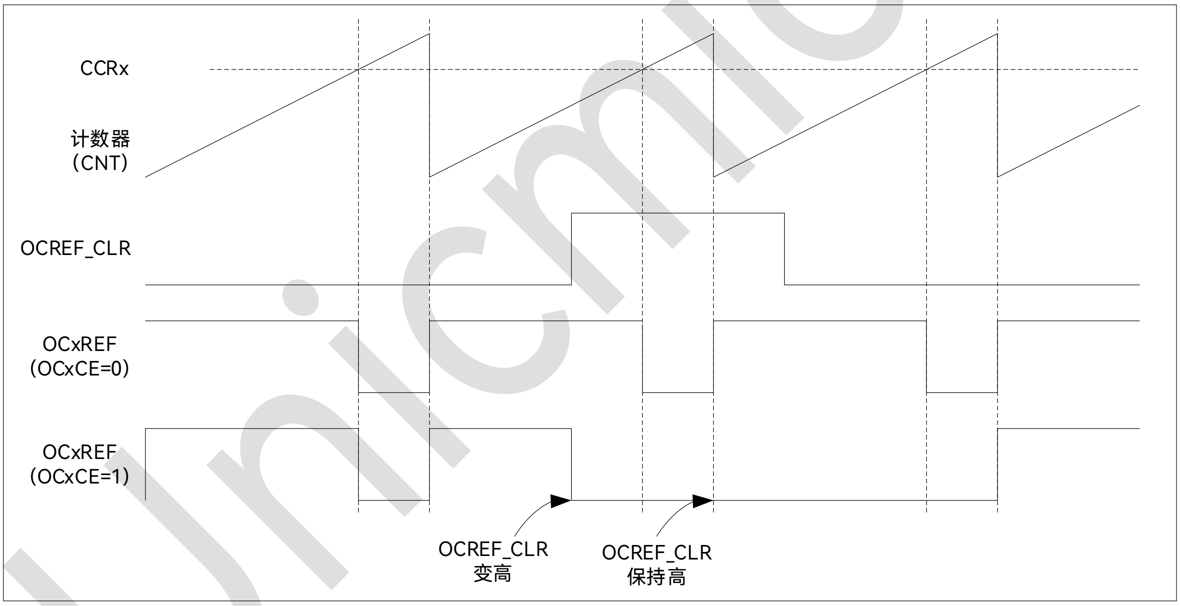


图 14-19：清零 OCxREF 信号时序

注：在 PWM 占空比为 100%的情况时（CCRx>ARR），OCREF 在下一次计数溢出时被再次使能。

14.3.12 输入异或功能

TIMx_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIMx_CH1、TIMx_CH2 和 TIMx_CH3。

异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。

14.3.13 TIMx 定时器和外部触发的同步

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

14.3.13.1 从模式：复位模式

在一个触发输入发生事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器（TIMx_ARR，TIMx_CCRx）都被更新。

在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽（在本例中，不需要任何滤波器，因此保持 IC1F=0000）。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=0 和 CC1NP=0 以确定极性（只检测上升沿）。
- 置 TIMx_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器。

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

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

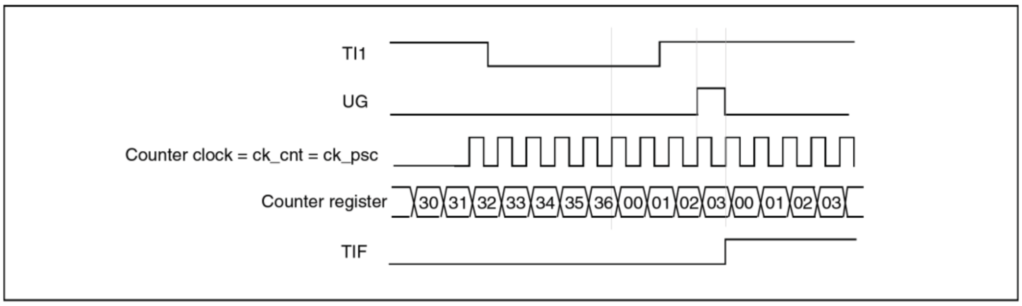


图 14-20：复位模式下的控制电路

14.3.13.2 从模式：门控模式

按照选中的输入端电平使能计数器。

在如下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽（本例中，不需要滤波，所以保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=1 和 CC1NP=0 以确定极性（只检测低电平）。
- 置 TIMx_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器（在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何）。

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

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

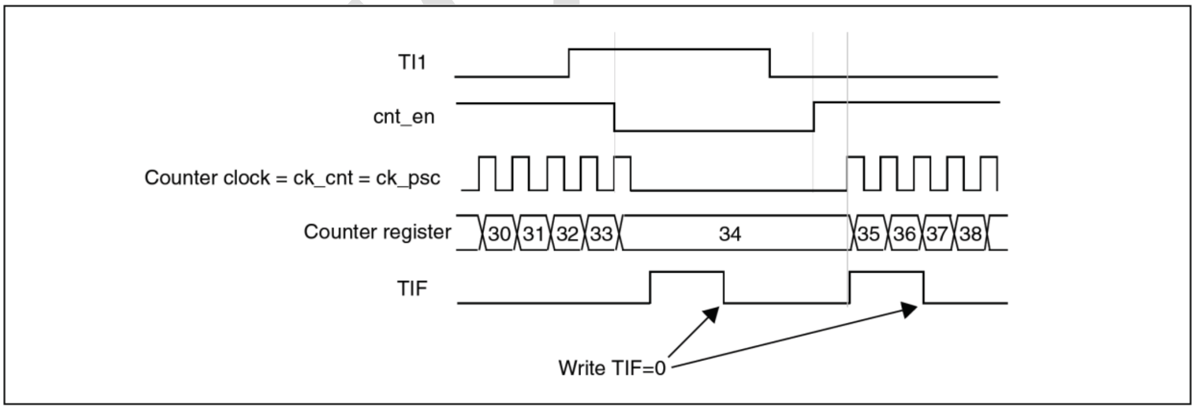


图 14-21：门控模式下的控制电路

14.3.13.3 从模式：触发模式

输入端上选中的事件使能计数器。

在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽（本例中，不需要任何滤波器，保持 IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIMx_CCMR1 寄存器中 CC2S=01。置 TIMx_CCER 寄存器中 CC2P=1 和 CC1NP=0 以确定极性（只检测低电平)。
- 置 TIMx_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIMx_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

  当 TI2 出现一个上升沿时，计数器按内部时钟开始计数，同时设置 TIF 标志。

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

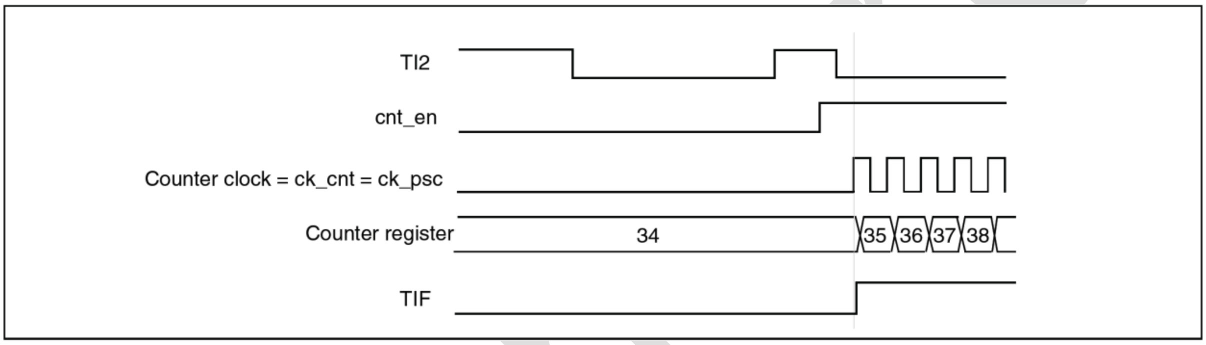


图 14-22：触发模式下的控制电路

14.3.14 调试模式

当微控制器进入调试模式时（Cortex-M0 内核停止），根据 DBG 模块中 DBG_TIMx_STOP 的设置，TIMx 计数器可以或者继续正常操作，或者停止。

14.4 相关寄存器

14.4.1 TIM 控制寄存器 1（TIM18_CR1）

偏移地址：0x000

复位值：0x0000 0000

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

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

位	名称	描述
31:10	RSV	保留位，固定读为 0
9:8	CKD[1:0]	时钟分频（Clock division），通过修改此位段可以对 CK_INT 时钟进行分频，分频后的时钟作为输入滤波（Tix）采样的基准时钟。 00：$t_{DTS} = t_{CK\_INT}$ 01：$t_{DTS} = 2 * t_{CK\_INT}$ 10：$t_{DTS} = 4 * t_{CK\_INT}$ 11：保留，禁止使用
7	ARPE	自动重载预装载使能（Auto-reload preload enable）: 0: TIMx_ARR 寄存器没有缓冲 1: TIMx_ARR 寄存器有缓冲
6:3	RSV	保留位，固定读为 0
2	URS	更新请求源（Update request source）: 软件置位和清零，选择 UEV 事件的源 0：下述任一事件产生更新中断: - 计数器上溢 - 置位 UG 位 - 从模式（slave mode）控制器产生的更新 1：仅计数器上溢产生更新中断。
1	UDIS	禁止更新（Update disable）： 软件置位和清零，通过该位允许/禁止 UEV 事件的产生 0：允许 UEV。更新（UEV）事件由下述任一事件产生: - 计数器上溢 - 置位 UG 位 - 从模式（slave mode）控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。 1：禁止 UEV。不产生更新事件，影子寄存器（ARR、PSC、CCRx）保持它们的值。如果置位 UG 位或从模式（slave mode）控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。
0	CEN	使能计数器（Counter enable） 0：禁止计数器 1：使能计数器 注：只有软件设置了 CEN 位后，外部时钟、门控模式才能工作。但触发模式可以自动地通过硬件设置 CEN 位。

14.4.2 TIM 控制寄存器 2（TIM18_CR2）

偏移地址：0x004

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								TI1S	MMS[2:0]			Res.			
								rw	rw						

位	名称	描述
31:8	RSV	保留位，固定读为 0
7	TI1S	TI1 选择（TI1 selection): 0：TIMx_CH1 引脚连到 TI1 输入。 1：TIMx_CH1、TIMx_CH2 和 TIMx_CH3 引脚经异或（XOR）后连到 TI1 输入。
6:4	MMS[2:0]	主模式选择（Master mode selection) 这 3 位用于选择在主模式下送到从定时器的同步信息（TRGO)。可能的组合如下: 000：复位 – TIMx_EGR 寄存器的 UG 位被用于作为触发输出 (TRGO)。如果是触发输入产生的复位（从模式控制器处于复位模式），则 TRGO 上的信号相对实际的复位会有一个延迟。 001：使能 – 计数器使能信号 CNT_EN 被用于作为触发输出（TRGO)。可用于同时启动多个定时器或控制在一段时间内使能从定时器。在门控模式下，计数器使能信号是 CEN 控制位和触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式（见 TIMx_SMCR 寄存器中 MSM 位的描述)。 010：更新 – 更新事件被选为触发输入（TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 011：比较脉冲 – 在捕获发生发生时，CC1IF 标志被置位时（即使它已经为高），触发输出发送一个正脉冲（TRGO)。
3:0	RSV	保留位，固定读为 0

14.4.3 TIM 从模式控制寄存器（TIM18_SMCR）

偏移地址：0x008

复位值：0x0000 0000

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

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

位	名称	描述
31:7	RSV	保留位，固定读为 0
6:4	TS[2:0]	触发选择（Trigger selection): 这 3 位选择用于同步计数器的触发输入。 000：保留 001：保留 010：保留 011：保留 100：TI1 的边沿检测器（TI1F_ED） 101：滤波后的定时器输入 1 (TI1FP1) 110：滤波后的定时器输入 2 (TI2FP2) 111：保留 注：此位只有当 SMS[2:0]=000b 时才能修改。
3	RSV	保留位，固定读为 0
2:0	SMS[2:0]	从模式选择（Slave mode selection): 当选择了外部信号，触发信号（TRGI）的有效边沿与选中的外部输入极性相关（见输入控制寄存器和控制寄存器的说明） 000：关闭从模式 – 如果 CEN=1，则预分频器直接由内部时钟驱动。 100：复位模式 – 选中的触发输入（TRGI）的上升沿重新初始化计数器，并且产生一次更新寄存器。 101：门控模式 – 当触发输入（TRGI）为高时，计数器的时钟开启，一旦触发输入变为低，计数器停止（但不复位）。计数器的启动和停止都是受控的。 110：触发模式 – 计数器在触发输入（TRGI）的上升沿启动（但不复位），只有计数器的启动是受控的。 111：外部时钟模式 1 – 选中的触发输入（TRGI）的上升沿驱动计数器。

位	名称	描述
		其他：保留 注 1：如果 TI1F_ED 被选为触发输入（TS=100b）时，不要使用门控模式。这是因为，TI1F_ED 在每次 TI1F 变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。 注 2：在接收主定时器事件之前，从定时器的时钟必须使能，并且在接收主定时器的触发信号过程中，不能动态改变。

14.4.4 TIM DMA/中断使能寄存器（TIM18_DIER）

偏移地址：0x00C

复位值：0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Res.</td></tr>
<tr><td colspan="16"></td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="9">Res.</td><td>TIE</td><td colspan="2">Res.</td><td>CC3IE</td><td>CC2IE</td><td>CC1IE</td><td>UIE</td></tr>
<tr><td colspan="9"></td><td>rw</td><td colspan="2"></td><td>rw</td><td>rw</td><td>rw</td><td>rw</td></tr>
</table>

位	名称	描述
31:7	RSV	保留位，固定读为 0
6	TIE	触发中断使能 0：触发中断禁止 1：触发中断允许
5:4	RSV	保留位，固定读为 0
3	CC3IE	捕获/比较 3 中断使能 0: CC3 中断禁止 1: CC3 中断允许
2	CC2IE	捕获/比较 2 中断使能 0: CC2 中断禁止 1: CC2 中断允许
1	CC1IE	捕获/比较 1 中断使能 0: CC1 中断禁止 1: CC1 中断允许
0	UIE	更新中断使能 0：更新中断禁止 1：更新中断允许

14.4.5 TIMx 中断标志和清除寄存器（TIM18_SR）

偏移地址：0x010

复位值：0x0000 0000

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

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

位	名称	描述
31:12	RSV	保留位，固定读为 0
11	CC3OF	捕获/比较 3 重复捕获标志，参考 CC1OF 描述
10	CC2OF	捕获/比较 2 重复捕获标志，参考 CC1OF 描述
9	CC1OF	捕获/比较 1 重复捕获标志（Capture/Compare 1 overcapture flag）: 仅当相应的通道被配置为输入捕获时，该标记可由硬件置 1。 软件写 0 可清除该位。 0：无重复捕获产生 1：当 CC1IF 的状态已经为‘1’，计数器的值被捕获到 TIMx_CCR1 寄存器
8:7	RSV	保留位，固定读为 0
6	TIF	触发器中断标志（Trigger interrupt flag）: 当发生触发事件（当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿)时由硬件对该位置‘1’。它由软件清‘0’。 0：无触发器事件产生 1：触发中断等待响应
5:4	RSV	保留位，固定读为 0
3	CC3IF	捕获/比较 3 中断标志，参考 CC1IF
2	CC2IF	捕获/比较 2 中断标志，参考 CC1IF
1	CC1IF	捕获/比较 1 中断标志: **如果通道 CC1 配置为输出模式：** 当计数器值与比较值匹配时该位由硬件置 1。 它由软件清‘0’。 0：无匹配发生 1：TIMx_CNT 的值与 TIMx_CCR1 的值匹配。当 TIMx_CCR1 的内容大于 TIMx_

位	名称	描述
0		APR 的内容时，在向上时计数器上溢条件下，CC1IF 位变高 **如果通道 CC1 配置为输入模式:** 当捕获事件发生时该位由硬件置‘1’, 它由软件清‘0’或通过读 TIMx_CCR1 清‘0’。 0：无输入捕获产生 1：计数器值被捕获至 TIMx_CCR1（在 IC1 上检测到与所选极性相同的边沿）
0	UIF	更新中断标志（Update interrupt flag): 当产生更新事件时该位由硬件置‘1’。它由软件清‘0’。 0：无更新事件产生; 1：更新中断等待响应。当寄存器被更新时该位由硬件置‘1’: - 若 TIMx_CR1 寄存器的 URS=0、UDIS=0，当设置 TIMx_EGR 寄存器的 UG=1 进而实现软件对计数器 CNT 重新初始化时。 - 若 TIMx_CR1 寄存器的 URS=0、UDIS=0，当计数器 CNT 被触发事件重新初始化时。参考章节：TIMx 从模式控制寄存器（TIMx_SMCR)

14.4.6 TIMx 事件产生寄存器（TIM18_EGR）

偏移地址：0x014

复位值：0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Res.</td></tr>
<tr><td colspan="16"></td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="9">Res.</td><td>TG</td><td colspan="2">Res.</td><td>CC3G</td><td>CC2G</td><td>CC1G</td><td>UG</td></tr>
<tr><td colspan="9"></td><td>w</td><td colspan="2"></td><td>w</td><td>w</td><td>w</td><td>w</td></tr>
</table>

位	名称	描述
31:7	RSV	保留位，固定读为 0
6	TG	触发产生（Trigger generation）: 该位由软件置‘1’，用于产生一个事件，由硬件自动清‘0’。 0：无动作 1：TIMx_SR 中 TIF 标志置 1，若开启对应的中断，则产生相应的中断
5:4	RSV	保留位，固定读为 0
3	CC3G	捕获/比较 3 发生参考 CC1G 描述
2	CC2G	捕获/比较 2 发生参考 CC1G 描述

位	名称	描述
1	CC1G	捕获/比较 1 发生（Capture/Compare 1 generation): 该位由软件置‘1’，用于产生一个捕获/比较事件，由硬件自动清‘0’。 0：无动作 1：在通道 1 上产生一个捕获/比较事件 若通道 CC1 配置为输出: 设置 CC1IF=1，若开启对应的中断，则产生相应的中断。 若通道 CC1 配置为输入: 当前的计数器值被捕获至 TIMx_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断，则产生相应的中断。若 CC1IF 已经为 1，则设置 CC1OF=1。
0	UG	产生更新事件（Update generation): 该位由软件置‘1’，由硬件自动清‘0’。 0：无动作 1：重新初始化计数器，并产生一个（寄存器）更新事件。注意预分频器的计数器也被清‘0’（但是预分频系数不变）。

14.4.7 TIMx 捕获/比较模式寄存器 1【复用】（TIM18_CCMR1）

偏移地址：0x018

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15	OC2CE	输出比较 2 清 0 允许
14:12	OC2M[2:0]	输出比较 2 模式
11	OC2PE	输出比较 2 预分频允许
10	OC2FE	输出比较 2 快速使能
9:8	CC2S[1:0]	捕获/比较 2 选择（Capture/Compare 1 selection） 这 2 位定义通道的方向（输入/输出），及输入信号的选择: 00：CC2 通道被配置为输出 01：CC2 通道被配置为输入，IC2 映射在 TI1 上

位	名称	描述
		10：CC2 通道被配置为输入，IC2 映射在 TI2 上 11：保留 注：CC2S 仅在通道关闭时（TIMx_CCER 寄存器的 CC2E=0）才是可写的。
7	OC1CE	输出比较 1 清 0 允许 0: OC1REF 不受 OCREF_CLR 输入（比较器的输出）的影响; 1: 一旦检测到 OCREF_CLR 输入为高电平，清除 OC1REF=0。
6:4	OC1M[2:0]	输出比较模式 1（Output Compare 1 mode） 该 3 位定义了输出参考信号 OC1REF 的行为，而 OC1REF 决定了 OC1、OC1N 的电平。OC1REF 是高电平有效，而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用（此模式用于产生一个时基）。 001：匹配时，设置通道 1 为有效电平。当 TIMx_CNT = TIMx_CCR1 时，强制 OC1REF 为高。 010：匹配时，设置通道 1 为无效电平。当 TIMx_CNT = TIMx_CCR1 时，强制 OC1REF 为低。 011：翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。 100：强制为无效电平，强制 OC1REF 为低。 101：强制为有效电平，强制 OC1REF 为高。 110：PWM 模式 1 - 在向上计数时，当 TIMx_CNT<TIMx_CCR1 时，通道 1 为有效电平（OC1REF=1），否则为无效电平（OC1REF=0）。 111：PWM 模式 2 - 在向上计数时，当 TIMx_CNT<TIMx_CCR1 时，通道 1 为无效电平，否则为有效电平。 注 1：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。
3	OC1PE	输出比较 1 预装载允许（Output Compare 1 preload enable): 0：禁止 TIMx_CCR1 寄存器的预装载功能，可随时写入 TIMx_CCR1 寄存器，并且新写入的数值立即起作用。 1：开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。
2	OC1FE	输出比较 1 快速使能（Output Compare 1 fast enable): 该位用于加快 CC 输出对触发输入事件的响应。 0：CC1 的正常操作依赖于计数器与 CCR1 的值，即使工作于触发器状态。当触发器的输入有一个有效沿时，激活 CC1 输出的最小延时为 5 个时钟周期

位	名称	描述
		1：输入到触发器的有效沿的作用等同于发生了一次比较匹配。因此，OC 被设置为比较电平（独立于比较结果）。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。
1:0	CC1S[1:0]	捕获/比较 1 选择（Capture/Compare 1 selection）： 这 2 位定义通道的方向（输入/输出），及输入信号的选择： 00：CC1 通道被配置为输出 01：CC1 通道被配置为输入，IC1 映射在 TI1 上 10：CC1 通道被配置为输入，IC1 映射在 TI2 上 11：保留 注：CC1S 仅在通道关闭时（TIMx_CCER 寄存器的 CC1E=0）才是可写的。

14.4.8 TIMx 捕获/比较模式寄存器 1【复用】（TIM18_CCMR1）

偏移地址：0x018

复位值：0x0000 0000

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

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	

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:12	IC2F[3:0]	输入捕获 2 滤波器，参考输入捕获 1 滤波器的定义
11:10	IC2PSC[1:0]	输入捕获 2 预分频，参考输入捕 1 预分频器的定义
9:8	CC2S[1:0]	捕获/比较 2 选择（capture/compare 2 selection）： 该 2 位定义通道的方向（输入/输出），及输入脚的选择： 00：CC2 通道被配置为输出 01：CC2 通道被配置为输入，IC2 映射在 TI2 上。 10：CC2 通道被配置为输入，IC2 映射在 TI1 上。 11：保留 注：CC2S 仅在通道关闭时（TIMx_CCER 寄存器的 CC2E=0）才是可写的。

位	名称	描述
7:4	IC1F[3:0]	输入捕获 1 滤波器（Input capture 1 filter）： 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变： 0000：无滤波器，以 f_{DTS} 采样 0001：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=2 0010：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=4 0011：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=8 0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6 1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8 1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5 1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8 注：当 ICxF[3:0]=1、2 或 3 时，公式中的 f_{DTS} 由 CK_INT 替代。
3:2	IC1PSC[1:0]	输入捕获 1 预分频器： 这 2 位定义了 CC1 输入（IC1）的预分频系数。一旦 CC1E=0（TIMx_CCER 寄存器中），则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获。 01：每 2 个事件触发一次捕获。 10：每 4 个事件触发一次捕获。 11：每 8 个事件触发一次捕获。
1:0	CC1S[1:0]	捕获/比较 1 选择（Capture/Compare 1 selection）： 这 2 位定义通道的方向(输入/输出)，及输入信号的选择： 00：CC1 通道被配置为输出 01：CC1 通道被配置为输入，IC1 映射在 TI1 上 10：CC1 通道被配置为输入，IC1 映射在 TI2 上 11：保留 注：CC1S 仅在通道关闭时（TIMx_CCER 寄存器的 CC1E=0）才是可写的。

14.4.9 TIMx 捕获/比较模式寄存器 2【复用】（TIM18_CCMR2）

偏移地址：0x01C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
								rw	rw			rw	rw	rw	

位	名称	描述
31:8	RSV	保留位，固定读为 0
7	OC3CE	输出比较 3 清 0 允许 0: OC3REF 不受 OCREF_CLR 输入（比较器的输出）的影响。 1: 一旦检测到 OCREF_CLR 输入为高电平，清除 OC3REF=0。
6:4	OC3M[2:0]	输出比较模式 3 (Output Compare 1 mode) 该 3 位定义了输出参考信号 OC1REF 的行为，而 OC1REF 决定了 OC1、OC1N 的电平。OC1REF 是高电平有效，而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用（此模式用于产生一个时基）。 001：匹配时，设置通道 1 为有效电平。当 TIMx_CNT = TIMx_CCR1 时，强制 OC1REF 为高。 010：匹配时，设置通道 1 为无效电平。当 TIMx_CNT = TIMx_CCR1 时，强制 OC1REF 为低。 011：翻转。当 TIMx_CCR1=TIMx_CNT 时，翻转 OC1REF 的电平。 100：强制为无效电平。强制 OC1REF 为低。 101：强制为有效电平。强制 OC1REF 为高。 110：PWM 模式 1 - 在向上计数时，当 TIMx_CNT<TIMx_CCR1 时，通道 1 为有效电平 (OC1REF=1)，否则为无效电平（OC1REF=0）。 111：PWM 模式 2 - 在向上计数时，当 TIMx_CNT<TIMx_CCR1 时，通道 1 为无效电平，否则为有效电平。

位	名称	描述
		注 1：在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。
3	OC3PE	输出比较 3 预装载允许（Output Compare 1 preload enable）： 0：禁止 TIMx_CCR1 寄存器的预装载功能，可随时写入 TIMx_CCR1 寄存器，并且新写入的数值立即起作用。 1：开启 TIMx_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。
2	OC3FE	输出比较 3 快速使能（Output Compare 1 fast enable）： 该位用于加快 CC 输出对触发输入事件的响应。 0：CC3 的正常操作依赖于计数器与 CCR1 的值，即使工作于触发器状态。当触发器的输入有一个有效沿时，激活 CC3 输出的最小延时为 5 个时钟周期 1：输入到触发器的有效沿的作用等同于发生了一次比较匹配。因此，OC 被设置为比较电平（独立于比较结果）。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。
1:0	CC3S[1:0]	捕获/比较 3 选择（Capture/Compare 1 selection）： 这 2 位定义通道的方向（输入/输出），及输入信号的选择： 00：CC3 通道被配置为输出 01：CC3 通道被配置为输入，IC3 映射在 TI3 上 10：保留 11：保留 注：CC3S 仅在通道关闭时（TIMx_CCER 寄存器的 CC3E=0)才是可写的。

14.4.10 TIMx 捕获/比较模式寄存器 1【复用】（TIM18_CCMR2）

偏移地址：0x01C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
								rw				rw		rw	

位	名称	描述
31:8	RSV	保留位，固定读为 0
7:4	IC3F[3:0]	输入捕获 3 滤波器（Input capture 3 filter): 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变: 0000：无滤波器，以 f_{DTS} 采样 0001：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=2 0010：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=4 0011：采样频率 $f_{SAMPLING}=f_{CK\_INT}$，N=8 0100：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=6 1001：采样频率 $f_{SAMPLING}=f_{DTS}/8$，N=8 1010：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=5 1011：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率 $f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率 $f_{SAMPLING}=f_{DTS}/32$，N=8 注：当 ICxF[3:0]=1、2 或 3 时，公式中的 f_{DTS} 由 CK_INT 替代。
3:2	IC3PSC[1:0]	输入捕获 3 预分频器: 这 2 位定义了 CC3 输入（IC3）的预分频系数。一旦 CC3E=0（TIMx_CCER 寄存器中），则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01：每 2 个事件触发一次捕获 10：每 4 个事件触发一次捕获 11：每 8 个事件触发一次捕获
1:0	CC3S[1:0]	捕获/比较 3 选择（Capture/Compare 3 selection): 这 2 位定义通道的方向（输入/输出）及输入信号的选择: 00：CC3 通道被配置为输出 01：CC3 通道被配置为输入，IC3 映射在 TI3 上 10：保留 11：保留 注：CC3S 仅在通道关闭时（TIMx_CCER 寄存器的 CC3E=0）才是可写的。

14.4.11 TIMx 捕获/比较使能寄存器（TIM18_CCER）

偏移地址：0x020

复位值：0x0000 0000

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

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

位	名称	描述
31:12	RSV	保留位，固定读为 0
11	CC3NP	捕捉/比较 3 输出极性（Capture/Compare 3 complementary output Polarity） 通道 CC3 配置为输出时，CC3NP 必须设置为清除，CC3NP = 0。 通道 CC3 配置为输入时，CC3NP 与 CC3P 联合控制 TI3FP3 的极性。 参考 CC1P 描述。
10	RSV	保留位，固定读为 0
9	CC3P	捕获/比较 3 输出极性，详见 CC1P
8	CC3E	捕获/比较 3 输出使能，详见 CC1E
7	CC2NP	捕捉/比较 2 输出极性（Capture/Compare 2 complementary output Polarity） 通道 CC2 配置为输出时，CC2NP 必须设置为清除，CC2NP = 0。 通道 CC2 配置为输入时，CC2NP 与 CC2P 联合控制 TI2FP3 的极性。 参考 CC1P 描述。
6	RSV	保留位，固定读为 0
5	CC2P	捕获/比较 2 输出极性，详见 CC1P
4	CC2E	捕获/比较 2 输出使能，详见 CC1E
3	CC1NP	捕捉/比较 1 输出极性（Capture/Compare 1 complementary output Polarity） 通道 CC1 配置为输出时，CC1NP 必须设置为清除，CC1NP = 0。 通道 CC1 配置为输入时，CC1NP 与 CC1P 联合控制 TI1FP1 的极性。 参考 CC1P 描述。
2	RSV	保留位，固定读为 0
1	CC1P	捕获/比较 1 输出极性（Capture/Compare 1 output polarity） **CC1 通道配置为输出:** 0：OC1 高电平有效； 1：OC1 低电平有效。 **CC1 通道配置为输入:**

位	名称	描述
		CC1NP/CC1P 位选择在触发或捕获模式下 TI1FP1 和 TI2FP1 的有效极性。 00：非反相/上升沿 - TIxFP1 的上升沿（在复位、外部时钟或触发模式下的捕获或触发操作） - TIxFP1 非反相（在门控模式） 01：反相/下降沿 - TIxFP1 的下降沿（在复位、外部时钟或触发模式下的捕获或触发操作）， - TIxFP1 反相（在门控模式） 10：保留，此配置不用 11：非反相/上升或下降沿 - TIxFP1 的上升沿和下降沿（在复位、外部时钟或触发模式下的捕获或触发操作） - TIxFP1 非反相（在门控模式）
0	CC1E	捕获/比较 1 输出使能（Capture/Compare 1 output enable） **CC1 通道配置为输出：** 0：关闭—OC1 未激活，OC1N 的电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 这些位的功能。 1：开启—OC1 信号输出到对应的输出引脚，其输出依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 这些位的功能。 **CC1 通道配置为输入：** 本位用于决定捕获的计数器值是否要装载到捕获/比较寄存器 1（TIMx_CCR1）。 0：捕获禁止 1：捕获允许

14.4.12 TIMx 计数器（TIM18_CNT）

偏移地址：0x024

复位值：0x0000 0000

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	CNT[15:0]	计数器值

14.4.13 TIMx 预分频器（TIM18_PSC）

偏移地址：0x028

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	PSC[15:0]	预分频值（Prescaler value） 计数器的时钟频率（CK_CNT）= f_{CK_PSC}/(PSC[15:0]+1)。每次当更新事件产生时，PSC 的值被装入当前预分频器寄存器；更新事件包括计数器被 TIM_EGR 的 UG 位清‘0’或被配置为复位模式的触发器控制器清‘0’。

14.4.14 TIMx 自动重装载寄存器（TIM18_ARR）

偏移地址：0x02C

复位值：0x0000 FFFF

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	ARR[15:0]	自动重装载的值（Auto-reload value） ARR 包含了将要装载入的实际自动重装载寄存器的值。详细参考章节：14.3.1 时基单元，有关 ARR 的更新和行为。 当自动重装载的值为空时，计数器不工作。

14.4.15 TIMx 捕获/比较寄存器 1（TIM18_CCR1）

偏移地址：0x034

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw															

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	CCR1[15:0]	捕获/比较通道 1 的值 若 CC1 通道配置为输出: CCR1 是要加载到实际捕获/比较 1 寄存器的值（预装载值）。 如果在 TIMx_CCMR1 寄存器（OC1PE 位）中未选择预装载功能，写入的数值会永久加载。否则只有当更新事件发生时，此预装载值才加载至内部活动的捕获/比较 1 寄存器中。 活动的捕获/比较寄存器与计数器 TIMx_CNT 进行比较，并在 OC1 端口上产生输出信号。 若 CC1 通道配置为输入: CCR1 是上一次输入捕获 1 事件（IC1）捕获的计数器值。

14.4.16 TIMx 捕获/比较寄存器 2（TIM18_CCR2）

偏移地址：0x038

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw															

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	CCR2[15:0]	捕获/比较通道 2 的值 若 CC2 通道配置为输出: CCR2 是要加载到实际捕获/比较 2 寄存器的值（预装载值）。 如果在 TIMx_CCMR1 寄存器（OC2PE 位）中未选择预装载功能，写入的数值会永久加载。否则只有当更新事件发生时，此预装载值才加载至内部活动的捕获/比较 2 寄存器中。 活动的捕获/比较寄存器与计数器 TIMx_CNT 进行比较，并在 OC2 端口上产生输出信号。 若 CC2 通道配置为输入: CCR2 是上一次输入捕获 2 事件（IC2）捕获的计数器值。

14.4.17 TIMx 捕获/比较寄存器 2（TIM18_CCR3）

偏移地址：0x03C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw															

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	CCR3[15:0]	捕获/比较通道 3 的值 若 CC3 通道配置为输出: CCR3 是要加载到实际捕获/比较 3 寄存器的值（预装载值）。 如果在 TIMx_CCMR2 寄存器（OC3PE 位）中未选择预装载功能，写入的数值会永久加载。否则只有当更新事件发生时，此预装载值才加载至内部活动的捕获/比较 2 寄存器中。 活动的捕获/比较寄存器与计数器 TIMx_CNT 进行比较，并在 OC3 端口上产生输出信号。 若 CC3 通道配置为输入: CCR3 是上一次输入捕获 3 事件（IC3）捕获的计数器值。

15 基本定时器（TIM6）

15.1 TIM6 简介

该定时器基于一个 16 位自动重载递增计数器和一个 16 位预分频器。

15.2 特性

- 12 位分辨率。
- 16 位自动装载计数器
- 16 位可编程预分频器，支持 1 ~ 65535 之间的任意分频
- 下述事件触发中断：
    - 更新（Update）：计数器上溢

15.3 功能描述

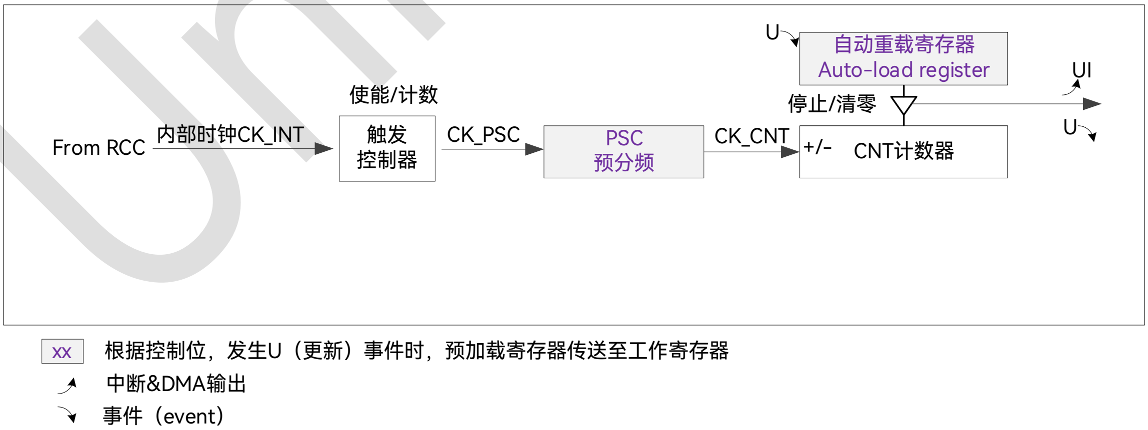


图 15-1：TIM6 模块框图

15.3.1 时基单元

TIM6 主要由 16 位计数器及相关自动重载寄存器构成，计数器支持向上计数。计数器时钟支持预分频。计数器寄存器、自动重载寄存器和预分频寄存器支持软件读写，即使计数器正在运行读写仍然有效。

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，读写自动重载寄存器将访问预装载寄存器。设置 TIMx_CR1 寄存器中的自动重载预装载使能位（ARPE），选择预装载寄存器的内容永久传送至缓存寄存器或在每次更新事件（UEV）传送至缓存寄存器。当计数器达到溢出条件并当 TIMx_CR1 寄存器中的 UDIS 位等于 0 时，产生更新事件。更新事件也可由软件产生。有关更新事件的产生，针对每种配置后续章节会详细描述。

计数器时钟由预分频后输出 CK_CNT 驱动，需置位 TIMx_CR1 寄存器中的计数器使能位（CEN）时，CK_CNT 才有效（请参阅从模式控制器描述以获得计数器使能的更多细节）。

注：设置 TIMx_CR 寄存器的 CEN 位，计数器会立即开始计数。

15.3.1.1 预分频

预分频器支持计数器时钟 1~65536 之间任意分频，基于 1 个 16 位的计数器，由 TIMx_PSC 寄存器中的 16 位寄存器控制。此寄存器内部带有缓存器，支持运行时修改；新修改的预分频值在下一次的更新事件（update event）发生时生效。

下图给出运行时更改预分频值，计数器的行为。

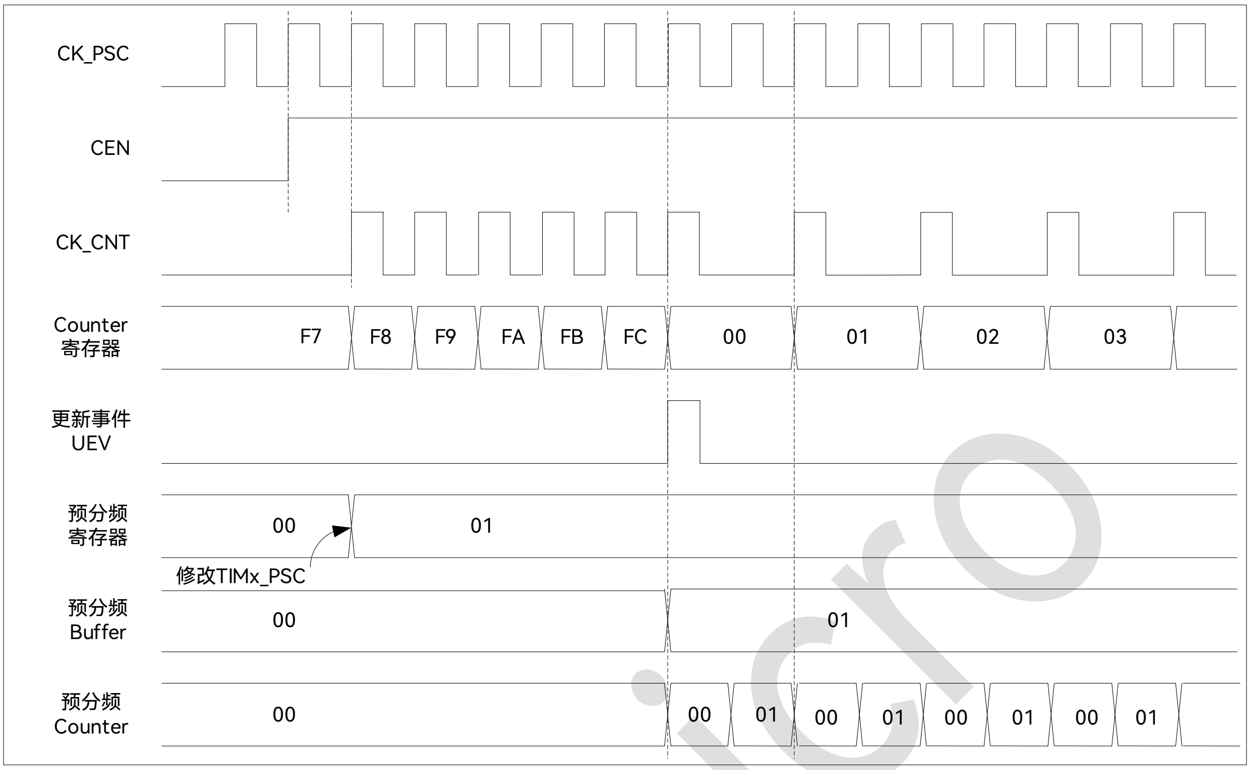


图 15-2：计数器时序图：预分频修改，从 1 到 2

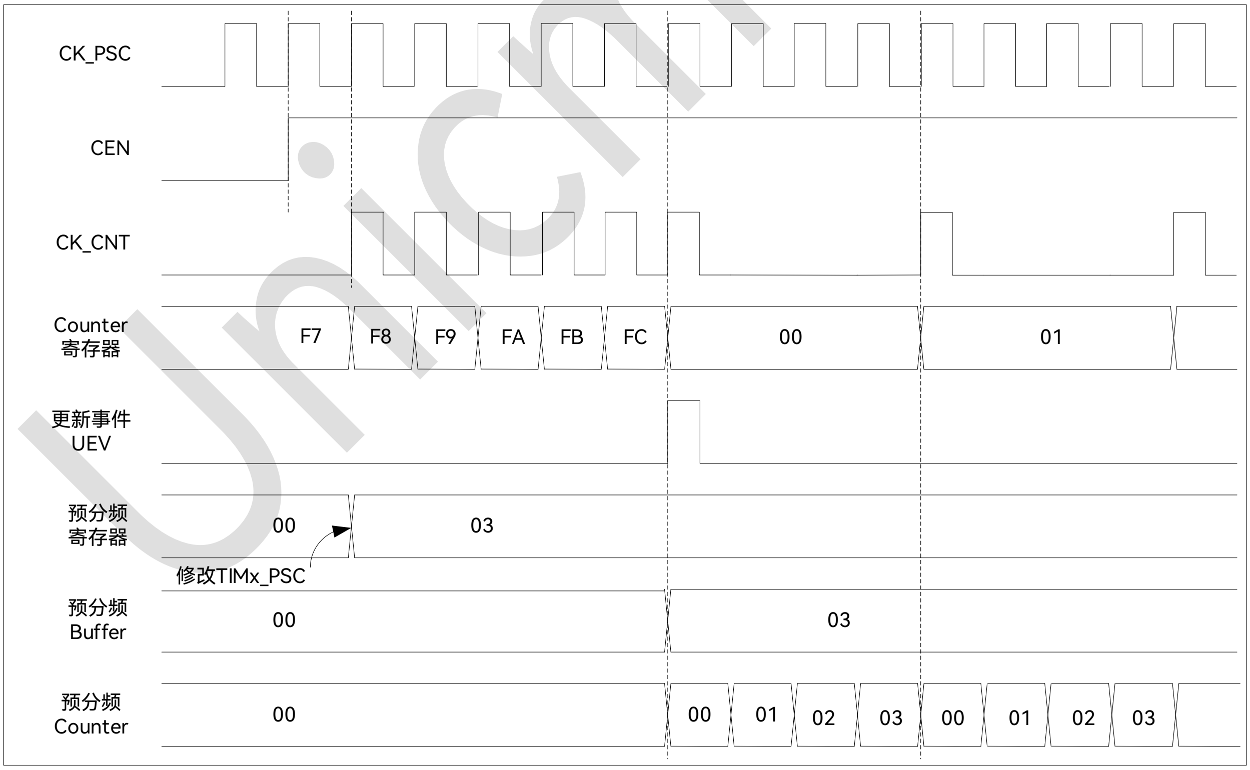


图 15-3：计数器时序图：预分频修改，从 1 到 4

15.3.2 计数器模式

在向上计数模式中，计数器从 0 计数到自动重载值（TIMx_ARR 计数器的值），然后重新从 0 开始计数并且产生一个计数器溢出事件（overflow）。

每次计数器溢出时都会产生更新事件。

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

通过软件置位 TIMx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新缓存寄存器。在 UDIS 位被清‘0’之前，将不产生更新事件。但是，在应该产生更新事件时，计数器会被清‘0’，同时预分频器的计数器也被清 0（但预分频器的数值不变）。

此外，如果置位 TIMx_CR1 寄存器中的 URS 位（update request selection），置位 UG 位将产生一个更新事件 UEV，但硬件不置位 UIF 标志（即不产生中断）。

当发生一个更新事件时，所有的寄存器都被更新，同时（依据 URS 位）置位更新标志位（TIMx_SR 寄存器中的 UIF 位）:

- 预分频器的缓冲区被写入预装载寄存器的值（TIMx_PSC 寄存器的值）
- 自动重载缓存寄存器更新为预装载寄存器的值（TIMx_ARR）

举例如下：TIMx_ARR=0x36 时，计数器在不同时钟频率下计数器的行为。

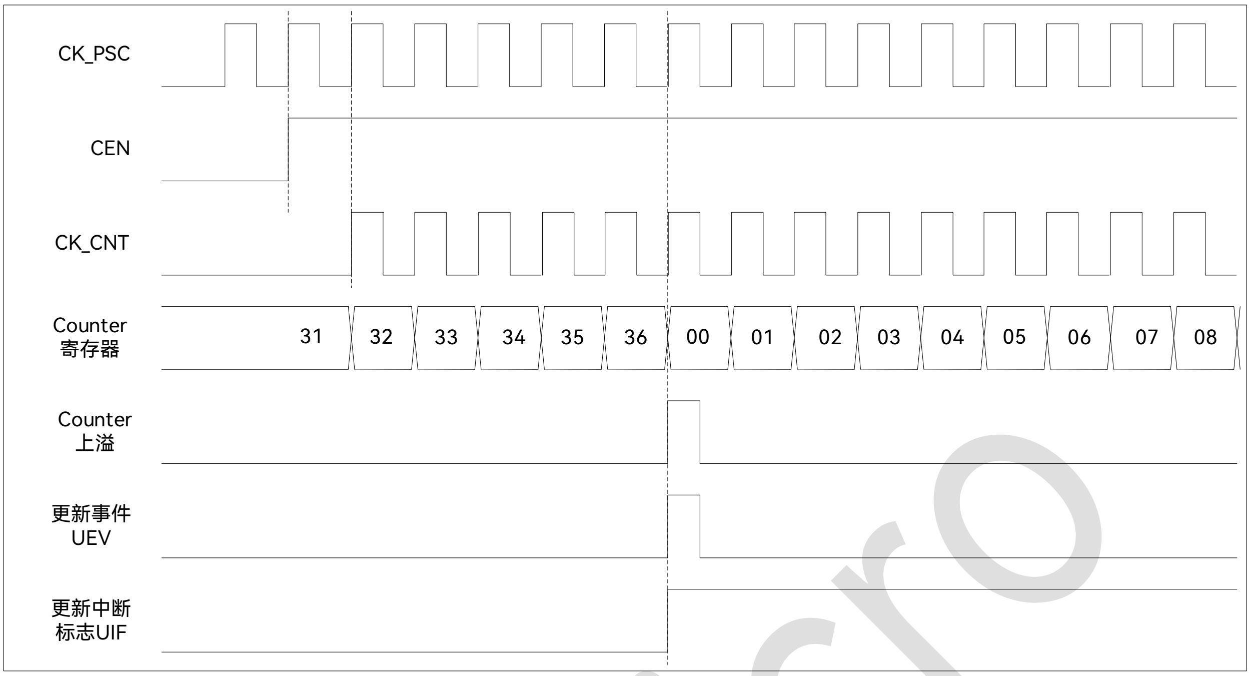


图 15-4：计数器时序图：预分频因子=1

图 15-5：计数器时序图：预分频因子=2

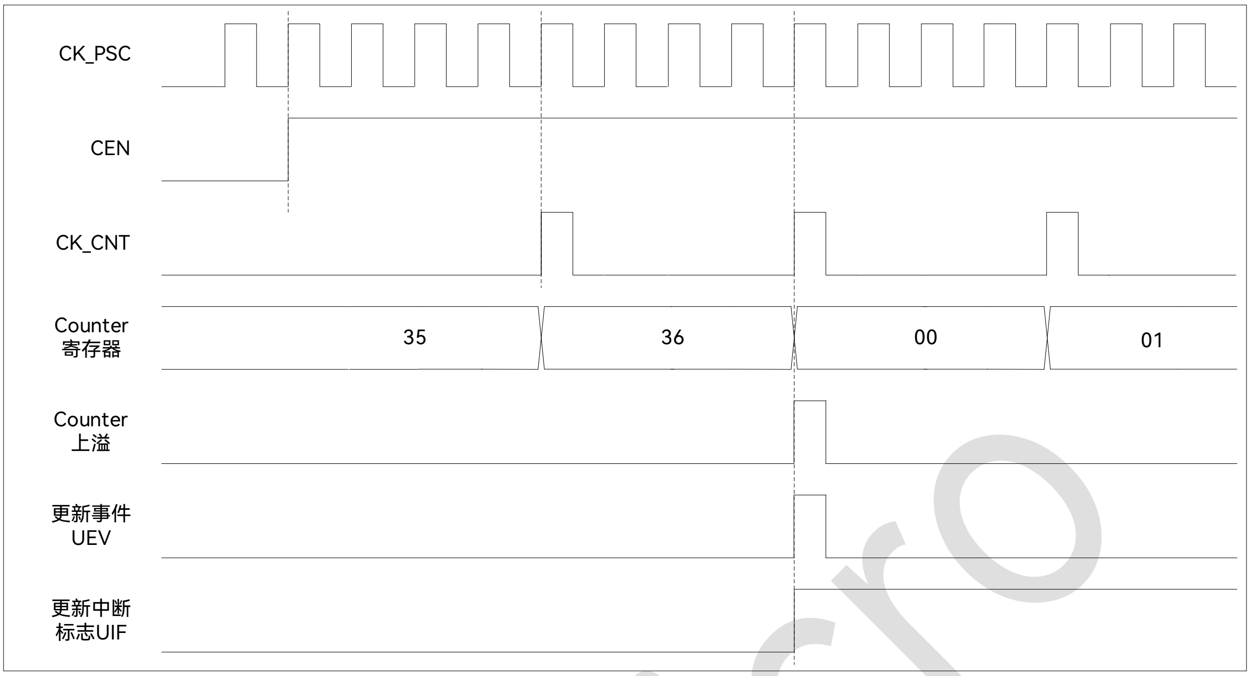


图 15-6：计数器时序图：预分频因子=4

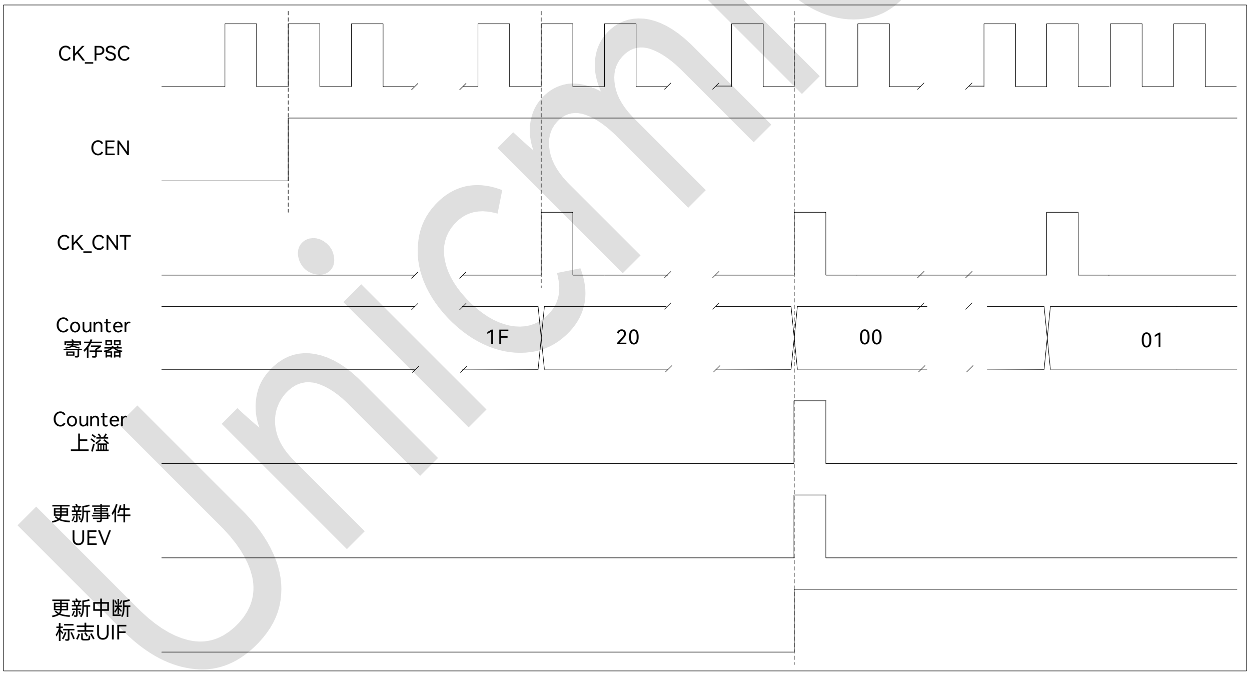


图 15-7：计数器时序图：预分频因子=N

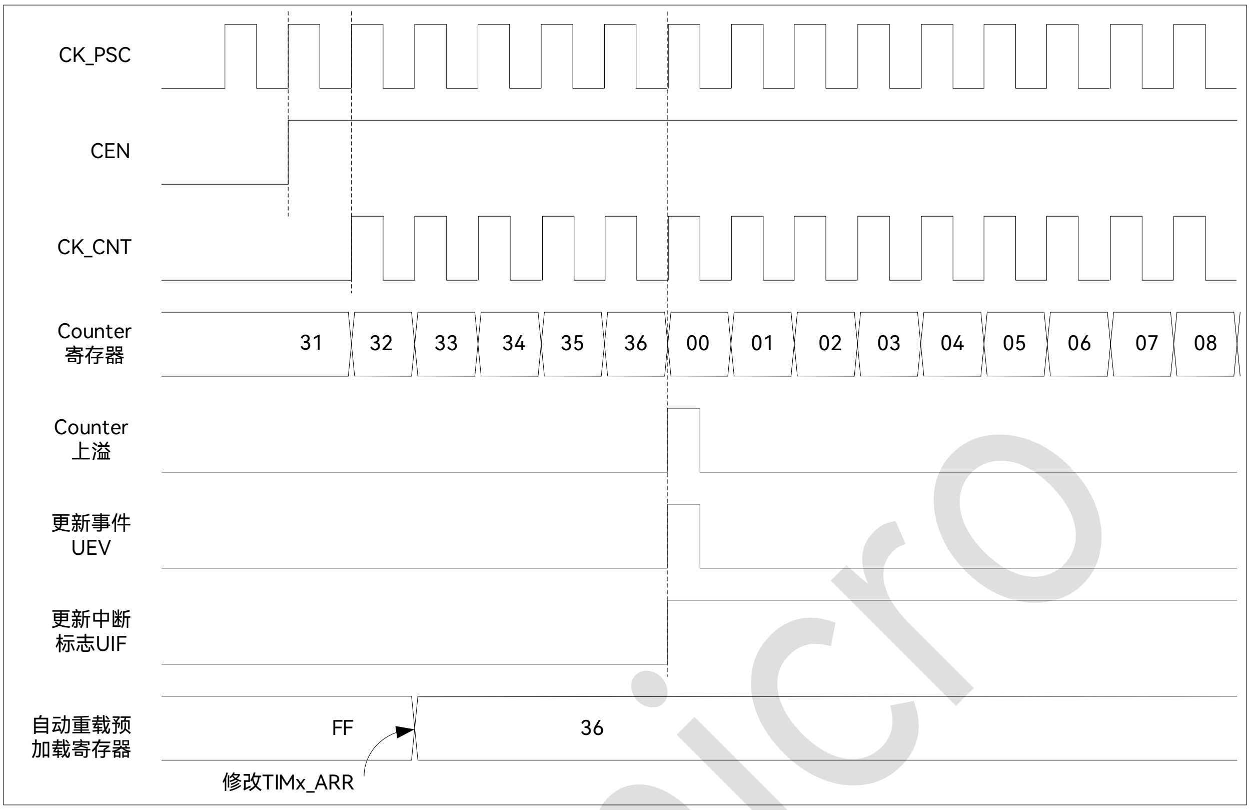


图 15-8：计数器时序图：预分频因子=1，ARPE=0（TIMx_ARR 没有预装载）

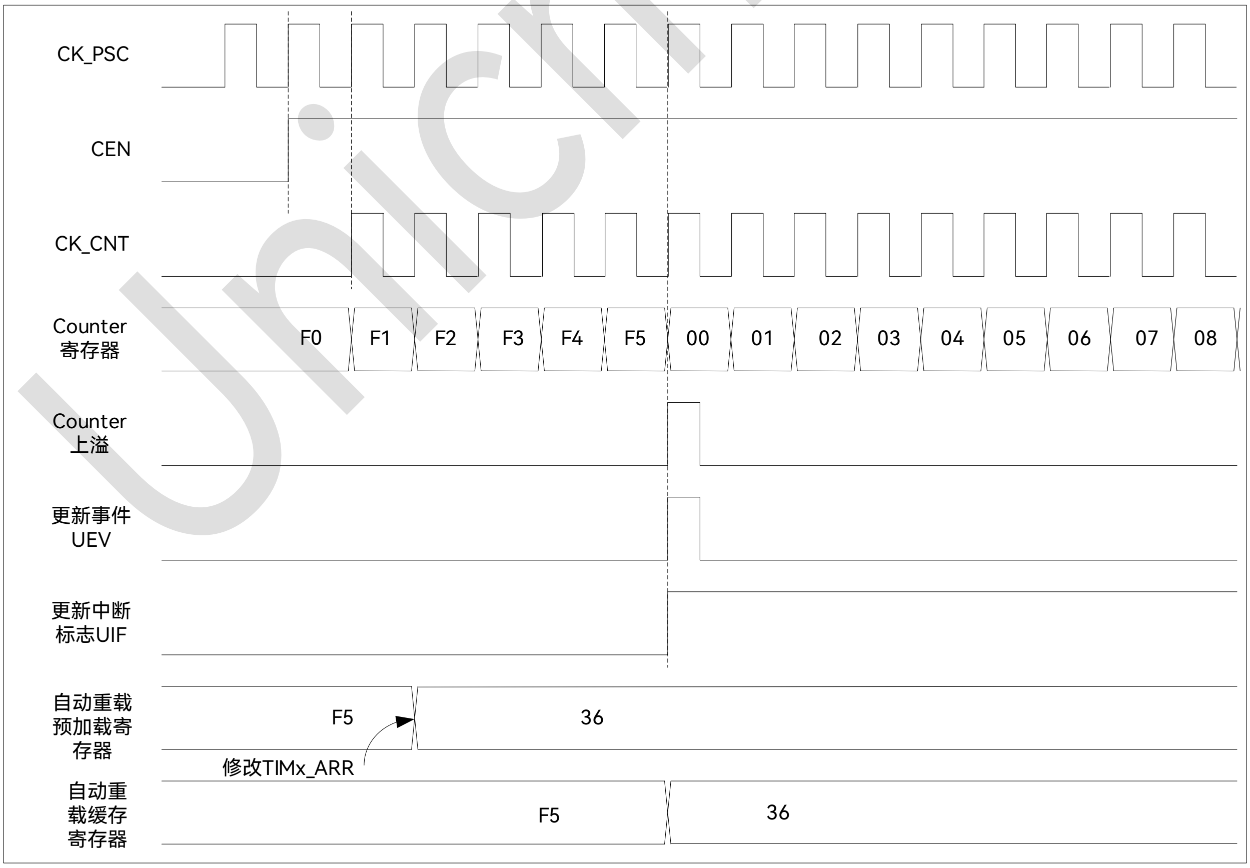


图 15-9：计数器时序图：预分频因子=1，ARPE=1（TIMx_ARR 已预装载）

15.3.3 时钟源

计数器时钟源包括：

- 内部时钟 CK_INT

15.3.3.1 内部时钟源（CK_INT）

CEN（TIMx_CR1 寄存器）和 UG 位（TIMx_ EGR 寄存器）为实际控制位，并且只能被软件修改（除了 UG 位保持自动被清除）。一旦 CEN 位被写成‘1’，预分频器的时钟就由内部时钟 CK_INT 提供。

图 15-10：计数器时序图：内部时钟，预分频因子=1，正常模式

15.3.4 调试模式

当微控制器进入调试模式时（Cortex-M0 内核停止），根据 DBG 模块中 DBG_TIMx_STOP 的设置，TIMx 计数器可以或者继续正常操作，或者停止。

15.4 相关寄存器

15.4.1 TIM 控制寄存器 1（TIM6_CR1）

偏移地址：0x000

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								ARPE	Res.				URS	UDIS	CEN
								rw					rw	rw	rw

位	名称	描述
31:8	RSV	保留位，固定读为 0
7	ARPE	自动重载预装载使能（Auto-reload preload enable）： 0: TIMx_ARR 寄存器没有缓冲 1: TIMx_ARR 寄存器有缓冲
6:3	RSV	保留位，固定读为 0
2	URS	更新请求源（Update request source） 软件置位和清零，选择 UEV 事件的源 0：下述任一事件产生更新中断： - 计数器上溢 - 置位 UG 位 1：仅计数器上溢产生更新中断。
1	UDIS	禁止更新（Update disable） 软件置位和清零，通过该位允许/禁止 UEV 事件的产生 0：允许 UEV。更新（UEV）事件由下述任一事件产生： - 计数器上溢 - 置位 UG 位 具有缓存的寄存器被装入它们的预装载值。 1：禁止 UEV。不产生更新事件，影子寄存器（ARR、PSC、CCRx）保持它们的值。如果置位 UG 位，则计数器和预分频器被重新初始化。
0	CEN	使能计数器（Counter enable）： 0：禁止计数器 1：使能计数器

15.4.2 TIM DMA/中断使能寄存器（TIM6_DIER）

偏移地址：0x00C

复位值：0x0000 0000

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

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

位	名称	描述
31:1	RSV	保留位，固定读为 0
0	UIE	更新中断使能： 0：更新中断禁止 1：更新中断允许

15.4.3 TIMx 中断标志和清除寄存器（TIM6_SR）

偏移地址：0x010

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															UIF
															rc_w0

位	名称	描述
31:1	RSV	保留位，固定读为 0
0	UIF	更新中断标志（Update interrupt flag） 当产生更新事件时该位由硬件置’1’。它由软件清’0’。 0：无更新事件产生

位	名称	描述
		1：更新中断等待响应。当寄存器被更新时该位由硬件置‘1’： - 若 TIMx_CR1 寄存器的 URS=0、UDIS=0，当设置 TIMx_EGR 寄存器的 UG=1，进而实现软件对计数器 CNT 重新初始化时。 - 若 TIMx_CR1 寄存器的 URS=0、UDIS=0，当计数器 CNT 被触发事件重新初始化时。

15.4.4 TIMx 事件产生寄存器（TIM6_EGR）

偏移地址：0x014

复位值：0x0000 0000

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

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

位	名称	描述
31:1	RSV	保留位，固定读为 0
0	UG	产生更新事件（Update generation） 该位由软件置‘1’，由硬件自动清‘0’。 0：无动作 1：重新初始化计数器，并产生一个（寄存器）更新事件。注意预分频器的计数器也被清‘0’（但是预分频系数不变）。

15.4.5 TIMx 计数器（TIM6_CNT）

偏移地址：0x024

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	CNT[15:0]	计数器值

15.4.6 TIMx 预分频器（TIM6_PSC）

偏移地址：0x028

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	PSC[15:0]	预分频值（Prescaler value） 计数器的时钟频率（CK_CNT）= f_{CK_PSC}/(PSC[15:0]+1)。每次当更新事件产生时，PSC 的值被装入当前预分频器寄存器；更新事件包括计数器被 TIMx_EGR 的 UG 位清‘0’或被配置为复位模式的触发器控制器清‘0’。

15.4.7 TIMx 自动重装载寄存器（TIM6_ARR）

偏移地址：0x02C

复位值：0x0000 FFFF

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	ARR[15:0]	自动重装载的值（Auto-reload value）: ARR 包含了将要装载入的实际自动重装载寄存器的值。详细参考时基单元，有关 ARR 的更新和行为。 当自动重装载的值为空时，计数器不工作。

16 独立看门狗（IWDG）

16.1 概述

本芯片内嵌一个独立看门狗模块，用于检测和解决某些软件故障问题，当看门狗计数器达到预定门限，将触发一个系统复位请求。

独立看门狗时钟由特定的 LSI 时钟驱动，在主时钟失效情况下，仍可以保持正常工作。

IWDG 最适用于：在主程序外，看门狗作为独立进程的应用，而且对时钟精度要求较低。

16.2 特性

- 自由运行的向下计数器
- 由独立的低频 RC 振荡器驱动，在 STOP 模式下可以工作
- 复位条件
- 向下计数器计数至 0x000，产生复位
- 向下计数器的值高于窗口值时产生一次重加载操作（向 IWDG_KR 中写入 0x0000AAAA），产生复位

16.3 功能描述

当向独立看门狗的关键字寄存器（IWDG_KR）写入启动指令 0x0000CCCC 的时候，看门狗计数器开始由复位值 0xFFF 向下计数。当计数值达到 0x000 的时候由独立看门狗发出复位信号（IWDG reset）。

任何时候将关键字 0x0000AAAA 写到 IWDG_KR 寄存器中，都会使得 IWDG_RL 寄存器中的值被重加载到看门狗计数器中，从而阻止即将发生的复位动作。

一旦开启，独立看门狗将不能被关闭。

16.3.1 IWDG 模块框图

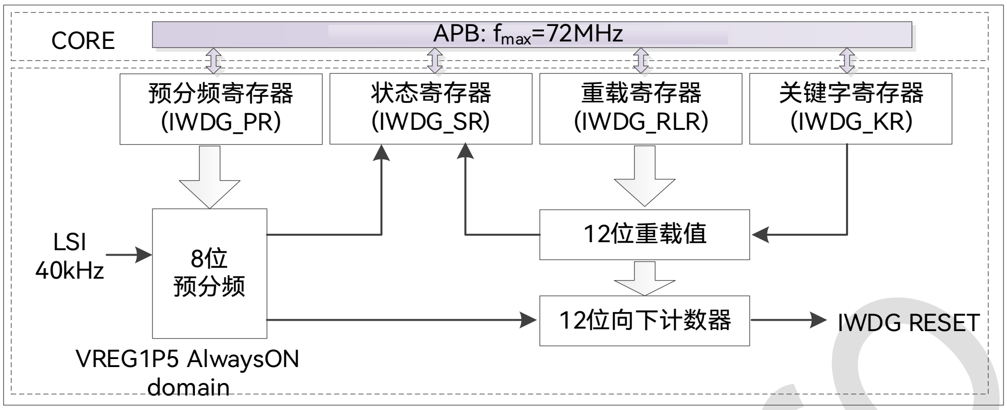


图 16-1：IWDG 框图

16.3.2 窗口功能

在 IWDG_WINR 寄存器中设置适当的窗口值，IWDG 也能够工作在窗口看门狗模式下。

如果重加载操作时，看门狗计数器的值高于窗口寄存器（IWDG_WINR)的值，将发生复位。

IWDG_WINR 的默认值是 0x00000FFF，所以如果没有改写它，那么窗口功能选项默认是关闭的。

窗口值一旦改变，立即就会引起看门狗计数器的一次更新动作，将其更新为 IWWDG_RLR 中所设置的值，从而一定程度上延缓目前到下次复位所需的时间周期。

窗口功能使能时 IWDG 配置：

1. 将 0x0000CCCC 写到 IWDG_KR 寄存器，使能 IWDG。
2. 向 IWDG_KR 寄存器 0x00005555 打开寄存器访问许可。
3. 向 IWDG_PR 写 0 ~ 7 的值，以配置 IWDG 的预分频器。
4. 配置重加载寄存器（IWDG_RLR)。
5. 等待状态寄存器 IWDG_SR 的值更新为 0x00000000。
6. 配置窗口寄存器 IWDG_WINR。这将会引起自动将 IWDG_RLR 的值更新到看门狗计数器。

注： 当 IWDG_SR 的值为 0x00000000 时，写窗口值的动作会使 RLR 的值更新至计数器。

窗口功能关闭时 IWDG 配置：

1. 将 0x0000CCCC 写到 IWDG_KR 寄存器，使能 IWDG。

2. 向 IWDG_KR 寄存器写 0x00005555 打开寄存器访问许可。
3. 向 IWDG_PR 写 0 ~ 7 的值，以配置 IWDG 的预分频器。
4. 配置重加载寄存器（IWDG_RLR）。
5. 等待状态寄存器 IWDG_SR 的值更新为 0x00000000。
6. 向 IWDG_KR 寄存器写 0x0000AAAA，IWDG_RLR 的值会重加载到看门狗计数器中。

16.3.3 硬件看门狗

如果在选项字节中打开了“硬件看门狗" 功能，那么在上电的时候看门狗就被自动打开。

如果要避免硬件看门狗产生复位，需要在以下条件向 IWDG_KR 寄存器写入 0x0000AAAA：

- 在看门狗向下计数器计数结束前
- 向下计数的值在窗口值范围内

16.3.4 寄存器访问保护

默认情况下，对 IWDG_PR、IWDG_RLR 和 IWDG_WINR 的写访问操作都是受保护的。若要修改这些寄存器，必须先向 IWDG_KR 写入 0x0000 5555 解锁。如果写入别的值，将会打破解锁状态，寄存器的访问保护重新生效。也就是说，在做重加载操作的时候（向该寄存器写入 0x0000 AAAA）就属于这种情况。

状态寄存器（IWDG_SR）可以指示预分频器的更新、看门狗计数器的重加载或窗口值的更新。

16.3.5 Debug 模式

当微控制器进入调试模式（内核停止）时，IWDG 计数器要么继续正常工作，要么停止，这取决于 DBG 模块的 DBG_IWDG_STOP 配置位。

16.4 相关寄存器

16.4.1 独立看门狗关键字寄存器（IWDG_KR）

偏移地址：0x000

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，始终读为 0
15:0	KEY[15:0]	关键值（只写，读的话会是 0x0000） 这些位必须周期性的由软件写入 0xAAAA，否则当计数器向下计数到 0 的时候会产生硬件复位请求。 写入 0x5555 会使能对 IWDG_PR, IWDG_RLR 和 IWDG_WINR 三个寄存器的访问许可。 写入 0xCCCC 启动看门狗(除非已经由选项字节在上电的时候就启动了它)。

16.4.2 独立看门狗喂狗寄存器（IWDG_PR）

偏移地址：0x004

复位值：0x0000 0000

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

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

位	名称	描述
31:3	RSV	保留位，始终读为 0
2:0	PR[2:0]	预分频器 这些位平时处于写保护状态，由软件写入，用来选择对输入时钟的预分频系数。 000: 4 分频 001: 8 分频 010: 16 分频 011: 32 分频 100: 64 分频 101: 128 分频 110: 256 分频 111: 256 分频

16.4.3 独立看门狗重加载寄存器(IWDG_RLR)

偏移地址：0x008

复位值：0x0000 0FFF

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

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

位	名称	描述
31:12	RSV	保留位，始终读为 0
11:0	RL[11:0]	看门狗计数器重置值 这些位平时处于写保护状态，这个值是由软件来设置的，并且每次向 IWDG_KR 寄存器写入 0xAAAA 的时候，这个值会被更新到看门狗计数器中。如果想延时长一点，这个值就该大一些。因为看门狗计数器正是从这个值开始向下计数。定时的长度是由这个值和预分频器的设置值来共同决定的。参见数据手册中关于超时的数据。

16.4.4 独立看门狗窗口寄存器（IWDG_WINR）

偏移地址：0x010

复位值：0x0000 0FFF

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

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

位	名称	描述
31:12	RSV	保留位，始终读为 0
11:0	WIN[11:0]	看门狗计数器窗口值 这些位平时处于写保护状态，这些位包含的是窗口值和向下计数器的比较上限。 为了阻止复位信号的产生，必须在向下计数器递减到窗口值和 0 之间的某个值时重加载它。

17 串行外设接口（SPI）

17.1 概述

SPI 接口支持基于 SPI 协议的通信。

串行外设接口（SPI）协议，支持半双工，全双工和单工同步等方式与外部设备的串行通信。该接口可配置为主机模式，在这种情况下，它向外部的从属设备提供通信时钟（SCK）。此接口也能够以多主机配置的方式操作。

17.2 SPI 特性

- 主机或从机模式
- 三线全双工同步传输
- 两线半双工同步传输（双向数据线）
- 两线单工同步传输（单向数据线）
- 4 位~16 位数据位宽选择
- 多主机模式
- 主机模式波特率分频器 8 档可选，波特率高达 $f_{PCLK}/2$
- 主机和从机模式下都可以由硬件或软件管理 NSS：主/从模式操作的动态变化
- 可编程时钟极性和相位
- 可编程数据位序：MSB-first 或 LSB-first 移位
- 专用的发送和接收状态标志，全部支持中断触发
- SPI 总线忙状态标志
- SPI Motorola 方式支持
- 主模式故障、溢出标志支持中断触发
- 32 位独立的发送和接收 FIFO：RX FIFO 和 TX FIFO

17.3 功能描述

SPI 支持 MCU 与外部设备的同步和异步通讯，应用中可以通过轮询状态标志或 SPI 中断来管理通讯过程。

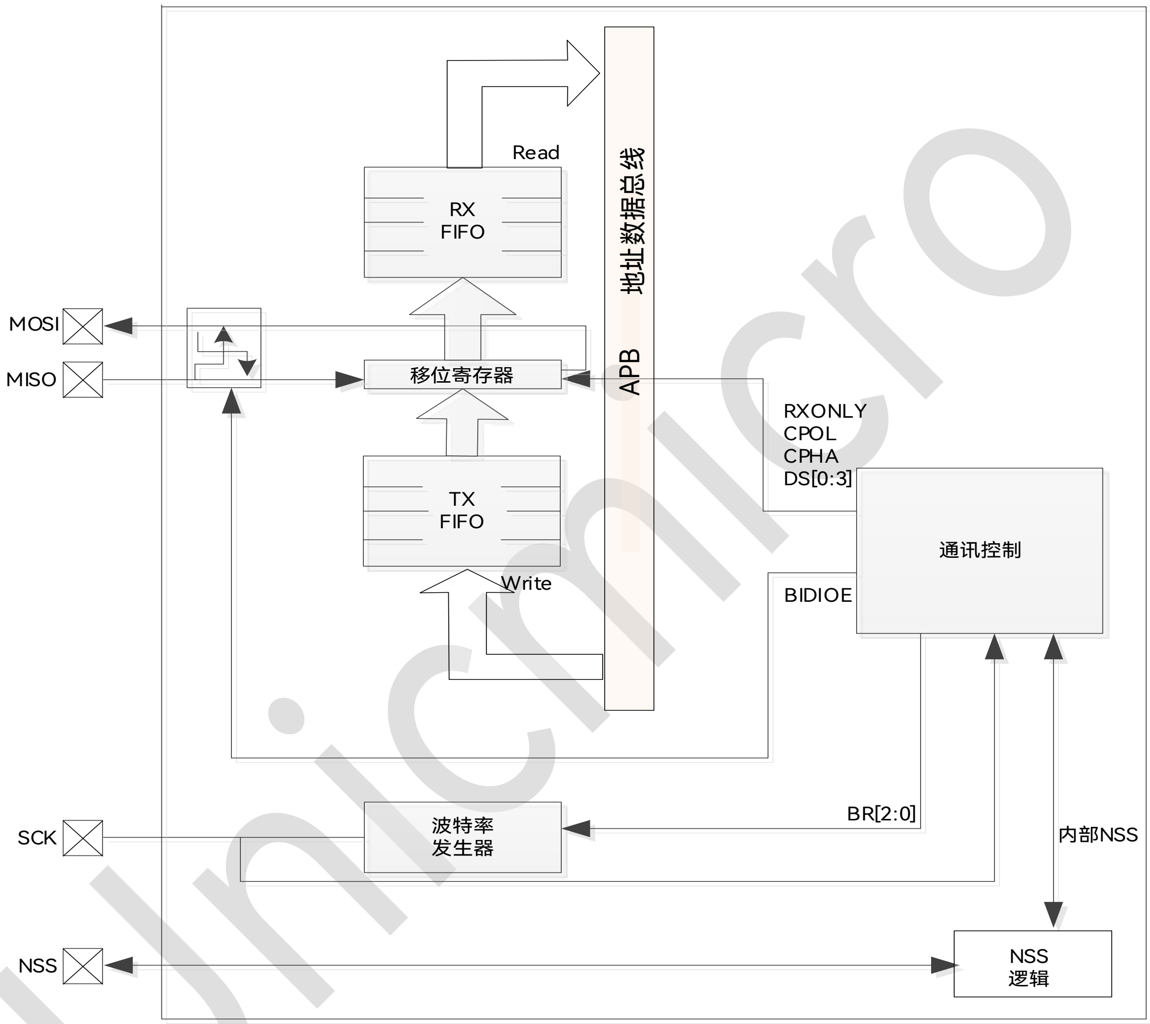


图 17-1：SPI 模块框图

通常 SPI 通过 4 个引脚与外部设备相连。

- MISO：主设备输入/从设备输出引脚。一般情况下，此引脚用于在从模式下的数据发送和主模式下的数据接收。
- MOSI：主设备输出/从设备输入引脚。一般情况下，此引脚用于在从模式下的数据接收和主模式下的数据发送。
- SCK：SPI 串行时钟输出引脚，作为主设备输出，从设备的输入。

- NSS：从机片选脚。根据 SPI 和 NSS 设置，此引脚可用于：
  - 选择一个从机来通信
  - 同步数据帧
  - 检测多个主机之间的冲突

  参见 17.3.3 从机选择（NSS）的引脚管理。

SPI 总线允许一个主设备和一个或多个从设备之间的通信。总线由至少有两条线——时钟信号和同步数据传输。其他信号基于 SPI 节点和主机间的数据交换目的来添加。

17.3.1 一个主设备和一个从机之间的通信

SPI 允许 MCU 根据目标的设备和应用程序的要求，使用不同的配置进行通信。这些配置使用 2 或 3 线（软件 NSS 管理）或 3 或 4 线（硬件 NSS 管理）。通信总是由主机发起的。

17.3.1.1 全双工通信

默认情况，SPI 被配置为全双工通信。在此配置中，主机和从机的移位寄存器通过两个单向线，MOSI 和 MISO 引脚进行连接。在 SPI 通信时，按照主机提供的 SCK 时钟沿进行同步数据传输。主机的数据经由 MOSI 发送到从机，从机的数据经由 MISO 线发送到主机。当数据帧传输完成（所有位转移）时，主机和从机间的信息交换就完成了。

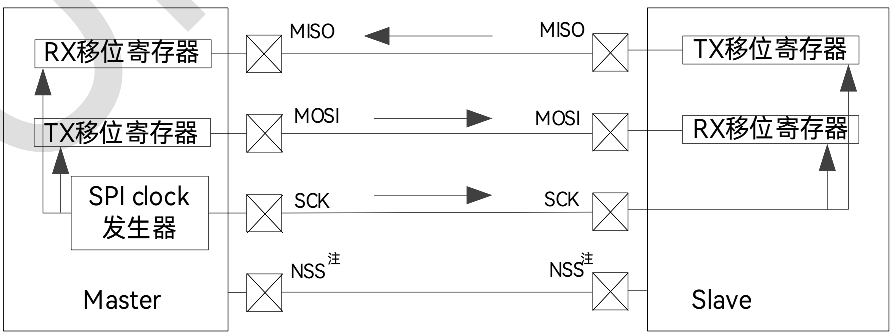


图 17-2：全双工单主/单从应用

注：NSS 引脚可以用于提供主/从机之间的硬件控制流，此引脚也可以悬空不接，主/机的控制使用其他信号来处理。

17.3.1.2 半双工通信

通过设置 SPIx_CR1 寄存器的 BIDIMODE 位，可以令 SPI 工作在半双工（half-duplex）模式下。在这种配置中，使用一条交叉连接线将主和从的移位寄存器连接在一起。在通信期间，数据按照 SPIx_CR1 寄存器的 BDIOE 位的设定，由主机移位寄存器同步 SCK 时钟沿传输到从机。在此配置中，主机的 MISO 引脚和从机的 MOSI 引脚都是自由的，可作为 GPIO 供其他应用程序使用。

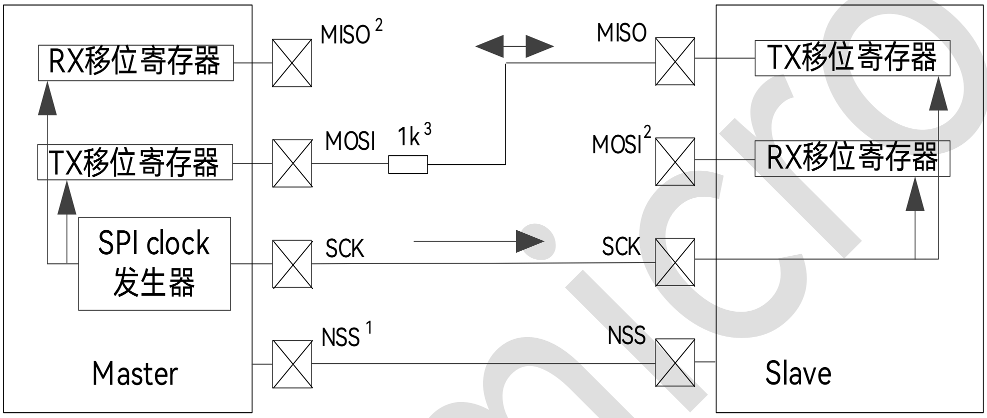


图 17-3：半双工单主/单从应用

注：

1. NSS 引脚可以用于提供主/从机之间的硬件控制流，此引脚也可以悬空不接，主/机的控制使用其他信号来处理。
2. 在此配置中，主机的 MISO 引脚和从机的 MOSI 引脚可以作为 GPIO 用。
3. 当工作在双向模式下的两个节点之间的通信方向发生了不同步的变化，新的发送端访问公共数据线，而旧的发送端仍然在数据线上保持相反的值（该值取决于 SPI 配置和通信数据）时，会发生电平竞争。建议此情况下 MOSI 和 MISO 串接电阻进行输出保护和限流。

17.3.1.3 简单通信

可以通过 SPIx_CR2 寄存器 RXONLY 位选择 SPI 工作在单工模式下，实现单发送或单接收通讯。在此配置中，使用一条交叉连接线将主和从的移位寄存器连接在一起。其余 MISO 和 MOSI 引脚不用于通信，并可以作为标准的 GPIO 使用。

- 单发送模式（RXONLY = 0）：配置设置和全双工相同。应用程序必须忽略在未使用的输入引脚上捕获的信息。这个脚可以作为一个标准的 GPIO 引脚。
- 单接收模式（RXONLY = 1）：应用程序可以设置 RXONLY 位以禁用 SPI 输出功能。在配置为从机时，MISO 输出被禁用，并且可以当作一个 GPIO 引脚来使用。当 slave 选择信号激活时，slave 持续从 MOSI 引脚接收数据（参见 17.3.3 从机选择（NSS）的引脚管理）。根据数据缓冲区的配置，会出现数据接收事件。在配置为主机时，MOSI 输出被禁用，并且可以当作一个 GPIO 引脚来使用。SPI 使能时，时钟信号会不断产生。要停止时钟输出，只有清除 RXONLY 位或 SPE 位，并等待直到从 MISO 引脚的输入样式的完成并填充数据缓冲区结构，这取决于其具体配置。

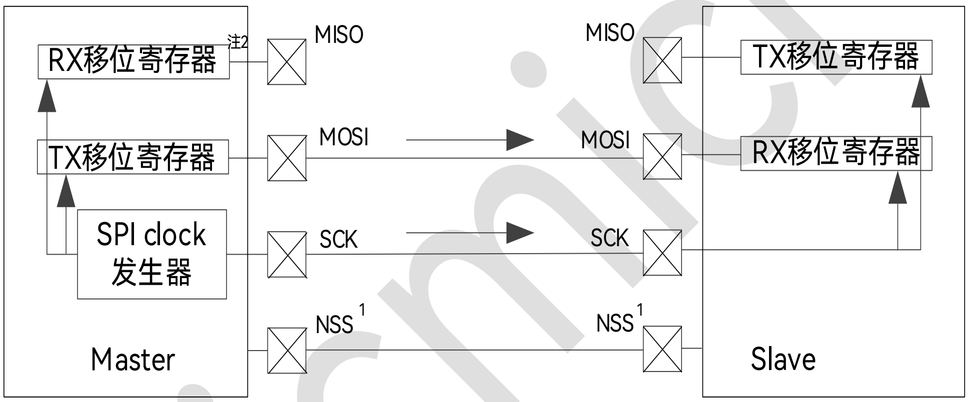


图 17-4：简单的单主/单从应用（主机仅发送/从机仅接收）

注：

1. NSS 引脚可以用于提供主/从机之间的硬件控制流，此引脚也可以悬空不接，但是主/机的控制需要使用各自内部的其他信号来处理。
2. 在移位寄存器中输入的信息被捕获，但在标准单发送模式下必须丢弃（例如，OVF 格式标志）。
3. 在此配置中，只要是 MISO 引脚都可用作 GPIO。
4. 简单通信模式可以被一个固定传输方向的半双工通信模式替代。

17.3.2 标准多从机通讯

在有两个或两个以上独立的从机的配置时，主机用 GPIO 引脚来管理每个从机的片选线。主机必须通过 GPIO 拉低其中一个从机的 NSS 引脚。一旦做到这一步，就会建立一个标准的主机和专用的通信渠道。

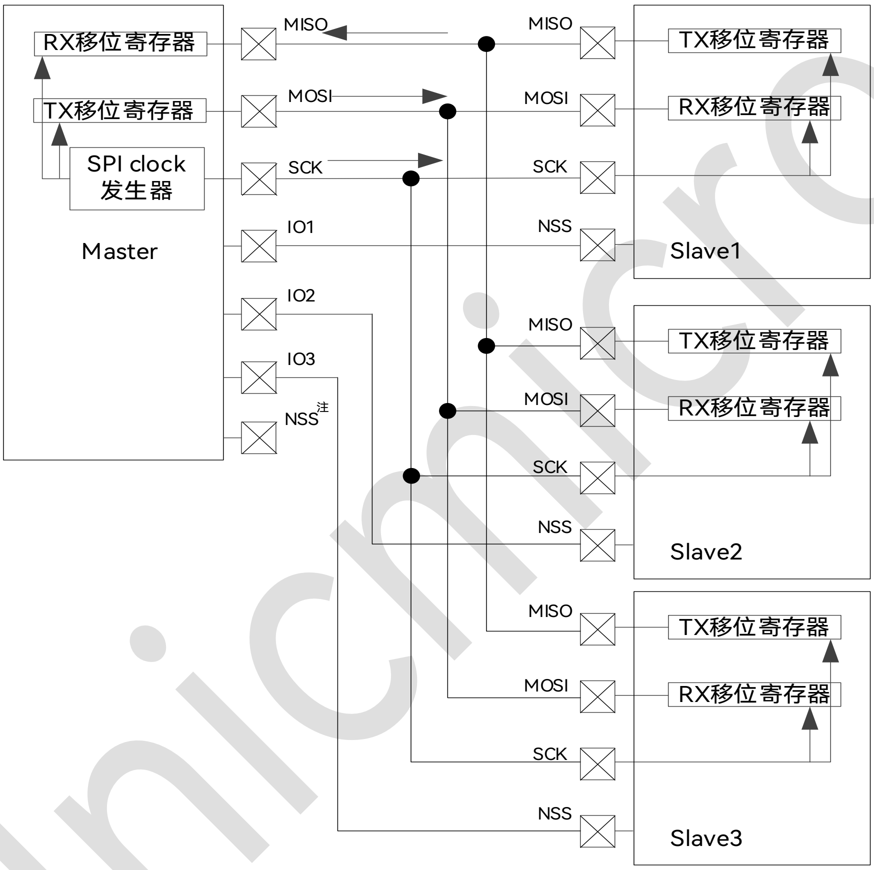


图 17-5：主机和 3 个独立的从机

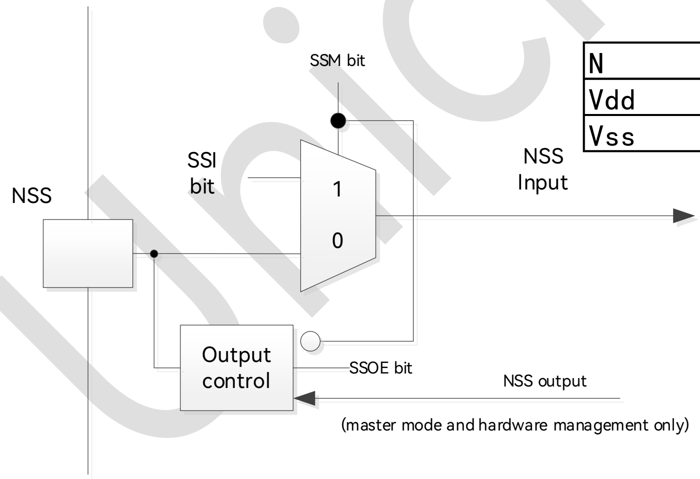
注：由于从机的 MISO 引脚连接在一起，所有的从机都必须将他们的 MISO 引脚设置为复用功能漏极开路状态（参见章节：6.3.6 I/O 复用功能输入/输出）。

17.3.3 从机选择（NSS）的引脚管理

在从机模式下时，NSS 作为一个标准的“片选”输入工作，并允许主从通信。在主机模式下，NSS 可以用来作为输入或输出。作为输入，它可以防止多主总线冲突；作为输出，它可以驱动一个单一的从机片选信号。

可以设置 SPIx_CR1 寄存器中的 SSM 位，来选择使用硬件或软件的从机选择管理：

- 软件 NSS 管理（SSM = 1）：在此配置中，从机选择信息由内部寄存器 SPIx_CR1 的 SSI 位值来驱动。外部 NSS 引脚可以由其他应用程序自由支配。
- 硬件 NSS 管理（SSM = 0）：在这种情况下，有两种可能的配置。这种设置由（寄存器 SPIx_CR1 SSOE 位）来决定 NSS 的输出。
  - NSS 输出使能（SSM = 0，SSOE = 1）：此配置仅用于当 MCU 作为主机时。NSS 引脚由硬件管理。在 SPI 作为主模式（SPE 的= 1）的同时 NSS 信号被拉低，并保持低直到 SPI 被禁止（SPE = 0）。如果 NSS 脉冲模式被打开（NSSP = 1），在连续通信间隔期间可以产生一个 NSS 脉冲。在这种 NSS 设置中，SPI 不能支持多重主机工作状态。
  - NSS 输出禁止（SSM = 0，SSOE = 0）：如果在总线上 MCU 作为主机，那么此配置就允许多主机通讯。如果在这种模式下 NSS 引脚被拉低，SPI 进入主模式故障状态，并自动重新配置为从机模式。在从机模式下，NSS 引脚作为标准的片选输入来工作，当 NSS 线被拉低时，从机被选中。



N	Master	Slave
Vdd	OK	Non active
Vss	Conflict	Active

图 17-6：硬件/软件从机选择管理

17.3.4 通讯格式

在 SPI 通信中，接收和发送操作同时进行。串行时钟（SCK）同步发送并同时采样数据线上的信息。通讯格式取决于时钟相位，时钟极性和数据帧格式。为了能够正常通讯，主机和从机必须遵循相同的通信格式。

17.3.4.1 时钟相位和极性控制

可以通过 SPIx_CR1 的 CPOL 和 CPHA 位用软件选择四种可能的时序关系。

CPOL（时钟极性）位控制着在没有数据在发送时的空闲状态的时钟输出电平。该位既针对主机模式也针对从机模式。如果 CPOL 被清零，SCK 引脚在闲置状态输出低电平。如果 CPOL 置 1，SCK 引脚在闲置状态输出高电平。

如果 CPHA 位被置 1，SCK 引脚上的第二沿对准的是第一位的捕获时机（如果 CPOL 位是 0，则为下降沿；如果 CPOL 位为 1，则为上升沿）。每次发生这个时钟切换时，数据被锁存。如果 CPHA 位为 0，SCK 引脚上的第一个沿对准第一位的捕获时机（如果 CPOL 位为 1，则为下降沿；如果 CPOL 位为 0，则为上升沿）。 每次发生这个时钟切换时，数据被锁存。

CPOL（时钟极性）和 CPHA（时钟相位）的选择共同决定数据采集时的时钟边沿。下图显示了 CPHA 和 CPOL 的四种组合的 SPI 全双工传输。

注：

1. 改变 CPOL / CPHA 位之前，SPI 必须通过清零 SPE 位先关闭。
2. 在 SCK 的空闲状态，必须符合在 SPIX_CR1 寄存器中对极性的选择（如果 CPOL = 1，上拉 SCK；如果 CPOL = 0，下拉 SCK）。

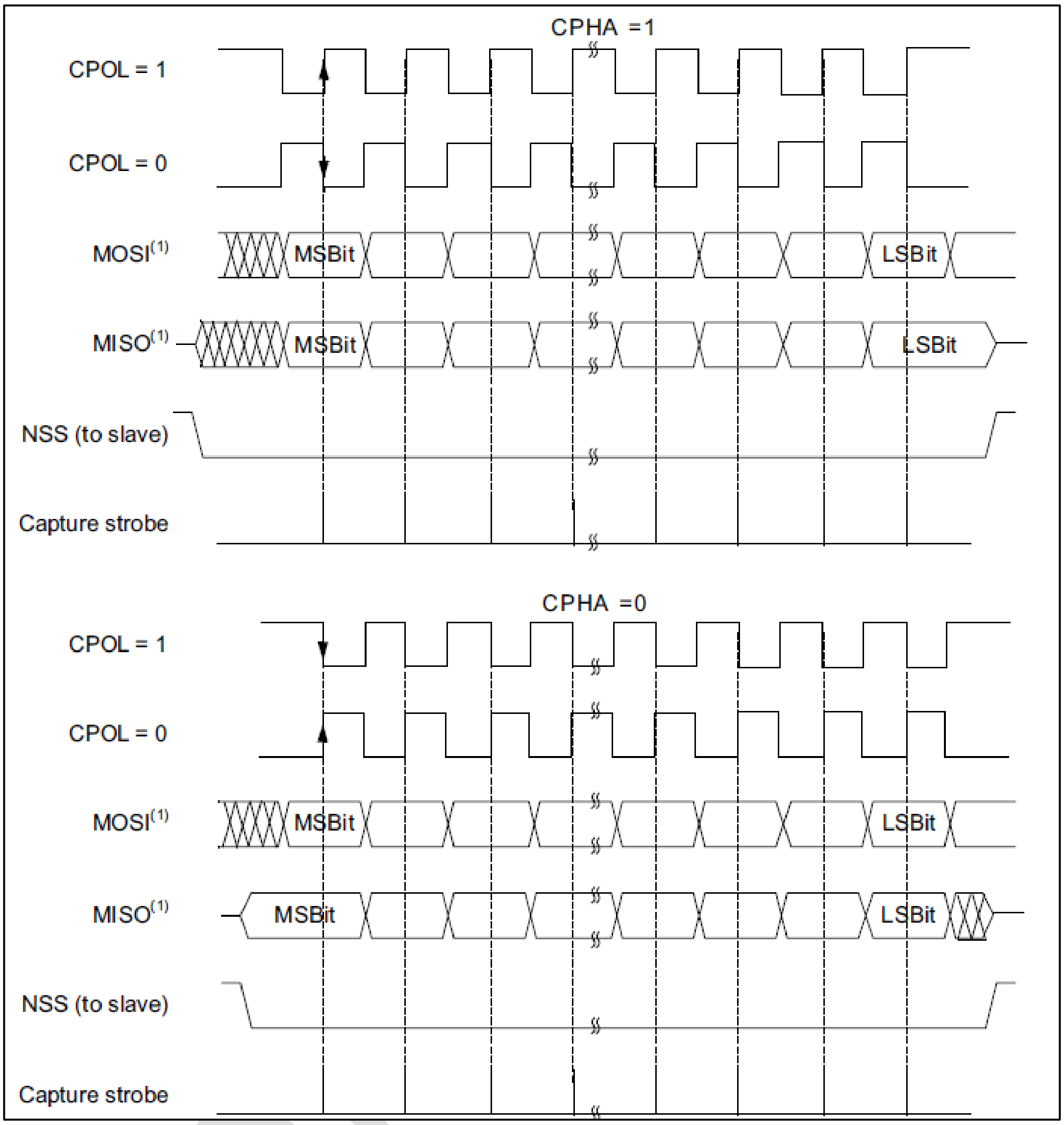


图 17-7：数据时钟时序图

注：这些时序显示的是 SPIx_CR1 寄存器的 LSBFIRST 位为 0 且 8 位字长（DS [3:0] = 0111）时的情形。

17.3.4.2 数据帧格式

SPI 移位寄存器可以设置 MSB 在前或 LSB 在前，取决于的 LSBFIRST 位的值。数据字长使用 DS[3:0]位选择，它可以设定从 4 位到 16 位的长度，同时适用于发送和接收。无论选定多大字长，对 FIFO 的读访问必须与 FRXTH 水平对齐。当访问 SPIx_DR 寄存器时，无论是一个字节（如果数据能够在一个字节中）还是一个字，数据帧总是右对齐（见下图）。通讯时，只有数据字长范围内的位会随时钟输出。

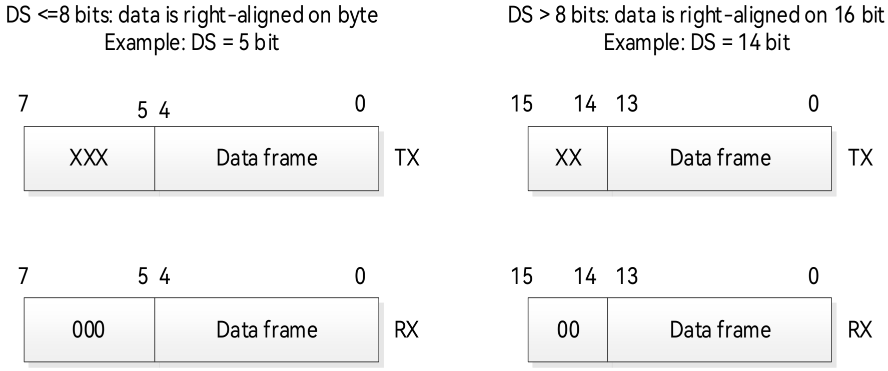


图 17-8：当数据长度不等于 8 位或 16 位时的数据对齐

注：最小的数据长度为 4 位。如果选择数据长度小于 4 位，会被硬件自动选择为 8 位的数据字长。

17.3.5 SPI 的初始化

主机和从机的初始化过程几乎是相同的。都是设置位配置寄存器 SPIx_CR1 和 SPIx_CR2：

1. 使用 BR [2:0]位选择串行时钟的波特率[注 1]。
2. 设置 CPOL 和 CPHA 位的组合，定义数据发送和串行时钟之间的四个关系[注 2]。
3. 配置 RXONLY，BIDIOE 和 BIDIMODE 选择传输模式[注 3]。
4. 设置 DS[3:0]位选择用于传送的数据长度。
5. 配置 LSBFIRST 位，定义帧格式[注 2]。
6. 根据应用的需要，设置 SSM，SSI 和 SSOE。在主机模式下，内部 NSS 信号必须在完整序列期间保持为高（请参见 17.3.3 从机选择（NSS）的引脚管理）。 在从机模式下，内部 NSS 信号必须在完整序列期间保持为低[注 2]。
7. 如果需要按 TI 协议，设置 FRF 位（见章节：TI 模式）。
8. 如果两个数据单位之间需要 NSS 脉冲，那么设置 NSSP 位以打开 NSS 脉冲模式。 此配置下 CHPA 位必须设置为 1 [注 2]。
9. 设置 FRXTH 位。RXFIFO 的阈值必须和对 SPIx_DR 寄存器的读访问的字长对齐。
10. 在 NSS 内部信号为高的时候设置 MSTR 位[注 1]（请参见 17.3.3 从机选择（NSS）的引脚管理）。
11. 设置 SPE 位，启用 SPI [注 3]。

注:

1. 这个步骤在从模式下不需要。
2. 这个步骤在 TI 模式中不要求。
3. 在任何主机单接收模式（RXONLY = 1 或 BIDIMODE = 1 & BIDIOE = 0），时钟在 SPI 使能后立即开始运行。

17.3.6 数据发送和接收流程

17.3.6.1 RXFIFO 和 TXFIFO

所有 SPI 数据换都通过 32 位的嵌入式 FIFO。这使 SPI 可以连续工作，防止短数据帧时的数据断流。每个方向都有它自己的 FIFO 称为 TXFIFO 和 RXFIFO。这些 FIFO 被用于除了单接收+CRC 模式外的所有的 SPI 模式（主从）（见章节：CRC 计算）。

对 FIFO 的处理会根据数据交换模式（双工，单工），数据帧格式（字长），对 FIFO 数据寄存器访问的大小（8 位或 16 位），以及是否按照数据包访问 FIFO（见章节：TI 的模式）。

对 SPIx_DR 寄存器的读访问，会返回存储在 RXFIFO 中但还未读的最老的数据。对 SPIx_ DR 的写访问会将新的数据放在发送队列的尾部。读访问必须总是和 SPIx_CR2 寄存器中的 FRXTH 位设置的 RXFIFO 门限对齐。FTVL [1:0]和 FRLVL [1:0]位，表明两个 FIFO 目前的缓冲存储程度。

对 SPIx_DR 寄存器的读访问必须由 RXNE 事件引发。这个时间在数据被存入 RXFIFO 并达到门槛（由 FRXTH 位定义）的时候被触发。在 RXNE 被清除时，认为是空的 RXFIFO 已经清空。类似的，写访问要由 TXE 事件来引发。当 TXFIFO 的存储状况少于或等于满额的一半的时候，将触发这个事件。否则 TXE 被清零，并且 TXFIFO 被认为是有数据。用这种方式，RXFIFO 可以存储多达 4 个数据帧，而 TXFIFO 在字长不大于 8 的时候也只可以存储多达三个数据帧。这种差异可以防止在已经有 3 个 8 位数据存在 TXFIFO 中的时候，软件试图在 16 位模式下向 TXFIFO 写入更多的数据而造成数据破坏。TXE 和 RXNE 事件都可以通过中断查询或处理。

如果接收到下一个数据时 RXFIFO 的是满的，将导致发生溢出事件（见章节《关于 OVR 标志的描述：状态标志》）。溢出事件可以查询处理或中断处理。

正在执行数据传输的时候，硬件会将 BSY 位置 1 来指示这个状态。当时钟信号连续运行时，如果是主机模式，BSY 标志在帧与帧之间一直保持为 1，而在从机模式时，BSY 信号在帧与帧之间会有一小段时间（1 个 SPI 时钟周期）为 0。

17.3.6.2 序列处理

多个数据字节可以顺序发送来组成一个消息。启用发送后，在主机 TXFIFO 中的数据会开始发送并连续发完。主机会连续输出时钟信号，直至 TXFIFO 变为空，然后停下来等待新增的数据。

在单接收模式，半双工（(BIDIMODE = 1，BIDIOE = 0)， 或简单发送（BIDIMODE = 0，RXONLY = 1）模式下，只要 SPI 和单接收模式被使能，主机会立即开始接收序列。主机连续提供时钟信号，直到主机关闭 SPI 或者关闭单接收模式之前都不会停下来。这时主机会连续接收数据。

数据帧开始后，从机无法控制或延迟数据序列。出于这个原因，从机必须在开始传输之前准备好数据，并总是一直保持有数据待传（存在 TXFIFO 中）。主机必须在每个序列之间给从机保留足够的时间以准备数据。如果可能的话，序列中的字节的数量应得到限制，以便从机完成自动的数据处理（通过使用 FIFO）。主机必须提供额外的时间给从机处理数据内容。

在多从机并行的系统中，每个序列应该由 NSS 脉冲来分隔，以将每个序列对应到不同的从机。在单从机系统中通过 NSS 控制从机就显得没那么有必要了，但这里有个脉冲还是更好，这可以令从机和每个数据序列完成同步。NSS 既可以由软件管理也可以由硬件管理（见第 25.3.4: 从机选择（NSS）的引脚管理）。

BSY 位被置 1 时，它标志着一个持续的传输。这一点，结合 FTLVL [1:0 位，可用于检查传输是否完成。在系统进入 HALT 模式前这是很有必要的，过早的进入 Halt 模式可能导致数据破坏。判断 BSY 位的另外一个目的是可以用关键来管理 NSS 信号。当 RXNE 标志被置 1，它意味着对当前的传输结束了，即最后一位刚刚采样完毕，以及完整的数据帧存储在 RXFIFO 中。

当停止提供传输数据后，主机会完成当前数据传输。在这种情况下，最后一个数据传输结束后时钟输出就会停止。在数据包模式中，奇数个数据传输完毕后要特别注意防止出现空的字节（见第 25.4.2：TI 模式）。当主机处于单接收模式下，只有禁用 SPI 或单接收模式，才能停止时钟。为了收

到正确数量的字节并且阻止任何无效的空数据，就得在最后一个字节正在传输的时候，把握正确的关闭接收的时机。关闭动作必须发生在首位的采样时间和下一个字节(不要的空数据)的首位之间。

17.3.6.3 禁用 SPI 的步骤

在一帧数据正在传输的时候，或者 TXFIFO 当中有数据的时候，主机不可以通过操作 SPE 位来关闭 SPI。如果发生这种情况，时钟信号将持续发送，直到外设被重新启用使得传输可全面完成。如果收到了数据，但是仍然遗留在 RXFIFO 中未及读取时关闭了 SPI 的话，那么在下一次打开 SPI 开始新的传输序列之前一定要先处理历史数据。为了防止这种情况，请确保 RXFIFO 是空的时候再去禁用 SPI。这个过程可以通过正确的流程来实现，或者通过专门的外设复位控制寄存器来执行软件复位命令，从而全面初始化 SPI 的寄存器（见 RCC_ APBiRSTR 中的 SPIiRST 位）。

正确的关闭流程是（除了使用单接收模式时）:

1. 等到 FTLVL [1:0] = 00（没有更多的数据传输）。
2. 等待，直到 BSY = 0（最后一个数据帧处理完毕）。
3. 读取数据直到 FRLVL [1:0] = 00（读取所有接收到的数据）。
4. 禁用 SPI（SPE = 0）。

某些单接收模式的正确关闭步骤是:

1. 在最后一个数据的传输期间的特定时间窗口中关闭单接受模式（RXONLY = 0 或 BIDIOE= 1）。
2. 等待直到 BSY = 0（最后一个数据帧处理完毕）。
3. 读取数据，直到 FRLVL [1:0] = 00（读取完所有接收到的数据）。
4. 禁用 SPI（SPE= 0）。

注: 如果用了打包模式，并且收到一个格式为小于或等于 8 位的奇数个数据的数据帧，当 FRLVL [1:0]= 1 时，FRXTH 必然被置 1，为的是产生 RXNE 事件以读取最后的奇数数据。

17.3.6.4 数据打包

当数据帧的大小适合一个字节（小于或等于 8 位），并且对 SPIx_DR 寄存器执行任何 16 位的读写访问时，数据会自动打包在一起。在这种情况下，可以并行处理双数据。SPI 先操作低 8 位，然后

操作高 8 位。图 17-9 提供了打包方式顺序处理数据的一个例子。在一次对 SPIx_DR 的 16 位写访问后，就会有 2 个字节的数据被发送出去。如果 RXFIFO 的阈值设置为 16 位（FRXTH = 0），该序列则只会生成一个接收 RXNE 事件，而不是两个。针对这种单个的 RXNE 事件的响应，接受器必须对 SPIx_DR 寄存器作一次 16 位的读访问才能够把数据全都取到。RXFIFO 的阀值和跟进的数据访问的位宽必须保持一致，否则就会丢数据。

如果出现奇数个字节数据，那就会出现特别的问题，这是一定要解决的。在发送端，用 8 位方式访问 SPIx_DR 将最后一个字节发出来就够了。在接收端必须改变 Rx_FIFO 门限，以便在传送技术字节的数据帧的最后字节时能够产生 RXNE 事件。

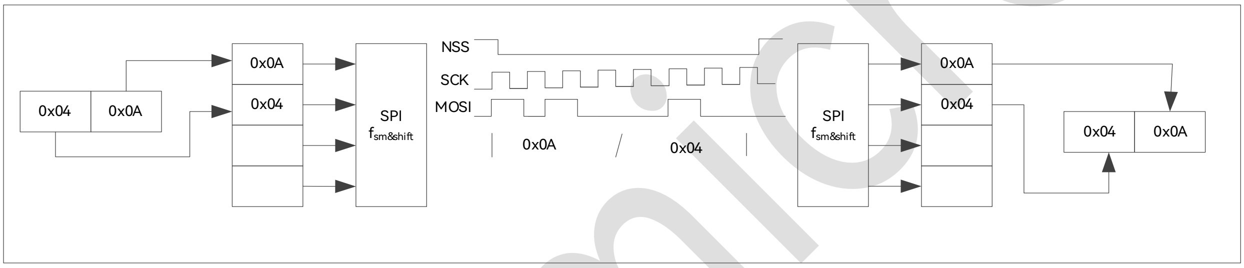


图 17-9：传输和接收 FIFO 中的数据包

17.3.7 状态标志

应用程序可通过 3 个状态标志来了解 SPI 总线的全部状态。

17.3.7.1 TX 缓冲器空标志（TXE）

当 TXFIFO 中有了足够的空间来存储发送数据时，TXE 标志被置 1。TXE 标志是连到 TXFIFO Level 的。在 TXFIFO 的存储水平低于或等于整个 FIFO 深度的一半的时候，这个标志会置高并且保持高。如果 SPIx_CR2 中的 TXEIE 位是 1，还会产生一个中断。如果 TXFIFO 的存储水平又高于 FIFO 深度的一半了，那这个位会自动的清零。

17.3.7.2 RX 缓冲非空（RXNE）

RXNE 标志的设置取决于对在 SPIx_CR2 寄存器 FRXTH 位值：

- 如果 FRXTH 为 1，RXNE 会在 RXFIFO 的存储水平大于或等于 1/4（8 位）的时候被置高，并且保持住。
- 如果 FRXTH 为 0，RXNE 会在 RXFIFO 的存储水平大于或等于 1/2（16 位）的时候被置高，并且保持住。

如果这时 SPIx_CR2 寄存器中的 RXNEIE 位是 1，那就会产生一个中断请求。当上述条件不再成立时，RXNE 由硬件自动清零。

17.3.7.3 忙标志（BSY）

BSY 标志由硬件设置和清零（软件改写这个标志是没有用的）。

当 BSY 为 1 时，它表示 SPI 正处于数据传输过程中（SPI 总线忙）。

在某些模式下可以使用的 BSY 标志来检测传输结束，从而软件可以在进入 HALT 模式之前禁用 SPI 及其外设时钟。这就避免了破坏最后的传输字节。

BSY 标志在防止多主机系统中的写碰撞也是很有用的。BSY 标志在下列条件之一达成时被清除：

- 当 SPI 被正确禁用时
- 当在主模式（MODF 位设置为 1）中检测到故障时
- 在主模式下，当完成了数据发送，并且没有新的数据要发送时
- 在从模式下，当两次数据传输之间间隔达到至少一个 SPI 时钟周期时

注：当下次传输可立即由主机来处理时（例如：如果主机是单接收模式或它的发送 FIFO 不为空），在主机这边的发送数据之间通讯不会中断，并且 BSY 标志仍然被保留为 1。尽管从机不会有这个情况，ST 总是建议使用 TXE 和 RXNE 标志（而不是标志的 BSY）来处理数据发送或接收操作。

17.3.8 错误标志

在将错误中断使能位 ERRIE 置 1 后，如果下列错误标志被置 1，就会产生 SPI 中断。

17.3.8.1 溢出标志（OVR）

当主机或从机在收到数据之后不去清除 RXNE 位，然后收到了新的数据之后，会引发溢出错误。当接收数据时 RXFIFO 中又没有足够的空间存储收到的数据时，溢出的情况也会发生。如果软件没有足够时间去读取 RXFIFO 中的前面收到数据，来为后面的数据腾出足够的空间的时候就会发生这种事。

溢出的情况发生时，新收到的值不会覆盖在 RXFIFO 的前一个数据。 新收到的值将被丢弃，随后传输的所有数据都将丢失。 在读了 SPIx_SR 寄存器后，再去读一下 SPIx_DR 寄存器， 就会清除 OVR 位。

17.3.8.2 模式故障位（MODF）

主机得知其内部 NSS 信号（NSS 引脚为硬件模式，或在 NSS 软件模式 SSI 位）被拉低时，发生模式故障位。这将自动设置 MODF 位。主机模式故障对 SPI 接口的影响包括以下方面:

- MODF 位被置 1，另外如果 ERRIE 位被置 1，会产生 SPI 中断。
- SPE 位被清零。这将阻止主机的数据输出，同时禁用 SPI 接口。
- MSTR 位被清除，从而迫使设备转到从机模式。

使用下面的软件序列，以清除 MODF 位:

1. 在 MODF 位被设置后，对 SPIx_SR 寄存器做一次读或写访问。
2. 然后写一下 SPIx_CR1 寄存器。

为了避免系统中的多个从机发生冲突，NSS 引脚必须在 MODF 位清零过程中拉高。SPE 和 MSTR 位可以在此清零过程后恢复到原来的状态。作为一种安全措施，硬件不允许在 MODF 位为 1 期间将 SPE 的 MSTR 位置 1。在从模式下，MODF 位永远不可能被置 1，除非是以前的多主机冲突的结果。

17.4 SPI 特殊功能

17.4.1 NSS 脉冲模式

在 SPIx_CR1 寄存器的 NSSP 位被置 1 时打开这个模式，只有在 SPI 接口被配置为摩托罗拉主模式，而且捕获第一个沿时（SPIx_CR1 CPHA＝0）才会生效。这时发送时，NSS 脉冲会在两个连续的数据帧之间产生，并且 NSS 至少会在高电平保持一个时钟周期的时间。这个模式允许从机锁存数据。NSSP 脉冲模式为单一的主从应用而设计。

下图说明 NSSP 脉冲模式时启用时的 NSS 引脚管理。

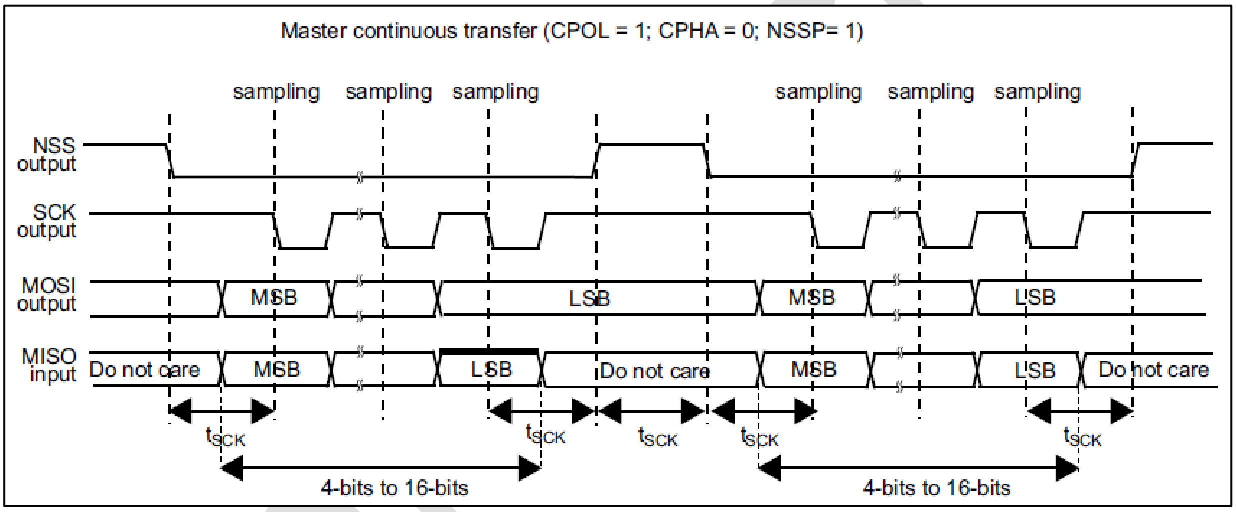


图 17-10：摩托罗拉 SPI 主模式下 NSSP 脉冲的产生

注：当 CPOL＝0 时动作会类似。这时，采样边沿是在 SCK 的上升沿，对 NSS 的采样也发生在这些沿。

17.5 SPI 中断

在 SPI 通信期间，中断可以由以下事件来产生：

- 发送寄存器待装填
- 在接收寄存器中有收到的数据
- 主模式故障
- 溢出错误

中断可以分别被启用和禁用。

表 17-1: SPI 中断请求

<table>
<tr><th>中断事件</th><th>事件标志</th><th>使能控制位</th></tr>
<tr><td>TXFIFO 待装填</td><td>TXE</td><td>TXEIE</td></tr>
<tr><td>RXFIFO 中有收到的数据</td><td>RXNE</td><td>RXNEIE</td></tr>
<tr><td>主模式故障</td><td>MODF</td><td rowspan="2">ERRIE</td></tr>
<tr><td>溢出错误</td><td>OVR</td></tr>
</table>

17.6 相关寄存器

17.6.1 控制寄存器（SPI_CR1）

偏移地址：0x000

复位值：0x0000 0008

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Res.</td></tr>
<tr><td colspan="16"></td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13 12 11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5 4 3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>BIDIMODE</td><td>BIDIOE</td><td>Res.</td><td>RXONLY</td><td>SSM</td><td>SSI</td><td>LSBFIRST</td><td>SPE</td><td>BR[2:0]</td><td>MSTR</td><td>CPOL</td><td>CPHA</td></tr>
<tr><td>rw</td><td>rw</td><td></td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td>rw</td></tr>
</table>

<table>
<tr><th>位</th><th>名称</th><th>描述</th></tr>
<tr><td>31:16</td><td>RSV</td><td>保留位，固定读为 0</td></tr>
<tr><td>15</td><td>BIDIMODE</td><td>双向数据模式使能：
这个位允许使用普通的单线双向数据线进行半双工通信。当双向模式激活时，保持 RXONLY 位清零。
0：选择 2 线单向数据模式
1：选择单线双向数据模式</td></tr>
<tr><td>14</td><td>BIDIOE</td><td>双向模式输出使能：
和 BIDIMODE 位一起决定在“单线双向”模式下数据的输出方向
0：输出禁止（单收模式）
1：输出使能（单发模式）
注：在主模式下，用 MOSI 引脚。在从模式下，用 MISO 引脚。</td></tr>
<tr><td>13:11</td><td>RSV</td><td>保留位，固定读为 0</td></tr>
<tr><td>10</td><td>RXONLY</td><td>单收模式使能
此位启动单工通讯，使用单线单工单收通讯。</td></tr>
</table>

位	名称	描述
		激活单收模式时，保持 BIDIMODE 位清零。此位也可用于多从设备系统中，此特定的从设备不会被访问，被访问的从设备输出不会被破坏。 0：全双工（发送和接收） 1：输出禁止（单收模式）
9	SSM	软件从机管理： 当 SSM 被置位时，NSS 引脚上的电平由 SSI 位的值决定。 0：禁用软件从设备管理 1：启用软件从设备管理
8	SSI	内部从设备选择： 此位只有当 SSM 位为 1 时才有效。此位的值被强制映射 NSS 引脚上，NSS 引脚上的 I/O 操作无效。
7	LSBFIRST	帧格式： 0：数据发送/接收 MSB 在前 1：数据发送/接收 LSB 在前 注：当通信在进行时不能改变该位的值。
6	SPE	SPI 使能： 0: 禁用 SPI 设备 1: 启用 SPI 设备 注：当关闭 SPI 设备时，请按照<禁用 SPI 步骤>操作。
5:3	BR[2:0]	波特率控制： 000：保留 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：当通信在进行时不能改变该位的值。
2	MSTR	主设备选择： 0: 配置为从设备 1: 配置为主设备 注：当通信在进行时不能改变该位的值。
1	CPOL	时钟极性： 0: 空闲状态时，SCK 保持低电平 1: 空闲状态时，SCK 保持高电平 注 1：当通信在进行时不能改变该位的值。

位	名称	描述
0	CPHA	时钟相位: 0: 第一个时钟沿对准第一位数据 1: 第二和时钟沿对准第一位数据 SPI 从机模式下不支持 CPOL=CPHA=1 的配置 注: 当通信在进行时不能改变该位的值。

17.6.2 控制寄存器 2（SPI_CR2）

偏移地址: 0x004

复位值: 0x0000 0700

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			FRXTH	DS[3:0]				TXEIE	RXNEIE	ERRIE	Res.	NSSP	SSOE	Res.	
				rw				rw	rw	rw		rw	rw		

位	名称	描述
31:13	RSV	保留位，固定读为 0
12	FRXTH	FIFO 接收门限: 此位用来设置触发 RXNE 事件时的 RXFIFO 的阈值。 0: 如果 FIFO 的存储水平大于或等于 1/2（16 位），产生 RXNE 事件。 1: 如果 FIFO 的存储水平大于或等于 1/4（8 位），产生 RXNE 事件。
11:8	DS[3:0]	数据位宽: 这些位配置 SPI 传输数据的位宽: 0000: 不使用 0001: 不使用 0010: 不使用 0011: 4 位 0100: 5 位 0101: 6 位 0110: 7 位 0111: 8 位 1000: 9 位 1001: 10 位

位	名称	描述
		1010：11 位 1011：12 位 1100：13 位 1101：14 位 1110：15 位 1111：16 位 注：如果软件试图写一个“不使用”的值，它们会被迫赋值“ 0111”（8 位）。
7	TXEIE	TX 缓冲器空中断使能: 0：TXE 中断屏蔽 1：TXE 中断没有被屏蔽。用于在 TXE 标志置 1 的时候产生一个中断请求。
6	RXNEIE	RX 缓冲区非空中断使能: 0：RXNE 中断屏蔽 1：RXNE 中断没有被屏蔽。用于在 RXNE 标志置 1 的时候产生一个中断请求。
5	ERRIE	错误中断使能: 这个位控制在出现错误事件时是否产生中断（SPI 模式中的 OVR、MODF）。 0: 错误中断屏蔽 1: 错误中断使能
4	RSV	保留位，固定读为 0
3	NSSP	NSS 脉冲管理: 该位仅在主模式下使用。它允许 SPI 在连续传输时，两个数据传输之间产生一个 NSS 脉冲。在单个数据传输的情况下，它会在传输结束后将 NSS 脚强制为高电平。在 CPHA=1 的时候，这个位没有意义。 0: 没有 NSS 脉冲 1: 产生 NSS 脉冲 注：该位必须在 SPI 被禁止（SPE = 0）的时候写入。
2	SSOE	SS 输出使能: 0：在主模式下 SS 输出被禁用，SPI 接口可以工作在多主机的配置下。 1：SPI 接口启用的同时在主模式下启用 SS 输出。SPI 接口不能在多主机环境下工作。
1:0	RSV	保留位，固定读为 0

17.6.3 状态寄存器（SPI_SR）

偏移地址：0x008

复位值：0x0000 0002

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

15:13	12:11	10:9	8	7	6	5	4:2	1	0
Res.	FTLVL[1:0]	FRLVL[1:0]	Res.	BSY	OVR	MODF	Res.	TXE	RXNE
	r	r		r	r	r		r	r

位	名称	描述
31:13	RSV	保留位，固定读为 0
12:11	FTLVL[1:0]	FIFO 发送水平： 由硬件置位或清零 00：FIFO 空 01：1/4 FIFO 10：1/2 FIFO 11：FIFO 满（当 FIFO 门限大于 1/2 时认为是满）
10:9	FRLVL[1:0]	FIFO 接收水平： 由硬件置位或清零 00：FIFO 空 01：1/4 FIFO 10：1/2 FIFO 11：FIFO 满
8	RSV	保留位，固定读为 0
7	BSY	忙标志，由硬件置位和清除。 0：SPI 不忙 1：SPI 通信忙或 TX 缓冲区不为空 注：必须谨慎使用的 BSY 标志（参见章节<17.3.7 SPI 状态标志>和<17.3.6.3 禁用 SPI 的步骤>）。
6	OVR	溢出标志： 0: 没有发生溢出 1: 发生溢出 此标志由硬件置位，由软件序列复位。
5	MODF	模式故障：

位	名称	描述
		0：无模式故障发生 1：模式故障发生 此标志由硬件置位，由软件序列复位。软件序列请参阅章节<17.3.8.2 模式故障（MODF）>。
4:2	RSV	保留位，固定读为 0
1	TXE	发送缓冲区空标志： 0：TX 缓冲区非空 1：TX 缓冲区空
0	RXNE	接收缓冲区非空标志： 0：RX 缓冲区空 1：RX 缓冲区非空

17.6.4 数据寄存器（SPI_DR）

偏移地址：0x00C

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:0	DR[15:0]	数据寄存器： 待发送或者已经收到的数据 数据寄存器作为 RX 和 TX FIFOs 的接口。当读取数据寄存器时，RXFIFO 会被访问，而写入数据寄存器则会访问 TXFIFO（见章节：数据发送和接收流程）。 注：数据始终是右对齐。当写寄存器的时候，没有用到的位会被忽略；读的时候，没有用到的位会是 0。接收门限的设置必须始终符合目前使用的读访问方式。

18 内部集成电路接口（I2C）

18.1 I2C 概述

本芯片内置 1 组 I2C 功能。

芯片内置 1 个 I2C 总线通讯模块，支持 7 位地址编码主从模式的 I2C 总线通讯。I2C 总线通讯接口为时钟线 SCL 和数据线 SDA 的双向两线接口，I2C 使能后复用的 I/O 端口用作 SCL/SDA，此时为输入/开漏输出口，输出时其内部上拉电阻控制位依然有效，可选择内部或外接合适的上拉电阻，以匹配选定的通讯速率（最高支持 400kbps）。

注：I2C 模块的时钟源为内部高频时钟 HSI，仅当 HSI 工作时，I2C 才可正常工作。

18.2 I2C 数据传输

总线空闲时，数据线 SDA 和时钟线 SCL 均为高电平。SDA 电平在 SCL 高电平期间由高变低的下降沿表示起始信号 START，而 SDA 电平在 SCL 高电平期间由低变高的上升沿则表示停止信号 STOP。START/STOP 信号以及 SCL 上的时钟信号均由主机发送，而数据线 SDA 上的数据则由主从双方同步于 SCL 时钟进行单向传输。数据传输时，SDA 电平在 SCL 高电平期间必须保持稳定，只有在 SCL 为低电平时，SDA 电平才允许变化。

一帧数据传输以一个起始信号 START 开始，以一个停止信号 STOP 或重复起始信号 RE-START 结束，一个重复起始信号 RE-START 也是下一帧数据传输的开始（需从机支持重复起始信号 RE-START），期间总线不被释放。

每一帧数据传输时需先由主机发送一个以 7 位从机地址和 1 位读/写命令组成的控制字节，再由主机或从机发送一个或多个数据字节。一个完整字节的传输需 9 个时钟，前 8 个时钟传输 8 位字节内容（最高位最先传输），第 9 个时钟则为应答时钟，此时 SDA 上的电平即为接收方返回的应答信号，低电平表示应答（ACK），高电平表示非应答（NACK）。

18.2.1 主机到从机的数据传输

主机先发送起始信号 START，再发送一个包含“写”命令的控制字节，从机返回一个 ACK，然后主机开始发送数据字节，从机接收完每一个数据字节后均返回一个 ACK，主机在接收到最后一个字节从机返回的 ACK 后，发送停止信号 STOP 结束本次数据传输。

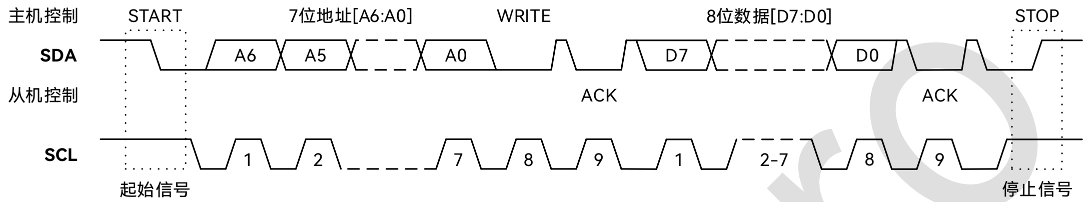


图 18-1：主机发送数据的时序

18.2.2 从机到主机的数据传输

主机先发送起始信号 START，再发送一个包含“读”命令的控制字节，从机返回一个 ACK，然后主机开始接收从机发送的数据字节，并在接收完每一个数据字节后均返回一个 ACK，当主机不再接收数据时，则在接收完从机上一个字节后返回一个 NACK，接着发送停止信号 STOP 结束本次数据传输。

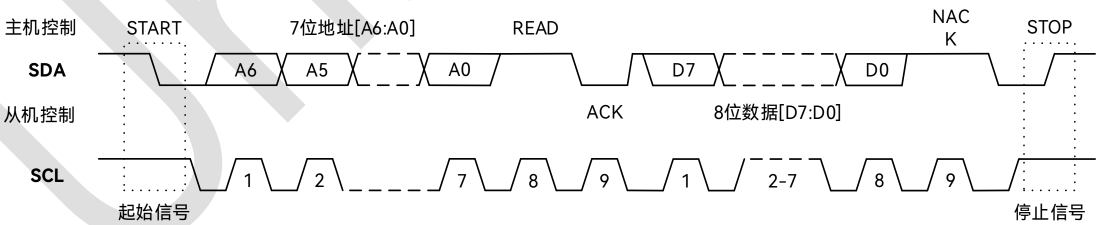


图 18-2：主机接收数据的时序

18.3 I2C 工作模式

I2C 总线接口模块支持 7 位地址编码的主从模式，可通过寄存器位 MOD 选择主机工作模式或从机工作模式。主机模式下，总线时钟从 SCL 端口输出，通讯速率可通过寄存器位 SPD 选择 100kbps

或 400kbps；从机模式下，总线时钟由 SCL 端口输入，通讯速率在同步采样方式（SYN=0）时需通过 SPD 选择 100kbps 或 400kbps，而在异步采样方式（SYN=1）时最高可支持 400kbps。

18.3.1 主机模式

主机模式下，I2C 模块可单独发送地址帧（包含起始信号）、数据帧、或停止信号。

主机发送地址帧时，会将地址寄存器 AR 中内容作为地址（高 7 位）和命令（最低位）发送。

主机发送数据帧时，会将发送缓冲器（TD_buf）中内容作为数据发送，而在接收数据帧时，则将接收的数据存入接收缓冲器（RD_buf）中，TD_buf 和 RD_buf 均需通过数据寄存器 DR 进行读/写操作，写 DR 为写 TD_buf，读 DR 则为读 RD_buf。

主机发送完地址或数据帧（第 9 位接收完从机应答信号）、主机接收完数据帧（第 9 位发送完应答信号）、或主机发送完停止信号后，中断标志 IF 将被置 1 触发 I2C 中断。

18.3.2 从机模式

从机模式下，I2C 模块在总线上检测到 I2C 起始信号后开始接收主机发送的地址及命令（缓存在接收缓冲器中），若地址与 AR 中预设的从机地址（若使能广播地址则还包括地址 00H）匹配，则在第 9 位应答时钟期间将寄存器位 TACKS 内的应答内容发送给主机。若发送的应答内容为 NACK，则发送完成后重置 I2C 模块以等待总线上的下一次起始信号；若发送的应答内容为 ACK，则发送完成后地址匹配标志位 SADMF 将被置 1，同时 I2C 模块会将 SCL 电平拉为低电平（下一次检测到 SCL 电平上升沿时 SADMF 将被清 0），中断标志 IF 将被置 1 触发 I2C 中断。

清 0 中断标志 IF 可将 SCL 线释放，此前可通过读写命令标志位 SCMDF 准备接收或发送数据。从机接收或发送数据帧时，也会将接收的数据存入接收缓冲器（RD_buf）中或将发送缓冲器（TD_buf）中内容发送。接收/发送完数据并发送/接收完应答信号后，数据帧收发完成标志位 SDTFF 将被置 1，同时 I2C 模块会再次将 SCL 电平拉为低电平（下一次检测到 SCL 电平上升沿时 SDTFF 将被清 0），中断标志 IF 将被置 1 触发 I2C 中断。清 0 中断标志 IF 释放 SCL 线，可使主机继续传输数据帧或发送停止信号。

检测到总线上的停止信号后，从机模式通讯结束。

注：从机模式地址匹配时，接收到的地址信息缓存于 RD_buf 中，因此若使能广播地址，则可通过 DR 读取 RD_buf 中的地址信息，判断当前所响应的地址是广播地址 00H 还是预设于 AR 中的从机地址。

从机模式还可选择同步或异步采样方式，同步方式可增加采样准确率，异步方式则支持休眠模式下检测总线信号，当有地址匹配或接收完成时可以唤醒 CPU。同步/异步方式的切换需在 SCL 被拉住后进行，且切换后需等待 t_{HIGH} 时间后才可释放 SCL 线。

应答寄存器位：

无论主机模式还是从机模式，发送应答信号时将发送寄存器位 TACKS 中内容、接收到的应答内容保存在寄存器位 RACKF 中。

18.4 I2C 时序说明

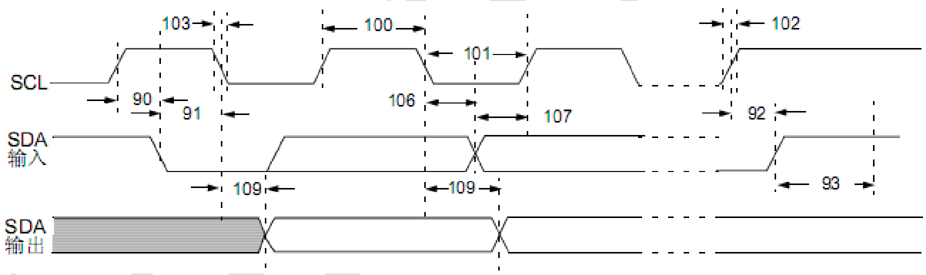


图 18-3：I2C 时序

表 18-1：主机模式信号输出时间定义

标号	参数	符号	典型值（Typ.）		单位
			100kbps	400kbps	
90	重复起始条件的建立时间	$t_{SU;STA}$	6	1.5	μs
91	起始条件的保持时间	$t_{HD;STA}$	4	1	μs
101	SCL 时钟的低电平周期	t_{LOW}	6	1.5	μs
100	SCL 时钟的高电平周期	t_{HIGH}	4	1	μs
102	SDA 和 SCL 上升沿时间	t_R	1000	300	ns
103	SDA 和 SCL 下降沿时间	t_F	300	300	ns
109	时钟输出有效时间	t_{AA}	5	1	μs
92	停止条件的建立时间	$t_{SU;STO}$	4	1	μs
93	停止条件的保持时间	$t_{HD;STO}$	6	1.5	μs

表 18-2：主机模式信号输入时间要求

标号	参数	符号	典型值（Typ.）		单位
			100kbps	400kbps	
106	数据输入保持时间	$t_{HD;DAT}$	1	0.5	μs
107	数据输入建立时间	$t_{SU;DAT}$	300	150	ns

表 18-3：从机模式信号输入时间要求

标号	参数	符号	典型值（Typ.）		单位
			100kbps	400kbps	
90	重复起始条件的建立时间	$t_{SU;STA}$	4.7	0.6	μs
91	起始条件的保持时间	$t_{HD;STA}$	4	0.6	μs
101	SCL 时钟的低电平周期	t_{LOW}	4.7	1.3	μs
100	SCL 时钟的高电平周期	t_{HIGH}	4	0.6	μs
107	数据输入建立时间	$t_{SU;DAT}$	300	150	ns
92	停止条件的建立时间	$t_{SU;STO}$	4	0.6	μs
93	停止条件的保持时间	$t_{HD;STO}$	4	0.6	μs

18.5 I2C 多机通讯

I2C 模块支持多主机通讯，可实现时钟同步和总线仲裁功能。

主机模式下，I2C 模块发送时钟的高电平后会检测 SCL 线，直至 SCL 为高后开始高电平宽度计时，在高电平保持时间达到后才发送时钟的低电平，若在高电平计时期间 SCL 线被其他设备拉低，I2C 模块则将直接进入低电平宽度计时，如此 SCL 线上时钟的低电平时间由所有主机中最长的低电平时长决定，而高电平时间由最短的高电平时长决定，从而实现多主机在 SCL 线上的时钟同步。

主机或从机模式下，I2C 模块发送数据的高电平后会进行总线仲裁，即检测 SDA 线上的电平状态，若仍为低电平，则表示高电平未发送成功，已有其他设备占用总线，此时 I2C 模块将释放总线，总线仲裁状态标志 ARLO 将被置 1。

主机模式下发送起始信号或重复起始信号时，I2C 模块释放 SCL 总线时检测到 SDA 为低电平和 I2C 模块拉低 SDA 时检测到 SCL 为低电平，总线仲裁状态标志 ARLO 将被置 1。

主机模式下发送停止信号时，I2C 模块释放 SDA 时检测到 SCL 为低电平和停止信号发送完毕后检测到 SDA 为低电平，总线仲裁状态标志 ARLO 将被置 1。

主机模式下在发送地址帧/数据帧、发送应答时，从机模式下在发送数据帧或发送应答时，向 SDA 发送高电平，但在 SCL 的上升沿却从 SDA 上检测到低电平，总线仲裁状态标志 ARLO 将被置 1。

只有使能位 EN 清 0 才能将状态标志 ARLO 清 0。

18.6 I2C 中断与低功耗唤醒

I2C 模块有多个标志位，包括总线仲裁状态标志 ARLO、START 信号检测状态标志 STRF、STOP 信号检测标志 STPF、从模式地址匹配标志 SADMF、从模式数据帧接收完成标志 SDTFF，这些标志中的任何一个置 1 都会导致 I2C 模块的中断标志 IF 置 1，从而触发中断。I2C 中的标志位为 1 时，软件可以通过对该标志位写 1 清 0。如果 IF 标志位为 1，软件对 IF 写 1 只能将 IF 清零，不会对其他标志位产生影响。

如果 I2C 工作在从机模式，并且使用异步采样时，可以通过地址匹配或数据接收完成标志将芯片从 STOP 模式中唤醒。芯片处于 STOP 模式时，I2C 检测到总线信号，硬件接收到一帧数据，如果是判断地址匹配则置起 SADMF，如果是数据则置起 SDTFF，SADMF 和 SDTFF 都可以产生一个内部唤醒事件。

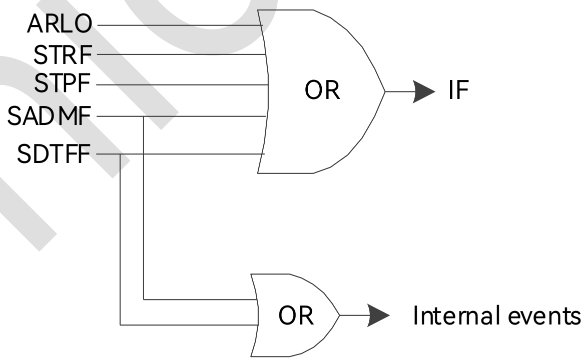


图 18-4：唤醒逻辑图

18.7 相关寄存器

18.7.1 控制寄存器 1（I2C_CR1）

偏移地址：0x00

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								EN	MD	SPD	SYN	GCE	Res.		RUS
								rw	rw	rw	rw	rw			rw

位	名称	描述
31:8	RSV	保留位，固定读为 0
7	EN	I2C 使能位： 0：关闭 I2C 1：使能 I2C
6	MD	I2C 工作模式选择位： 0：主机模式 1：从机模式
5	SPD	通讯速率选择位： 0：通讯速率为 100kbps 1：通讯速率为 400kbps
4	SYN	从机模式采样方式选择位： 0：同步采样 1：异步采样
3	GCE	从机模式广播呼叫使能位： 0：从机模式下不支持广播呼叫功能，即不响应广播地址 00H 1：从机模式下支持广播呼叫功能，即响应广播地址 00H
2:1	RSV	保留位，固定读为 0
0	RUS	内部强上拉（3K）电阻使能位 0：禁止内部强上拉电阻 1：使能内部强上拉电阻

18.7.2 控制寄存器 2（I2C_CR2）

偏移地址：0x04

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								TACKS	IE	Res.	MNASTPE	MTSAA	MTSTP	MTDTA	MRDTA
								rw	rw		rw	rw	rw	rw	rw

位	名称	描述
31:8	RSV	保留位，固定读为 0
7	TACKS	应答信号发送内容选择位 0：待发送的应答信号为应答（ACK） 1：待发送的应答信号为非应答（NACK）
6	IE	中断使能位 0：禁止中断 1：使能中断
5	RSV	保留位，固定读为 0
4	MNASTPE	主机模式发送完 NACK 或接收到 NACK 后自动发送 STOP 信号使能位： 0：发送完 NACK 或接收到 NACK 后，不自动发送 STOP 信号 1：发送完 NACK 或接收到 NACK 后，自动发送 STOP 信号 注：此位必须 EN 和 MD 位为 1 时才能使能。
3	MTSAA	主机模式发送 START 信号和地址帧再接收应答控制位： 0：操作未开始或已完成，写 1 开始发送起始信号和地址帧再接收从机应答信号 1：发送起始信号和地址帧再接收从机应答信号操作中，完成后自动清 0
2	MTSTP	主机模式发送 STOP 停止信号控制位： 0：操作未开始或已完成，写 1 开始发送停止信号 1：发送停止信号操作中，完成后自动清 0
1	MTDTA	主机模式发送数据帧再接收应答控制位： 0：操作未开始或已完成，写 1 开始发送数据帧再接收从机应答信号 1：发送数据帧再接收从机应答信号操作中，完成后自动清 0
0	MRDTA	主机模式接收数据帧再发送应答控制位： 0：操作未开始或已完成，写 1 开始接收从机数据帧再发送应答信号 1：接收从机数据帧再发送应答信号操作中，完成后自动清 0

18.7.3 状态寄存器（I2C_SR）

偏移地址：0x08

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								IF	ARLO	STRF	STPF	SADMF	SDTFF	SCMDF	RACKF
								rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	r

位	名称	描述
31:8	RSV	保留位，固定读为 0
7	IF	中断标志位： 0：未触发 IIC 中断 1：已触发 IIC 中断，需软件清 0 注：该标志位为总标志位，“ISR”中“ARLO、STRF、STPF、SADMF、SDTFF”任意一个标志触发都能置起该总标志。
6	ARLO	总线仲裁状态标志位： 0：总线上未检测到仲裁失败 1：总线上检测到仲裁失败（仅在 I2C 关闭后清 0）
5	STRF	总线 START 信号检测状态标志位： 0：总线上未检测到起始信号 1：总线上检测到起始信号（检测到停止信号后清 0）
4	STPF	总线 STOP 信号检测状态标志位： 0：总线上未检测到停止信号 1：总线上检测到停止信号（检测到起始信号后清 0）
3	SADMF	从机模式地址匹配状态标志位： 0：从机模式接收到的地址帧中地址不匹配，或接收到数据帧 1：从机模式接收到地址帧，且地址与本机预设地址匹配
2	SDTFF	从机模式数据帧收发完成状态标志位： 0：从机模式数据帧接收/发送中，或未接收到数据帧 1：从机模式数据帧接收/发送完成
1	SCMDF	从机模式读写命令接收内容标志位： 0：从机模式接收到的命令为“读”命令（逻辑“0”） 1：从机模式接收到的命令为“写”命令（逻辑“1”）

位	名称	描述
0	RACKF	应答信号接收内容标志位： 0：接收到的应答信号为应答（ACK） 1：接收到的应答信号为非应答（NACK）

18.7.4 地址寄存器（I2C_AR）

偏移地址：0x0C

复位值：0x0000 0000

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

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

位	名称	描述
31:8	RSV	保留位，固定读为 0
7:1	ADDR[6:0]	I2C 通讯地址： 主机模式下为发送的地址 从机模式下为预设的从机地址
0	WR	主机模式下发送的读/写命令控制位 0：发送写命令 1：发送读命令

18.7.5 数据寄存器（I2C_DR）

偏移地址：0x10

复位值：0x0000 0000

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

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

位	名称	描述
31:8	RSV	保留位，固定读为 0
7:0	DATA[7:0]	I2C 通讯数据： 写操作为写发送缓冲器数据 读操作为读接收缓冲器数据 注：对 DR 读和写操作的目的寄存器不是同一寄存器。

18.7.6 辅地址寄存器（I2C_AAR）

偏移地址：0x14

复位值：0x0000 0000

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

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

位	名称	描述
31:8	RSV	保留位，固定读为 0
7:1	ADDR[6:0]	I2C 通讯辅地址，主机模式下无效，从机模式下为预设的从机辅地址 注：AAR 为 0 时，不响应 I2C 的 0 地址，0 地址对应广播地址。
0	RSV	保留位，固定读为 0

18.7.7 辅地址掩码寄存器（I2C_AMR）

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.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								AM6	AM5	AM4	AM3	AM2	AM1	AM0	Res.
								rw	rw	rw	rw	rw	rw	rw	

位	名称	描述
31:8	RSV	保留位，固定读为 0
7:1	AMx	从机模式 IIC 通讯从地址掩码控制位（x=0~6）: 0：从机模式下，检验预设地址 AAR 寄存器中 ADDR[6:0]与接收的对应地址位是否匹配 1：从机模式下，不检验接收的对应地址位
0	RSV	保留位，固定读为 0

19 通用异步收发器（UART）

19.1 概述

器件内置有一个通用异步收发器（UART），支持工业标准 NRZ 编码，可实现 MCU 与设备的全双工异步通讯。

UART 支持单线半双工通信模式以及多处理器通信模式。

表 19-1：UART 功能

UART 模式/功能	UART1
多机通讯	√
单线半双工通讯	√
双时钟驱动和从 Stop 模式唤醒	√

注：√ = 支持

19.2 特性

- 全双工异步通信
- NRZ 标准格式（mark/space 校验）
- 可配置的 16 位过采样方法提供速度和时钟容忍度间的灵活选择
- 通用可编程收发波特率可达 4.5Mbit/s（@时钟 72MHz，16 倍过采样）
- 支持双时钟域：
  - 从 STOP 模式下唤醒
  - 方便的波特率编程，独立于 PCLK 时钟
- 数据字节长度可编程（8 或 9 位）
- 可编程数据位序，LSB 在前或 MSB 在前
- 停止位可编程（1 个或 2 个停止位）
- 接收器和发送器独立使能位

- 接收器和发送器独立极性控制
- 可配置 TX/RX 引脚互换
- 通讯控制/错误检测标志
- 奇偶控制
  - 发送奇偶位
  - 接收数据字节奇偶检查
- 14 个中断源和中断标志
- 多机通信
  - 若地址不匹配，进入静默模式
- 静默模式唤醒（检测到线路空闲或检测到地址标记）

19.3 功能描述

任何 UART 双向通讯要求最少有两个引脚：接收数据输入（RX）和发送数据输出（TX）。

- RX：接收数据输入是串行数据的输入口。使用过采样技术来完成数据恢复，以区别输入数据和噪声。
- TX：数据发送输出。当发送器被禁止，输出脚回到其 I/O 口配置状态。当发送器被使能，但不发送数据时，TX 脚为高电平输出。

数据帧包含：

- 在发送和接收之前为空闲状态
- 起始位
- 数据字（8 或 9 位），LSB 在前
- 1/2 个停止位，表明帧的结束
- 一个波特率发生器（UART_BRR）
- 接收和发送数据寄存器（UART_RDR，UART_TDR）

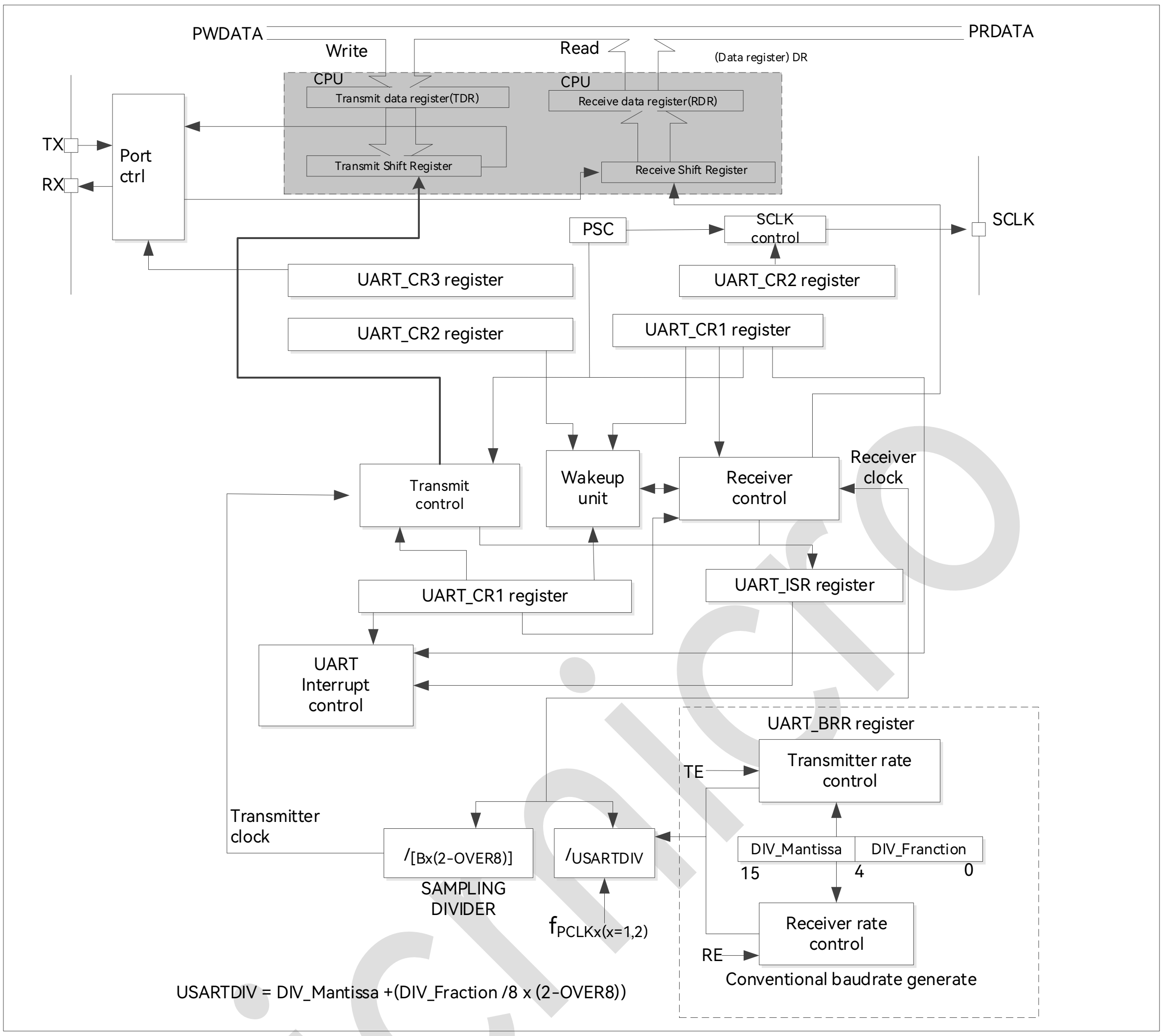


图 19-1：UART 模块框图

注：

1. 有关在 UART_BRR 寄存器中编码 UARTDIV 的详细信息，请参阅 19.3.4 UART 波特率产生。

2. f_{CK} 可以为 f_{HSI}，f_{SYS}。

19.3.1 UART 符号描述

配置 UART_CR1 寄存器（参见下图）中的 M0 位可选择 8 位或 9 位数据长度。

- M0=0：8 位字符长度
- M0=1：9 位字符长度

默认设置中，发送起始位期间，信号（RX 和 TX）都是低电平，发送停止位期间都是高电平。

这个逻辑可以在极性控制中单独设置为反向。

空闲帧，被视为完全由 '1' 组成的完整的数据帧 ('1'的位数也包括了停止位的位数)。

发送和接收由一个共用的波特率发生器驱动，当发送器和接收器的使能位分别置 1 时，分别为其产生时钟。

下面是每个模块的详细说明：

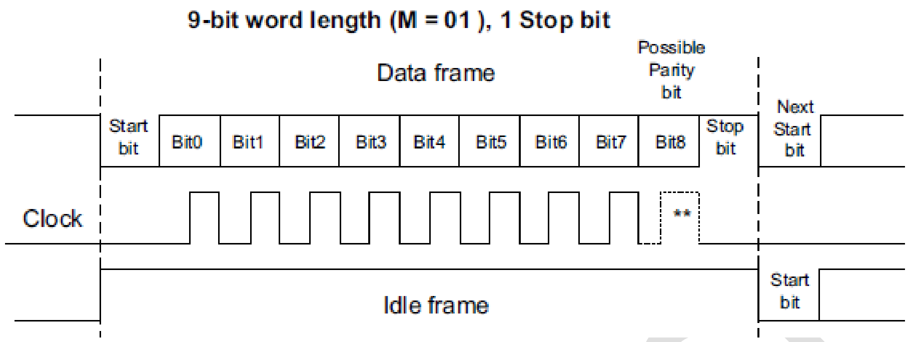


图 19-2：字长编程（M=01）

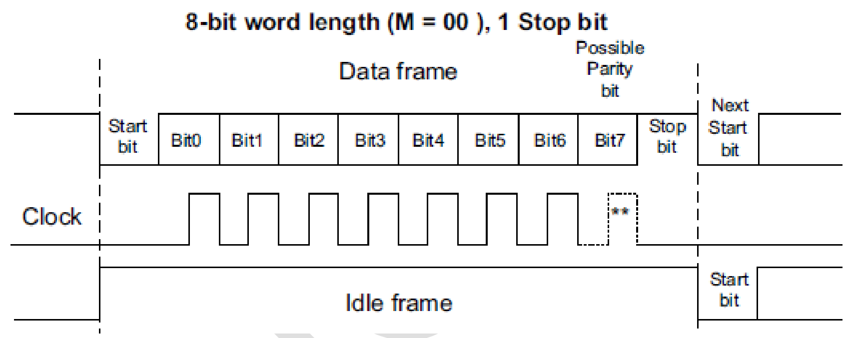


图 19-3：字长编程（M=00）

19.3.2 UART 发送器

发送器根据 M0 位的状态发送 8 位或 9 位的数据字。

发送使能位（TE）必须置 1 以打开发送功能。发送移位寄存器中的数据在 TX 脚上输出。

19.3.2.1 字符发送

在 UART 发送期间，在 TX 引脚上首先移出数据的最低有效位（默认配置下）。在此模式里，UART_TDR 寄存器充当了一个内部总线和发送移位寄存器之间的缓冲器 (TDR) (见图 19-4)。

每个字符之前都有一个低电平的起始位，之后跟着停止位，停止位的数目是可选择的。

UART 支持 2 种停止位的选择：1 和 2 个停止位。

注：

1. 在向 UART_TDR 写数据之前必须先令 TE 位为 1。
2. TE 位在数据传输过程中不应复位。在传输过程中复位 TE 位将破坏 TX 引脚上的数据，因为波特率计数器将被冻结。当前正在传输的数据将丢失。
3. 在 TE 位被置 1 后将会发送一个空闲帧。

19.3.2.2 可配置的停止位

每个数据帧发送的停止位的位数可以通过寄存器 UART_CR2 的位 STOP[1:0]进行编程。

- 1 个停止位：停止位的位数的默认值。
- 2 个停止位：可用于常规 UART 模式、单线模式。

空闲帧包括了停止位。

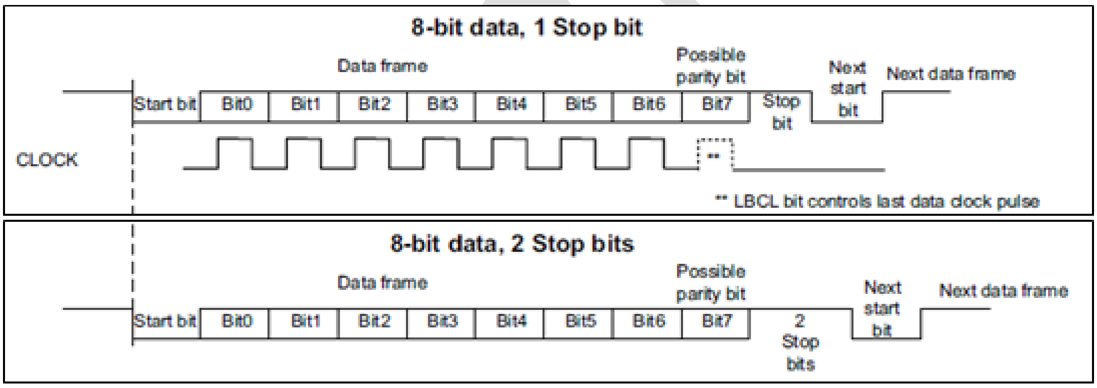


图 19-4：可配置的停止位

配置步骤：

1. 设置 UART_CR1 的 M0 位来定义字长。
2. 利用 UART_BRR 寄存器选择预期的波特率。
3. 在 UART_CR2 中设置停止位的位数。
4. 通过将 UART_CR1 寄存器的 UE 位置 1 来使能 UART。
5. 设置 UART_CR1 中的 TE 位，发送一个空闲帧作为第一次数据发送。
6. 把要发送的数据写进 UART_TDR 寄存器（此动作将清除 TXE 位）。在只有一个缓冲器的情况下，

对每个待发送的数据重复步骤 7。重复步骤 7 之前应等待 TXE 变成 1。

7. 在 UART_TDR 寄存器中写入最后一个数据字后，要等待 TC=1，它表示最后一个数据帧的传输结束。例如当关闭 UART 或需要进入低功耗模式之前，需要确认传输结束，避免破坏最后一次传输。

19.3.2.3 单字节通信

清零 TXE 位总是通过对数据寄存器的写操作来完成的。

TXE 位由硬件来设置，它表明:

- 数据已经从 TDR 移送到移位寄存器，数据发送已经开始。
- UART_TDR 寄存器被清空。
- 下一个数据可以被写进 UART_TDR 寄存器而不会覆盖先前的数据。

如果 TXEIE 位被设置，此事件将产生一个中断请求。

如果此时 UART 正在发送数据，对 UART_TDR 寄存器的写操作把数据存进 TDR 寄存器，并在当前传输结束时把该数据复制进移位寄存器。

如果此时 UART 没有在发送数据，对 UART_TDR 寄存器的写操作将导致直接把数据放进移位寄存器，数据传输开始，TXE 位则立即被置起。

当一帧字节发送完成（停止位发送后）且 TXE 已置 1 时，TC 位会被置 1。如果 UART_CR1 寄存器中的 TCIE 位是 1 时，则会产生中断。

在 UART_TDR 寄存器中写入了最后一个数据字后，在关闭 UART 模块之前或设置微控制器进入低功耗模式之前，必须先等待 TC=1（见图 19-5）。

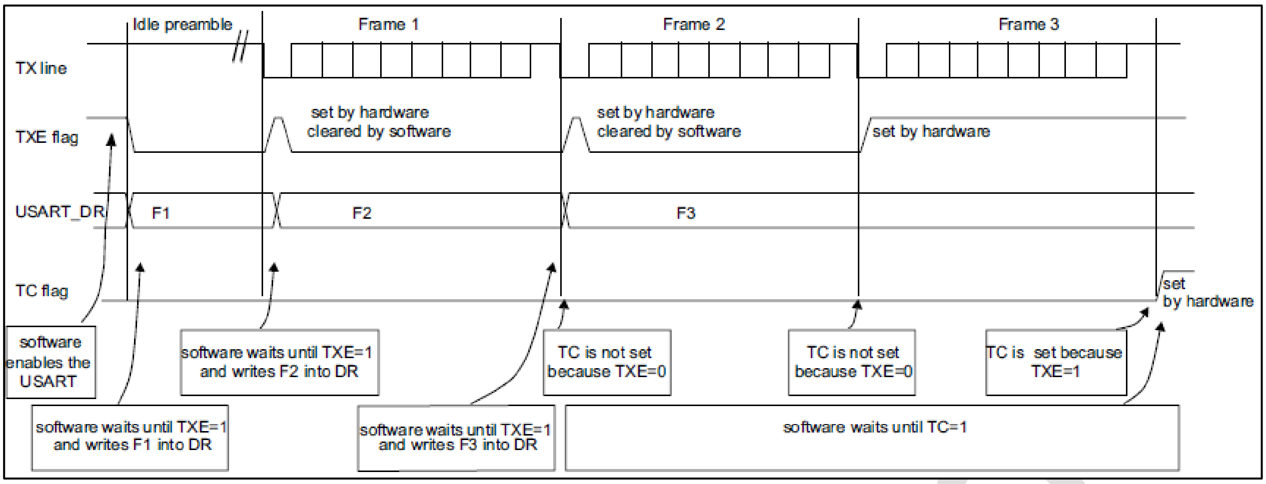


图 19-5：发送时 TC/TXE 行为

关闭 UART 的正确步骤如下：

1. 清除 TE 位（如果有数据正在传输或者 UART_TDR 寄存器中还有待发送的数据，会在关闭动作生效前发送完）。
2. TC 位会在发送完毕后被置 1。
3. 在 TC=1 后清除 UE 位。

19.3.2.4 空闲符号（Idle Characters）

将 TE 置 1 将使得 UART 在第一个数据前发送一空闲符号。

19.3.3 UART 接收器

接收器根据 UART_CR1 寄存器中 M0 位的状态接收 8 位或 9 位的数据字。

19.3.3.1 起始位侦测

过采样率为 16，起始位侦测序列相同。

在 UART 中，如果识别出一个特定的采样序列，那么就认为侦测到一个起始位。

该序列为：1 1 1 0 X 0 X 0 X 0 X 0 X 0 X 0。

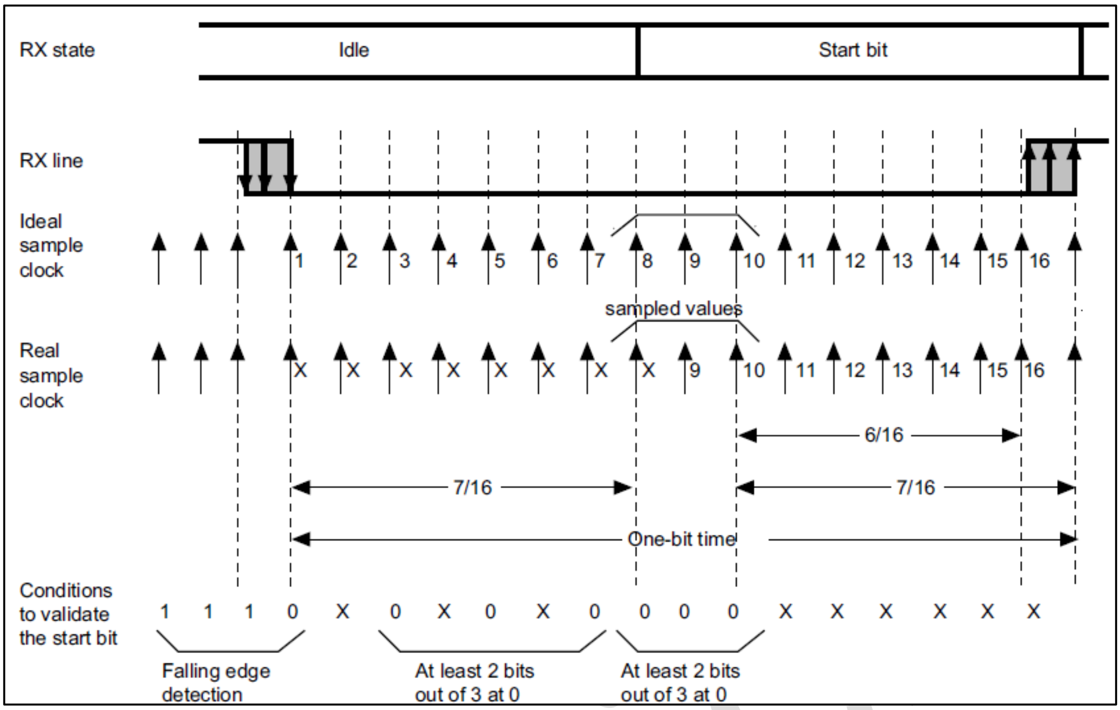


图 19-6：过采样率为 16 时的起始位侦测

注：如果该序列不完整，那么接收端将退出起始位侦测并回到空闲状态（不置标志位）开始等待下降沿。

如果 3 个采样点都为‘0’（在第 3、5、7 位的第一次采样，和在第 8、9、10 的第二次采样都为‘0’），则确认收到起始位，这时 RXNE 标志会由硬件置 1，如果这时 RXNEIE=1，则会产生中断请求。

如果下面 2 个条件成立，那么起始位仍然是有效的（RXNE 标志会置 1，如果 RXNEIE=1，则会产生中断请求），但是会设置 NE 噪声标志位。

- 两组采样（第 3、5、7 位的采样点和第 8、9、10 位的采样点）中，3 个采样点有 2 次采样为‘0’。
- 两组采样中的一组（第 3、5、7 位的采样点或第 8、9、10 位的采样点），3 个采样点有 2 次采样为‘0’。

如果两个条件都不满足，则起始位检测终止，接收器返回到空闲状态（不置位标志位）。

19.3.3.2 字符接收

在 UART 接收期间，数据的最低有效位（默认情况下）首先从 RX 脚移进。在此模式里，UART_RDR 寄存器充当了一个位于内部总线和接收移位寄存器之间的缓冲器。

配置步骤：

1. 设置 UART_CR1 的 M0 位来定义字长。
2. 利用波特率寄存器 UART_BRR 选择预期的波特率。
3. 在 UART_CR2 中设置停止位的位数。
4. 通过将 UART_CR1 寄存器的 UE 位置 1 来激活 UART。
5. 将 UART_CR1 的 RE 位置 1。这将激活接收器，使它开始寻找起始位。

当一个字符被接收到时：

- RXNE 位被置 1。它表明移位寄存器的内容被转移到 RDR。换句话说，数据已经被接收并且可以被读出（包括与之有关的错误标志 ）。
- 这时如果 RXNEIE 位是 1，将会引起中断请求。
- 在接收期间如果检测到帧错误，噪音或溢出错误，错误标志将被置起。PE 标志也会和 RXNE 一起被置 1。
- 由软件读 UART_RDR 寄存器完成对 RXNE 位清除。RXNE 位必须在下一字符接收结束前被清零，以避免溢出错误。

19.3.3.3 空闲符号

当一空闲帧被检测到时，其处理步骤和接收到普通数据帧一样，但如果 IDLEIE 位为 1 将产生一个中断请求。

19.3.3.4 溢出错误

如果 RXNE 还没有被复位就又接收到了一个字符，则发生溢出错误。数据只有当 RXNE 位被清零后才能从移位寄存器转移到 RDR 寄存器。

RXNE 标志是接收到每个字节后被置 1 的。如果下一个数据已被收到，RXNE 标志仍是置起的，会产生溢出错误。当溢出错误产生时：

- ORE 位被置 1。
- RDR 内容将不会丢失，读 UART_RDR 寄存器仍能得到先前的数据。

- 移位寄存器中以前的内容将被覆盖，随后接收到的数据都将丢失。
- 如果 RXNEIE 位被置为 1 或 EIE 位被置为 1，会产生中断。
- ORE 位可以通过将 ICR 寄存器中的 ORECF 位置 1 的方式清除。

注：当 ORE 位置 1 时，表明至少有 1 个数据已经丢失。有两种可能性：

- 如果 RXNE=1，尚一个有效数据还在接收寄存器 RDR 上，可以被读出。
- 如果 RXNE=0，这意味着上一个有效数据已经被读走，RDR 已经没有东西可读。 当上一个有效数据在 RDR 中被读取的同时又接收到新的（也就是丢失的）数据时，此种情况是可能发生的。

19.3.3.5 选择时钟源和适当的过采样方法

时钟源的选择要通过时钟控制系统（参见章节：5 复位与时钟控制（RCC））。必须在使能 UART 之前将 UE 位置 1，从而打开 UART 的时钟源。

时钟源的选择需要依据两个标准：

- 在低功耗模式下使用串口的可能性
- 通讯速度

时钟源的频率是 f_{CK}。

当使用双时钟域支持 Stop 模式唤醒的功能时，时钟源可以是下列时钟源中的一个：PCLK（默认）、HSI 或 SYSCLK。

除此之外，UART 的时钟源是 PCLK。

选择 HSI 作为时钟源，可以使得 UART 在 MCU 处于低功耗状态的时候还能接收数据。基于唤醒模式选择和接收到的数据，UART 会将 MCU 唤醒，并由软件将收到的数据从 UART_RDR 寄存器中读走。

如果用其它的时钟源，必须先将其打开以保证 UART 通讯。

通讯速度范围（特别是最大通讯速度）是由所选的时钟源来决定的。

接收器根据用户设定的过采样率来执行数据恢复，从而将接收数据和噪声区分开来。这需要在最大通讯速度和噪声抗扰度及对波特率偏差的敏感度之间做取舍。

- 过采样率是波特率时钟的 16 倍，可提高时钟偏差容忍度。这种情况下，最高通讯速度就会被限

制在 $f_{CK}/16$，其中，f_{CK} 就是时钟源的频率。

UART_CR3 中的 ONEBIT 位用来选择判断逻辑电平的方法。有两种选择：

- 根据接收数据位中央的三个采样结果进行多数票决。这种情况下，当发现三个参与票决的采样结果不等时，NF 位会被置 1。
- 根据接收数据位中央的单个采样结果来直接裁决
- 根据应用：
  - 在噪声干扰环境中应该选择三个采样多数票决的方式，可以排除检测到噪声的数据， 因为那说明在数据采样的时候有干扰。
  - 在不担心噪声的情况下，应选择单采样裁决（ONEBIT=1），可提高接收器对时钟偏差的容忍度（参见章节：19.3.5 UART 波特率对时钟偏差的容忍），这种情况下 NF 位永远都不会置 1。

当一帧数据中检测到噪声时：

- 在 RXNE 位的上升沿，NF 位会被置 1。
- 有问题的数据仍会从移位寄存器转移到 UART_RDR 寄存器。
- 单字节通讯的时候不会产生中断。 然而与这一位同时上升的 RXNE 是会产生中断的。在多缓冲区通讯的情况下，只要 UART_CR3 中的 EIE 位是高就会产生中断。

将 ICR 寄存器中的 NFCF 位置 1 就可以清除 NF 标志。

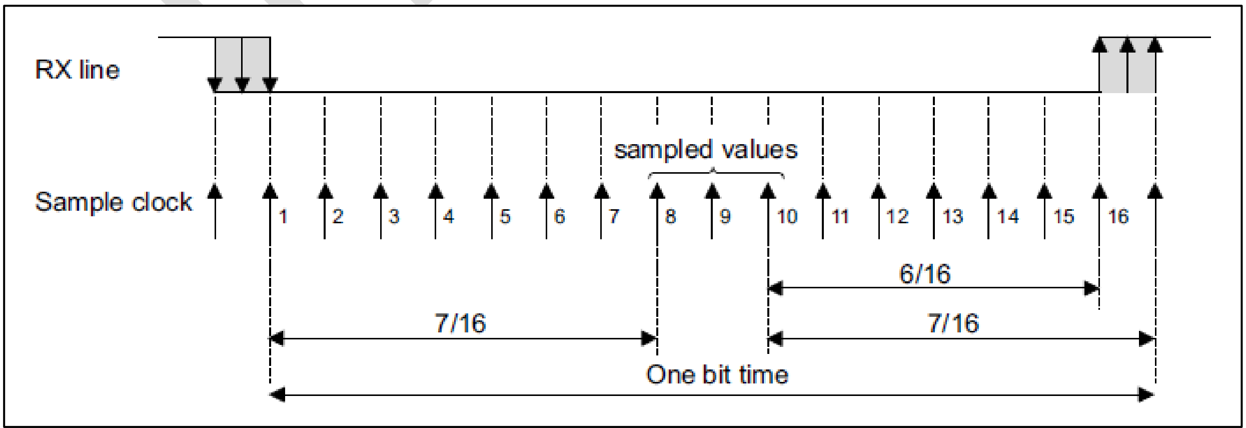


图 19-7：过采样率为 16 的时候的数据采样

表 19-2：从采样数据中检测噪声

采样值	NF 状态	接收到的位值
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

19.3.3.6 帧错误

当以下情况发生时检测到帧错误：由于没有同步上或大量噪音的原因，停止位没有在预期的时间上接收和识别出来。

当帧错误被检测到时：

- FE 位被硬件置 1
- 有问题的数据仍会从移位寄存器转移到 UART_RDR 寄存器。
- 单字节通讯的时候不会产生中断。 然而与这一位同时上升的 RXNE 是会产生中断的。在多缓冲区通讯的情况下，只要 UART_CR3 中的 EIE 位是高就会产生中断。

将 UART_ICR 寄存器中的 FECF 置 1 就可以清除 FE 标志。

19.3.3.7 接收期间的可配置的停止位

被接收的停止位的个数可以通过 UART_CR2 的控制位来配置，在正常模式时，可以是 1 或 2 个：

- 1 个停止位：对 1 个停止位的采样在第 8，第 9 和第 10 采样点上进行。
- 2 个停止位：对 2 个停止位的采样是在第一停止位的第 8，第 9 和第 10 个采样点完成的。如果第一个停止位期间检测到一个帧错误，帧错误标志将被置 1。第二个停止位不再检查帧错误。在第一个停止位结束时 RXNE 标志将被设置。

19.3.4 UART 波特率产生

接收器和发送器的波特率相同，在 UART_BRR 寄存器中设置。

公式 1：标准 UART（包括 SPI 模式）的波特率（OVER8=0/1）

如果过采样为 16，等式为：Tx/Rx_Baud = f_{CK} / (UARTDIV)

如果过采样为 8，等式为：Tx/Rx_Baud = 2 x f_{CK} / (UARTDIV)

UARTDIV 是一个无符号的定点数。值设置在 UART_BRR 寄存器：

当 OVER8=0，BRR = UARTDIV。

当 OVER8=1，BRR[2:0]的值由 UARTDIV[3:0]右移 1 位所得；BRR[3]位必须保持为 0；BRR[15:4=UARTDIV[15:4]。

注：在写入 UART_BRR 之后，波特率计数器会立即被波特率寄存器的新值替换。因此，不要在通信过程中改变波特率寄存器的数值。

19.3.4.1 如何从 UART_BRR 寄存器值得到 UARTDIV

例 1：获得 9600 波特率@f_{CK}=8MHz

如果过采样为 16：UARTDIV = 8000000/9600；BRR = UARTDIV = 833d = 0341h

例 2：获得 921600 波特率@f_{CK}=48MHz

如果过采样为 16：UARTDIV = 48000000/921600；BRR = UARTDIV = 52d = 34h

表 19-3：波特率时的误差计算（f_{CK}=72MHz，过采样率为 16/8）

	16 倍采样			8 倍采样		
目标波特率 (bps)	实际波特率 (bps)	BRR 值	Error (%)	实际波特率 (bps)	BRR 值	Error (%)
2400	2400	0x7530	0	2400	0xEA60	0
9600	9600	0x1D4C	0	9600	0x3A94	0
19200	19200	0x0EA6	0	19200	0x1D46	0
38400	38400	0x0753	0	38400	0x0EA3	0
57600	57600	0x04E2	0	57600	0x09C2	0
115200	115200	0x0271	0	115200	0x04E1	0
230400	230769	0x0139	0.16	230400	0x0270	0

	16 倍采样			8 倍采样		
目标波特率 (bps)	实际波特率 (bps)	BRR 值	Error (%)	实际波特率 (bps)	BRR 值	Error (%)
460800	461538	0x009C	0.16	460064	0x0134	0.16
921600	923076	0x004E	0.16	923077	0x0096	0.16
2M	2M	0x0024	0	2M	0x0044	0
3M	3M	0x0018	0	3M	0x0030	0
4M	4M	0x0012	0	4M	0x0022	0
4.5M	4.5M	0x0010	0	4.5M	0x0020	0
5M	N/A	N/A	N/A	4965517	0x0016	0.69
6M	N/A	N/A	N/A	6M	0x0014	0
7M	N/A	N/A	N/A	6857143	0x0012	2.04
8M	N/A	N/A	N/A	8M	0x0011	0
9M	N/A	N/A	N/A	9M	0x0010	0

注：

1. % Error = (实际波特率 - 目标波特率)/目标波特率
2. CPU 的时钟频率越低，则某一特定波特率的精度也越低。可以达到的波特率上限可以由这组数据得到。

19.3.5 UART 波特率对时钟偏差的容忍

只有当总的时钟偏差小于 UART 接收器的容忍度，UART 异步接收器才能正常地工作。影响这些变化的因素有：

- DTRA：由于发送器误差而产生的变化（包括发送器端振荡器的变化）
- DQUANT：接收器端波特率取整所产生的误差
- DREC：接收器端振荡器的变化
- DTCL：由于传输线路产生的变化（通常是由于收发器在由低变高的转换时序与由高变低转换时序之间的不对称性所造成）
  - DTRA + DQUANT + DREC + DTCL + DTU < UART 接收器的容忍度
- 其中，DWU 是使用停止模式唤醒时由于采样点偏差引起的误差。
  - 当 M0 = 1：DWU = t_{WUUART} / (11 * t_{bit})

- 当 M0 = 0：DWU = t_{WUUART} / (10 * t_{bit})

- t_{WUUART} 是从检测到唤醒事件到时钟（由外设请求）和调节器（LDO）准备就绪之间的时间。在本产品中，t_{WUUART} 对应于数据表中提供的 t_{WUSTOP} 值。
- 根据以下选择，UART 接收器可以在下面表格中指定的最大耐受偏差范围内正确接收数据:
  - 由 UART_CR1 寄存器的 M0 位定义 10 位或 11 位字符长度。
  - UART_BRR 寄存器的比特 BRR[3:0]等于或不等于 0000。
  - UART_CR3 寄存器的 ONEBIT 位定义的，是 1 位采样还是 3 位采样。

表 19-4：串口接收容忍度（当 BRR[3:0]= 0000b）

M[0]	OVER8 = 1	
	ONEBIT = 0	ONEBIT = 1
0	2.50%	3.75%
1	2.27%	3.41%

表 19-5：串口接收容忍度（当 BRR[3:0] != 0000b）

M0	OVER8 = 1	
	ONEBIT = 0	ONEBIT = 1
0	2%	3%
1	1.82%	2.73%

注：表格中指定的数据在特殊情况下可能会略有不同，当接收到的帧包含一些恰好是 10 位持续时间的空闲帧时，M0 bits = 0（当 M0 bits =1 时 11 位位长）。

19.3.6 UART 多机通信

多机通讯中，下面这些位需要清零:

- UART_CR3 寄存器中的 HDSEL、IREN 位

可以将多个 UART 连接成一个网络来实现多机通讯。例如，一个 UART 设备可以是主机，它的 TX 输出和其他 UART 的 RX 输入相连接；其他 UART 作为从设备，它们各自的 TX 输出‘逻辑与’在一起，并且和主设备的 RX 输入相连接。

在多机配置中，我们通常希望只有被寻址的接收者才被激活，来接收预期的数据，这样就可以减少未被寻址的接收器的参与带来多余的 UART 服务开销。

未被寻址的设备可启用其静默功能进入静默模式。通过 UART_CR1 寄存器的 MME 位置 1，启动静默模式功能。

在静默模式：

- 任何接收状态位都不会被置 1。
- 所有接收中断被禁止。
- UART_CR1 寄存器中的 RWU 位被置 1。RWU 可以被硬件自动控制或在一定条件下由软件通过 UART_RQR 寄存器的 MMRQ 位写入。

根据 UART_CR1 寄存器中的 WAKE 位状态，UART 可以使用一下两种方式之一进入或退出静默模式：

- 如果 WAKE 位为 0：进行空闲总线检测。
- 如果 WAKE 位为 1：进行地址标记检测。

19.3.6.1 空闲总线检测（WAKE=0）

当 MMRQ 位被置 1，且 RWU 被自动置 1 时，UART 进入静默模式。

当检测到一个空闲帧的时候会唤醒，然后 RWU 位被硬件清零但 UART_ISR 寄存器中的 IDEL 位没有被置位。

下图给出了利用空闲总线检测来唤醒和进入静默模式的一个例子：

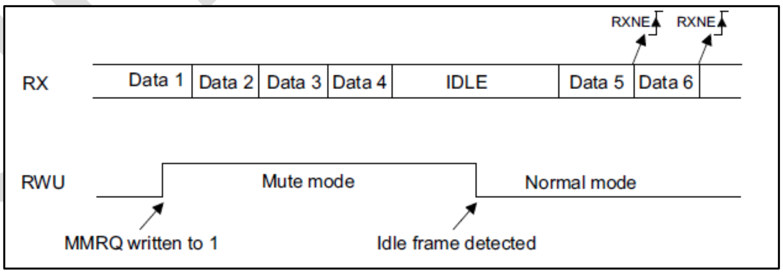


图 19-8：静默模式（使用空闲总线检测）

注：

1. 如果在空闲字符已经过去之后才置位 MMRQ，将不会退出静默模式（RWU 没被置 1）。
2. 如果在线路空闲期间激活 UART，在一个空闲帧时长之后才会检测到空闲状态（不仅仅是在收到一个数据帧之后）。

19.3.6.2 4-bit/7-bit 地址标记检测（WAKE=1）

在这个模式里，如果 MSB 是 1，该字节被认为是地址，否则被认为是数据。在一个地址字节中，目标接收器的地址被放在 4-bit 或 7-bit 位域中。设置 ADDM7 位来选择用 4 位地址还是用 7 位地址。接收器将这 4-bit 或 7-bit 地址同它自己地址做比较，接收器的地址在 UART_CR2 寄存器的 ADD 位设置。

注：在 9-bit 数据模式下，地址检测分别按 8 位地址（ADD[7:0]）操作。

如果接收到的地址字节与它的编程地址不匹配时，UART 进入静默模式。此时，硬件将 RWU 位置 1。当 UART 进到静默模式，此地址字节不会置位 RXNE 标志，也不会产生中断。

将 MMRQ 写 1 也会令 UART 进入静默模式。这时 RWU 位也被自动置 1。

如果接收到的字节与它的编程地址匹配，UART 将退出静默模式。然后 RWU 位被清零，后续的字节会被正常接收。自 RWU 位被清零之后，RXNE 位会因为地址字节的接收而被置 1。

下图给出了利用地址标记检测来退出静默模式的一个举例。

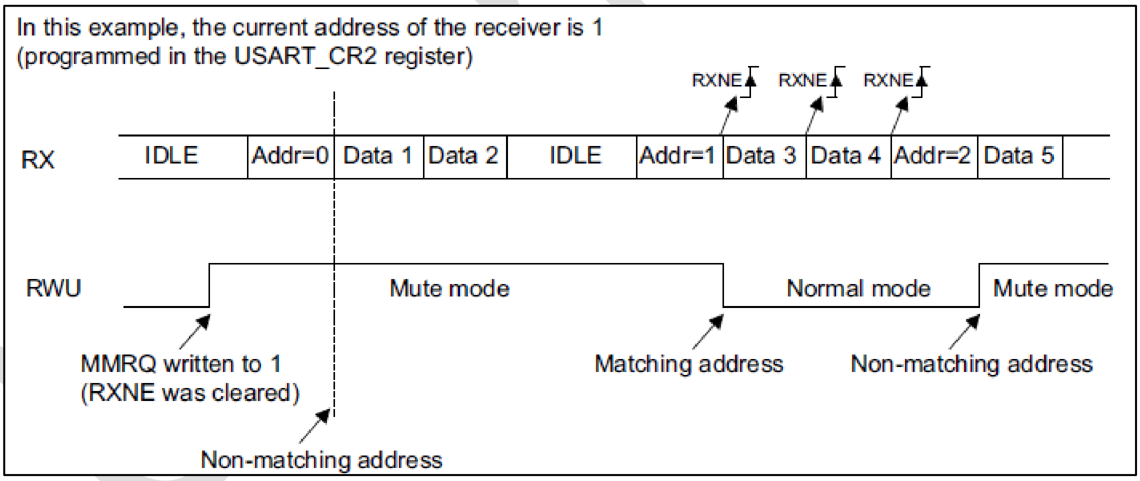


图 19-9：静默模式（使用地址标记检测）

19.3.7 UART 检验控制

设置 UART_CR1 寄存器上的 PCE 位，可以使能校验控制（发送时生成一个校验位，接收时进行校验检查)。根据 M 位定义的帧长度，UART 帧格式罗列在下表。

表 19-6：帧格式

M 位	PCE 位	UART 帧*
0	0	Start bit+8-bit data+stop bit
0	1	Start bit+7-bit data+parity bit+stop bit
1	0	Start bit+9-bit data +stop bit
1	1	Start bit+8-bit data+parity bit+stop bit

注*：在数据寄存器中，parity bit 总是占据 MSB 位置（第 9 位、第 8 位，取决于 M 位值）。

19.3.7.1 偶校验

计算校验位，使得一帧中的 LSB 数据位加上校验位中‘1’的总个数为偶数。LSB 数据位可以是 7 或 8 位，由 M 位控制。

例如: 数据=00110101, 有 4 个‘1’, 如果选择偶校验（UART_CR1 中的 PS = 0), 校验位将是‘0’。

19.3.7.2 奇校验

计算校验位，使得一帧中的 LSB 数据位加上校验位中‘1’的总个数为奇数。LSB 数据位可以是 7 或 8 位，由 M 位控制。

例如: 数据=00110101, 有 4 个‘1’, 如果选择奇校验（UART_CR1 中的 PS = 1), 校验位将是‘1’。

19.3.7.3 接收时的校验检查

如果校验检查失败，UART_ISR 寄存器中的 PE 标志会被置 1，如果 UART_CR1 寄存器中的 PEIE 为 1，将产生相应中断。软件对 UART_ICR 寄存器的 PECF 位写 1，可以清除 PE 标志。

19.3.7.4 发送时的校验生成

如果 UART_CR1 的 PCE 位被置 1，写进数据寄存器的数据的 MSB 位被校验位替换后发送出去（如果选择偶校验（PS=0）偶数个‘1’，如果选择奇校验（PS=1）奇数个‘1’）。

19.3.7.5 多缓冲器通信中的错误标志和中断产生

在多缓冲器通信的情况下，通信期间如果发生任何错误，会在当前字节传输后将错误标志置 1。如果中断使能位被置 1，将产生中断。在单个字节接收的情况下，帧错误、溢出错误和噪音标志和 RXNE 一起被置位，有一个独立的错误标志中断使能位（UART_CR3 寄存器的 EIE 位）；如果此 EIE 被置位，任一错误发生都会在当前字节传输结束后产生中断。

19.3.8 STOP 模式唤醒

本节内容仅在 STOP 模式唤醒功能支持时有效。请参见 UART 具体功能配备。

当 UART 时钟被设置为 HSI 时，并置位 UESM 位就可以使用 UART 将 MCU 从 Stop 模式唤醒。时钟设置参考章节：5 复位与时钟控制系统（RCC）。

- UART 时钟源为 HSI
- 如果在停止模式下，HSI 时钟被关闭，当 UART 接收线上的下降沿被检测到时，UART 请求打开 HSI 时钟，然后 HSI 时钟用于帧接收。
- 如果确认唤醒事件，则 MCU 从低功耗模式唤醒，数据接收正常。
- 如果唤醒事件未被验证，则 HSI 时钟再次关闭，MCU 不被唤醒，保持在低功耗模式，释放时钟请求。

可以用标准的 RXNE 中断来唤醒 STOP 模式。这时，在进入 Stop 模式之前 RXNEIE 位必须先设置为 1。另外需要在 WUS 位指定这个中断。

必须在进入 Stop 模式之前先置位 UART_CR1 中的 UESM 位，否则唤不醒。

当检测到唤醒事件，WUF 表示会被硬件置 1。若 WUFIE 位置 1，将会产生中断。

注：

1. 在进入 Stop 模式之前，软件必须检查 UART 没有正在发数据，这可以通过 UART_SR 寄存器中的 BUSY 标志来判断。
2. 在收到唤醒事件时，WUF 标志肯定会被置 1，这不受 MCU 所处的工作模式影响，无论是在 Stop 模式还是在工作模式都可以。

3. 当刚刚初始化好接收器的时候就进入 Stop 模式时，需要检查 REACK 位以确定 UART 确实已经被打开了。

19.3.8.1 Stop 模式下使用静默功能

如果在进入 Stop 模式之前，UART 已经处于静默状态下：

- 不可以使用基于空闲检测退出静默的方式，因为空闲检测在 Stop 模式下不工作。
- 如果使用地址匹配的方式退出静默状态，唤醒源也就只能是地址匹配事件。如果在进入 Stop 模式时设置了 RXNE 标志位，则 UART 接口在地址匹配时保持静音模式，并在 Stop 模式下唤醒。
- 如果 UART 被设置为起始位检测就唤醒 MCU，WUF 标志会置 1，但 RXNE 标志就不会置位。

19.3.8.2 确定允许从停止模式正确唤醒的最大 UART 波特率（UART 时钟源是 HIS）

允许从停止模式正确唤醒的最大波特率取决于：

- 数据手册中的 t_{WUUART}。
- UART 接收器容忍，参见 19.3.5 UART 波特率对时钟偏差的容忍。

举例：OVER8 = 0, M bits = 1, ONEBIT = 0, BRR[3:0] = 0000。

此条件下，根据表 19-4：串口接收容忍度（当 BRR[3:0]= 0000b），UART 接收器容忍为 3.41%。

DTRA + DQUANT + DREC + DTCL + DWU < UART 接收器容忍

DWU max. = t_{WUUART} / (11 x t_{bit} min.)

t_{bit} min. = t_{WUUART} / (11 x DWU max.)

如果我们考虑理想情况，参数 DTRA、DQUANT、DREC 和 DTCL 为 0%，DWU max.为 3.41%；在实际中，我们至少需要考虑 HSI 的不准确性。

假设 HSI 误差 = 1%，t_{WUUART} = 3 µs（值为示例，正确值请参考芯片数据手册）：

DWU max. = 3.41% - 1% = 2.41%

t_{bit} min. = 3 µs / (11 x 2.41%) = 11.32 µs

此条件下，允许从停止模式正确唤醒的最大波特率为 1/11.32 µs = 88.36KBaud。

19.4 UART 低功耗模式

表 19-7：UART 低功耗模式

模式	描述
SLEEP	无影响，UART 中断唤醒 SLEEP 模式
STOP	当 UESM 位被设置并且 UART 时钟被设置为 HSI 时，UART 能够从 STOP 模式唤醒 MCU。MCU 从 STOP 模式唤醒可以使用任一标准 RXNE 或 WUF 中断。

19.5 UART 中断

表 19-8：UART 中断表

中断事件	事件标志	使能控制位
发送数据寄存器空	TXE	TXEIE
发送完成	TC	TCIE
接收数据寄存器非空（数据 ready 可读）	RXNE	RXNEIE
溢出错误检测	ORE	
空闲线检测	IDLE	IDLEIE
奇偶错误	PE	PEIE
噪声标志，溢出错误和多缓冲区通讯中的帧错误	NF or ORE or FE	EIE
从 Stop 模式唤醒	WUF*	WUFIE

注*：WUF 中断只在 Stop 模式中有用。

UART 中断时间全部连接到同一个中断向量（见图 19-10：UART 中断映射图）。

- 发送期间：发送完成，发送数据寄存器空中断。
- 接收期间：空闲线检测、溢出错误、接收数据寄存器非空、校验错误、噪声标志（仅在多缓冲区通讯时）、帧错误（仅在多缓冲区通讯）、字符匹配，等等。

如果设置了相应的使能控制位，这些事件都可以引起中断。

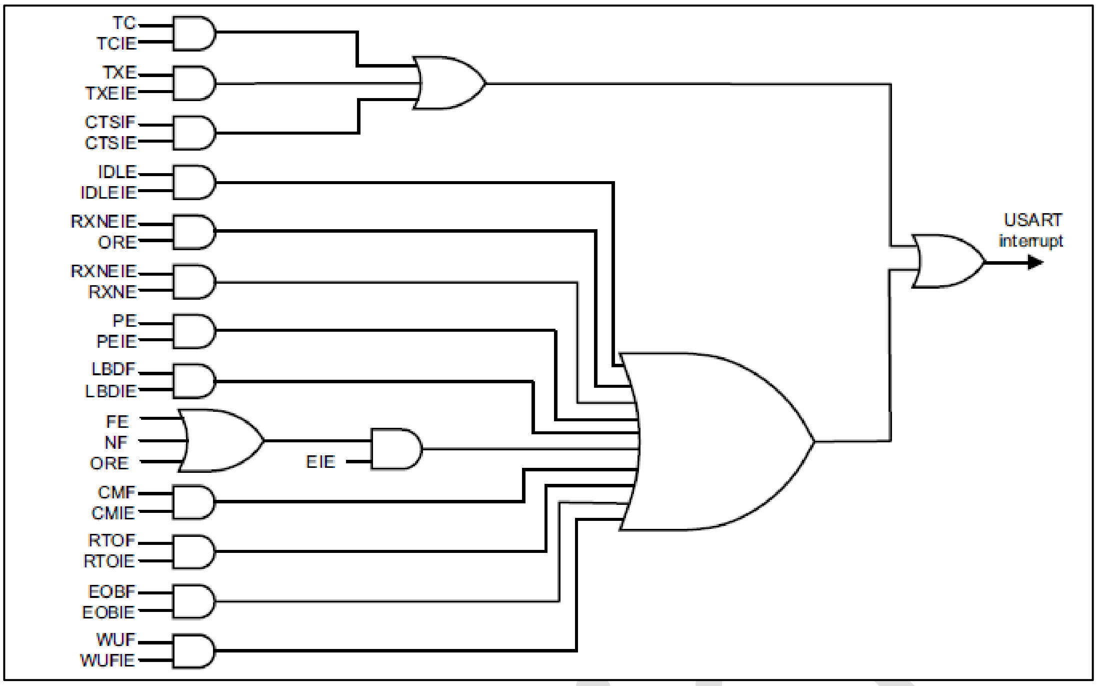


图 19-10：UART 中断映射图

19.6 相关寄存器

19.6.1 控制寄存器（UART_CR1）

偏移地址：0x00

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	Res.	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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15	OVER8	过采样模式： 0：16 倍过采样 1：8 倍过采样 此位只能在 UART 禁用时（UE=0）改写。
14	RSV	保留位，固定读为 0

位	名称	描述
13	MME	静默模式使能，由软件置 1 和清零。 这个位开启 UART 的静默模式功能。当置为 1 时，UART 可以在工作模式和静默模式之间切换，和 WAKE 位的定义一样。 0：接收器保持在工作模式（active mode） 1：接收器可以在工作模式和静默模式间切换
12	M0	字长（word length），由软件置 1 和清零。 0：1 个起始位，8 个数据位，n 个停止位 1：1 个起始位，9 个数据位，n 个停止位 此位只能在 UART 禁用时（UE=0）改写。
11	WAKE	接收器唤醒方式： 这个位决定 UART 从静默模式唤醒的方式，由软件置 1 和清零。 0：空闲线 1：地址标记 此位只能在 UART 禁用时（UE=0）改写。
10	PCE	校验控制使能，由软件置 1 和清零。 这个位选择硬件校验控制（产生和检测）功能。当校验控制被打开，计算好的校验位被插入到最高位 MSB（M0=1 时是第 9 位，M0=0 时是第 8 位），并检测接收数据的校验位。一旦此位（PCE）被置 1，在当前字节（包括收发）之后激活校验控制。 0: 校验控制禁止 1: 校验控制使能 此位只能在 UART 禁用时（UE=0）改写。
9	PS	校验选择： 当校验生成和检测开启（PCE=1），这个位选择使用奇校验还是偶校验。由软件置 1 和清零。校验方式会在当前字节结束后生效。 0: 偶校验 1: 奇校验 此位只能在 UART 禁用时（UE=0）改写。
8	PEIE	PE 中断使能，由软件置 1 和清零。 0: 中断禁止 1: 在 UART_ISR 寄存器中的 PE 被置 1 的时候会产生 UART 中断。
7	TXEIE	TXE 中断使能，由软件置 1 和清零。 0: 中断禁止 1: 在 UART_ISR 寄存器中的 TXE 被置 1 的时候会产生 UART 中断。
6	TCIE	TC 中断使能，由软件置 1 和清零。 0: 中断禁止 1: 在 UART_ISR 寄存器中的 TC 位被置 1 的时候会产生 UART 中断。

位	名称	描述
5	RXNEIE	RXNE 中断使能，由软件置 1 和清零。 0：中断禁止 1：在 UART_ISR 寄存器中的 ORE 或 RXNE 被置 1 的时候会产生 UART 中断。
4	IDLEIE	空闲中断使能，由软件置 1 和清零。 0：中断禁止 1：在 UART_ISR 寄存器中的 IDLE 位被置 1 的时候会产生 UART 中断。
3	TE	发送器使能： 这个位打开发送器，由软件置 1 和清零。 0：发送器关闭 1：发送器打开 注：在发送期间，TE 位上的一个‘0’脉冲（‘0’后面跟一个‘1’）会引起当前字发送完后跟着再发送一个线路空闲信息出去。要产生一个空闲字符，TE 位不可以立即被写 1。为了确保所需的持续时间，软件可以轮询 UART_ISR 寄存器中的 TEACK 位。
2	RE	接收器使能： 这个位打开接收器，由软件置 1 和清零。 0：接收器关闭 1：接收器打开并开始侦测起始位
1	UESM	Stop 模式下 UART 使能： 当这个位为 0，UART 不能将 MCU 从 Stop 模式下唤醒。 当这个位为 1，UART 可以将 MCU 从 Stop 模式下唤醒，条件是 UART 时钟选择 HIS（在 RCC 模块），由软件置 1 和清零。 0：UART 不能从 Stop 模式中唤醒 MCU。 1：UART 可以从 Stop 模式中唤醒 MCU。当这个功能被打开，UART 的时钟源必须为 HIS（参见章节：5 复位与时钟控制（RCC）） 注 1：推荐在进入 Stop 模式前将这个 UESM 位置 1，并在退出 Stop 模式后将它清零。 注 2：如果 UART 不支持从 Stop 模式唤醒功能，该位为保留位并由硬件强制为零。参见章节：UART 功能配备
0	UE	UART 使能： 当这个位被清零，UART 的预分频器和输出都立即停止，并且当前的操作也被取消。UART 的配置会被保留，但 UART_ISR 中所有的状态标志都会被复位。此位由软件置 1 和清零。 0: UART 预分频器和输出关闭，低功耗模式 1: UART 开启 注：为了进入低功耗模式而不在线路上产生错误，需要在清零 UE 位之前先清零 TE 位，并等待 UART_ISR 中的 TC 位被置 1。

19.6.2 控制寄存器 2（UART_CR2）

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD[7:4]				ADD[3:0]				Res.				MSBFIRST	DATAINV	TXINV	RXINV
rw				rw								rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	Res.	STOP[1:0]		Res.							ADDM7	Res.			
rw		rw									rw				

位	名称	描述
31:28	ADD[7:4]	UART 节点的地址： 这个位给出 UART 节点的地址或等待确认的字符码。 这用于静音模式或停止模式期间的多处理器通信，用于 7 位地址标记检测唤醒。发送器发出的字符的最高位（MSB）应该为 1。 这个位只能在接收器被关闭（RE=0）或者在 UART 被关闭的时候（UE=0）才能改写。
27:24	ADD[3:0]	UART 节点的地址： 这用于静音模式或停止模式期间的多处理器通信，用于地址标记检测唤醒。 这个位域只能在接收器被关闭（RE=0）或者在 UART 被关闭的时候（UE=0）才能改写。
23:20	RSV	保留位，固定读为 0
19	MSBFIRST	高位在前（Most significant bit first）： 由软件置 1 和清零 0：数据发送和接收，起始位+数据位 LSB 在前（bit 0）。 1：数据发送和接收，起始位+数据位 MSB 在前（bits 7/8/9 取决 M[1:0]）。 这个位只能在 UART 禁止时（UE=0）改写。
18	DATAINV	按位反向（Binary data inversion）： 由软件置 1 和清零 0：数据寄存器中的逻辑数据在发送和接收时，采用正/直接逻辑。(1=H, 0=L) 1：数据寄存器中的逻辑数据在发送和接收时，采用负/反向逻辑。(1=L, 0=H)。校验位也一样反向。 这个位只能在 UART 禁止时（UE=0）改写。
17	TXINV	TX 脚有效电平反向，由软件置 1 和清零： 0：TX 脚信号工作于标准逻辑电平（VDD =1/idle, Gnd=0/mark） 1：TX 脚信号被反向（VDD =0/mark, Gnd=1/idle）

位	名称	描述
		这可以用于 TX 线上带有外部反相器应用。 这个位只能在 UART 禁止时（UE=0）改写。
16	RXINV	RX 引脚有效电平反向，由软件置 1 和清零: 0：RX 脚信号工作于标准逻辑电平（VDD =1/idle, Gnd=0/mark） 1：RX 脚信号被反向（VDD =0/mark, Gnd=1/idle)。这可以用于 RX 线上带有外部反相器应用。 这个位只能在 UART 禁止时（UE=0）改写。
15	SWAP	交换 TX/RX 引脚，由软件置 1 和清零: 0：TX/RX 引脚按照标准引脚分配来使用。 1：TX 和 RX 的引脚功能交换，用于和其它 UART 口进行交叉互联的时候。 这个位只能在 UART 禁止时（UE=0）改写。
14	RSV	保留位，固定读为 0
13:12	STOP[1:0]	停止位: 这些位用来配置停止位的个数。 00：1 个停止位: 01：保留 10：2 个停止位 11：保留 这个位只能在 UART 禁止时（UE=0）改写。
11:5	RSV	保留位，固定读为 0
4	ADDM7	7 位地址监测或 4 位地址监测（Address Detection）: 这个位用来选择使用 4 位地址检测还是 7 位地址检测。 0：4 位地址检测 1：7 位地址检测（在 8 位数据模式下） 这个位只能在 UART 禁止时（UE=0）改写。 注：在 7 位和 9 位数据模式下，地址检测分别按 6 位和 8 位地址（ADD[5:0] 和 ADD[7:0]）操作。
3:0	RSV	保留位，固定读为 0

19.6.3 控制寄存器 3（UART_CR3）

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.									WUFIE	WUS[1:0]		Res.			
									rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			OVRDIS	ONEBIT	Res.										EIE
			rw	rw											rw

位	名称	描述
31:23	RSV	保留位，固定读为 0
22	WUFIE	从 Stop 模式唤醒中断使能，由软件置 1 和清零. 0：中断禁止 1：在 UART_ISR 寄存器中的 WUF 被置 1 的时候会产生 UART 中断。 注: 1. WUFIE 位必须在进入 Stop 模式之前设置。 2. WUF 中断只在 Stop 模式中有用。 3. 若 UART 不支持从 Stop 模式唤醒功能，该位为保留位并由硬件强制为零。
21:20	WUS[1:0]	从 Stop 模式唤醒中断标志选择这个位域制定激活 WUF 标志的事件。 00：在发生地址匹配事件的时候激活 WUF 01：保留 10：WUF 在检测到起始位的时候激活 11：WUF 在得到接受数据寄存器非空事件时激活 这个位域只能在 UART 未被使能的时候（UE=0）改写。 注：如果 UART 不支持从 Stop 模式唤醒功能，该位为保留位并由硬件强制为零。
19:13	RSV	保留位，固定读为 0
12	OVRDIS	溢出检测禁止，这个位用于禁止对接收溢出现象的检测: 0：当之前接收到的数据没有在新接收到数据之前读走时，会引起溢出错误标志 ORE 被硬件置 1。 1：溢出检测功能关闭。如果在新的接收数据到来时，RXNE 标志仍然是 1，但 ORE 标志还不是 1 时，新的数据会将 UART_RDR 中以前的内容覆盖掉。 这个位域只能在 UART 未被使能的时候（UE=0）改写。 注：这个控制位允许读取数据的时候检查通讯数据流。
11	ONEBIT	单次采样方式使能: 这个位允许用户选择采样方式。当选择单次采样方式的时候，噪声监测标志（NF）就被禁止了。 0：三次采样方式 1：单次采样方式 这个位域只能在 UART 未被使能的时候（UE=0）改写。
10:1	RSV	保留位，固定读为 0
0	EIE	错误中断使能: 在允许帧错误，溢出错误或噪声错误产生中断请求时要打开这个开关。

位	名称	描述
		0：中断禁止 1：当 UART_ISR 寄存器中的 FE=1 或 ORE=1 或 NF=1 时，会产生中断。

19.6.4 波特率寄存器（UART_BRR）

偏移地址：0x00C

复位值：0x0000 0000

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:4	DIV Mantissa[11:0]	UART DIV 的整数部分: 这 12 位定义 UART 分频器触发因子的整数部分。 注：此位段只能在 UE=0 时软件才能修改。
3:0	DIV Mantissa[3:0]	UART DIV 分配器触发因子的小数部分: 这 4 位定义 UART 分频器触发因子的小数部分，当 OVER 8 = 1 时，DIV_Fraction3 位是没有用的，并且必须保持为 0。 注：此位段只能在 UE=0 时软件才能修改。

19.6.5 请求寄存器（UART_RQR）

偏移地址：0x018

复位值：0x0000 0000

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

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

位	名称	描述
31:3	RSV	保留位，固定读为 0
2	MMRQ	静默模式请求： 对这个位写 1 将导致 UART 进入静默模式，同时置位 RWU 标志。
1:0	RSV	保留位，固定读为 0

19.6.6 中断和状态寄存器（UART_ISR）

偏移地址：0x01C

复位值：0x0000 00C0

31–23	22	21	20	19	18–17	16
Res.	REACK	TEACK	WUF	RWU	Res.	BUSY
	r	r	r	r		r

15–8	7	6	5	4	3	2	1	0
Res.	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
	r	r	r	r	r	r	r	r

位	名称	描述
31:23	RSV	保留位，固定读为 0
22	REACK	接收使能通知标志： 这个位有硬件控制，当接收使能信号被 UART 读到的时候，会给出一个回执。这可以在进入 Stop 模式之前，用来确认 UART 是不是已经准备好接收了。 注：若 UART 不支持从 Stop 模式唤醒功能，该位为保留位并由硬件强制为零。
21	TEACK	发送使能通知标志： 这个位有硬件控制，当发送使能信号被 UART 读到的时候，会给出一个回执。这可以在写 UART_CR1 寄存器的 TE=0 以产生一个空闲帧请求，跟着又将 TE 写成 1 时，用于保证 TE=0 的最小周期的时候用。
20	WUF	从 Stop 模式唤醒标志： 当检测到唤醒事件时由硬件置 1。具体时间由 WUS 位域来定义。由软件向 UART_ICR 寄存器的 PECF 位写 1，可以清除这个标志。如果 UART_CR3 寄存器的 WUF=1，会产生中断请求。 注： 1. 当 UESM 被清零，WUF 标志也会被清零。 2. WUF 中断只在 Stop 模式中有用。 3. 若 UART 不支持从 Stop 模式唤醒功能，该位为保留位并由硬件强制为零。

位	名称	描述
19	RWU	接收器从静默模式唤醒： 这个位表示 UART 处于静默模式时，当唤醒和静默模式切换时，由硬件清零和置 1。静默模式控制顺序（地址还是空闲）用 UART_CR1 寄存器的 WAKE 位来选择。如果选择由空闲信号唤醒，那这个位就只能由软件置 1 了，方法是向 UART_RQR 寄存器置 1。 0：接收器处于工作模式 1：接收器处于静默模式 注：若 UART 不支持从 Stop 模式唤醒功能，该位为保留位并由硬件强制为零。
18:17	RSV	保留位，固定读为 0
16	BUSY	忙标志： 由硬件置 1 和清零。当 RX 线在通讯时（成功检测到起始位），这个位被硬件置 1。接收结束后（不管成功与否）会由硬件清零。 0：UART 处于空闲（没接收） 1：正在接收数据
15:8	RSV	保留位，固定读为 0
7	TXE	发送数据寄存器空： 当 UART_TDR 寄存器中的值被取到移位寄存器的同时，这个位被硬件置 1。再向 UART_TDR 寄存器写数据就会同时清掉这个位。 如果 UART_CR1 寄存器中的 TXEIE 位被置起时，则会产生中断。 0：没有数据被传到移位寄存器 1：有数据被传到移位寄存器，发送数据寄存器为空。 注：这个位在单缓冲区发送的时候会用到。
6	TC	发送完成： 在 TXE 为 1 的条件下，数据发送完成时这个位会被硬件置 1。如果 UART_CR1 寄存器中的 TCIE 位是 1，就会产生中断请求。向 UART_TDR 寄存器再次写入数据，或者向 UART_ICR 寄存器的 TCCF 位写 1，都可以清除这个标志。如果 UART_CR1 寄存器中的 TCIE 位是 1，就会产生中断请求。 0: 发送未完成 1: 发送完成 注：如果 TE 位被清零，并且没有在发送数据，那 TC 位会立即被置 1。
5	RXNE	接收数据寄存器非空： 当接收移位寄存器的内容被传递到 UART_RDR 寄存器中时，这个位被硬件置 1。读取 UART_TDR 寄存器的数据就会同时清掉这个位。RXNE 标志也可以通过对 UART_RQR 寄存器中的 RXFRQ 位写 1 来清除。如果 UART_CR1 寄存器中的 RXNEIE 位是 1，就会产生中断请求。 0: 没收到数据 1: 收到的数据已经可读

位	名称	描述
4	IDLE	空闲线检测: 当检测到线路空闲时由硬件置 1。如果 UART_CR1 寄存器中的 IDLEIE 位是 1，就会产生中断请求。由软件向 UART_ICR 寄存器的 IDLECF 位写 1，可以清除这个标志。 0: 没有检测到线路空闲 1: 检测到线路空闲 注: 1. 除非 RXNE 位被置 1，否则 IDLE 位不会再次置 1（例如：一个新的线路空闲信号来临）。 2. 如果静默模式被使能了（MME=1），只要 UART 没有处于静默状态（RWU=0），那么无论在 WAKE 位上用什么方式设置静默功能，IDLE 就会被置 1。如果 RWU=1，那 IDLE 就不会被置 1 了。
3	ORE	溢出错误: 在 RXNE=1 的条件下（也就是上次数据还没有读走），串口接收寄存器又接收好了一个字节的数据并准备转移到 RDR 寄存器的时候，硬件会将这个位置 1；软件会向 UART_ICR 寄存器的 ORECF 位写 1，可以清除这个标志。 如果 UART_CR1 寄存器中的 RXNEIE 位或 EIE 位是 1，就会产生中断请求。 0：没有溢出错误 1：检测到溢出错误 注: 1. 当这个位被置 1，RDR 寄存器中的数据不会丢，但移位寄存器中的（那个新的）数据就会蒸发掉。如果在多缓冲区通讯时 EIE 位是 1，并且 ORE 标志被置 1 的话，就会同步引起一个中断请求。 2. 如果 UART_CR3 寄存器中的 OVRDIS 位是 1，那么这个位就会被长期强制为零（失去溢出检测功能）。
2	NF	噪声检测标志: 当一帧数据接收完成的时候检测到噪声，这个位会由硬件置 1；由软件向 UART_ICR 寄存器的 NFCF 位写 1，可以清除这个标志。 0：没有检测到噪声 1：检测到噪声 注: 1. 这一位不会产生中断，因为和它同时为 1 的 RXNE 能够产生中断。如果在多缓冲区通讯时 EIE 位是 1，并且 NF 标志被置 1 的话，也会同步引起一个中断请求。 2. 当线路不会受到噪声干扰，可以将这个噪声检测功能关闭，从而降低 UART 对时钟偏差的敏感度，具体做法是将 ONEBIT 位写成 1。（参见章节 19.3.5 UART 波特率对时钟偏差的容忍）

位	名称	描述
1	FE	帧错误： 当一个不同步现象、强噪声或一个断开符号被检测到的时候，这个位由硬件置 1；由软件向 UART_ICR 寄存器的 FECF 位写 1，可以清除这个标志。 如果 UART_CR1 寄存器中的 EIE 位是 1，会产生中断请求。 0：没有检测到帧错误 1：有检测到帧错误或者有收到断开字
0	PE	校验错误标志： 当在接收数据的时候发现校验错误，这个位会由硬件置 1；由软件向 UART_ICR 寄存器的 PECF 位写 1，可以清除这个标志。如果 UART_CR1 寄存器中的 PEIE 位是 1，会产生中断请求。 0：没有校验错误 1：有校验错误

19.6.7 中断标志清除寄存器（UART_ICR）

偏移地址：0x020

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									TCCF	Res.	IDLECF	ORECF	NCF	FECF	PECF
									w		w	w	w	w	w

位	名称	描述
31:21	RSV	保留位，固定读为 0
20	WUCF	从 Stop 模式唤醒的标志的清除： 对这个位写 1，会清除 UART_ISR 寄存器中的 WUF 标志位。 注：如果 UART 不支持从 Stop 模式唤醒功能，该位为保留位并由硬件强制为零。
19:7	RSV	保留位，固定读为 0
6	TCCF	发送完成标志的清除： 对这个位写 1，会清除 UART_ISR 寄存器中的 TC 标志
5	RSV	保留位，固定读为 0

位	名称	描述
4	IDLECF	线路空闲检测标志的清除： 对这个位写 1，会清除 UART_ISR 寄存器中的 IDLE 标志位。
3	ORECF	溢出错误标志的清除： 对这个位写 1，会清除 UART_ISR 寄存器中的 ORE 标志位。
2	NCF	噪声检测标志的清除： 对这个位写 1，会清除 UART_ISR 寄存器中的 NF 标志位。
1	FECF	帧错误标志的清除： 对这个位写 1，会清除 UART_ISR 寄存器中的 FE 标志位。
0	PECF	校验错误标志的清除： 对这个位写 1，会清除 UART_ISR 寄存器中的 PE 标志位。

19.6.8 数据接收寄存器（UART_RDR）

偏移地址：0x024

复位值：0x0000 0000

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

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

位	名称	描述
31:9	RSV	保留位，固定读为 0
8:0	RDR[8:0]	接收数据的值： 包含所收到的字节。 RDR 寄存器提供输入移位寄存器和内部总线间的并行接口（见图 19-1：UART 模块框图）。当接收数据时打开了校验位，读这个寄存器得到的最高位是校验位。

19.6.9 数据发送寄存器（UART_TDR）

偏移地址：0x028

复位值：0x0000 0000

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

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

位	名称	描述
31:9	RSV	保留位，固定读为 0
8:0	TDR[8:0]	发送数据的值： 用于写入要发送的数据字节。 TDR 寄存器提供发送移位寄存器和内部总线间的并行接口（见图 19-1：UART 模块框图）。 当发送的时候设置了校验功能（UART_CR1 中的 PCE=1），向最高位（位 7 还是位 8 取决于设置的字长）写入的信息是无效的，因为它总是要被校验位代替了之后再发送的。 注：这个寄存器只能在 TXE=1 的时候才能执行写操作。

20 模拟电压比较器（CMP）

20.1 概述

本产品内嵌 1 个正相 5 通道（3 外部+2 内部通道）切换的比较器（CMP1）和 1 个正相 3 通道（1 外部+2 内部通道）的比较器（CMP2），每个比较器都有反相端内部比较电压可选功能，施密特窗口档位选择，输出信号滤波及极性改变功能。同时最终的输出信号可以反馈到芯片管脚，内部中断同时也能接入 MCP、TIMx 与其产生联动，或是作为保护信号接入 MCP 的刹车功能。

20.2 特性

- 比较器 1 支持正相 5 通道选择
- 比较器 2 支持正相 3 通道选择
- 支持反相端多级电压可选
- 支持施密特窗口档位配置
- 支持输出滤波和极性控制
- 输出信号连接至 MCP_BKIN 等信号进行保护控制（PWM 刹车）

20.3 功能描述

2 路比较器，以下为单个比较器的系统框图，CMP_INTERRUPT 由 ARM 外部中断单元统一控制，CMPx_OUT 引脚由 GPIO 复用功能单元统一控制。

注：在电机控制应用中，建议比较器 1 作为三相反电动势比较，比较器 2 作为过流保护比较。

BEMF_MID 信号是由比较器的正相输入引脚 CMP1_INP1、CMP1_INP2 和 CMP1_INP3 三个信号平均所得。三个电阻值均为 10K，S1、S2 和 S3 三个开关只有当比较器反相输出端选择 BEFM_MID 信号时才会导通，其他情况处于断开的状态。

此功能主要应用于 BLDC 方波控制时，三相反电动势经过分压后分别接到 CMP1_INP1、CMP1_INP2 和 CMP1_INP3，比较器反相端选择 BEMF_MID，开关导通，即比较器的反相端为虚拟中心点。正相端通过分时复用选择三相反电动势，通过比较器输出得到过零点。

内部参考电压 ANAVREF 是内部提供给模拟模块使用的参考基准，典型值为 2V。使用前需要将比较器模块中的 CMP_CPANA 寄存器中的 ANAVREFEN 位设置为 1，并等待 100μs 使得 ANAVREF 建立完成。

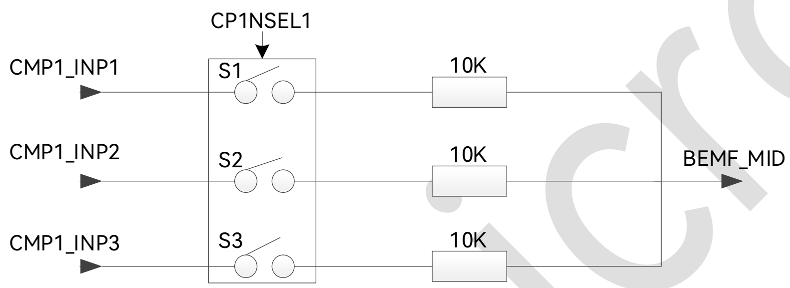


图 20-1：三相反电动势

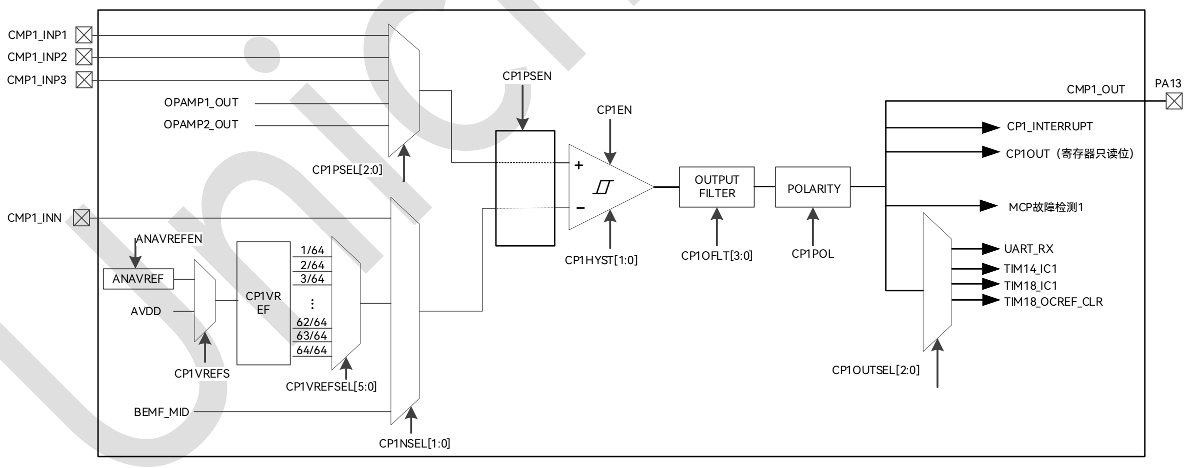


图 20-2：比较器 1 模块框图

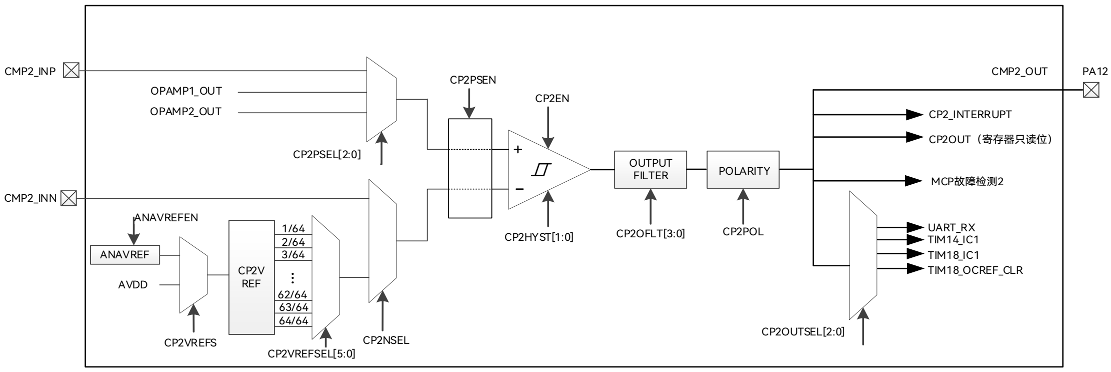


图 20-3：比较器 2 模块框图

20.3.1 比较器开窗控制功能

当设置比较器控制寄存器 CPxCR 中的 CPxWINEN 位为 1 时，表示开启了比较器的开窗比较功能。比较器会根据 MCP 的通道输出电平来决定是否执行比较功能。当 CPxWINSEL 位为 0 时，在 MCP 的通道输出为有效电平时，比较器执行比较功能，而其他时间则保持原输出不变但不执行比较功能。当 CPxWINSEL 位为 1 时，在 MCP 的通道输出为无效电平时，比较器执行比较功能，而其他时间则保持原输出不变但不执行比较功能。

20.3.2 比较器时钟及唤醒 STOP 模式

比较器 1 和比较器 2 的时钟由 APB1 clock 提供，而比较器的输出滤波时钟固定由 HIS（18MHz）提供。当配置完比较器的参数后，关闭比较器时钟，不影响比较器的比较功能。如果比较器的输出无滤波，即使在没有时钟情况下，比较器仍然可以正常工作。

比较器的输出可以将 MCU 从 STOP 模式下唤醒。

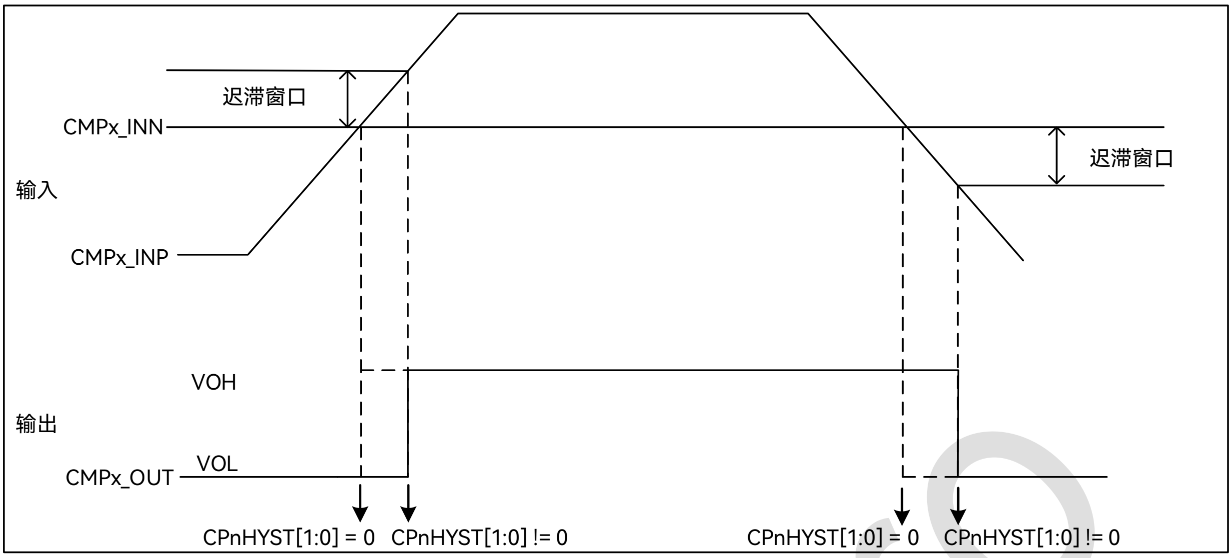


图 20-4：比较器施密特窗口及输出电压关系

20.3.3 引脚映射

表 20-1：比较器引脚映射

实际引脚	比较器 2 对应模拟端口	复用类型
PA10	CMP2_INP	模拟复用功能
PA15	CMP2_INN	模拟复用功能
PA12	CMP2_OUT	数字复用功能 AF6
实际引脚	**比较器 1 对应模拟端口**	**复用类型**
PA7	CMP1_INP1	模拟复用功能
PA8	CMP1_INP2	模拟复用功能
PA9	CMP1_INP3	模拟复用功能
PA0	CMP1_INN	模拟复用功能
PA13	CMP1_OUT	数字复用功能 AF6

20.4 相关寄存器

20.4.1 比较器 1 控制寄存器（CMP_CP1CR）

偏移地址：0x000

复位值：0x003E 0000

31	30	29 28 27	26	25	24	23	22 21 20 19 18 17	16
CP1LCK	CP1OUT	Res.	CP1PSEN	CP1WINSEL	CP1WINEN	Res.	CP1VREFSEL[5:0]	Res.
rw	r		rw	rw	rw		rw	

15	14 13 12 11	10 9 8	7 6	5 4 3	2 1	0
CP1POL	CP1OFLT[3:0]	CP1OUTSEL[2:0]	CP1NSEL[1:0]	CP1PSEL[2:0]	CP1HYST[1:0]	CP1EN
rw	rw	rw	rw	rw	rw	rw

位	名称	描述
31	CP1LCK	比较器 1 寄存器锁定控制位： 该位软件只能置“1”不能清“0”，只有系统复位后才能清“0”。当置“1”时令比较器 1 所有控制位变为只读。 0：比较器 1 寄存器的所有控制位维持自己的读写属性（默认） 1：比较器 1 寄存器的所有控制位为只读 注：当 CP1LCK 为 1 时，CP1CR 和 CP1CAL 寄存器均不能修改。当系统复位或对 RCC_APB2RSTR 中的 CMPRST 位先写 1 再写 0 可以复位 CP1LCK 位。
30	CP1OUT	比较器 1 数字输出值： 该位只读，反映应比较器 1 输出状态。 0：输出低（默认） 1：输出高
29:26	RSV	保留位，固定读为 0
25	CP1WINSEL	比较器 1 开窗控制选择： 0：当 MCP 的 CH1、CH2、CH3 中任何通道输出高电平时比较器开启比较功能 1：当 MCP 的 CH1、CH2、CH3 中任何通道输出低电平时比较器开启比较功能
24	CP1WINEN	比较器 1 开窗控制使能位： 0：比较器开窗控制功能禁止 1：比较器开窗控制功能使能

位	名称	描述
23	RSV	保留位，固定读为 0
22:17	CP1VREFSEL[5:0]	比较器 1 内部参考电压选择位： 这几位控制比较器 1 参考电压的档位选择。 000000：1/64*参考电压 CP1VREF 000001：2/64*参考电压 CP1VREF 000010：3/64*参考电压 CP1VREF 000011：4/64*参考电压 CP1VREF …… 011111：32/64*参考电压 CP1VREF（默认） …… 111100：61/64*参考电压 CP1VREF 111101：62/64*参考电压 CP1VREF 111110：63/64*参考电压 CP1VREF 111111：64/64*参考电压 CP1VREF
16	RSV	保留位，固定读为 0
15	CP1POL	CP1POL –比较器 1 输出极性控制位： 该位控制比较器 1 的输出极性。 0：保持输出（默认） 1：取反输出
14:11	CP1OFLT[3:0]	比较器 1 输出滤波周期选择位： 这几位控制比较器 1 的输出滤波，达到连续的模块时钟周期后比较器输出不变则认为是有效的结果，否则保持不变，使用芯片的系统时钟。 0000：无滤波 0001：0.2μs 0010：0.5μs 0011：1μs 0100：2μs 0101：4μs 0110：6μs 0111：8μs 1000：16μs 1001：32μs 其他：32μs 注：滤波时钟固定使用 HIS（18MHz）。
10:8	CP1OUTSEL[2:0]	比较器 1 内部输出功能选择位： 这些位用来选择比较器 1 输出方向对应的功能。

位	名称	描述
		000：不输出 001：UART 的 RX 引脚 010：定时器 14 输入捕获 1 通道（TIM14_IC1） 011：定时器 18 输入捕获 1 通道（TIM18_IC1） 100：定时器 18 OCREF CLEAR 输入（TIM18_OCREF_CLR） 其他：保留 注：此位与“CP2CR”中的“CP2OUTSEL[2:0]”选择一致时，优先 CP1CR 的功能效果。
7:6	CP1NSEL[1:0]	比较器 1 反相输入选择位: 该位用于选择连接到比较器 1 反相输入的信号源。 00：CMP1_INN 引脚连接到比较器 1 的反相输入端（默认） 01：内部参考电压连接到比较器 1 的反相输入端 1x：BEMF_MID 信号连接到比较器 1 的反相输入端
5:3	CP1PSEL[2:0]	比较器 1 正相输入选择位: 这几位用于选择连接到比较器 1 正相输入的信号源。 000：CMP1_INP1 比较器外部正相管脚 1（默认） 001：CMP1_INP2 比较器外部正相管脚 2 010：CMP1_INP3 比较器外部正相管脚 3 011：OPAMP1_OUT 运算放大器 1 输出电压 100：OPAMP2_OUT 运算放大器 2 输出电压 其他：保留
2:1	CP1HYST[1:0]	比较器 1 迟滞功能选择位: 这几位控制比较器 1 的迟滞电压。 00：0mV（默认） 01：15mV 10：30mV 11：90mV
0	CP1EN	比较器 1 模块工作使能位: 该位控制比较器 1 模块的打开/关闭。 0：关闭比较器 1（默认） 1：打开比较器 1 注：此位与“CP1CAL”中的“CP1CALEN”逻辑互斥，不能同时为“1”；即当“CP1CALEN”为“1”此时对“CP1EN”写“1”无效，当“CP1EN”为“1”此时对“CP1CALEN”写“1”无效

20.4.2 比较器 2 控制寄存器（CMP_CP2CR）

偏移地址：0x004

复位值：0x003E 0000

31	30	29 28 27	26	25	24	23	22 21 20 19 18 17	16
CP2LCK	CP2OUT	Res.	CP2PSEN	CP2WINSEL	CP2WINEN	Res.	CP2VREFSEL[5:0]	Res.
rw	ro		rw	rw	rw		rw	

15	14 13 12 11	10 9 8	7	6	5 4 3	2 1	0
CP2POL	CP2OFLT[3:0]	CP2OUTSEL[2:0]	Res.	CP2NSEL	CP2PSEL[2:0]	CP2HYST[1:0]	CP2EN
rw	rw	rw		rw	rw	rw	rw

位	名称	描述
31	CP2LCK	比较器 2 寄存器锁定控制位： 该位软件只能置“1”不能清“0”，只有系统复位后才能清“0”。当置“1”时令比较器 2 所有控制位变为只读。 0：比较器 2 寄存器的所有控制位维持自己的读写属性（默认） 1：比较器 2 寄存器的所有控制位为只读 注：当 CP2LCK 为 1 时，CP2CR 和 CP2CAL 寄存器均不能修改。当系统复位或对 RCC_APB2RSTR 中的 CMPRST 位先写 1 再写 0 可以复位 CP2LCK 位。
30	CP2OUT	比较器 2 数字输出值： 该位只读，反映应比较器 2 输出状态。 0：输出低（默认） 1：输出高
29:26	RSV	保留位，固定读为 0
25	CP2WINSEL	比较器 2 开窗控制选择： 0：当 MCP 的 CH1、CH2、CH3 中任何通道输出高电平时比较器开启比较功能 1：当 MCP 的 CH1、CH2、CH3 中任何通道输出低电平时比较器开启比较功能
24	CP2WINEN	比较器 2 开窗控制使能位： 0：比较器开窗控制功能禁止 1：比较器开窗控制功能使能
23	RSV	保留位，固定读为 0
22:17	CP2VREFSEL[5:0]	比较器 2 内部参考电压选择位： 这几位控制比较器 2 参考电压的档位选择。

位	名称	描述
		000000：1/64*参考电压 CP2VREF 000001：2/64*参考电压 CP2VREF 000010：3/64*参考电压 CP2VREF 000011：4/64*参考电压 CP2VREF …… 011111：32/64*参考电压 CP2VREF（默认） …… 111100：61/64*参考电压 CP2VREF 111101：62/64*参考电压 CP2VREF 111110：63/64*参考电压 CP2VREF 111111：64/64*参考电压 CP2VREF
16	RSV	保留位，固定读为 0
15	CP2POL	CP2POL – 比较器 2 输出极性控制位: 该位控制比较器 2 的输出极性。 0: 保持输出（默认） 1: 取反输出
14:11	CP2OFLT[3:0]	比较器 2 输出滤波周期选择位: 这几位控制比较器 2 的输出滤波，达到连续的模块时钟周期后比较器输出不变则认为是有效的结果，否则保持不变，使用芯片的系统时钟。 0000：无滤波 0001：0.2µs 0010：0.5µs 0011：1µs 0100：2µs 0101：4µs 0110：6µs 0111：8µs 1000：16µs 1001：32µs 其他：32µs 注：滤波时钟固定使用 HSI（18MHz）
10:8	CP2OUTSEL[2:0]	比较器 2 内部输出功能选择位: 这些位用来选择比较器 2 输出方向对应的功能。 000：不输出 001：UART 的 RX 引脚 010：定时器 14 输入捕获 1 通道（TIM14_IC1） 011：定时器 18 输入捕获 1 通道（TIM18_IC1）

位	名称	描述
		100：定时器 18 OCREF CLEAR 输入（TIM18_OCREF_CLR） 其他：保留 注：此位与“CP1CR”中的“CP1OUTSEL[2:0]”选择一致时，优先 CP1CR 的功能效果
7	RSV	保留位，固定读为 0
6	CP2NSEL	比较器 2 反相输入选择位: 该位用于选择连接到比较器 2 反相输入的信号源。 0：CMP2_INN 比较器外部反相管脚（默认） 1：内部参考电压
5:3	CP2PSEL[2:0]	比较器 2 正相输入选择位: 这几位用于选择连接到比较器 2 正相输入的信号源。 000：CMP2_INP 比较器外部正相管脚（默认） 001：保留 010：保留 011：OPAMP1_OUT 运算放大器 1 输出电压 100：OPAMP2_OUT 运算放大器 2 输出电压 其他：保留
2:1	CP2HYST[1:0]	比较器 2 迟滞功能选择位: 这几位控制比较器 2 的迟滞电压。 00：0mV（默认） 01：15mV 10：30mV 11：90mV
0	CP2EN	比较器 2 模块工作使能位: 该位控制比较器 2 模块是打开/关闭。 0：关闭比较器 2（默认） 1：打开比较器 2 注：此位与“CP2CAL”中的“CP2CALEN”逻辑互斥，不能同时为“1”；即当“CP2CALEN”为“1”此时对“CP2EN”写“1”无效，当“CP2EN”为“1”此时对“CP2CALEN”写“1”无效。

20.4.3 比较器模拟控制寄存器（CMP_CPANA）

偏移地址：0x008

复位值：0x0000 0300

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

15	14	13	12	11	10 9 8	7 6 5 4 3	2	1	0
Res.	OUT3VREFEN	OUT2VREFEN	OUT1VREFEN	OUTVREFS	OUTVREFSEL[2:0]	Res.	CP2VREFS	CP1VREFS	ANAVREFEN
	rw	rw	rw	rw	rw		rw	rw	rw

位	名称	描述
31:3	RSV	保留位，固定读为 0
2	CP2VREFS	比较器 2 参考电压选择位: 该位控制比较器 2 的参考电压的来源（VREF=2V，VDDA 为模拟电源上的输入电压）。 0：选择外部 AVDD 作为参考电压 CRV（默认） 1：选择内部 VREF 作为参考电压 CRV
1	CP1VREFS	比较器 1 参考电压选择位: 该位控制比较器 1 的参考电压的来源（VREF=2V，VDDA 为模拟电源上的输入电压）。 0：选择外部 AVDD 作为参考电压 CRV（默认） 1：选择内部 VREF 作为参考电压 CRV
0	ANAVREFEN	内部模拟参考电压使能位: 该位控制内部模拟参考电压是打开/关闭，打开参考电压后需要等待 100μs 等待模拟信号建立稳定。 0：关闭内部参考电压 ANAVREF（默认） 1：打开内部参考电压 ANAVREF

20.4.4 比较器 1 校准寄存器（CMP_CP1CAL）

偏移地址：0x00C

复位值：0x0000 2020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP1CALEN	CP1SYNC	Res.													
rw	r														

15 14	13 12 11 10 9 8	7 6	5 4 3 2 1 0
Res.	CP1CALDATP[5:0]	Res.	CP1CALDATN[5:0]
	rw		rw

位	名称	描述
31	CP1CALEN	比较器 1 自动校准功能使能位: 该位控制比较器 1 自动校准功能。由软件置 1 时自动启动校准，表示比较器 1 正在校准中; 完成时硬件自动清 0 表示校准完成并停止。 0: 比较器 1 自动校准功能停止/完成（默认） 1: 比较器 1 自动校准功能启动/进行中 注 1: 此位与 CPCR 中的 CPAMP1EN 逻辑互斥，不能同时为 1; 即当 CP1CALEN 为 1 时对 CPAMP1EN 写 1 无效，当 CPAMP1EN 为 1 时对 CP1CALEN 写 1 无效。 注 2: 校准时需要打开 HSI 或是 HSE 的时钟, 校准时间用时约 3ms（按照 8MHz HSI 时钟）。
30	CP1SYNC	比较器 1 校准值同步状态标志位: 该位表示比较器 1 校准值的同步标志。当对 CP1CALDATP 或 CP1CALDATN 写入数值后，由硬件自动置 1 时，表示比较器 1 正在将寄存器数值同步到内部电路中; 当完成时硬件自动清 0 表示同步完成。 0: 比较器 1 校准值完成 1: 比较器 1 校准值正在同步中
29:14	RSV	保留位，固定读为 0
13:8	CP1CALDATP[5:0]	比较器 1 正端校验结果数据值: 这几位用于显示自动校验结果，用户可根据数据内容进行微调。 BIT5 代表符号位: 0: 表示增加电压 1: 表示减少电压 BIT[4:0]代表数据档位: 1: LSB 表示一格模拟设计电压档位（1LSB 约 0.5mV） 注: 只有 CP1CALEN 为 0 时，可以对此区域的内容进行修改; 当 CP1CALEN 为 1 时，此区域无法写入任何内容。
7:6	RSV	保留位，固定读为 0
5:0	CP1CALDATN[5:0]	比较器 1 负端校验结果数据值: 这几位用于显示自动校验结果，用户可根据数据内容进行微调。 BIT5 代表符号位: 0: 表示增加电压 1: 表示减少电压 BIT[4:0]代表数据档位: 1: LSB 表示一格模拟设计电压档位（1LSB 约 0.5mV） 注: 只有 CP1CALEN 为 0 时，可以对此区域的内容进行修改; 当 CP1CALEN 为 1 时，此区域无法写入任何内容。

20.4.5 比较器 2 校准寄存器（CMP_CP2CAL）

偏移地址：0x010

复位值：0x0000 2020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CP2CALEN	CP2SYNC	Res.													
rw	r														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		CP2CALDATP[5:0]						Res.		CP2CALDATN[5:0]					
		rw								rw					

位	名称	描述
31	CP2CALEN	比较器 2 自动校准功能使能位: 该位控制比较器 2 自动校准功能。由软件置“1”时自动启动校准，同时表示比较器 2 正在校准中; 当完成时硬件自动清“0”表示校准完成并停止。 0：比较器 2 自动校准功能停止/完成（默认） 1：比较器 2 自动校准功能启动/进行中 注 1：此位与 CPCR 中的 CPAMP2EN 逻辑互斥，不能同时为 1；即当 CP2CALEN 为 1 时对 CPAMP2EN 写 1 无效，当 CPAMP2EN 为 1 时对 CP2CALEN 写 1 无效。 注 2：校准时需要打开 HSI 或是 HSE 的时钟，校准时间用时约 3ms（按照 8MHz HSI 时钟）
30	CP2SYNC	比较器 2 校准值同步状态标志位: 该位表示比较器 2 校准值的同步标志。当对 CP2CALDATP 或 CP2CALDATN 写入数值后，由硬件自动置 1 时，此时表示比较器 1 正在将寄存器数值同步到内部电路中；当完成时硬件自动清 0 表示同步完成。 0：比较器 2 校准值完成 1：比较器 2 校准值正在同步中
29:14	RSV	保留位，固定读为 0
13:8	CP2CALDATP[5:0]	比较器 2 正端校验结果数据值: 这几位用于显示自动校验结果，用户可根据数据内容进行微调。 BIT5 代表符号位: 0：表示增加电压 1：表示减少电压

位	名称	描述
		BIT[4:0]代表数据档位: 1：LSB 表示一格模拟设计电压档位（1LSB 约 0.5mV） 注：只有 CP2CALEN 为 0 时，可以对此区域的内容进行修改；当 CP2CALEN 为 1 时，此区域无法写入任何内容。
7:6	RSV	保留位，固定读为 0
5:0	CP2CALDATN[5:0]	比较器 2 负端校验结果数据值: 这几位用于显示自动校验结果，用户可根据数据内容进行微调。 BIT5 代表符号位: 0：表示增加电压 1：表示减少电压 BIT[4:0]代表数据档位: 1：LSB 表示一格模拟设计电压档位（1LSB 约 0.5mV） 注：只有 CP2CALEN 为 0 时，可以对此区域的内容进行修改；当 CP2CALEN 为 1 时，此区域无法写入任何内容。

21 运算放大器（OPAMP）

21.1 概述

本产品内嵌 2 个高性能运算放大器（OPAMP1、2），每个运算放大器都有内部固定倍数放大功能，施密特窗口档位选择（2、4、8、16、32 倍），反相端与输出口可以灵活选择接线方式，配合外部电路达到一些应用效果。输出信号可以接入 ADC 端口进行数据采样，也可以接入比较正端作为比较源头，同时也能输出到芯片管脚作为其它使用。

21.2 特性

- 内嵌 2 路运算放大器
- 支持内部固定放大倍数选择
- 支持施密特窗口档位选择
- 反相输入通道灵活选择
- 输出可接入 ADC 输入通道和比较器正相输入通道

21.3 功能描述

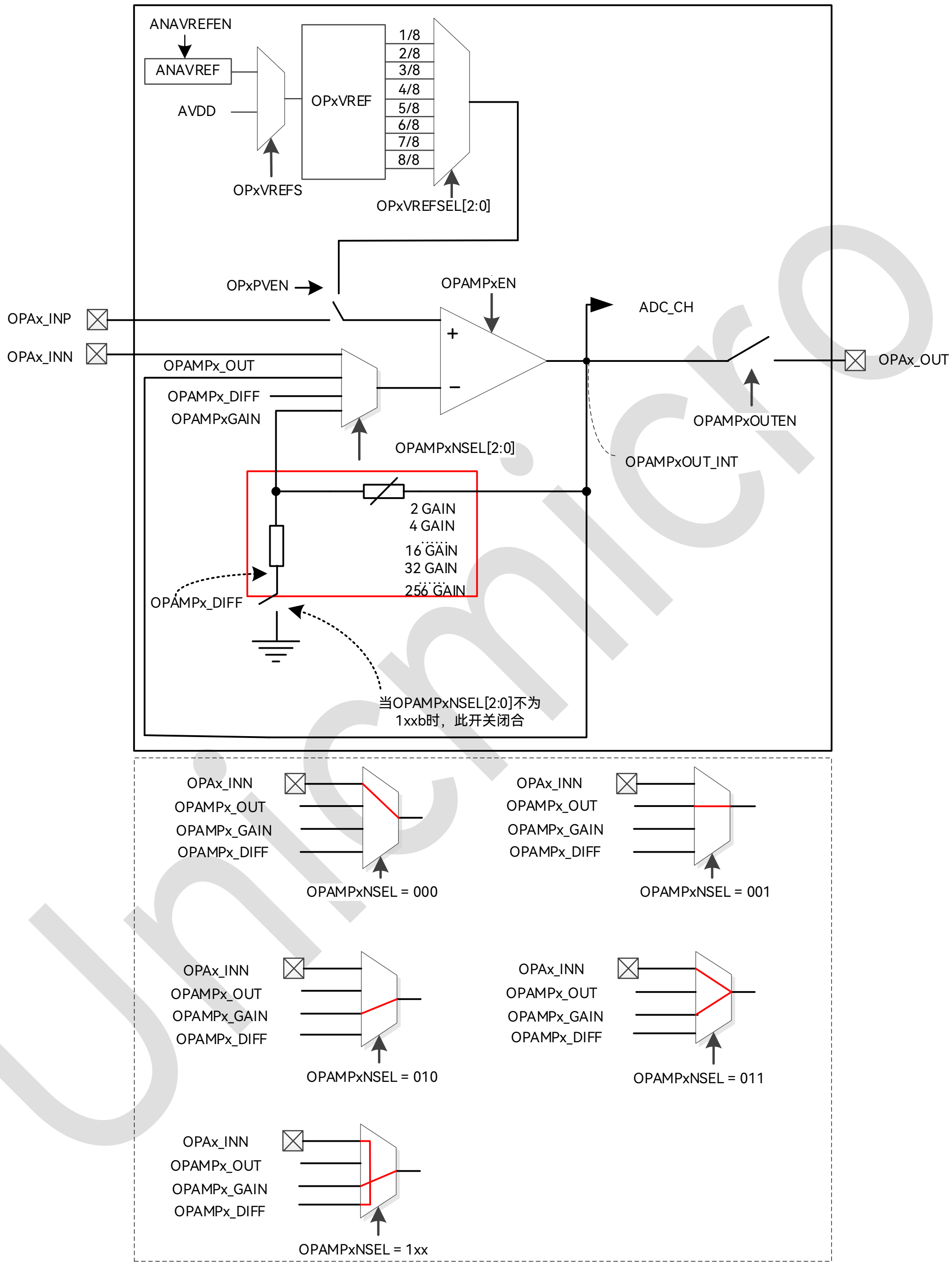


图 21-1：运算放大器模块框图

运放的正向端可以加偏置电压，通过 OPAMP_OPCR 寄存器中的 OPxPVEN 位选择是使能还是禁止正向偏置电压。偏置电压的档位可以通过 OPxVREFSEL [2:0]进行设置，偏置电压的输入源可以

通过 OPxVREFS 选择 AVDD 或者是 ANAVREF。 其中内部参考电压 ANAVREF 是内部提供给模拟模块使用的参考基准，典型值为 2V。使用前需要将比较器模块中的 CMP_CPANA 寄存器中的 ANAVREFEN 位设置为 1，并等待 100μs 使得 ANAVREF 建立完成。

注：OPAMPxNSEL=11B 模式保留，不推荐用户使用。

21.3.1 引脚映射

表 21-1：运算放大器引脚映射

实际引脚	运算放大器 2 对应模拟端口	复用类型
PA10	OPA2_INP	模拟复用功能
PA11	OPA2_INN	模拟复用功能
PA12	OPA2_OUT	模拟复用功能
实际引脚	**运算放大器 1 对应模拟端口**	**复用类型**
PA7	OPA1_INP	模拟复用功能
PA8	OPA1_INN	模拟复用功能
PA9	OPA1_OUT	模拟复用功能

21.3.2 与 ADC 模块连接

运算放大器的输出信号需要内部对接 ADC 通道进行采样。

ADC 内部输入通道 18~19，分别对应 OPAMP1~2，关系如下：

- ADC channel 18 -> OPAMP1 内部模拟输出
- ADC channel 19 -> OPAMP2 内部模拟输出

通过 ADC_CFGR1 的控制位“AWDCH[4:0]”选择对应的 OPAMPx 通道：

- 10010：由模拟看门狗监控的 ADC 模拟输入通道 18
- 10011：由模拟看门狗监控的 ADC 模拟输入通道 19

OPAMPx 对应的通道是否被转换，可能通过 ADC_SEQCHSELR1 和 ADC_SEQCHSELR2 寄存器中的 SEQCHx[4:0]位段进行设置，当 SEQCHx[4:0]设置为 10010b 或 10011b 时，表示选择 OPAMP1 或 OPAMP2 对应的通道进行转换，转换结果存储在对应的结果寄存器 ADC_DRx 中。

详细 ADC 配置请参见 ADC 章节。

21.4 相关寄存器

21.4.1 运算放大器控制寄存器（OPAMP_OPCR）

偏移地址：0x000

复位值：0x0300 0300

31	30 29	28	27	26 25 24	23	22 21 20	19 18 17	16
OPAMPLOCK	Res.	OP2PVEN	OP2VREFS	OP2VREFSEL[2:0]	OPAMP2OUTEN	OPAMP2GAIN[2:0]	OPAMP2NSEL[2:0]	OPAMP2EN
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.	OP1PVEN	OP1VREFS	OP1VREFSEL[2:0]	OPAMP1OUTEN	OPAMP1GAIN[2:0]	OPAMP1NSEL[2:0]	OPAMP1EN
	rw	rw	rw	rw	rw	rw	rw

位	名称	描述
31	OPAMPLOCK	运算放大器寄存器锁定控制位： 该位软件只能置 1 不能清 0，只有系统复位后才能清 0。当置 1 时令所有运算放大器控制位变为只读。 0：所有运算放大器寄存器的控制位维持自己的读写属性（默认） 1：所有运算放大器寄存器的控制位为只读
30:29	RSV	保留位，固定读为 0
28	OP2PVEN	运算放大器 2 正向端偏置电压使能位： 0：运算放大器 2 正向端偏置电压禁止 1：运算放大器 2 正向端偏置电压使能
27	OP2VREFS	运算放大器 2 参考电压选择位： 该位控制运算放大器 2 的参考电压的来源（ANAVREF=2V，VDDA 为模拟电源上的输入电压）。 0：选择外部 AVDD 作为参考电压 OP2VREF（默认） 1：选择内部 ANAVREF 作为参考电压 OP2VREF
26:24	OP2VREFSEL[2:0]	运算放大器 2 正向端偏置电压选择位： 这几位控制运算放大器 2 正向端偏置电压的档位选择。 000：1/8*参考电压 OP2VREF 001：2/8*参考电压 OP2VREF 010：3/8*参考电压 OP2VREF

位	名称	描述
		011：4/8*参考电压 OP2VREF 100：5/8*参考电压 OP2VREF 101：6/8*参考电压 OP2VREF 110：7/8*参考电压 OP2VREF 111：8/8*参考电压 OP2VREF
23	OPAMP2OUTEN	运算放大器 2 输出端口使能位: 该位控制运算放大器 2 的端口输出是否有效。 0: 运算放大器 2 的输出端口不连接到 GPIO 1: 运算放大器 2 的输出端口连接到 GPIO 注 1：运算放大器的输出端口是否连接到 GPIO，不仅使能该位，同时对应 GPIO 还要选择为“模拟功能”。 注 2：使用者请注意，如果对应其它模拟功能也打开（如：ADC 通道），该管脚的 GPIO 会同时具备运放输出功能和 ADC 采样功能。
22:20	OPAMP2GAIN[2:0]	运算放大器 2 增益倍数选择位: 这几位控制运算放大器 2 的增益倍数。 000：2 GAIN（默认） 001：4 GAIN 010：8 GAIN 011：16 GAIN 100：32 GAIN 其他值：保留
19:17	OPAMP2NSEL[2:0]	运算放大器 2 反相输入选择位: 这几位用于选择连接到运算放大器 2 正相输入的信号源。 000：OPAMP2_INN 运算放大器 2 反相输入（默认） 001：OPAMP2_OUT 运算放大器 2 输出 010：OPAMP2 的 GAIN 增益倍数 011：OPAMP2_INN 运算放大器 2 反相输入与 GAIN 增益倍数短接再输入 1xx：保留
16	OPAMP2EN	运算放大器 2 模块工作使能位: 该位控制运算放大器 2 模块是打开/关闭。 0：关闭运算放大器 2（默认） 1：打开运算放大器 2 注：此位与 OP2CAL 中的 OP2CALEN 逻辑互斥，不能同时为 1；即当 OP2CALEN 为 1 时对 OPAMP2EN 写 1 无效，当 OPAMP2EN 为 1 时对 OP2CALEN 写 1 无效。
15:13	RSV	保留位，固定读为 0

位	名称	描述
12	OP1PVEN	运算放大器 1 正向端偏置电压使能位： 0：运算放大器 1 正向端偏置电压禁止 1：运算放大器 1 正向端偏置电压使能
11	OP1VREFS	运算放大器 1 参考电压选择位： 该位控制运算放大器 1 的参考电压的来源（ANAVREF=2V，VDDA 为模拟电源上的输入电压）。 0：选择外部 AVDD 作为参考电压 OP1VREF（默认） 1：选择内部 ANAVREF 作为参考电压 OP1VREF
10:8	OP1VREFSEL[2:0]	运算放大器 1 正向端偏置电压选择位： 这几位控制运算放大器 1 正向端偏置电压的档位选择。 000：1/8*参考电压 OP1VREF 001：2/8*参考电压 OP1VREF 010：3/8*参考电压 OP1VREF 011：4/8*参考电压 OP1VREF 100：5/8*参考电压 OP1VREF 101：6/8*参考电压 OP1VREF 110：7/8*参考电压 OP1VREF 111：8/8*参考电压 OP1VREF
7	OPAMP1OUTEN	运算放大器 1 输出端口使能位： 该位控制运算放大器 1 的端口输出是否有效。 0：运算放大器 1 的输出端口不连接到 GPIO 1：运算放大器 1 的输出端口连接到 GPIO 注 1：运算放大器的输出端口是否连接到 GPIO，不仅使能该位，同时对应 GPIO 还要选择为“模拟功能”。 注 2：使用者请注意，如果对应其它模拟功能也打开（如：ADC 通道），该管脚的 GPIO 会同时具备运放输出功能和 ADC 采样功能。
6:4	OPAMP1GAIN[2:0]	运算放大器 1 增益倍数选择位： 这几位控制运算放大器 1 的增益倍数。 000：2 GAIN（默认） 001：4 GAIN 010：8 GAIN 011：16 GAIN 100：32 GAIN 101：64 GAIN 110：128 GAIN 111：256 GAIN
3:1	OPAMP1NSEL[2:0]	运算放大器 1 反相输入选择位：

位	名称	描述
		这几位用于选择连接到运算放大器 1 正相输入的信号源。 000：OPAMP1_INN 运算放大器 1 反相输入（默认） 001：OPAMP1_OUT 运算放大器 1 输出 010：OPAMP1 的 GAIN 增益倍数 011：OPAMP1_INN 运算放大器 1 反相输入与 GAIN 增益倍数短接再输入 1xx：OPAMP1_DIFF 与 OPAMP1_INN 短接,GAIN 增益倍数输入到反相输入端
0	OPAMP1EN	运算放大器 1 模块工作使能位: 该位控制运算放大器 1 模块是打开/关闭。 0：关闭运算放大器 1（默认） 1：打开运算放大器 1 注：此位与 OP1CAL 中的 OP1CALEN 逻辑互斥，不能同时为 1；即当 OP1CALEN 为 1 时对 OPAMP1EN 写 1 无效，当 OPAMP1EN 为 1 时对 OP1CALEN 写 1 无效。

21.4.2 运算放大器 1 校准寄存器（OPAMP_OP1CAL）

偏移地址：0x04

复位值：0x0000 2020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OP1CALEN	OP1SYNC	Res.													
rw	r														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		OP1CALDATP[5:0]						Res.		OP1CALDATN[5:0]					
		rw								rw					

位	名称	描述
31	OP1CALEN	运算放大器 1 自动校准功能使能位: 该位控制运算放大器 1 自动校准功能。由软件置 1 时自动启动校准，同时表示运算放大器 1 正在校准中；当完成时硬件自动清 0 表示校准完成并停止。 0：运算放大器 1 自动校准功能停止/完成（默认）

位	名称	描述
		1：运算放大器 1 自动校准功能启动/进行中 注 1：此位与 OPCR 中的 OPAMP1EN 逻辑互斥，不能同时为 1；即当 OP1CALEN 为 1 时对 OPAMP1EN 写 1 无效，当 OPAMP1EN 为 1 时对 OP1CALEN 写 1 无效。 注 2：校准时需要打开 HSI 或是 HSE 的时钟，校准时间用时约 3ms（按照 8MHz HSI 时钟）。
30	OP1SYNC	运算放大器 1 校准值同步状态标志位： 该位表示运算放大器 1 校准值的同步标志。当对 OP1CALDATP 或 OP1CALDATN 写入数值后，由硬件自动置 1 时，此时表示运算放大器 1 正在将寄存器数值同步到内部电路中；当完成时硬件自动清 0 表示同步完成。 0：运算放大器 1 校准值完成 1：运算放大器 1 校准值正在同步中
29:14	RSV	保留位，固定读为 0
13:8	OP1CALDATP[5:0]	运算放大器 1 正端校验结果数据值： 这几位用于显示自动校验结果，用户可根据数据内容进行微调。 BIT5 代表符号位： 0：表示增加电压 1：表示减少电压 BIT[4:0] 代表数据档位： 1：LSB 表示一格模拟设计电压档位 注：只有 OP1CALEN 为 0 时，可以对此区域的内容进行修改；当 OP1CALEN 为 1 时，此区域无法写入任何内容。
7:6	RSV	保留位，固定读为 0
5:0	OP1CALDATN[5:0]	运算放大器 1 负端校验结果数据值： 这几位用于显示自动校验结果，用户可根据数据内容进行微调。 BIT5 代表符号位： 0：表示增加电压 1：表示减少电压 BIT[4:0] 代表数据档位： 1：LSB 表示一格模拟设计电压档位 注：只有 OP1CALEN 为 0 时，可以对此区域的内容进行修改；当 OP1CALEN 为 1 时，此区域无法写入任何内容。

21.4.3 运算放大器 2 校准寄存器（OPAMP_OP2CAL）

偏移地址：0x08

复位值：0x0000 2020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OP2CALEN	OP2SYNC	Res.													
rw	r														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		OP2CALDATP[5:0]						Res.		OP2CALDATN[5:0]					
		rw								rw					

位	名称	描述
31	OP2CALEN	运算放大器 2 自动校准功能使能位： 该位控制运算放大器 2 自动校准功能。由软件置 1 时自动启动校准，同时表示运算放大器 2 正在校准中；当完成时硬件自动清 0 表示校准完成并停止。 0：运算放大器 2 自动校准功能停止/完成（默认） 1：运算放大器 2 自动校准功能启动/进行中 注 1：此位与 OPCR 中的 OPAMP2EN 逻辑互斥，不能同时为 1；即当 OP2CALEN 为 1 时对 OPAMP2EN 写 1 无效，当 OPAMP2EN 为 1 时对 OP2CALEN 写 1 无效。 注 2：校准时需要打开 HSI 或是 HSE 的时钟，校准时间用时约 3ms（按照 8MHz HSI 时钟）。
30	OP2SYNC	运算放大器 2 校准值同步状态标志位： 该位表示运算放大器 2 校准值的同步标志。当对 OP2CALDATP 或 OP2CALDATN 写入数值后，由硬件自动置 1 时，此时表示运算放大器 1 正在将寄存器数值同步到内部电路中；当完成时硬件自动清 0 表示同步完成。 0：运算放大器 2 校准值完成 1：运算放大器 2 校准值正在同步中
29:14	RSV	保留位，固定读为 0
13:8	OP2CALDATP[5:0]	运算放大器 2 正端校验结果数据值： 这几位用于显示自动校验结果，用户可根据数据内容进行微调。 BIT5 代表符号位： 0：表示增加电压 1：表示减少电压

位	名称	描述
		BIT[4:0]代表数据档位： 1：LSB 表示一格模拟设计电压档位 注：只有 OP2CALEN 为 0 时，可以对此区域的内容进行修改； 当 OP2CALEN 为 1 时，此区域无法写入任何内容。
7:6	RSV	保留位，固定读为 0
5:0	OP2CALDATN[5:0]	运算放大器 2 负端校验结果数据值： 这几位用于显示自动校验结果，用户可根据数据内容进行微调。 BIT5 代表符号位： 0：表示增加电压 1：表示减少电压 BIT[4:0]代表数据档位： 1：LSB 表示一格模拟设计电压档位 注：只有 OP2CALEN 为 0 时，可以对此区域的内容进行修改； 当 OP2CALEN 为 1 时，此区域无法写入任何内容。

22 调试支持（DBGMCU）

22.1 概述

本器件基于 Cortex-M0 内核，内嵌高级调试功能，支持内核在取值（breakpoint）和访问数据（watchpoint）时停止。内核停止时，可查询内核状态和系统外部状态。完成查询，内核和系统可恢复，程序继续执行。

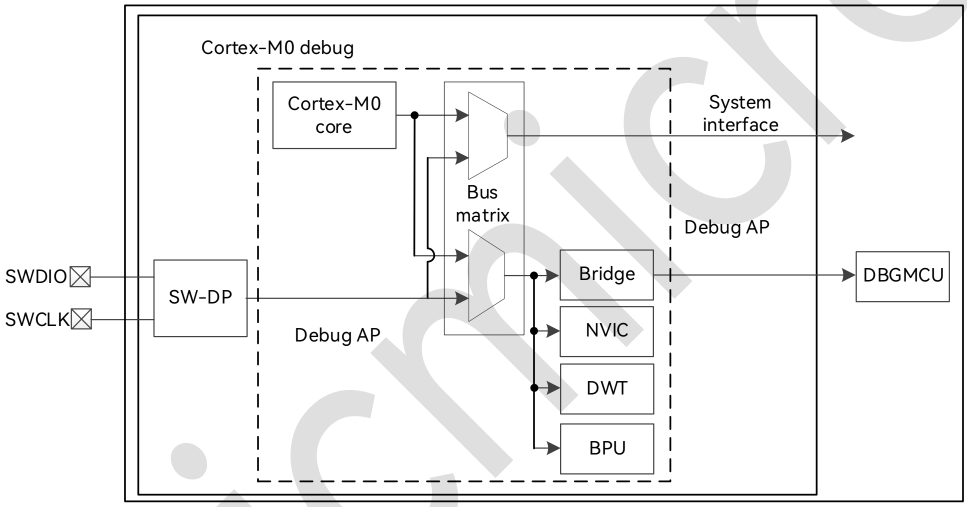


图 22-1：DEBUG 框图

ARM Cortex-M0 内核片上集成了调试接口，包括：

- SW-DP：串行调试接口
- BPU：断点单元
- DWT：数据观察点触发

注：有关 ARM Cortex®-M0 内核调试功能的更多信息，请参阅 The Cortex®-M0 Technical Reference Manual。

相关文档参考：

- <Cortex®-M0 Technical Reference Manual (TRM)>
- <ARM Debug Interface V5>
- <ARM CoreSight Design Kit revision r1p1 Technical Reference Manual>

22.2 引脚分布和调试引脚

表 22-1：SW debug 端口

SW-DP 引脚名	I/O 类型	说明	引脚分配
SWDIO	I/O	串行数据输入/输出	PA13
SWCLK	I	串行时钟	PA14

复位后（SYSRESETn 或 PORRESETn），SW-DP 端口被指定为 SW 功能，调试器主机可以立即使用这些引脚。

另外，MCU 也提供了禁用 SWD 端口的操作，这样可以释放相关引脚为通用 IO（GPIO）。详细禁用操作信息，请参考 6.3.2 I/O 复用功能映射。

22.2.1 SWD 引脚内部上拉/下拉

保证 SWD 输入引脚非浮空状态，内嵌上拉/下拉：

- SWDIO：内部上拉
- SWCLK：内部下拉

一旦 SWD 引脚被用户代码释放，GPIO 控制器将取得控制权，IO 口恢复到复位态：

- SWDIO：输入内部上拉
- SWCLK：输入内部下拉

注：内嵌电阻可以消除片外增加电阻的需求。

22.3 ID 码和锁定机制

芯片包含多个 ID 码，用户可通过 debug 接口或应用程序访问。

22.3.1 CHIP_ID 存储地址

MCU 内置 CHIP_ID 寄存器，包含芯片代号和版本号信息。此 ID 存储的起始地址为 0x4001 00FC，长度为 4 字节，默认值为 0x0000B28X。

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

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

位	名称	描述
31:16	RSV	保留位，固定读为 0
15:4	DEV[11:0]	芯片 ChipID，读出为 0xB28
3:0	REV[3:0]	芯片版本信息，复位值反应实际版本号

22.3.2 DBGMCU_IDCODE 存储地址

DBGMCU_IDCODE 存放于 DBGMCU 模块的起始地址，占用 4 个字节，具体功能参考 22.6 节 BGMCU 模块的寄存器描述。

22.3.3 Cortex SW IDCODE

CPU 有一个 SW IDCODE，通过 SW-DP 访问，包含 SW-DP 的 partnumber 和 JEDEC-106 ID 标识码。在 SW 通讯连接时作为 connect 连接判断及 JTAG/SW 区分。

22.3.4 CPUID

CPU 内置一个 CPUID 寄存器，用于确定 CPU 的类型和版本。

22.3.5 ID 编码汇总

表 22-2：ID 编码汇总表

ID 名称	地址	值
CHIP_ID	0x4001 00FC	0x0000B28x（x=0,1,2…）
DBGMCU_IDCODE	0x4001 5800	0x10006444
CPUID	0xE000 ED00	0x410CC200 (r0p0)
SW IDCODE	DPnAP=1，地址 0b00，read	0x0BB11477

注：CPUID 和 SW IDCODE 是 cortex-m0 内核设计包含的，详情请参考<Cortex®-M0 Technical Reference Manual (TRM)>。

22.4 SWD 接口

详情请参考<Cortex®-M0 Technical Reference Manual (TRM)>。

22.5 MCU 调试模块（DBGMCU）

MCU 调试模块协助调试器提供以下功能：

- 低功耗模式
- 在断点时提供定时器，看门狗的时钟控制

22.5.1 低功耗模式的调试支持

通过使用 WFI 和 WFE 可以进入低功耗模式。

MCU 支持多种低功耗模式，分别可以关闭 CPU 时钟，或降低 CPU 的能耗。

内核不允许在调试期间关闭 FCLK 或 HCLK。这些时钟对于调试操作是必要的，因此在调试期间，它们必须工作。MCU 使用一种特殊的方式，允许用户在低功耗模式下调试代码。

为实现这一功能，调试器必须先设置一些配置寄存器来改变低功耗模式的行为：

- 在 Sleep 模式下，FCLK 和 HCLK 保持工作状态，因此，这种模式不会对标准的调试特性施加任何限制。
- 在 Stop 模式下，调试器必须先置位 DBG_STOP 位。这将在 Stop 模式下激活内部 RC 振荡器，为 FCLK 和 HCLK 提供时钟。

22.5.2 Timer、看门狗的调试支持

在产生断点时，有必要根据定时器和看门狗的不同用途选择计数器的工作模式：

- 在产生断点时，计数器继续计数。这在输出 PWM 控制电机时常常要用到。
- 在产生断点时，计数器停止计数。这对于看门狗的计数器是必需的。

22.6 相关寄存器

22.6.1 DBGMCU ID 寄存器（DBGMCU_IDCODE）

偏移地址：0x00

复位值：0x1000 6444

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">REV_ID[15:0]</td></tr>
<tr><td colspan="16">r</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">DEV_ID[15:0]</td></tr>
<tr><td colspan="16">r</td></tr>
</table>

位	名称	描述
31:16	REV_ID[15:0]	版本信息，读出值固定为 0x1000
15:0	DEV_ID[15:0]	ID 信息，读出值固定为 0x6444

22.6.2 控制寄存器（DBGMCU_CR）

偏移地址：0x04

复位值：0x0000 0000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Res.</td></tr>
<tr><td colspan="16"></td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">Res.</td><td>DBG_STOP</td><td>Res.</td></tr>
<tr><td colspan="14"></td><td>rw</td><td></td></tr>
</table>

位	名称	描述
31:2	RSV	保留位，固定读为 0
1	DBG_STOP	调试停机模式（Debug Stop mode）: 0:（FCLK 关，HCLK 关）在停机模式时，时钟控制器禁止一切时钟（包括 HCLK 和 FCLK）。当从 STOP 模式退出时，时钟的配置和复位之后的配置一样（由 18MHz 的内部振荡器（HSI）提供时钟）。因此，软件必须根据应用需求重新配置时钟控制系统。 1:（FCLK 开，HCLK 开）在停机模式时，RC 振荡器保持开启，FCLK 和 HCLK 时钟由内部 RC 振荡器（HSI 18M）提供。退出 STOP 模式时，软件必须根据应用需求重新配置时钟控制系统。
0	RSV	保留位，固定读为 0

22.6.3 冻结寄存器（DBGMCU_APB1）

偏移地址：0x08

复位值：0x0000 0000

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

15 14 13	12	11 10	9	8	7 6 5 4 3 2 1 0
Res.	DBG_IWDG_STOP	Res.	DBG_TIM6_STOP	DBG_TIM14_STOP	Res.
	rw		rw	rw	

位	名称	描述
31:13	RSV	保留位，固定读为 0
12	DBG_IWDG_STOP	调试模式下 CPU 停止，独立看门狗停止（Debug IWDG stopped when core is halted）: 0：内核停止，独立看门狗继续工作 1：内核停止，独立看门狗停止
11:10	RSV	保留位，固定读为 0
9	DBG_TIM6_STOP	调试模式下 CPU 停止，TIM6 计数器停止（Debug TIM6 stopped when core is halted）: 0：内核停止，TIM6 计数器继续工作 1：内核停止，TIM6 计数器停止
8	DBG_TIM14_STOP	调试模式下 CPU 停止，TIM14 计数器停止（Debug TIM14 stopped when core is halted）

位	名称	描述
		0：内核停止，TIM14 计数器继续工作 1：内核停止，TIM14 计数器停止
7:0	RSV	保留位，固定读为 0

22.6.4 APB2 冻结寄存器（DBGMCU_APB2）

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.												DBG_TIM18_STOP	Res.		
												rw			

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

位	名称	描述
31:20	RSV	保留位，固定读为 0
19	DBG_TIM18_STOP	调试模式下 CPU 停止，TIM18 计数器停止（Debug TIM18 stopped when core is halted）： 0：内核停止，TIM18 计数器继续工作 1：内核停止，TIM18 计数器停止
18:13	RSV	保留位，固定读为 0
12	DBG_MCP_STOP	调试模式下 CPU 停止，MCP 计数器停止（Debug MCP stopped when core is halted）： 0：内核停止，MCP 计数器继续工作 1：内核停止，MCP 计数器停止
11:0	RSV	保留位，固定读为 0

23 器件电子签名（UID）

设备的电子签名存储在闪存模块的系统内存区, 可以通过调试接口读取, 也可以通过 CPU 读取。它包含工厂编程的识别和校准数据，允许用户固件或其他外部设备读取，用以自动匹配微控制器的特性。

23.1 UID 器件唯一码

UID 用于:

- 用作序列号（举例，USB 字符串序列号或其他终端应用）
- 用作安全密钥，使用此唯一码结合软件加密算法，可以提高代码安全性
- 用作激活自举过程的安全检测机制

UID 为 96 位的唯一硬件标识符，由芯片出厂时确定，用户不可改写。

23.2 UID 存储地址

96 位 UID 在芯片出厂时被存储在系统存储区，起始地址为 0x1FFF F7AC。

23.2.1 UID 数据 1

基地址：0x00

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[31:15]															
r															

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

位	名称	描述
31:0	UID [31:0]	X and Y coordinates on the wafer（BCD 码格式）

23.2.2 UID 数据 2

基地址：0x04

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[63:48]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[47:40]								UID[39:32]							
r								r							

位	名称	描述
31:8	UID[63:40]	Lot number（ASCII 码格式）
7:0	UID[39:32]	Wafer number（8-bit 无符号数）

23.2.3 UID 数据 3

基地址：0x08

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[95:80]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[79:64]															
r															

位	名称	描述
31:0	UID[95:64]	Lot number(ASCII 码格式)

23.3 存储区大小

FLASH 空间大小可以通过访问系统存储区内指定地址获取当前型号芯片的 FLASH 容量值，硬件使用 2 个字节表示存储区的容量，存储数据的起始地址为 0x1FFF F7CC。

23.3.1 FLASH 空间大小数据

偏移地址：0x00

复位值：0xXXXX

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

位	名称	描述
15:0	FLASH_SIZE[15:0]	FLASH 空间大小: 这个位字段表示设备 FLASH 内存的大小，以 KBytes 为单位 举例：0x0020 为 32KBytes，0x0040 为 64KBytes

24 版本维护

版本	日期	描述
V1.0	2025.05.20	初始版